UNIVERSITY OF VERONA

DEPARTMENT OF

Computer Science

DOCTORAL SCHOOL

Area: Natural Sciences and Engineering

DOCTORAL PROGRAM IN

Computer Science

Cycle XXXV, 2019

Explainable temporal data mining techniques to support the prediction task in Medicine

S.S.D. INF/01 - INFORMATICS

Beatrice Amico

Supervisor:
Prof. Carlo Combi

PhD program chair:
Prof. Ferdinando Cicalese

Explainable temporal data mining techniques to support the prediction task
in Medicine
Beatrice Amico
Ph.D. Thesis
Verona, Italy, May 9, 2023

Abstract

In the last decades, the increasing amount of data available in all fields raises the necessity to discover new knowledge and explain the hidden information found.

On one hand, the rapid increase of interest in, and use of, artificial intelligence (AI) in computer applications has raised a parallel concern about its ability (or lack thereof) to provide understandable, or explainable, results to users. In the biomedical informatics and computer science communities, there is considerable discussion about the " un-explainable" nature of artificial intelligence, where often algorithms and systems leave users, and even developers, in the dark with respect to how results were obtained. Especially in the biomedical context, the necessity to explain an artificial intelligence system result is legitimate of the importance of patient safety.

On the other hand, current database systems enable us to store huge quantities of data. Their analysis through data mining techniques provides the possibility to extract relevant knowledge and useful hidden information. Relationships and patterns within these data could provide new medical knowledge. The analysis of such healthcare/medical data collections could greatly help to observe the health conditions of the population and extract useful information that can be exploited in the assessment of healthcare/medical processes. Particularly, the prediction of medical events is essential for preventing disease, understanding disease mechanisms, and increasing patient quality of care. In this context, an important aspect is to verify whether the database content supports the capability of predicting future events.

In this thesis, we start addressing the problem of explainability, discussing some of the most significant challenges need to be addressed with scientific and engineering rigor in a variety of biomedical domains. We analyze the "temporal component" of explainability, focusing on detailing different perspectives such as: the use of temporal data, the temporal task, the temporal reasoning, and the dynamics of explainability in respect to the user perspective and to knowledge.

Starting from this panorama, we focus our attention on two different temporal data mining techniques.

The first one, based on trend abstractions, starting from the concept of Trend-Event Pattern and moving through the concept of prediction, we 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.

The second one, based on functional dependencies, 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. We then discuss the concept of approximation, the data complexity of deriving an APFD, the introduction of two new error measures, and finally the quality of APFDs in terms of coverage and reliability.

Exploiting these methodologies, we analyze intensive care unit data from the MIMIC dataset.

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction: the general picture

Motivation and context

Modern technologies allow storing huge amounts of data. Traditional manual data analysis has become insufficient, and methods for efficient computer-based analysis are indispensable. The ability to extract useful hidden knowledge and perform actions on the basis of the discovered knowledge is becoming increasingly important. Knowledge discovery in databases is frequently defined as a process consisting of the following steps: understanding the domain, forming the data set and cleaning the data, extracting regularities hidden in the data, thus formulating knowledge seen as patterns or models (data mining), post-processing of discovered knowledge, and exploiting the results [66]. With the further objectives of clustering, classification, and prediction, discovering new information, becomes a characteristic widespread in different domains, such as finance, information systems security, and, in particular, medicine [167].

Data mining in medicine has been receiving considerable attention since it provides a way of revealing useful information hidden in clinical data. The explosion of the number of clinical datasets available for the research activity is accompanied by the necessity to extract useful knowledge from these data. Modern technologies give the possibility to clinicians to collect automatically huge amounts of data. The analysis of such healthcare/medical data collections could greatly help to gain a deeper insight into the health conditions of the population and extract useful information that can be exploited in the assessment of healthcare/medical processes. The massive amount of data derived from electronic medical records is often under-utilized. Such datasets can be characterized by their incompleteness because of missing parameter values, incorrectness as a consequence of systematic or random noise in the data, sparseness as a result of few and/or non-representable patient records available, and inexactness in the selection of parameters for the given task [101].

Another important aspect that has to be considered is the intrinsic necessity to take into account the temporal dimension. This aspect needs to be considered when representing information within computer-based systems, when querying information about temporal features of the represented real world, when reasoning about time-oriented data, during the design process of analysis tools for prediction, personalized medicine, and therapy support. Analyzing time-oriented data enables researchers to discover new temporal knowledge and gain an understanding of the temporal behavior and temporal associations present in the data. One of the most meaningful examples of the clinical domain, where the analysis of the temporal aspect is fundamental, is given by the intensive care unit domain. In this environment, the health condition of the patients is constantly monitored, and it is subjected to several changes over time. The possibility to analyze the patients' behavior over the time, comparing different conditions and situations, discovering hidden information among the collected data, could greatly increase patient care. It is particularly interesting to extract not only the recurrent information, but also have the possibility to predict some specific event that could occur in the future.

Problem statement and contributions

Faced with this amount of data to be analyzed and understood, the use of diagnostic AI systems to support clinicians has become an important building block. Recent advances in artificial intelligence present a relevant opportunity to improve healthcare. However, the translation of research techniques to effective clinical deployment represents a challenge. The concept of *explainability* is often the center of debates. In biomedical informatics and computer science communities, a central point of discussion is the un-explainable nature of Artificial Intelligence (AI), referring to AI algorithms and systems as black-box. Explainability is a notion that is intrinsically complex and requires a cooperation among the different involved stakeholders [111]. Such systems in medicine are often related to decision-support tasks, where data may be incomplete, uncertain, ambiguous, or missing. Thus, robust, prospective clinical evaluation is essential to ensure that AI systems are safe and effective.

In this thesis, in Chapter 3, we focus our attention on the description of the current panorama in the use of explainable AI (XAI) in medicine, examining the most recent research activities that apply these techniques. We also gave space to some reflections on the terminologies, and on how to facilitate communication between the different stakeholders. We consider the rapid increase of interest in, and use of, artificial intelligence (AI) in computer applications which has raised a parallel concern about its ability (or lack thereof) to provide understandable, or explainable, output to users. This concern is especially legitimate in biomedical contexts, where patient safety is of paramount importance. We explore in depth the concept of explainable AI, or XAI, offering a functional definition and conceptual framework or model that can be used when considering XAI. We also consider a series of desiderata for attaining explainability in AI, each of which touches upon a key domain in biomedicine. An essential aspect that need to be discussed moving closer to the content of this thesis is the temporal aspect of XAI. We introduce and decline with examples some important issues, according to which how we can decline temporality

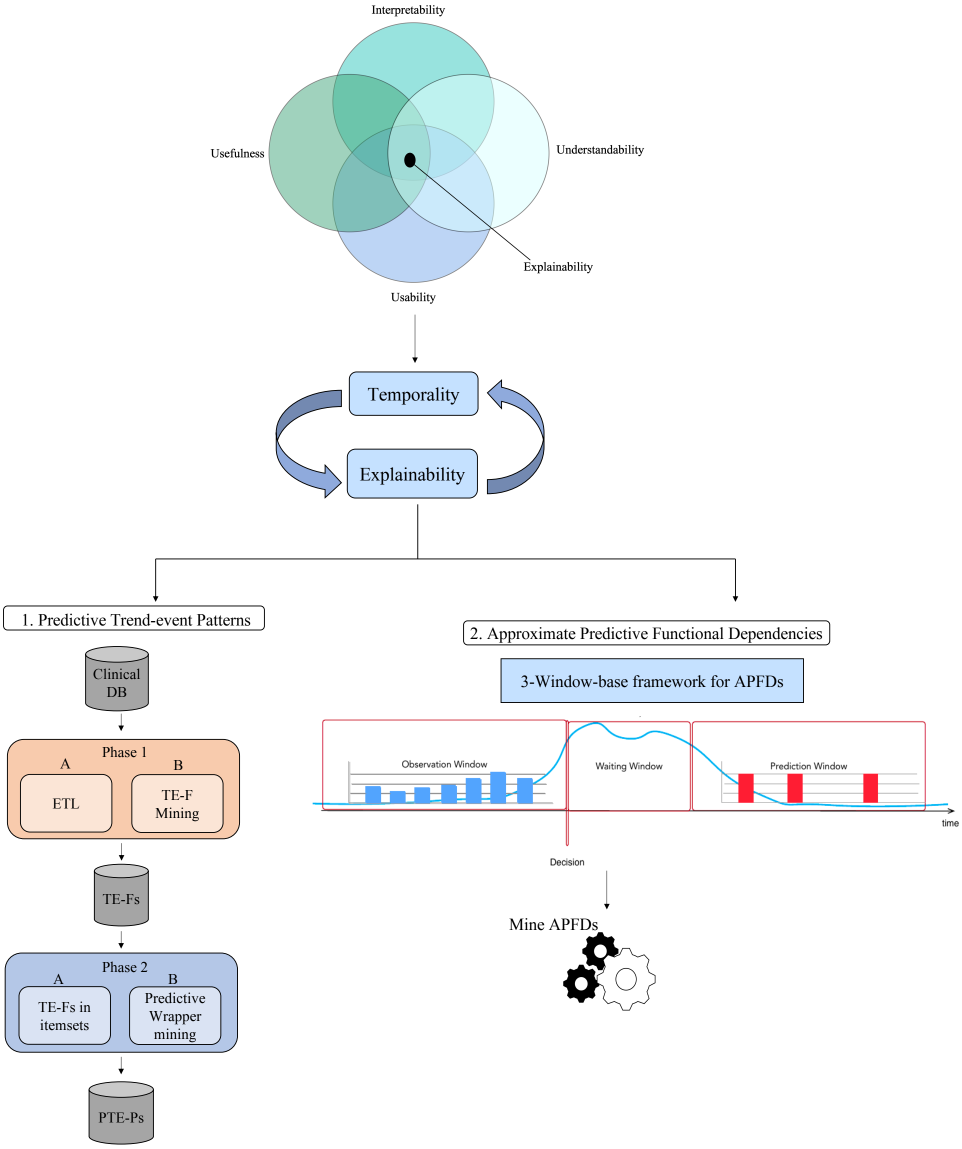


Figure 1-1: Graphical structure of this thesis.

in the XAI context. The main concepts of this chapter have been published in [44].

With the aim of integrating explainability in the context of temporal data mining, in this thesis we present two different techniques that decline some of the aspects of temporal XAI, mentioned in Chapter 3.

In Chapter 4, we propose a methodology for deriving a new kind of predictive temporal patterns, called *predictive trend-event patterns* (*PTE*-Ps), that consists of predictive patterns composed by event occurrences and trends of vital signs, they could influence. *PTE*-Ps are extracted using a classification model that considers and combines various predictive pattern candidates and selects only those that are relevant to improve the performance of the prediction of a specific class (e.g., only those patterns important to predict sepsis). We provide an original algorithm to mine *PTE*-Ps and describe the tool we implemented for retrieving them, published in [117].

The second approach proposed in Chapter 5, is centered on a new type of functional dependency, where the final goal is mining temporal functional dependencies that predict a future event. We show how temporal data mining, precisely mining of functional dependencies, can be fruitfully exploited to improve prediction, using a motivating example from medicine.

To develop an early prediction model, a window-based data aggregation approach could be a good starting point, therefore we introduce a new temporal framework based on three temporal windows designed to extract predictive information. 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 Observation Window (OW), Waiting Window (WW), and Prediction Window (PW). We define a Predictive Functional Dependency, discuss the concept of approximation for such PFDs, introduce two new error measures, address the (data) complexity of deriving an APFD, and finally discuss the quality of APFD in terms of coverage and reliability. A part of the concepts of this chapter has been already published in [10], where we won the “Mario Stefanelli Best student paper” award.

We summarize the general overview of this thesis in Figure 1-1.

Structure of the thesis

The overall structure of the thesis is the following:

- Chapter 2 - Background and Related work. In this chapter, we explain the two medical problems on which we base our preliminary experiments describing in detail, the main source used throughout the thesis, the MIMIC dataset. Secondly, we collect the most important contributions in literature, meaningful for the methods reported in this thesis. We decide to focus mainly on different data mining techniques temporal oriented. We particularly detail the functional dependencies and its different aspects;

- Chapter 3 - Towards Explainable Artificial Intelligence in Medicine. In this chapter, we address one of the major challenges that must be faced by researchers in the last years, namely the explainability of AI algorithms. We report a research paper published recently *A manifesto on explainability for artificial intelligence in medicine*, in Artificial Intelligence in Medicine. Specifically, with Prof. Carlo Combi and John H Holmes, I contributed on the foundation of the general idea, comprehensive of the Venn diagram, the different aspects to analyze, and the different applications in clinical fields. In addition, I investigated the temporal aspects of explainability, discussing some issues on how temporalities and explainability are intertwined;

- Chapter 4 - Discovering Predictive Trend-Event Patterns. We formalize a new type of temporal predictive pattern. The idea is to extract a compact set of complex predictive patterns, composed by interval-based data (i.e., trends) and instantaneous events, through a classification model. We describe the research work *Discovering predictive trend-event patterns in temporal clinical data* published at SAC (Symposium on Applied Computing) 2021;

- Chapter 5 - A framework for the discovery of predictive temporal functional dependencies. We introduced and discussed a novel framework for extracting APFDs. The approach fits well into the context of the approximate temporal functional dependencies, adding a new aspect that has never been formalized before. It differs from the previous work because we dealt with the potential predictiveness of the approximate temporal functional dependencies, considering the possibility to exploit data dependencies for the prediction. We describe the already published *A 3-Window Framework for the Discovery and Interpretation of Predictive Temporal Functional Dependencies*, at International Conference on Artificial Intelligence in Medicine (AIME 2022), and additional novel contributions.

- Chapter 6 - Conclusions. We conclude by summarizing the research work presented in this thesis, discussing limitations, and providing a general overview of future work.

Chapter 2

Background and Related work

In this chapter, we recall some basic notions of the clinical domains used in the thesis. We describe the MIMIC dataset, the data source used for our experiments. In the second part, we introduce the importance of the temporal component in the data, delineating two main sections, temporal abstraction and pattern discovery and temporal functional dependencies with their approximation. We recap the most meaningful examples of temporal abstractions, recalling the Time Interval Related Patterns (TIRPs), different trend abstractions, and temporal association rules. Moreover, we give an overview of different types of approximate functional dependencies, and on their temporal extension. We select different techniques according to the possibility of explaining the results and their temporal component, a fundamental aspect throughout the entire thesis.

2.1 The medical domains and data sources

2.1.1 The clinical domains

The increasing incidence of morbidity and mortality caused by multiple organ failure has paralleled improvements in life-support technologies available to patients admitted to an intensive care unit (ICU) [28]. Critically ill patients are typically cared for in ICUs, which specialize in providing continuous monitoring and advanced therapeutic and diagnostic technologies. Physicians have the access to a large quantity of data for each patient collected in the electronic medical records. In these contexts, it is very difficult to identify the most important information for clinical decisions. Within this panorama, it is crucial to exploit this amount of data, in order to discover novel information possibly useful for clinicians.

Sepsis

Sepsis is a systemic, deleterious host response to infection leading to severe sepsis (acute organ dysfunction secondary to documented or suspected infection) and septic shock (severe sepsis plus hypotension not reversed with fluid resuscitation). Severe sepsis and septic shock are major healthcare problems,

General variables
Fever (>38.3°C))
Hypothermia (core temperature<36°C)
Heart rate>90 min-1 or more than two SD above the normal value for age
Tachypnea
Altered mental status
Significant edema or positive fluid balance (>20 mL/kg over 24 h)
Hyperglycemia (plasma glucose>140 mg/dL or 7.7 mmol/L) in the absence of diabetes
Inflammatory variables
Leukocytosis (WBC count>12,000 $\mu L^{-}1$)
Leukopenia (WBC count<4,000 $\mu L^{-}11$)
Normal WBC count with greater than 10
Plasma C-reactive protein more than two SD above the normal value
Plasma procalcitonin more than two SD above the normal value
Hemodynamic variables
Arterial hypotension (SBP<90 mmHg, MAP<70 mmHg, or an SBP decrease>40 mmHg in adults or less than two SD below normal for age)
Organ dysfunction variables
Arterial hypoxemia (PaO2/FiO2<300)
Acute oliguria (urine output<0.5 mL kg^{-1} h^{-1} for at least 2 h despite adequate fluid resuscitation)
Creatinine increase<0.5 mg/dL or 44.2 lmol/L
Coagulation abnormalities (INR>1.5 or aPTT>60 s)
Ileus (absent bowel sounds)
Thrombocytopenia (platelet count<100,000 $\mu L^{-}1$)
Hyperbilirubinemia (plasma total bilirubin>4 mg/dL or 70 lmol/L)
Tissue perfusion variables
Hyperlactatemia (>1 mmol/L)
Decreased capillary refill or mottling

Table 2.1: Diagnostic criteria for sepsis from [61]

Severe sepsis
Severe sepsis definition = sepsis-induced tissue hypoperfusion or organ dysfunction (any of the following thought to be due to the infection)
Sepsis-induced hypotension
Lactate above upper limits laboratory normal
Urine output<0.5 mL kg^{-1} h^{-1} 1 for more than 2 h despite adequate fluid resuscitation
Acute lung injury with PaO2/FiO2<50 in the absence of pneumonia as infection source
Acute lung injury with PaO2/FiO2<200 in the presence of pneumonia as infection source
Creatinine>2.0 mg/dL (176.8 μmol/L)
Bilirubin>2 mg/dL (34.2 μmol/L)
Platelet count<100,000 lL
Coagulopathy (international normalized ratio>1.5)

Table 2.2: Diagnostic criteria for severe sepsis from [61]

affecting millions of people around the world each year, killing one in four (and often more), and increasing in incidence [61, 28].

Sepsis is defined as the presence (probable or documented) of infection together with systemic manifestations of infection. In Table 2.1, we report the clinical guidelines for the diagnosis of sepsis found in literature [61]. While in Table 2.2, we report the criteria for the *severe sepsis*, defined as sepsis plus sepsis-induced organ dysfunction or tissue hypoperfusion.

Acute kidney injury (AKI)

Acute kidney injury (AKI), previously known as acute renal failure (ARF), is a syndrome characterized by sudden kidney failure or kidney damage that occurs within a few hours or a few days and rarely has a sole and distinct pathophysiology. The need to evaluate the adequacy and efficacy of different therapeutic protocols, in addition to the possibilities of prevention and/or limitation of the damage, has led to formulate a classification of the AKI that also includes slight alterations in the renal function. AKI is not a single-organ failure clinical event, but a syndrome where the kidney plays an active role in the progress of multi-organ dysfunction, with different critical clinical conditions ranging from a slight increase in creatinine to anuria, namely the complete cessation of urine flow [116].

The early detection, the prompt treatment and the anticipated interventions, are elements that likely provide benefits for the patient which has the possibility to avoid temporary support from a dialysis machine or death itself. AKI is often a quickly-evolving clinical event with high morbidity that represents an important complication in patients admitted to hospital (10–15% of all hospitalizations). The mortality rate can be very high, between 50% and 80%, especially for patients in the ICU, where it sometimes exceeds 50% [153].

The major challenge to AKI diagnosis and treatment is that specific syndromes often coexist, without the immediate onset of alarming symptoms such as chest pain, dyspnea, palsy or blindness; hence, diagnosis requires specific technical assessments. One of the most important index of illness is the Glomerular filtration rate (GFR), widely accepted as the best indicator of renal function in both healthy and ill patients. It measures the amount of plasma filtered through the glomeruli in a given period of time. However, GFR is difficult to measure and is commonly estimated from the serum levels of endogenous filtration markers, such as creatinine. In details, the increase in serum creatinine levels represents a marker of kidney excretory function, while a decrease in the urinary output represents a quantitative marker of urine production.

Different criteria have been used to gather accurate conclusions on the epidemiology of this syndrome. The first one was the International consensus criteria introduced by the Acute Dialysis Quality Initiative [23], and afterwards modified by the AKI Network [123], until 2012 when Kidney Disease Improving Global Outcomes (KDIGO) [106] provided the new guidelines. The KDIGO releases their clinical practice guidelines for acute kidney injury (AKI), which build off of the RIFLE criteria and the AKIN criteria to evaluate, classify and manage patients with renal insufficiency. The KDIGO definition of AKI has been broadly used by researchers and physicians and it is defined as any of the following conditions:

- an increase in serum creatinine by ≥ 0.3 mg/dl ($\geq 26.5\, \mu mol/l$) within 48 h;
- an increase in serum creatinine to ≥ 1.5 times baseline within the previous 7 days;
- a urine volume ≤ 0.5 ml/kg/h for 6 hours.

The KDIGO criteria stage patients according to changes in serum creatinine and urine output, rather than changes in glomerular filtration rates, apart from in children younger than 18 years. In the Table 2.3, we report the AKI severity stages.

2.1.2 MIMIC

In recent years, the digitalization of the health care data has become a central focus. In the following, we present the two clinical datasets that have been

Stage	Serum creatinine	Urine output
1	1.5–1.9 times baseline OR $\geq$ 0.3 mg/dl ($\geq$ 26.5 mmol/l) increase	<0.5 ml/kg/h for 6–12 hours
2	2.0–2.9 times baseline	<0.5 ml/kg/h for $\geq$ 12 hours
3	3.0 times baseline OR Increase in serum creatinine to $\geq$ 4.0 mg/dl ($\geq$ 353.6 mmol/l) OR	<0.3 ml/kg/h for $\geq$ 24 hours OR Anuria for $\geq$ 12 hours

Table 2.3: AKI is staged for severity according to the following criteria.

used to infer knowledge through techniques described in the next chapters.

MIMIC-III

MIMIC-III is a relational database containing data related to patients, who were hospitalized within the intensive care units at Beth Israel Deaconess Medical Center between 2001 and 2012; in particular, 53.423 distinct hospital admissions for adult patients who were hospitalized to critical care units from 2001 to 2012, and 7.870 neonates admitted between 2001 and 2008.

Before the insertion in the database, a patient was first deidentified in accordance with Health Insurance Portability and Accountability Act (HIPAA) standards using structured data cleansing and date shifting. They removed all the identifying data elements, such as name, address, telephone number. In addition, dates were shifted into the future by a random offset for each individual patient in a consistent manner to preserve intervals. Figure 2-1 represents the general overview of the database.

As a general introduction to the relational database, MIMIC-III consists of 26 tables, provided as a collection of comma-separated values (CSV) files, linked by identifiers which usually have the suffix “ID”. HADM_ID refers to a unique hospital admission, SUBJECT_ID refers to a unique patient, and ICUSTAY_ID characterizes a single stay in ICU. Data came from two different critical care information systems: Philips CareVue Clinical Information System and iMDsoft MetaVision ICU. Except data relating to fluid intake, whose tables are characterized by a suffix denoting the data source data, the other information was merged into one dataset when building the database tables. MIMIC-III tables are divided in four groups: tables that track the the patient’s history, tables related to the critical care unit, tables which contain data regarding the hospital record system, and the group of dictionaries pre-fixed with “D_” which provide definitions for identifiers.

The tables useful to track the patient’s history are the following:

- ADMISSIONS: Every unique hospitalization for each patient in the database (defines HADM_ID);
- CALLOUT: Information regarding when a patient was cleared for ICU discharge and when the patient was actually discharged;
- ICUSTAYS: Every unique ICU stay in the database (defines ICUSTAY_ID);
- PATIENTS: Every unique patient in the database (defines SUBJECT_ID);

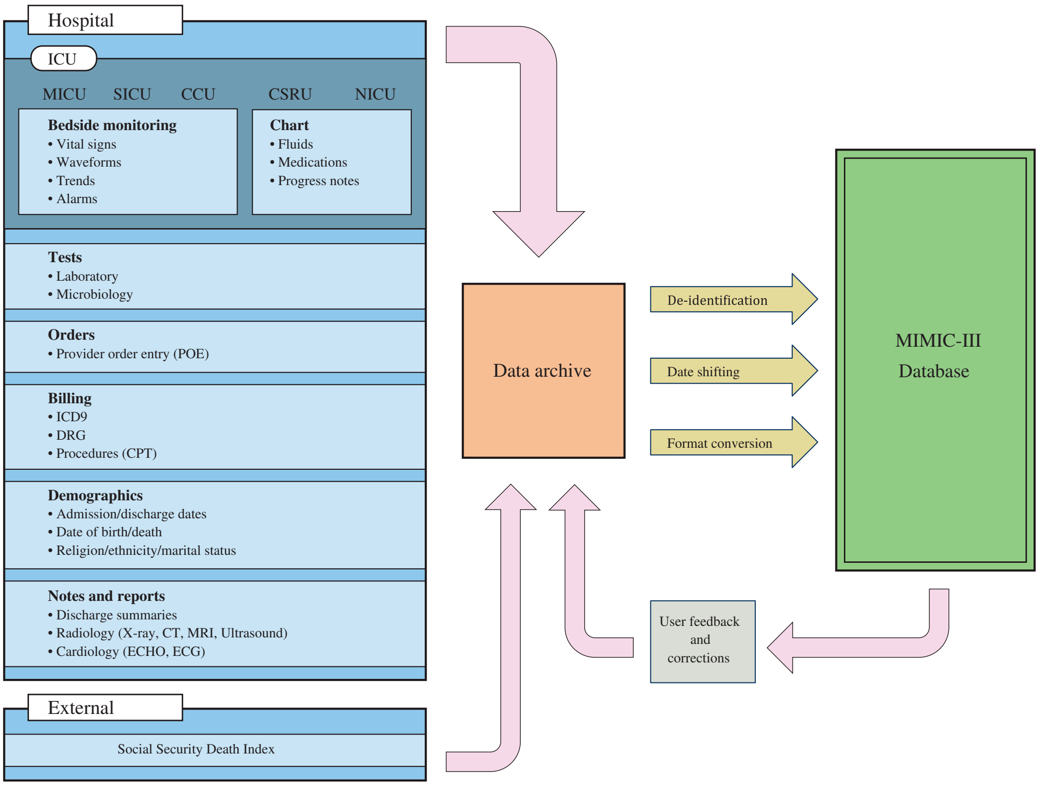


Figure 2-1: Overview of the MIMIC-III critical care database from [100]. *MICU:Medical Intensive Care Unit, SICU: Surgical Intensive Care Unit, CCU: Coronary Care Unit, CSRU: Cardiac Surgery Recovery Unit, NICU: Neonatal Intensive Care Unit.*

- SERVICES: The clinical service under which a patient is registered;
- TRANSFERS: Patient movement from bed to bed within the hospital, including ICU admission and discharge.

Of note, each ICUSTAY_ID characterizes a single HADM_ID and a single SUBJECT_ID. A single SUBJECT_ID can correspond to multiple HADM_ID, and multiple ICUSTAY_ID (multiple ICU stays either within the same hospitalization, or across multiple hospitalizations, or both).

All the tables referred to data collected in the critical care unit are:

- CAREGIVERS: Every caregiver who has recorded data in the database (defines CGID);
- CHARTEVENTS: All charted observations for patients;
- DATETIMEEVENTS: All recorded observations which are dates, for example time of dialysis or insertion of lines;
- INPUTEVENTS_CV: Intake for patients monitored using the Philips CareVue system while in the ICU;

- INPUTEVENTS_MV: Intake for patients monitored using the iMDSoft Metavision system while in the ICU;
- NOTEEVENTS: Deidentified notes, including nursing and physician notes, ECG reports, imaging reports, and discharge summaries;
- OUTPUTEVENTS: Output information for patients while in the ICU;
- PROCEDUREEVENTS_MV: Patient procedures for the subset of patients who were monitored in the ICU using the iMDSoft MetaVision system.

Data regarding the hospital record system are recorded in:

- CPTEVENTS: Procedures recorded as Current Procedural Terminology (CPT) codes;
- DIAGNOSES_ICD: Hospital assigned diagnoses, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system;
- DRGCODES: Diagnosis Related Groups (DRG), which are used by the hospital for billing purposes;
- LABEVENTS: Laboratory measurements for patients both within the hospital and in out patient clinics;
- MICROBIOLOGYEVENTS: Microbiology measurements and sensitivities from the hospital database;
- PRESCRIPTIONS: Medications ordered, and not necessarily administered, for a given patient;
- PROCEDURES_ICD: Patient procedures, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system.

Another important part is related to dictionaries:

- D_CPT: High-level dictionary of Current Procedural Terminology (CPT) codes;
- D_ICD_DIAGNOSES: Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD) codes relating to diagnoses;
- D_ICD_PROCEDURES: Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD) codes relating to procedures;
- D_ITEMS: Dictionary of ITEMIDs appearing in the MIMIC database, except those that relate to laboratory tests;
- D_LABITEMS: Dictionary of ITEMIDs in the laboratory database that relate to laboratory tests.

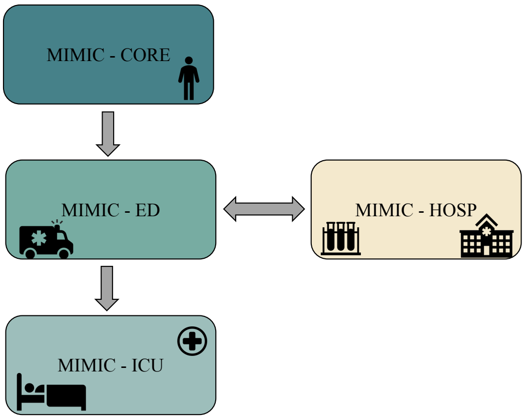


Figure 2-2: MIMIC modules structure

MIMIC-IV

MIMIC-IV (version 1.0) represents an update to MIMIC-III, which incorporates contemporary data and improvements on numerous aspects of MIMIC-III. MIMIC-IV adopts a modular approach to data organization, highlighting data provenance and facilitating both individual and combined use of disparate data sources. MIMIC-IV is sourced from two hospital database systems: a custom hospital wide electronic health record (EHR) and an ICU specific clinical information system. The entire dataset is divided in four modules: *CORE, HOSP, ED*, and *ICU*. The general structure is visualized in in Figure 2-2.

As shown in Figure 2-3, patients are identified using three identifiers: subject_id is the anonymized version of a patient's medical record number, hadm_id is the identifier assigned to each patient hospitalization, transfer_id is the artificially generated identifier uniquely assigned to a ward stay for a patient. The *CORE module* contains three tables:

- patients: here we have basic information about gender, age, and date of death (if exists);
- admissions: this table records data about the patient's admission to the hospital, including for example timing information for admission, discharge, and demographic information;
- transfers: this table contains detailed information about patients' unit transfers that is, the physical locations of each patient during the hospital stay.

The *ED* and *ICU modules* have a star schema (Figure 2-4 and Figure 2-5).

The *ED module* contains data for emergency department patients collected while they are in the ED. The tables involved are:

- edstays: it is the main tracking table for emergency department visits. It provides the admission time to, and the discharge time from the emergency department.

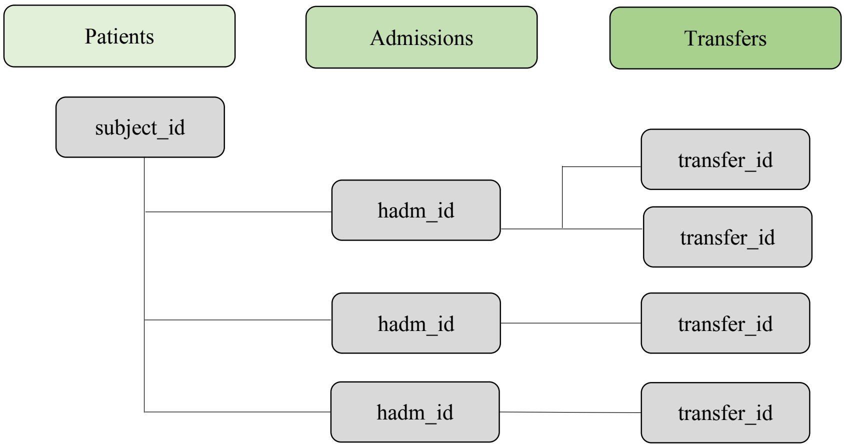


Figure 2-3: MIMIC *CORE module* structure

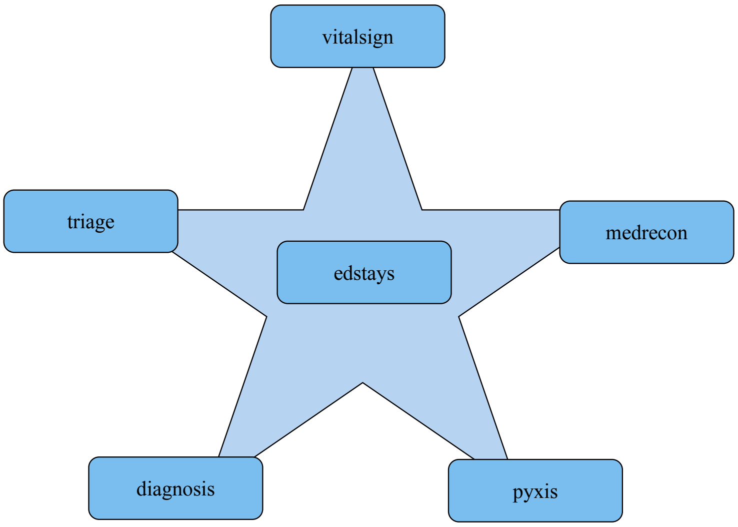


Figure 2-4: MIMIC *ED module* structure

- diagnosis: it provides information about billed diagnoses for each patient;
- triage: it contains information about the patient when he was first triaged in the emergency department;
- vitalsign: it collects all the vital signs of the patient;
- medrecon: it contains all the information about the medications that a patient has already taken, before the admission to the emergency department;
- pyxis: it provides information on medicine dispensations made via the Pyxis system, the emergency information system.

The *ICU module* contains data sourced from the clinical information system at the Beth Israel Deaconess Medical Center (BIDMC): MetaVision (iMDSoft).

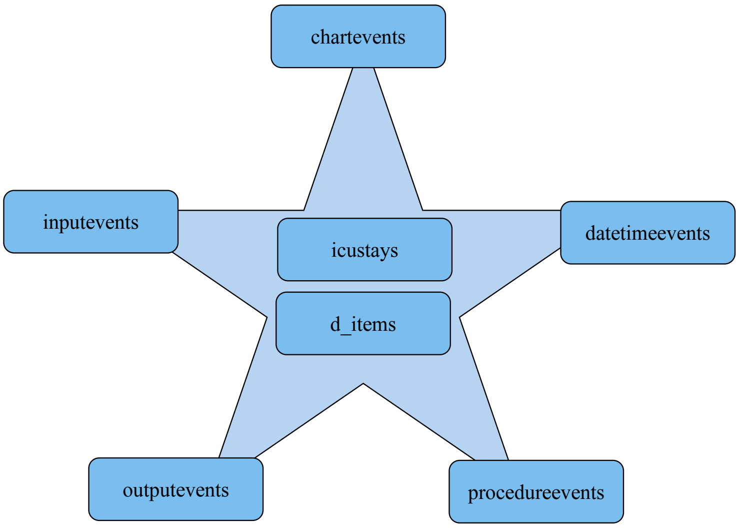


Figure 2-5: MIMIC *ICU module* structure

MetaVision tables were denormalized to create a star schema where the icustays and d_items tables link to a set of data tables all suffixed with ”events”. All events tables contain a stay_id column allowing identification of the associated ICU patient in icustays, and an itemid column allowing identification of the concept documented in d_items. The tables within this module are detailed below:

- d_items and icustays: the first table is the dimension table describing itemid, while the second one contains the tracking information for ICU stays of each patient. These two tables are linked to the rest of the “events” tables;
- outputevents: it contains patient outputs;
- inputevents: it contains intravenous and fluid inputs;
- chartevents: it contains charted information occurring during ICU stays;
- datetimeevents: it contains information documented as a date or time;
- procedureevents: it contains procedures documented during the ICU stay (e.g. ventilation), though not necessarily conducted within the ICU.

The *HOSP module* contains data derived from the hospital wide EH which, as shown in Figure 2-6 divided into four sections, namely billing, measurements, medications, and admin. The tables are detailed below:

- hpcsevents: it contains billed events occurring during the hospitalization;
- d_hcpcs: it describes CPT codes;
- diagnoses_icd: it contains the billed ICD-9/ICD-10 diagnoses for hospitalizations;

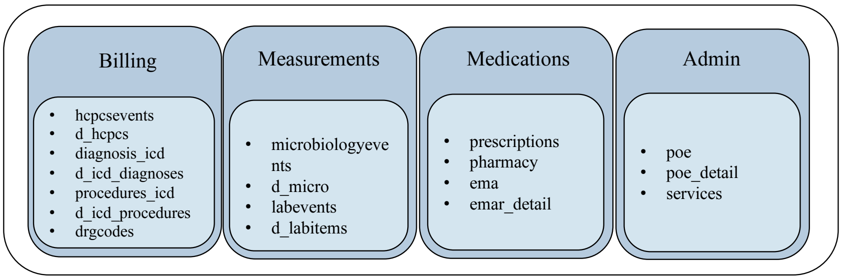


Figure 2-6: MIMIC hosp module structure

- d_icd_diagnoses: it provides a description of ICD-9/ICD-10 billed diagnoses;
- procedures_icd: it contains billed procedures for patients during their hospital stay;
- d_icd_procedures: it is description of ICD-9/ICD-10 billed procedures;
- drgcodes: it contains the billed DRG codes for hospitalizations;
- microbiologyevents: it records microbiology cultures;
- d_micro: it describes microbiology test codes;
- labevents: it stores the results of all laboratory measurements made for a single patient;
- d_labitems: it provides a description of all lab items;
- prescriptions: it provides information about prescribed medications;
- pharmacy: it provides detailed information regarding filled medications which were prescribed to the patient;
- emar and emar_detail: they record all administrations and information of a given medicine to an individual patient;
- poe and poe_detail: they record all treatments and procedures ordered via (Provider order entry) POE, the general interface used by the care providers;
- services: it describes the service_s that a patient was admitted under.

2.2 Temporal abstractions and pattern discovery

Healthcare organizations are increasingly collecting large amounts of data related to their day-by-day activities. The analysis of such healthcare databases could greatly help to gain a deeper insight into the health condition of the population and to extract useful information that can be exploited in the assessment of healthcare processes. In clinical databases, the temporal features play the primary role [55]. Sometimes clinical data are represented by a set of time series of numeric values. In order to get a uniform representation of these data as temporal sequences of events, the clinical data need first to undergo a pre-processing procedure.

Analyzing time-oriented data enables researchers to discover new temporal knowledge and gain understanding regarding the temporal behavior and temporal associations of such data, with the further objectives of clustering, classification, and prediction. To enhance the capabilities of temporal data analysis, it is necessary to perform a preprocessing step of meaningful summarization and interpretations of the time-stamped raw data into interval-based abstractions, known as temporal abstractions. In general, Temporal Abstraction is the abstraction and aggregation of a time point series into a symbolic time intervals series-based representation, suitable for the purposes of data mining. The temporal-abstraction process can use knowledge-based approaches, which exploit domain-specific knowledge, or data-driven, domain-independent discretization methods.

2.2.1 Temporal abstractions

The first conceptual model based on temporal abstractions (TAs), was proposed by Shahar in [164], with a method called the Knowledge-Based Temporal Abstraction (KBTA). This method exploits domain specific knowledge, to generate state abstractions, as well as more complex abstractions, such as gradients. In [165], Shahar and Musen defined a general problem-solving method for interpreting data in time-oriented, knowledge-intensive domains, such as clinical ones: the knowledge-based temporal-abstraction (KBTA) method. The KBTA framework emphasizes the explict representation of the knowledge required for abstractions of time oriented clinical data, and facilitates its acquisition, maintenance, reuse and sharing. This method is able to acquire the relevant knowledge and to define the domain ontology (e.g., security ontology) based on five KBTA entities and the relations between them (primitive parameters, abstract parameters, contexts, events and patterns). Five inference mechanisms are then applied in parallel to derive the high level abstractions from the raw data. The inputs to these five mechanisms are the primitive parameters and the events, which are related to raw data, and the outputs are contexts, abstractions, and patterns.

Time Interval Related Patterns (TIRPs)

TAs can be further mined to discover temporal patterns. These patterns essentially characterize frequent temporal pathways, trajectories, or journeys, within a given population of entities described by longitudinal multivariate data. Moskovitch and Shahar [130] propose the KarmaLego algorithm, a highly efficient method for fast mining of interval-based patterns, by exploiting the transitivity of temporal relations and with a highly efficient indexing scheme. It is a Time Interval Related Patterns (TIRPs) discovery algorithm with a highly efficient indexing scheme. The KarmaLego algorithm outputs an enumeration tree of all of the frequent TIRPs discovered in the given database. Each discovered TIRP can be found at a leaf of the tree, while its components (symbolic intervals and temporal relations) appear on the path from the root of the tree to the leaf. Each TIRP (path) represents a cluster of entities having similar qualitative temporal relations among their multivariate variables.

Over the years, KarmaLego showed limitations; it referred only to the discovery of the first instance of a symbolic time interval, or TIRP's instance, omitting potentially frequent TIRPs. In 2021, Harel and Moskovitch [78] had introduced TIRPClo, an efficient algorithm for the complete discovery of only the frequent closed TIRPs, a compact subset of all the frequent TIRPs based on which their complete information can be revealed. The algorithm utilizes a memory-efficient index, and a novel method for data projection, which makes it the first algorithm to guarantee a complete discovery of frequent closed TIRPs.

In [167], we find a recent example of the TIRPs use in the clinical context. The authors applied their methodology to three different medical domains: oncology, infectious hepatitis, and diabetes. They explore the similarities among frequent pattern sets of different subject populations, demonstrating that similar patient populations share frequent TIRPs. The discovered TIRPs can indeed be used for various tasks, such as classification or prediction.

In [134] we find another recent example of the use of TIRPs, specifically a new extension of them. The authors introduce the concept of interger-TIRPs (iTIRPs), a novel representation for temporal data consisting of frequent TIRPs' instances for temporal Artificial Neural Networks (ANNs). They focus on the prediction of the mortality in Type 2 diabetes patients with complications from Chronic Kidney Disease (CKD), using iTIRPs as features for the classification. The detected TIRPs' instances are aggregated by counting their appearance in each time-stamp, which is represented by a vector that represents the TIRP's occurrences along time. They demonstrate the potential use of temporal abstraction for the prediction.

Frequent temporal patterns

Another important topic in temporal data analytics is temporal knowledge discovery through frequent temporal patterns. Typically, the task is to discover items that co-occur, or appear together, in basket bags,or other collections. Agrawal and Srikant [7] introduced the AprioriAll algorithm, which is still useful in sequential or time intervals mining, defining that a pattern can be frequent only when its components are initially frequent. In order to find for example that the pattern "ab" is frequent, first we have to verify that "a" is frequent and "b" is frequent, otherwise there is no chance that "ab" can be frequent.

In [17] we find a usage example, where the authors proposed a temporal pattern mining approach for analyzing electronic medical records data. The key step is defining a language that can adequately represent the temporal dimension of the data. They rely on temporal abstractions [164] and temporal logic [9] to define patterns able to describe temporal interactions among multiple time series. They use the class information and mine frequent temporal patterns for each class label separately using local minimum supports as opposed to mining frequent temporal patterns from the entire data using a single global minimum support. The mining algorithm performs an Apriori-like levelwise search [7]. Applying frequent temporal pattern mining on data usually results in a very large number of patterns, most of which may be unimportant for the classification task. If P is frequent, all instances covered by P are also covered by all of its subpatterns, which are also in the result of the frequent pattern mining method. This nested structure causes the problem of spurious patterns, namely a pattern P that is predictive when evaluated by itself, but it is redundant given one of its subpatterns. In order for a pattern mining method to be useful for knowledge discovery, the method should provide the user with a small set of understandable patterns that are able to capture the important information in the data.

2.2.2 Trend abstractions

Among different types of temporal abstractions, we find the trend abstractions which focus on detecting changes in the temporal evolution of a parameter. In literature, there exist many contributions on time series and temporal trends.

Haimowitz et al. [76] propose a temporal pattern-matching system, called TrenDx. TrenDx focuses on using efficient general methods to represent and detect predefined temporal features in raw time-stamped data. Trend templates characterize typical clinical temporal patterns, such as specific types correlated to functional states, disease states or normal growth development, by representing these patterns as temporal and measurement constraints. The TrenDx system has been developed mainly within the domain of pediatric growth monitoring, although examples from other domains have been presented to demonstrate its more general potential.

Wijsen [179]proposes a temporal constraints called trend dependencies (TDs), which permits to express significant temporal trends. The temporal dimension is captured by trend dependencies through the concept of time accessibility relation, which can also express time granularities in a simple and elegant way. Trend dependencies can compare attribute values via some comparison operators.

Mantovani et al. [119] proposed a new kind of temporal patterns called *Trend-Event Patterns* (TE-Ps), namely a family of temporal patterns focused on the interaction of trends and events. There are many different temporal abstractions, one of them being represented by trends. Trend abstraction aims at detecting relevant changes and change rates in the temporal evolution of a parameter. Trend abstraction entails merge and persistence abstraction, in order to derive the extents where no change is observed in the value of the considered parameter. A TE-P is a pattern formed by an event E and two different trends for the same parameter: one before *E* and one after *E*,

called $trend^{pre}$ and $trend^{post}$, respectively; for example: “The increasing trend of the body temperature of a patient before the administration of Paracetamol and the drug administered determines the decreasing trend of the body temperature of the same patient after such administration occurs” becomes $[Increasing; Paracetamol; Decreasing]$.
To derive a TE-P from this scenario, we need to start from the event and more specifically from the time when such event happened. The parameter values are thus partitioned according to their *timestamp*. Every value before the event could potentially be part of $trend^{pre}$, while every value after the event could be in $trend^{post}$. In both trends the first parameter value is denoted as t_{start}, while the last one is t_{end}. The tuple in $trend^{pre}$ that is closer to the event is called t^{pre}_{end}, because it is the last tuple of it; while the tuple in $trend^{post}$ that is closer to the event is called $t_{star}t^{post}$, as it represents the beginning of such trend. Given the definition of TE-P: a TE-P is a pattern with an expression of the form: $[trend^{pre};E; trend^{post}]$.

2.2.3 Temporal association rules (TAR)

Association rule mining is another method for identifying correlations or dependencies between the elements or values (items) in a dataset [163]. They are commonly evaluated by making use of the classical measures support and confidence. The support of an *itemset I* (which will be called *Sup(I)*) is defined as the frequency with which I appears in a dataset. Based on this definition, the measures support and confidence of the rule $X \to Y$ are defined as follows:

$$Support(X \to Y) = \frac{Sup(XY)}{|N|}, Confidence(X \to Y) = \frac{Sup(XY)}{Sup(X)}$$

where $|N|$ is the number of examples/transactions in the dataset, and *Sup(XY)* and *Sup(Y)* are the support of the itemsets XY and X, respectively.

Typically, such relations are expressed in terms of if-then rules consisting of different rule antecedents (conditions) and consequents (targets). It represents a technique with a lot of popularity in data mining research, including medical data mining. The strength of this technique is the possibility to completely explore all patterns that occur in the data. The disadvantage is that the number of association rules could be very large and the outputs could be difficult to deal with. Hence, it is desirable to reduce the mined rule set as much as possible while preserving the most important relations found in the data.

Similar to the previously cited TIRPs, there exists the Temporal Association Rules (TARs). In TIRPs mining the temporal operator is applied among each interval that builds up the pattern. While, according to the definition, any TAR involves only two elements as we will see in details shortly.

Temporal Association Rules (TARs) are association rules of the kind $A \to C$, where the antecedent (A) is related to the consequent (C) by some kind of temporal operator. TARs mining algorithms are aimed at extracting frequent associations, where frequency is evaluated on the basis of suitable indicators, the most utilized being support and confidence. The support gives an indication of the proportion of cases verifying a specific rule in the population; confidence instead represents the probability that a subject verifies the rule given that it verifies its antecedent.

Combi et al. [53] exploited knowledge-based Temporal Abstractions (TAs) to shift from a time point quantitative representation of time series to a qualitative interval-based description of the available data.
A temporal fact f represents a class of episodes of the same type. Each episode e is associated to the interval when the episode holds. $e.start, e.end$ denote the starting and ending point of the interval associated to e, respectively. E_f denotes the set of episodes of a temporal fact f.
Informally we can enunciate: A *temporal association rule (TAR)* is a temporal pattern that exists between episodes of *temporal facts* belonging to a reference set Facts of Interest (FoI). To specify a TAR it is necessary to introduce the concept of temporal precedence: relation *precedes* between two intervals a and b: $a \preceq b \iff a.start \leq b.start \wedge a.end \leq b.end.$

Definition 1 (Temporal Association Rule (TAR)). *A TAR is an implication of the form* $\{a_1, ..., a_n\} \overset{p}{\to} c$ *where* $\{a_1, ..., a_n\} \subset FoI, c \in FoI$ *with* $c \notin \{a_1, ..., a_n\}$, *and* $p = \langle LS, GAP, RS \rangle$ *is the parameter set determining the relation between the antecedent and the consequent.*

To determine an occurrence of the antecedent, there must exist a non empty intersection between all the episodes e_i corresponding to facts a_i, respectively. More specifically, a (composite) antecedent occurrence has interval $[maxStart,$ $\min End]$ where [155]:

$$max.Start \overset{\text{def}}{=\!=} \max(e_i.start|\ 1 \leq i \leq n), minEnd \overset{\text{def}}{=\!=}\ min(e_i.end|\ 1 \leq i \leq n)$$

.

Set p is composed by the following parameters: (i) *Left Shift (LS)*: maximum distance allowed between $maxStart$ and *c.start*; (ii) *Gap (GAP)*: maximum distance allowed between $minEnd$ and *c.star*t; (iii) *Right Shift (RS)*: maximum distance allowed between $minEnd$ and *c.end.*

Definition 2 (Occurrence of a TAR). *An episode set* $\{e_1, ..., e_n, e_c\}$ *is an occurrence of a TAR* $\{a_1, ..., a_n\} \overset{p}{\to} c$ *with* $p = \langle LS, GAP, RS \rangle$, *if* ($i$) $\{e_1, ..., e_n\}$ *is an antecedent occurrence and* e_c *is a consequent episode; (*ii*) the antecedent occurrence precedes the consequent episode, i.e.,* $[maxStart,\ \min End] \preceq c$*; (*$iii$*) all the quantitative constraints imposed by* p *are satisfied.*

In [155], Sacchi et al. present an approach to pre-process and interpret clinical time series. Their idea is to filter the original time series using temporal abstractions an then to interpret the new and derived time series by both statistical and artificial intelligent methods. Patterns of interest can be specified on the basis of domain knowledge into a set called Abstractions of Interest, and rules containing such pattern in the antecedent and in the consequent are extracted. After the development of a TARs mining framework mainly oriented to the analysis of clinical data, the framework had been extended to incorporate also administrative healthcare information into the data set. The autors, starting from the framework of knowledge-based Temporal Abstraction [164], propose a new kind of temporal association rule working on the extraction of the frequent temporal precedence occurrences between patterns. Discovering occurrences of temporal relationships between patterns characterizing a time series needs the accomplishment of three conceptual and procedural steps. First of all, it is necessary to define the patterns and retrieve them in the time series; then a formal definition of the relationships of interest must be given and, finally, an algorithm to search for frequent occurrences of such relationships in the dataset must be designed, implemented and run. Intuitively, a pattern is a behavior or property that we may want to distinguish in the data. In temporal data, a pattern is usually associated to a time interval in which such behavior occurs. Moreover, a pattern is often related to a qualitative representation of the property that we are looking for, which may be interesting in the problem domain. Here, according to the data model proposed in [20], temporal data are represented as time-stamped entities, called events, while their abstract representation is given by TAs as a sequence of intervals, called episodes. Each episode corresponds to a specific behavior of interest detected in the time course of the data. TA tasks could be divided into two subtasks, each one solved by specific mechanisms: Basic TA, solved by mechanisms that abstract time-stamped data into intervals, and Complex TAs, solved by mechanisms that abstract intervals into other intervals. Complex TAs are used to detect patterns characterized by behaviors which cannot be represented by basic TAs. The episode set is evaluated according to a pattern specified between episodes of the two composing episode sets. The complex TA patterns are based on temporal relationships: more specifically, the temporal relationships investigated correspond to the 13 temporal operators defined in Allen's algebra [9]. They included: BEFORE, FINISHES, OVERLAPS, MEETS, STARTS, DURING, their corresponding inverse relations, and the EQUALS operator. We can thus exploit this kind of TA to detect a great variety of patterns. Fig.2-7 shows patterns of complex shape which have been detected both on a single time series, and on multiple time series.

We find another application of the temporal association rules in medicine, where the authors used the repository of the Regional Healthcare Agency (ASL) of Pavia, that maintains a central data repository which stores healthcare data about the population of Pavia area [55]. They specifically focus the analysis on a sample of patients suffering from Diabetes Mellitus. They perform the mining of Temporal Association Rules (TARs) over a set of temporal sequences of hybrid events, i.e.,events characterized by heterogeneous temporal nature. Starting from the idea considering a pattern as the occurrence of one or more contemporary events, here a TAR is defined as a relationship specified through a temporal operator which holds between an antecedent, consisting in a pattern of single or multiple cardinality, and a consequent, consisting in a pattern with single cardinality. A recent example of TARs application [136],

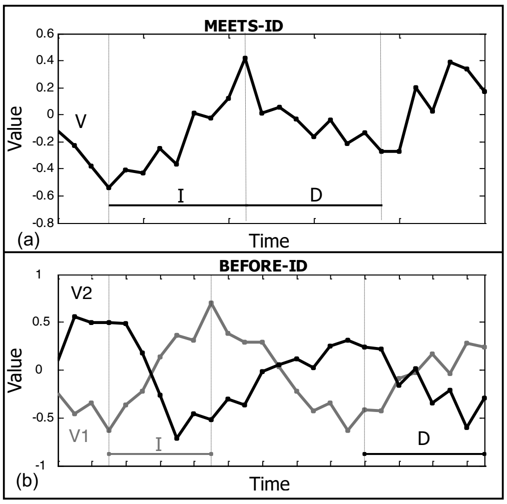


Figure 2-7: Complex TAs used to detect patterns of complex shape both **(a)** on a single time series MEETS-ID, and **(b)** on multidimensional time series BEFORE-ID: in this case an I (increase) episode in V1 occurs before a D (decrease) episode in V2 (from [155]).

where the authors describe the incorporation of frequent TARs as features to the Naive Bayes classifier for the coronary heart disease (CHD) diagnosis. The discovered TARs represent associations that combine symbolic time intervals using exclusively the temporal relation Precedes, which is a disjunctive temporal relation of a number of Allen's interval relations.

2.3 Temporal functional dependency (TFD) and Approximate functional dependency (AFD)

2.3.1 Functional dependencies

The concept of fuctional dependency [43] is defined as follows:

Definition 3 (Functional Dependency (FD))**.** *Let r be a relation over the relational schema $R(U)$: let $X, Y \subseteq R$ be sets of attributes of U. We assert that r fulfills the functional dependency $X \to Y$ (written as $r \models X \to Y$) if the following condition holds: $\forall t, t' \in r(t[X] = t'[X] \Rightarrow t[Y] = t'[Y])$*

Informally, for all the couples of tuples t and t' showing the same value(s) on X, the corresponding value(s) on Y are (or must, if we are specifying a constraint) identical.

Through the use of functional dependencies, we can express concepts such as: *"for each drug with a given symptom the disease does not change"*:

$$\textit{Drug, Symptom} \to \textit{Disease.}$$

2.3.2 Temporal functional dependency (TFD)

When considering temporal aspects, we talk about temporal functional dependency (TFD). In literature [181], several kinds of TFDs have been proposed, and various representation formalisms have been developed [176, 97, 174, 182, 180, 47], usually as temporal extensions of the widely known atemporal functional dependencies.

In [176], the authors present the notion of TFD which concentrates on temporal granularity and compares tuples valid during the same granule of some temporal granularity. They define a temporal functional dependency as:

Definition 4. *A temporal functional dependency (TFD) over U is an expression $X \rightarrow_{\mathcal{H}} Y$, where $X, Y \subseteq U$ and $\mathcal{H}$ is a granularity. A $\mathcal{G}$-relation I over U satisfies $X \rightarrow_{\mathcal{H}} Y$ if for each granule $H \in \mathcal{H}$, the relation $\{t[U] | t \in I, t(T) \subseteq H\}$ satisfies the FD $X \rightarrow Y$.*

In [47], the authors propose a new formalism for the representation of TFDs, involving multiple time granularities.

Let U be a set of atemporal attributes and VT be a temporal attribute, representing the valid time of a tuple. A temporal relation r is a relation on a temporal relation schema R with attributes $U \cup \{VT\}$ ($r \in R$). Given a tuple $t \in r$ and an attribute $A \in U \cup \{VT\}$, we denote by $t[A]$ the value that t assumes on A. The temporal attribute VT specifies the valid time of a tuple, and it takes its value over the time domain T, that is, $t[VT] \in T$.

They propose a new formalism, whereby a TFD is an expression of the following form:

$$[E - Exp(R), t - Group]X \rightarrow Y$$

where $E - Exp(R)$ is a relational expression on R, called *evolution expression*, $t - Group$ is a mapping $\mathbb{N} \rightarrow 2^{\mathbb{N}}$, called *temporal grouping* which specifies how to group tuples on the basis of the values they take on the temporal attribute VT , when $X \rightarrow Y$ is evaluated, and $X \rightarrow Y$ is a functional dependency.

Four different relevant classes of TFD have been identified:

- *Pure temporally grouping* TFD: E-Exp(R) returns the original temporal relation r. Dependencies of this class force the FD $X \rightarrow Y$, where $X, Y \subseteq U$, to hold over all the maximal sets which include all the tuples whose VT belongs to the same temporal grouping;

- *Pure temporally evolving* TFD: E-Exp(R) collects all the tuples modelling the evolution of an object. No temporal grouping exists: that is, the temporal grouping collects all the tuples of r in one unique set;

- *Temporally mixed* TFD: the expression E-Exp(R) collects all the tuples modelling the evolution of the object. The temporal grouping is applied to the set of tuples generated by E-Exp(R);

- *Temporally hybrid* TFDs: First, the evolution expression E-Exp(R) selects those tuples of the given temporal relation that contribute to the modeling of the evolution of a real-world object (that is, it removes isolated tuples); then, temporal grouping is applied to the resulting set of tuples.

In [122], the authors face another temporal aspect, which stems from the observation that frequent constraint violations in a database may be related to the fact that the considered (mini) world is changing, while the specified constraints remain static. FDs violated by current data are then identified and some approaches are proposed to suitably modify the given FD according to the new reality represented through the current data.

2.3.3 Approximate functional dependency

In many modern applications it might be necessary to extract properties and relationships that are not captured through FDs, due to the necessity to admit exceptions, or to consider similarity rather than equality of data values. More recently, AFDs have been included in the wider scenario of relaxed FDs (RFDs), where not only exceptions (i.e., violating tuples) are considered, but also similarities among attribute values and conditional constraints [36, 38].

In literature [107, 90, 91], the issue of discovering approximate functional dependencies from data has been largely studied.

Given a set of attributes U, a functional dependency over U is an expression $X \to Y$, where $X, Y \subseteq U$. If u is a relation over U, i.e., a finite set of mappings (called rows or tuples) from U to some domain, then the dependency $X \to Y$ holds in u, or u satisfies $X \to Y$, if all pairs of tuples that agree on X, agree also on Y. We say that a pair (u, v) of tuples of Y violates the dependency, or is a violating pair for it, if $u[X] = v[X]$ but $u[Y] \neq v[Y]$).

Given a relation u where an FD holds for most of the tuples in u , we may identify some tuples for which that FD does not hold. In [107], Kivinen and Mannila introduced three different measures, known as G_1, G_2 and G_3:

- G_1: it considers the number of violating couples of tuples. Formally:

  $$G_1(X \to Y, r) = |\{(t, t') \ : \ t, t' \in r \wedge t[X] = t'[X] \wedge t[Y] \neq t'[Y]\}|$$

  The related *scaled measurement* g_1 is defined as follows:

  $$g_1(X \to Y, r) = G_1(X \to Y, r)/|r|^2$$

  where $|r|$ is the cardinality of the relation r, i.e., the number of tuples belonging to the relation r.

- G_2: it considers the number of tuples which violate the functional dependency. Formally:

  $$G_2(X \to Y, r) = |\{t \ : \ t \in r \wedge \exists t'(t' \in r \wedge t[X] = t'[X] \wedge t[Y] \neq t'[Y])\}|$$

The related *scaled measurement* g_2 is defined as follows:

$$g_2(X \to Y, r) = G_2(X \to Y, r)/|r|$$

- G_3: it considers the minimum number of tuples in r to be deleted for the FD to hold. Formally: $G_3(X \to Y, r) = |r| - \max\{|s||s \subseteq r \wedge s \models X \to Y\}$. The related scaled measurement g_3 is defined as:

$$g_3(X \to Y, r)p = G_3(X \to Y, r)/|r|$$

Therefore, we can define a AFD as follows:

Definition 5 (Approximate Functional Dependency (AFD)). *Let r be a relation over the relational schema R: let $X, Y \subseteq R$ be sets of attributes of R. Relation r fulfills the functional dependency $X \overset{\varepsilon}{\to} Y$ (written as $r \models X \overset{\varepsilon}{\to} Y$) if $G_3(X \to Y, r) \leqslant \varepsilon$, where ε is the maximum acceptable error defined by the user.*

Among the several AFDs that can be identified over a relation r, the minimal AFD is of particular interest, as many other AFDs can then be derived from the minimal one. We thus define the minimal AFD as follows:

Definition 6 (Minimal Approximate Functional Dependency). *Given an AFD over r, we define $X \overset{\varepsilon}{\to} Y$ to be minimal for r if $r \models X \overset{\varepsilon}{\to} Y$ and $\forall X' \subset X$ we have that $r \not\models X' :\overset{\varepsilon}{\to} Y$.*

Through the use of approximate functional dependencies, we can express concepts such as *"for each drug with a given symptom the received diagnosis does not usually change"*:

$$\textit{Drug, Symptom} \overset{\varepsilon}{\to} \textit{Diagnosis.}$$

In [38], the authors present an overview of the most important RFDs, focusing on the relaxation criteria. As you can see in Figure 2-8 , they divide the different proposals in two main categories: *attribute comparison* using approximate matching paradigms to compare the attribute values on the left-hand side (LHS) and the implied attribute values on the right-hand side (RHS); *extent*, which indicates whether an RFD is satisfied by a subset or all the tuples.

Since our proposal is more similar to the FDs relaxing on the *extent*, in this section we discuss only the literature proposals related to the extent. All these proposals differ on the method used to specify the subset of tuples for which the RFD is satisfied, the coverage measure. In general, they use a different coverage measure defined as:

$$\mathbb{D}_{TRUE} : X_{EQ} \xrightarrow{\Psi_{err(0)}} Y_{EQ}$$

where Ψ represents the coverage measure, and EQ is the equality constraint.

The first proposal is the *purity dependencies* (PUDs) [168], which generalize the canonical FDs based on the notion of impurity measure. Since the attribute values of a database relation induce a partition on the set of tuples, we can observe that a canonical FD $X \to Y$ induces two partitions on attribute sets X and Y, respectively, without impurity. On the other hand, when the impurity induced by the attribute sets X and Y is greater than 0, but below a given threshold, a PUD can be used. Formally, let r be a database instance, X and Y be two sets of attributes, π_X and π_Y be two partitions of the set of tuples of r induced by the values of X and Y in r, respectively (i.e., π_X and π_Y correspond to the groups returned by SQL clauses group by X and group by Y). A PUD is defined as:

$$\mathbb{D}_{TRUE} : X_{EQ} \xrightarrow{\Theta(\pi_X,\pi_Y)\leq\epsilon} Y_{EQ}$$

where Θ is a concave and subadditive function that computes the largest impurity measure on the blocks in π_X relative π_Y.

Another proposal is the *numerical dependencies* (NUDs). They are FDs relaxing on the extent by means of a cardinality constraint. Given a relation R, and $X, Y \subseteq attr(R)$, specifies that each tuple $t[X]$ is associated to at most k different tuples on Y , for some constant k. Formally, a NUD on a relation R is defined as:

$$\mathbb{D}_{TRUE} : X_{EQ} \xrightarrow{card(X,Y)\neq k} Y_{EQ}$$

where $card(X, Y) = |\pi_Y(\sigma_{(X=t[X])}(r))|$, and EQ is the equality constraint..

Constrained functional dependencies (CDs) [115] is another type of RFDs, which generalizes traditional dependencies, such as functional dependencies, by expressing that the dependency applies not to an entire relation, but to a subset of the tuples in the relation described by a constraint. Formally, a CD on a relation R and class of constraints $\mathcal{L}$, has the form:

$$\mathbb{D}_c : X_{EQ} \xrightarrow{\Psi_{err(0)}} Y_{EQ}$$

where $X, Y \subseteq attr(R)$, $\mathbb{D}_c \subseteq dom(R)$ represents the tuples satisfying the constraint $c \in \mathcal{L}$ with variables from $attr(R)$.

A specialization of CDs, are the *conditional functional dependencies* (CFDs) [27]. They use conditions to specify the subset of tuples on which a dependency holds. Conditions are less general than CD constraints, since they only enable the specification of constraints based on the equality operator.

$$\mathbb{D}_{T_r} : X_{EQ} \xrightarrow{\Psi_{err(0)}} Y_{EQ}$$

where $\mathbb{D}_{T_r}$ is the domain of values satisfying a pattern tableau T_r with attributes in X and Y.

Another example of approximate functional dependencies are the *pattern functional dependencies* (PFDs) [147]. These are born from the necessity of relaxing the traditional integrity constraints which work on the entire attribute values. Relaxing the constraints of FDs operating on entire attribute values, they introduce a new type of dependencies 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

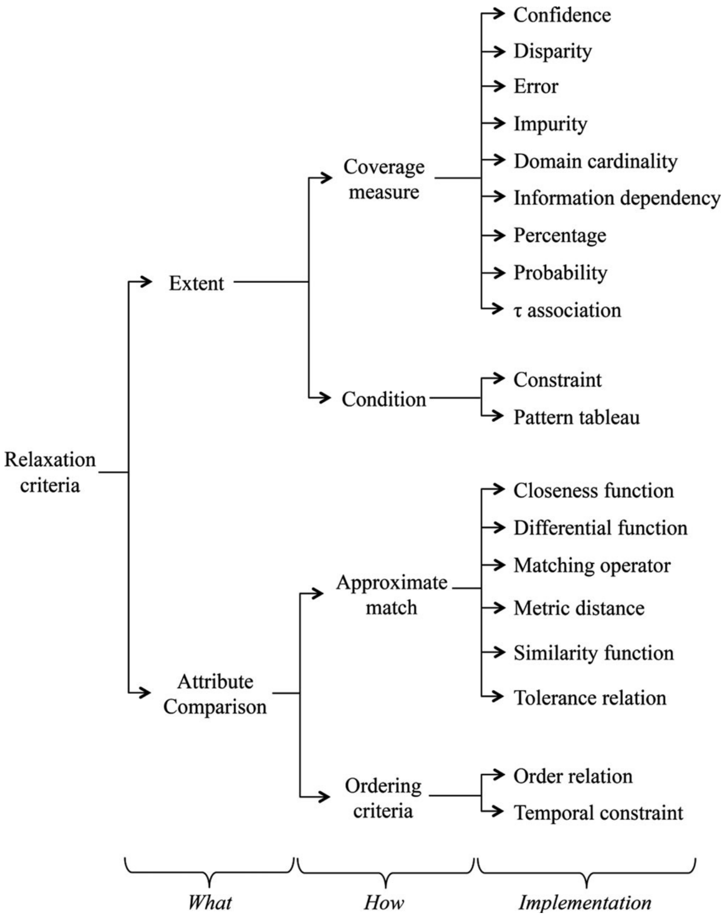


Figure 2-8: Characteristics of relaxation criteria for RFDs from [38]

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 $R(X \to Y; T_p)$, where:

- X and Y are sets of attributes from R;
- $X \to Y$ is a standard FD, called an embedded FD
- 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.

Frequent constraint violations in a database may be related to the fact that the considered (mini) world is changing, while the specified constraints remain static. Therefore it becomes necessary to identify the FDs violated by the current data and propose some approaches to appropriately modify the FDs according to the new reality represented by the current data.

In [122], considering possible mutations of the original database over time, the authors aim at modifying the integrity constraints so the semantics of the database will adhere as much as possible to the changing reality. The goal of their method was firstly to understand which FDs are violated, and secondly to repair these FDs by adding attributes to the antecedent of the dependency. Given a relation schema R, an instance r of R, and all the FDs defined on it, for each FD $F : X \rightarrow Y$, we compute its confidence. If the result is lower than 1 then the FD is not satisfied and, to repair it, we look for a set of attributes U in $R\ XY$ such that, if added to the antecedent of F, generate a new dependency $FD\ F^U : XU \rightarrow Y$ whose confidence is 1.

Regarding the dynamism of the data, but not related to the approximation of FDs, in literature we find another example [159] where the authors deal with the problem of continuously discovering FDs on dynamic datasets in an efficient way, and propose an incremental approach to solve it. They propose D$_{YN}$FD, the first algorithm that maintains the complete and exact set of minimal, non-trivial FDs on dynamic data, monitoring any changes in the data (insertion, update, deletion), and incorporating them as batches into sets of minimal functional dependencies.

2.3.4 Approximate temporal functional dependency

To conclude the panorama, moving from the definitions of FD, TFD, and AFD, we need to introduce the concept of approximate temporal functional dependencies (ATFDs). In [46], the authors introduce for the first time the ATFDs, which are defined and measured either on temporal granules or on sliding windows, and apply them to mine data from psychiatry and pharmacovigilance domains. In addition, they introduce a new error measure G_4, which considers the minimum number of tuples in r which must be modified for the plain TFD to hold on all the tuples of r.

According to the taxonomy previously defined, they considered pure temporally grouping TFDs of the form $[r, t - Group]X \rightarrow Y$, where $t - Group$ consists of *granularity (Gran) or sliding window (SW) grouping.* A temporal *granularity* is a partition of a temporal domain in indivisible non-overlapping groups, i.e., granules, of time points: minutes, hours, days, months, years as well as working days are *granularities* [45].

Definition 7 (Grouping by Gran(i))**.** *Two tuples $t_1, t_2 \in r$ belong to the same temporal group Gran(i) iff $t_1[VT], t_2[VT] \in$ Gran(i) where Gran(i) is the i^{th} granule of granularity Gran.*

A sliding window $SW(i, k)$ includes all the time points in interval $[i....i + k$ -1]. Thus, once we fix the length of the SW over relation r (i.e., k in the example), every SW over r will feature that length, and will - at most - include k elements (if relation r has tuples for all the time points of interval $[i...i + k$ -1]).

Definition 8 (Grouping by $SW(i, k)$)**.** *Two tuples $t_1, t_2 \in r$ belong to the same sliding window $SW(i, k)$ iff $t_1[VT], t_2[VT] \in [i...i + k$ -1].*

They evaluate a new error measure G_4, which considers the minimum number of tuples in r which must be modified for the plain TFD to hold on all the tuples of r.

$$G_4([r, t{-}Group]X \rightarrow Y, r) = \min\{|s|\ |s \subseteq r \wedge ((r{-}s) \cup w) \models [r, t{-}Group]X \rightarrow Y\}$$

where the set w is the minimal one for which the following formula holds:

$$\forall t \in s(\exists t' \in w(t[U - Y] = t'[U - Y] \wedge t[VT] = t'[VT]))$$

The related scaled measurement g_4 is defined as

$$g_4(X \rightarrow Y, r) = G_4(X \rightarrow Y, r)/|r|$$

The ATFD with different temporal grouping are defined as follows:

Definition 9 (ATFD with Gran grouping)**.** *Let r be a relationship over the relational schema R ($U \cup \{VT\}$): let $X, Y \subseteq U$ be attribute sets of R. Let Gran be the reference granularity, $[r, Gran]X \overset{\varepsilon}{\rightarrow} Y$ holds on relation r if the introduced error $g_3([r, Gran]X \ \rightarrow Y, r) \leq \varepsilon$ is less than the given threshold ε.*

A temporal granularity is a partition of a temporal domain in indivisible non-overlapping granules.

Definition 10 (Minimal ATFD with Gran grouping)**.** *An ATFD in the form of $[r, Gran]X \ \overset{\varepsilon}{\rightarrow}\ Y$ is said to be minimal for r iff $r \models [r, Gran]X \ \overset{\varepsilon}{\rightarrow}\ Y$ and $\forall X' \subset X$ we have that $r \not\models [r, Gran]X' :\overset{\varepsilon}{\rightarrow}\ Y$.*

Definition 11 (ATFD with SW grouping)**.** *Let r be a relationship over the relational schema R ($U \cup \{VT\}$): let $X, Y \subseteq U$ be attribute (sub) sets of R. Let $\{i...i + k - 1\}$ be a sliding window (SW) of length k. The approximate temporal functional dependency $[r, \{i...i + k - 1\}]X \ \overset{\varepsilon}{\rightarrow}\ Y$ holds on relation r if the introduced error $g_4([r, \{i...i + k - 1\}]X \ \rightarrow Y, r) \leq \varepsilon$ is lower than the given threshold ε.*

A sliding window (i, K) includes all the time points in interval $[i...i + k - 1]$ where k is the sliding window length. Fixed a length of an SW over a relation r, every SW contains at most k elements.

Definition 12 (Minimal ATFD with SW grouping)**.** *Given an ATFD over $[r, \{i...i + k - 1\}]$, we define $X \ \overset{\varepsilon}{\rightarrow}\ Y$ to be minimal for r iff $r \models [r, \{i...i + k - 1\}]X \ \overset{\varepsilon}{\rightarrow}\ Y$ and $\forall X' \subset X$ we have that $r \not\models [r, \{i...i + k - 1\}]X' \ \overset{\varepsilon}{\rightarrow}\ Y$.*

Another example of ATFDs is [2], where the authors present AETAS, a system for the discovery of approximate temporal functional dependencies. The discovered TFDs are mainly pure temporally grouping TFDs with moving windows, according to the classification proposed in [46]. They mine the duration that lead 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. In addition, they consider rules with constants (similar to conditional functional dependencies) such that specific durations can be used for specific entities, where the moving window may have different values according to specific values of atemporal attributes. The authors applied these in two different clinical domains: the first one referred to psychiatry, collecting data about contacts between patients and psychiatrists, psychologists, and social workers; the second was pharmacovigilance, collecting data about drug administrations and adverse reactions.

The first domain concerned the Verona Psychiatric Case Register (PCR). The National Health Service in trust with the University of Verona offers a public Community-based Psychiatric Service (CPS), providing psychiatric care to mentally ill as well as psychological care and responses to social needs. Data about patients are collected in the information system PCR, which has recorded information about patients' accesses to this service since 1979. PCR contained patients' personal data, patients' medical record, contact information, records education, employment, professional status, type of accommodation, and marital status. PCR is used as a basis to evaluate the direct management costs for groups of patients, and to monitor the effects coming from changes in resources, organization, and needs. The clinical purposes include monitoring of patients to plan future contacts at regular time intervals, and providing clinicians with reports about admissions and contacts for every patient in a given time period.These temporal data can then be used by psychiatrists, e.g., to identify the number of contacts in different time periods with respect to different factors such as age, diagnosis. A meaningful example discussed in this context was: $[133days]HealthStructure \rightarrow ContactType$.

The second one is Pharmacovigilance (PhV) that collects, analyzes, and prevents adverse reactions induced by drugs (ADR). The spontaneous reporting of ADRs identifies unexpected reactions and informs the regulating authority about them. This practice is valuable, provides early warnings, and requires limited economic and organizational resources. It also has the advantage of covering every drug on the market and every category of patient. PhV considers possible relationships between one or more adverse reactions and one or more drugs, mainly focusing on unknown or completely undocumented relationships. Reports suggest a cause-effect link among ADRs and drugs. Each report includes patient's information and the description of the occurred adverse relation. These temporal data are used to investigate any cause-effect relationship among drugs and reaction(s) in different time periods, or according to the time frame of the exposure. A meaningful example discussed in this context was: $[30days]Drug, AdverseReaction \rightarrow Outcome$.

Chapter 3

Towards Explainable Artificial Intelligence in Medicine

This chapter is mainly based on "A manifesto on explainability for artificial intelligence in medicine" [44], which aim is explore in depth the concept of explainable AI, offering a functional definition and conceptual framework or model that can be used when considering XAI. This is followed by a series of desiderata for attaining explainability in AI, each of which touches upon a key domain in biomedicine. To conclude, considering that one of the central themes in this thesis is "time", we discuss the meaning of *temporality* in XAI context, explicitly declining different aspects of it.

3.1 Introduction

There is considerable discussion in the biomedical informatics and computer science communities about the "un-explainable" nature of artificial intelligence (AI), in that much is made of so-called "black-box" algorithms and systems that leave users, and even developers, in the dark as to how results were obtained. As a result, there is growing skepticism about the potential limits of AI, even in the face of burgeoning interest that at times reflects over-optimism about it. At the same time, there is a growing community of researchers who are working to address this skepticism through their work in making AI explainable, and thus useful and potentially usable to those who employ AI in their work. This is especially welcome in the domain of biomedicine, where explainable AI is critically important for clinicians in their daily practice.

As AI (including Machine Learning) becomes increasingly ubiquitous, there are growing concerns and questions, such as:

- How does an AI algorithm work - what is it doing?

- Does an AI system work as well as an expert?

- Does an AI system do what a user would do, were she in the same situation?

- Why can't the system tell a user how it arrived at a conclusion or made a decision?

These concerns are of urgent importance and need to be addressed with scientific and engineering rigor in a variety of biomedical domains, including clinical decision support systems, patient monitoring, public health surveillance, and biomedical research. However, we in the informatics community are uniquely positioned to take leadership roles in developing and implementing strategies for improving the explainability of AI systems.

The primary goal of this paper is to present a compelling case for the need to address gaps in the explainability of AI software and the results presented to users. We hope to meet this goal by means of a rigorously developed conceptual model for thinking about explainable AI, or *XAI*, through a thorough exposition of the work to date and identification of gaps in research and application of XAI, and a proposition for how these gaps could be addressed. Even though many definitions and concepts we will introduce and discuss are general and may be applicable to many different domains, in the following we will focus on XAI in Medicine and Health. Indeed, these domains have special requirements that make XAI quite idiosyncratic and worthy of particular attention.

We have structured this chapter as follows: after an introduction to the problem of explainability, in Section 3.5 we illustrate how temporal elements can ba part of XAI models; in Section 3.2 we discuss some background on how informatics and computer science describe the problem, approaches to explainability, and applications of XAI to a variety of key clinical domains; Section 3.3 contains a proposal for a conceptual framework and foundational definition of XAI; Section 3.4 presents a set of desiderata that would be important to address XAI moving forward; finally, Section 3.6 sketches some conclusions and future directions.

3.2 A research field's description of the current landscape of AI

In this section, we will briefly introduce the main aspects that have been discussed about XAI in general, in the areas of Computer Science and Artificial Intelligence. Then, we will move to the main specific issues of XAI in Medicine, ending with some non-exhaustive examples of XAI approaches in clinical domains.

The concept of explainability has a long story in AI. Indeed, since the first proposals of the so-called "expert systems", there was the need of having an explanation of why and how some conclusions were reached by the system in a complex decision-support task. Such a requirement was, and remains, extremely important in medicine, as physicians needed to understand why the system was proposing, for example, a specific diagnosis or treatment regimen. The need of having some explanation about the output, an AI-based system provides, has recently been exacerbated by the adoption of machine learning (ML) approaches, where the reasoning task is often performed by “black-box” systems that do not allow one to understand clearly why a specific result has been reached [111].

In principle, explainability is related to understanding, i.e., having a mental model of what we are observing. With a slightly different terminology, we may say that explaining/interpreting consists of providing causes of observed phenomena in a comprehensible manner through a linguistic description of its logical and causal relationships [85, 111]. In the context of XAI, we need to understand the conclusions of a system that is reasoning on some data to reach some result. Such systems in medicine are often related to a decision-support task, where data may be incomplete, uncertain, ambiguous, or missing. Moreover, such data have a high complexity and heterogeneity, being expressed as often interrelated and intertwined data in various formats such as structured, semi-structured, or unstructured alphanumeric data, movies, images, sounds, waveform signals, and so on.

Methods proposed to support explainability are often divided into *ante-hoc* and *post-hoc* approaches. *Ante-hoc* approaches are related to systems that allow one to directly understand their mechanisms in providing a result such as a conclusion (e.g., a diagnosis) or a recommendation (e.g., a treatment option). Decision trees, rule-based models, and linear approximations are, for example, commonly considered to be implicitly explainable. Post-hoc approaches try to provide some explanation to the results reached by ML models, such as those based on deep neural networks, random forests, support vector machines, and many others. Post-hoc approaches are, in principle, applicable to different kinds of AI systems. The difference between these two approaches is that post-hoc approaches are not considered when designing a system, but deal with the extraction of explanatory information from an already existing system, which is usually based on ML “black-box” models. As we will see in this section, the distinction between post-hoc and ante-hoc approaches is sometimes subtle and has to be informed by further considerations.

Explainability is thus an inherently multifaceted concept, which still needs some more effort to have a precise characterization, also from the terminological point of view [111]. Let us now consider some dimensions of analysis that have been recently discussed in the literature.

The content of explanation: What is being explained? Independently from being either post-hoc or ante-hoc, XAI systems have to be specified and developed with respect to the subject of the provided explanation. Indeed, sometimes it is the reasoning mechanism itself that has to be explained. In this case, explanation focuses on the mechanics of the path that allowed the system to reach a specific result. Both generic and specific medical knowledge could be used to this regard. On the other side, explanatory information could be provided without any reference to the reasoning approach of the system, but focusing on deriving some form of association/relationship (causality) between data and corresponding results.

The stakeholders of explanation: Who needs explainability? Any kind of explanation needs to be tailored according to its recipients. It was recently highlighted that many possible stakeholders may be closely related to any XAI system [111]. In the medical and healthcare settings, among the possible stakeholders we consider a broad community of users, including clinicians, technicians, nurses, general practitioners, administrative staff, different kinds of students, healthcare policy makers, medical informaticians, and patients. The background knowledge of such stakeholders is often deeply different and often requires different user-centric solutions and techniques for a successful explanation.

The goal for explanation: Why is explainability required? Considering different stakeholders is not sufficient. We have to consider not only who is the recipient of the explanation, but also *why* the explanation is required. Indeed, the same stakeholder may have different motivations and requirements with respect to XAI systems. As an example, a physician may have different desiderata that include, variously, *education and experience*, *fairness*, *ethics*, *satisfaction*, *trust*, or *controllability*, while developers would consider *system acceptance*, possibly in addition to those required by a physician. Often such desiderata are not completely disjoint and may co-exist in a single XAI-system [111]. According to different desiderata, stakeholders could be looking for an answer to different questions related to explainability [85]: Why did the algorithm do that? Can I trust these results? How can I correct an error? Are data meaningful with respect to the required task?

The moment, the duration and the frequency of explanation: When, how long, and how frequently. A further, under-evaluated, issue is related to when and how frequently an explanation is requested of the system. Indeed, while naïve and occasional users often require frequent explanations at any stage of use of the supported AI system, experienced users who are supposed to use the system in the daily clinical routines, may require less frequent explanations, possibly focusing on rare or unexpected situations. The level of detail and thus the duration of the explanation may also be different, according to the specific needs of different stakeholders in different contexts, with different goals.

The modalities of explanation: How is explainability represented? Different choices are possible when deciding *how* to explain. A first option is to support *perceptive interpretability* [170]. This concept refers to interpretations that can be humanly perceived, (1) through the highlighting (often visual) of important input features with respect to a given output (*saliency*), (2) through the observation of the stimulation of neurons or groups of neurons (*signal interpretability*), and (3) through the composition of logical statements or sentences that can explain, even indicating causality (*verbal interpretability*). Often, perceptive interpretability is founded on an abstraction of the task at hand, which focuses on the most important aspects that explain the reached solution. Systems based on perceptive interpretability work with different techniques with respect to the ones used for the given task. For example, a fuzzy rule-based system may be coupled with an artificial neural network (ANN) system in diagnosing electrocardiographic (ECG) signals [30]. As for perceptive interpretability through visual and graphical systems, a widely acknowledged distinction exists between directly understandable data, which are visualized through one or two dimensional representations, and multi-dimensional representations, which are not directly understandable [85]. A second option is to consider *interpretability by mathematical structures.* In this case, either simple mathematical models are used, or different data-oriented approaches are used to highlight hidden features of data, such as data clustering, perturbations, data dependencies. Systems which support interpretability via mathematical structures consider outputs (which are ultimately perceptive) that require deeper cognitive processes and background knowledge, before being interpretable [170].

Further distinctions about the modalities of explanations supported by different XAI systems consider *model-agnostic approaches* and *model-specific approaches.* While the first approaches attempt to provide explanatory information only by observing input/output associations, model-specific approaches consider also specific features of the model under explanation [111]. A last aspect to consider for XAI systems is their *scope.* Indeed, some contributions focus on single predictions/classifications of the supported system (i.e., a single pair of inputs/output). Such systems have a *local scope* [3, 8], in comparison with other approaches that have a *global scope*, which are designed to explain the overall reasoning mechanism of the model.

Moving closer to applications in medicine, some aspects of AI have been identified that make XAI systems in medicine challenging but worthy of rigorous investigation. Factors as risk and responsibilities, accountability, and trustworthiness, even though already considered in non-medical domains, become here prominent and multifaceted. As an example, while explainability is a strong requirement in the clinical domain, as for acceptance, accountability, and legal compliance, a certain level of opaqueness can be acceptable for some clinical users, provided that some functional understanding of the model is supported, disregarding a possible low-level algorithmic understanding [85, 87, 170].

As XAI in medicine is in an early stage of investigation, some further issues have to be faced. Among them, the evaluation of XAI systems with actual end-users will help understand, represent, and satisfy user requirements [111]. *Causability* is the term proposed in [85] to explicitly highlight the need of measurements for the quality of explanations. In this direction, explanation interfaces have to make the results obtained through the explainable model both usable and useful to the considered stakeholder. Causability is thus a measure for the usability of such a human-AI interface.

All the previous arguments we discussed lead to a further, recently highlighted consideration [111]. Researching and developing XAI in medicine is an interdisciplinary task, which requires the active participation of different stakeholders, to cover different perspectives. Methodologies for the design of XAI systems in medicine would require skills from different scientific domains, such as AI, medical informatics, software engineering, medicine, healthcare, and cognitive sciences.

3.2.1 Applications

We find many examples in the literature of research activities that are devoted to exploring XAI in medical domains. Here we report some recent examples regarding different techniques.

ML algorithms such as neural networks are inherently non-explainable and are typically referred to as "black-box" models. However, there are some examples where neural network models can be shown to produce explanatory descriptions to support the interpretability of the output. In one study, the authors proposed a modular framework, CEFEs (CNN Explainability Framework for ECG signals), a post-hoc tri-modular evaluation structure that provides local interpretations and explanations from convolutional neural networks [121]. The evaluation of the model's capacity is performed through quantitative interpretability, where the metrics represent the features learned by the model. In addition, the visualization of the features allows visually correlating the features. Pennisi *et al.* employed a novel lung-lobe segmentation network to identify CT scans of COVID-19 patients and automatically categorize specific lesions [140]. They integrate the pipeline into a web application to support radiologists in the investigation of this disease.

In recent years, ensemble learning has achieved excellent results incorporating explainability. Yeboah *et al.* present an ensemble clustering-based XAI model for traumatic brain injury (TBI) prognostic and diagnostic analysis [183]. The goal is to identify patient subgroups and key phenotypes that delineate these subgroups using tomography data, exploring the features' relevance. In another example, the authors proposed an auxiliary decision support system that combined ensemble learning with case-based reasoning (CBR) to help physicians improve the accuracy of breast cancer recurrence prediction [72]. They use extreme gradient boosting (XGBoost) to predict the risk of breast cancer recurrence, and then use CBR to explain the reason for the prediction. Of note, they conducted a survey of 32 oncologists to assess the utility of the system as perceived by users, measuring the evaluation of the system through a questionnaire, leading to a positive assessment by the users of the system.

There are different examples of the usage of systems that exploit the explanations through rules-based systems extracted from medical data. They generated explanations in a human-understandable format, increasing the trust to believe the results given by the support system. El-Sappagh, *et al.*, proposed a system of fuzzy IF-THEN rules [63]. It integrates reasoning with fuzzy reasoning over an ontology. They proposed and implemented a new semantically interpretable fuzzy rule-based system framework for diabetes diagnosis that is able to provide accurate decision support as a result. Kavya *et al.* developed an Allergy Diagnosis Support System (ADSS) [103]. They applied several ML algorithms and then selected the best-performing algorithm using k-fold cross-validation. In terms of the XAI method, they developed a rule-based approach by building a random forest. Each path in a tree is represented as an IF-THEN rule, and these rules are stored in a rule base for expert assessment. Additionally, the authors developed a mobile application, which can assist junior clinicians in confirming the diagnostic predictions.

Although the user represents a central aspect in the approaches we have just seen, the creation of an explainable system to use in a particular context requires a multi-disciplinary collaboration, involving collaboration with the stakeholders. Schoonderwoerd *et al.* presented a case study of an application of a human-centered design approach for AI-generated explanations [160]. The approach consisted of three components:

(i) Domain analysis to define the concept and context of explanations;

(ii) Requirements elicitation and assessment to derive the use cases and explanation requirements; and

(iii) The consequential multi-modal interaction design and evaluation to create a library of design patterns for explanations.

They apply this system in the context of child health. Dragoni, *et al.* proposed an XAI system based on logical reasoning that supports the monitoring of users' behaviors and persuades them to follow healthy lifestyles [62]. In this case, the authors first assessed the usability of the application with questionnaires filled out by the user. Second, they validated the correctness of the explanation generated by the system. Finally, the last evaluation included an effectiveness analysis of the generated explanations.

3.3 Towards a foundational definition of XAI in Medicine

We propose a conceptual framework for XAI that captures the intersection of four characteristics that are typical of any information system, statistical model, or software application. These characteristics are *Interpretability*, *Understandability*, *Usability*, and *Usefulness*, respectively. *Interpretability* is the degree to which a user can intuit the cause of a decision and thus the ability of a user to predict a system's results [151]. *Understandability* is the degree to which a user can ascertain how the system works, and leads directly to user confidence in the system's output. *Usability* is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component. Usability thus asks the question "Can one use the system easily?". *Usefulness*, on the other hand, asks the question "Will one use the system because it meets a user's needs?", and is seen as the practical worth or applicability of a system. A system is unlikely to be useful if it is not usable, however. As a result, usability is generally a first-order requirement of any information system or software application [110].

However, when it comes to AI, we are not talking about *any* information

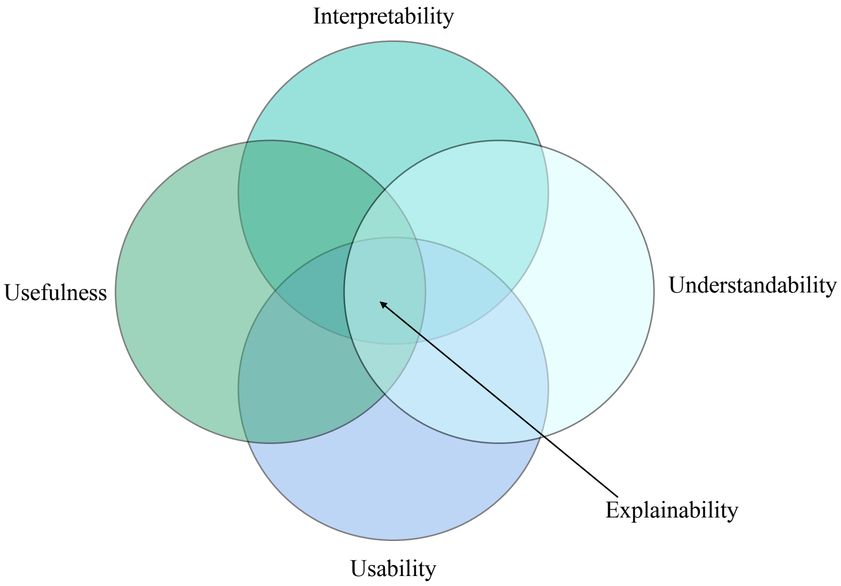


Figure 3-1: The Venn diagram of Explainability as intersection of Usability, Usefulness, Interpretability, and Understandability.

system. Rather, AI systems and applications typically realize some kind of reasoning task, to support some kind of decision-making, such as proposing a clinical diagnosis or controlling a task in an engineering operation, or to derive new knowledge/information in some specific context, as mining hidden patterns in patients' clinical histories. Perhaps unique to AI applications, we need two additional dimensions in order to realize an ability to provide user confidence that the decision was correct, but even more so, the ability for a user to ascertain how the system works. Thus, we propose that *understandability* is one such dimension, and furthermore that it is, in our framework, complementary to usability. That is, usability is enhanced via understandability: an AI application that is understandable is more likely to be usable.

The first characteristic of AI systems that we consider to be central to our framework is *interpretability*, which we construe as the degree to which a user can intuit the cause of a decision; in addition, it is the degree to which a human can consistently predict a model's results, based on her experience with the application. Just as understandability and usability are complementary, we propose that interpretability and usefulness are complementary as well. For example, a user of an AI application is more likely to find it useful, something that would meet her needs for a given purpose, if the result or decision made by the application is interpretable in the face of a real-life contingency.

This framework is illustrated as a Venn diagram in which these four characteristics overlap various points of articulation, but most importantly in the center, where all are needed when considering explainability, as is shown in Figure 3-1.

Where these four characteristics intersect is that smallest, yet richest, segment of the Venn diagram, *explainability*. Due to the intersectionality of the four characteristics just described, explainability is a complex concept. It is not merely a characteristic of the model, but rather something that emerges from the intersection of the four characteristics we addressed here. As a result, we maintain that it is best to describe explainability in a multidimensional way through addressing a series of seven questions through the lens of others who have worked extensively in this domain.

The proposed foundational definition of XAI does not explicitly contain any specific reference to the medical and health domains. Indeed, the concepts introduced here are general and can be applied to any domain. However, we would stress here that, to the best of our knowledge, the definition of explainability as the intersection of four different characteristics is both original and particularly well suited for medicine and health AI.

As for the novelty of our definition, we identify here two different aspects: (i) from one general side, we explicitly distinguish the concepts of *interpretability*, *understandability*, and *explainability*. Such distinction is not clearly discussed in the existing literature, where, for example, *interpretable* and *explainable* are often taken as synonyms (see, for example, [73, 111, 170, 120]). On the other hand, we explicitly introduced *usability* and *usefulness* as first principles of explainability. Such user-oriented aspects of explainability, even though considered and highlighted in the considered literature, have not been discussed as main component of a complex concept as complex as that of explainability.

The highlighted novelty of our foundational definition is also the leverage for making it especially well-suited for medicine and healthcare. Indeed, in our view, Medicine and healthcare are characterized by some specific features, which need to be considered as central for XAI. The first feature consists in the presence of *distributed, heterogeneous decision-making tasks* and a second can be defined as *knowledge-intensive domain*. The presence of *distributed, heterogeneous decision-making tasks* and of the corresponding XAI systems justify the presence of usability and usefulness in the definition of explainability. Indeed, usability and usefulness have to be evaluated according to different users and tasks. They are not absolute concepts and need to be assessed "on the field". The usability of systems that have to be adopted by specialized physicians in some intensive clinical setting requires it to be evaluated by the pertinent clinical stakeholders, while, for example, the usability of XAI systems supporting the communication and shared decision-making among clinicians, general practitioners, and patients (e.g., in a web app supporting the mental health monitoring of home patients) should be suitably assessed according to different explainability requirements, corresponding to different background knowledge and roles of the involved stakeholders.

Moreover, such *knowledge-intensive* and decision-intensive tasks require one to distinguish between interpretability and understandability. Indeed, while the concept of interpretability is related to the capability of predicting a system's result, even without being aware of the "internal" structure and functioning of the system, understandability refers to the capability of being aware of how the system works. In many intensive decision-based tasks, such as the prompt reaction to some unexpected change in an ICU patient's condition, the interpretability of an AI-based system may emerge as an indispensable feature. Indeed, the clinician has to be able to recognize how recorded vital signs are related to the alarms triggered by an AI-based system. It is worthwhile to stress that interpretability does not mean that the AI-based system is not important or useful as the user is able to predict the system's result. Indeed, the capability of predicting the system's result, does not mean that a human can process all the required data in an acceptable way, according to the requirements either related to the number of patients to consider or to the real-time results.

On the other hand, interpretability alone is often not sufficient to attain a necessary level of explainability. Understandability requires that the stakeholders have to be able to understand how the AI-based system works. In many medical and healthcare AI-based systems it may be important to have a deep understanding of the system internal behavior, in a way comprehensible to the specific clinical stakeholder. Let us continue with the example of an AI-based system for patient monitoring in ICU. While the AI-based system supporting real-time monitoring requires some kind of interpretability, the same AI-based system in the reporting and data analytics part could require more explicitly some kind of understandability. Indeed, when doing off-line data analysis it may be important to understand how the system is able to derive even unexpected results. As these results have to be related to existing and evolving medical knowledge, a deep comprehension of system technicalities and behaviors would also support a suitable elicitation of new medical knowledge.

3.4 Questions, propositions, and desiderata in the quest to attain XAI in medicine

After the proposal of our foundational definition of XAI in Medicine, supported by some simple examples in clinical domains, let us now move to more concrete issues that are necessary to consider in the practical development and use of (explainable) AI-based systems in medicine and healthcare. In the following we will touch on several different issues. After considering the design of XAI systems in Medicine (What are the requirements for XAI? How can we evaluate the goodness of the provided explanation?), we will introduce some further motivation supporting the distinction between understandability and explainability (If an AI system's output is understandable, is it automatically explainable?). Then, we will deal with the importance of modeling the considered medical domains (What is the role of domain understanding in achieving XAI in medical applications?). We will then continue with some more abstract aspects, as they relate to the evolution from data to wisdom through explainability (Can explainability draw us closer to wisdom?), to (Can an AI system that is not explainable be trustworthy?) and that connecting explainability and trustworthiness (Can an AI system that is not explainable be trustworthy?). We will end this section by answering the (usually hidden) question: Is XAI in medicine always required?

The questions we will deal with in this section complement the foundational definition we proposed in the previous section and apply such definition with respect to real-world aspects of XAI in clinical contexts.

3.4.1 What are the requirements for XAI? How can we evaluate the goodness of the provided explanation?

Proposition: There are tangible, instantiable, user-centered requirements that must be met in order to achieve an XAI system; more specifically, there is the need to measure, interpret, and understand usability vs. usefulness, and interpretability vs. understandability, and how those two relate to each other in the context of use and users, particularly in the context of AI in medicine.

Similar to any information system, systems that employ AI can and should be developed and evaluated using state-of-the-art methods that can be extended to the domain of explainability. While validation and verification have been part of the canon for evaluating AI systems for several decades, these focus on operability and the accuracy of knowledge representation and inference. However, neither validation nor verification have fully taken into account the explainability or interpretability of the results from a user's perspective. Proposed here are desiderata in two broad domains of requirements for XAI that would serve to further the development of AI systems that help users to understand how such systems reach conclusions or offer advice. These domains are linking the cognitive to the explainable, and the evaluation of explainability.

- *Linking the cognitive to the explainable: the role of theory.* Knowledge elicitation has long been the central purpose of knowledge engineering, but it focuses on developing a knowledge base that does not address the needs of users as they interact with an AI system. This lacuna is especially evident with regard to the user interface. It is argued here, and supported in the literature, that *qualitative inquiry driven by theoretical frameworks* is needed to develop user-centered interfaces for ML in healthcare applications [15]. Theory-driven user interface design that takes into account the cognitive and behavioral aspects of users is foundational to achieving true explainability. This extends traditional principles of user interface design to include aspects of what influences user interpretation. Such aspects include attitudes and beliefs that may bias interpretability and subsequently influence users' confidence and understanding of the system and its results. In their recent survey of models for achieving explainability, Markus *et al.* provide a framework for choosing the type of explainable interface between model-, attribution-, or example-based explanations [120]. They advocate for methods for achieving explainability that are sensitive to the requirements of the problem domain, and that these should drive the choice of approach, rather than enforcing a single paradigm of explainability. In a word, they call for an *agile* approach to attaining and evaluating explainability, which is very much in line with the best practices of information system development in general. One agile approach to attaining explainability in AI systems turns to fuzzy set theory and its application to fuzzy reasoning systems. Such systems provide a plausible paradigm for modeling explainability, since natural language is one defining characteristic of fuzzy systems. Alonso Moral *et al.* argue for this paradigm, showing how user-centered explainability is connected to fuzzy modeling [124]. Finally, any effort to establish explainability needs to be linked to the cognitive aspects of human inference. It is arguable that there is no more urgent need for this in medical decision making. An example of this kind of cognition is seen in the principle of *ex adiuvantibus*, which is the inference leading to a conclusion, such as the cause of a diseases, that is based on evidence that the disease responded to a treatment. As an example, one might infer that a migraine headache was caused by exposure to a specific allergen because an antihistamine was shown to prevent the headache. Such causal inferences many or may not be correct in practice, but they are made frequently in clinical practices, and in fact this type of reasoning is at the heart of allopathic medicine.

- *A user-oriented perspective of explainability.* The growing research community in XAI has already developed a number of highly successful XAI methods [185]. Explainability in this context highlights technically decision-relevant parts of machine representations and machine models. For example, parts that contributed to model accuracy during training or to a particular prediction are visualized by a heatmap, a good and proven example being the very well known Layer Wise Relevance Propagation (LRP) method [127]. However, this visualization does not refer to a human model. For this purpose, the concept of causability was introduced, which is defined as the measurable extent to which an explanation reaches a certain level of causal understanding for a human end-user [85]. Since this concept refers to a human model, it can be used very well to design and evaluate future human-AI interfaces [87]. These future Human-AI interfaces must provide a successful mapping between Explainability and causability and foster contextual understanding and allow the expert to ask questions and counterfactuals (”what-if” questions) [86]. At the same time such question-answer interfaces can make use of a human-in-the-loop, who can bring human experience and conceptual knowledge to AI processes - something that the best AI algorithms available still lack. An example that is important for medical AI is the classification of entities into several classes, where typically, taking into account the uncertainty about the membership of the classes, entities are classified as “yes”, “no”, or “maybe”. However, in doing so, it is desirable – especially in medical problems – to indicate the propensity or probability of a classification to belong to a single yes or no category. Neural networks have proven their high performance in crisp classification, however, as we know, the solution is not comprehensible and therefore difficult or impossible for a human expert (e.g. a physician) to interpret and understand. Rule-based systems are in principle explainable, however they are based on formal inference structures and also have problems with interpretability due to their high complexity. We must emphasize that even human experts sometimes cannot explain, but construct mental models of the problem and use these models to select the best possible solution. Hudec *et al.* propose a classification by aggregation functions of mixed behavior through the variability of ordinal sums of conjunctive and disjunctive functions [89]. In this way, domain experts should assign only the most relevant observations regarding the considered attributes. Consequently, the variability of the functions provides room for ML to learn the best possible option from the data. Such a solution is tractable, reproducible and explainable to domain experts.

- *Evaluating explainability.* Ultimately, explainability is in the eye of the beholder, i.e., the user. As such it is incumbent on those who aim to develop XAI systems to account for their usability, but also their usefulness. *Usability* can be measured using modifications of such instruments as the System Usability Scale (SUS). A detailed retrospective examination of the SUS is provided in [33]. Modification to this scale would need to account for the interpretability of the system, including both inputs and outputs. Another approach to usability assessment is one that focuses on causality [85, 83]. This approach allows users and developers to trace inferential pathways and evaluate them for plausibility. As such, not only can inferential errors be identified rapidly, the reasoning behind them can, as well. Using this scale, a deep assessment of usability can be obtained throughout the system development life cycle. Yet another approach to assessing usability focuses on user-centered reporting of results, such that users provide important input on and influence over what is reported by the system. This was shown to be an effective way to ensure that random forest results were reported in a way that users found them to be interpretable [141]. However, none of these approaches to evaluating explainability address the issue of *usefulness*. While a system may be usable, it is not necessarily useful, meaning that the system addresses some important task, telling a user something they did not already know or infer from available facts or knowledge. To assess usefulness, one needs to turn to long-term, post-hoc qualitative and quantitative evaluation of how, when, and why the system is being used and in what contexts does it fit (or fail to fit) workflows. Another consideration for usefulness is whether or not a system is used in practice to replace another. This is especially important in busy clinical settings, where AI systems might be used to augment medical decision making. However, if a system is not useful, practitioners will not use it, even though it might be very usable, or they will use the system but develop workarounds to make it more useful, sometimes with consequences that are potentially catastrophic to patients. For this type of evaluation, the frameworks mentioned above can inform the development of strategies and methods for observing the use of AI systems in these contexts in real-time, and the framework-driven analysis of data obtained during this endeavor.

3.4.2 If an AI system's output is understandable, is it automatically explainable?

Proposition: Understanding the output from an AI system is foundational to explainability, but it is only one requirement that has to be merged with usability, usefulness, and interpretability to compose explainability.

A central goal of ML is to build a model which summarizes linear and/or nonlinear patterns in a dataset. Good models are useful for making predictions in new data and thus have the quality of generalizability, which in turn makes them useful. Most ML models, such as those derived from neural networks or gradient boosting, have an underlying mathematical foundation. For example, a neural network model can be written as a summation of products of weights and inputs from data and hidden layer nodes. Thus, our knowledge of the mathematical foundation of a model makes it inherently understandable in that we know the function that relates the data inputs to the outcome being predicted. Our understanding can be improved by conducting experiments on the model by, for example, perturbing inputs and/or model components to observe their effects on model quality metrics. We can even decompose the model into linear and nonlinear components using these kinds of perturbation experiments when combined with entropy-based measures from information theory, for example. In this way, it is possible to gain a good understanding of a model. But does understanding translate to explainability?

As previously described, characteristics of XAI include usability, usefulness, interpretability, and understandability. Knowing the mathematical basis of a model does not necessarily make it useful. For example, a neural network model might do a good job of predicting 30-day hospital readmissions following surgery. Further, the model might generalize well to clinical data from other hospitals. The model is understandable because the mathematical basis is known and can be described. Although the model is predictive and understandable, it might not be useful for reducing readmissions if the features include patient demographics such as gender and zip code which can not be changed to improve the outcome. As another example, consider a neural network model relating gene expression features to risk of disease, where the predictive features include a number of housekeeping genes required for the maintenance and function of all cells. The model might be understandable and useful, but it might not be interpretable. In other words, it may be difficult for the domain expert to come up with an explanation for why this set of genes contributes to disease risk when they impact every cell in the body. This in turn would limit the ability of a pharmacologist to develop a therapeutic intervention.

Understanding an ML model is thus a first step toward XAI. While complementary, usability, usefulness, interpretability, and understandability can be synergistic. For example, a domain-specific knowledge graph can make a model more understandable and more interpretable by informing the user of biological relationships among the features [125]. Further, biomedical ontologies can facilitate both understanding and interpretation because the feature relationships have been described through a synthesis of multiple knowledge sources that capture their semantic meaning [56].

3.4.3 What is the role of domain understanding in achieving XAI in medical applications?

Proposition: XAI-based systems need to start from modeling the biomedical and clinical domain in order to obtain a true understanding of the context in which these systems will be used.

As stated by several authors, a key aspect of building biomedical (and in particular clinical) AI-based systems is to understand the context. For example, understanding the context of clinical decisions means to model the patients' careflow: identify the key actors of care and the decision-makers, explicitly define the timing of decisions, and clarify the data collection phases and their critical elements, including the potential sources of missing data. Only by deeply analyzing all these aspects it will be possible to design a successful AI-based system and to properly identify the explainability components. The real importance of an AI system in medicine is to support the planning and delivery of medical treatment more than just perform diagnostic labeling [5]. XAI is essential to achieving this goal, in addition to the strategies to induce trust in AI-supported decisions.

To this end, there is the need for integrating stakeholders and users into entire AI development life-cycle. Following the approach proposed by Bellazzi and Zupan in [21], a potential strategy is to apply in the design of AI-based systems the same conceptual model proposed for data mining models by the Cross Industry Standard Process for Data Mining (*CRISP-DM*) process model. CRISP-DM has six phases that are helpful to obtain explainable systems "by design":

1. Business understanding;
2. Data understanding;
3. Data preparation;
4. Modeling;
5. Evaluation;
6. Deployment.

While data preparation, modeling, and evaluation are now reported in all ML textbooks, very often little attention is given to business understanding, data understanding, and finally deployment. All of those are related to understanding the biomedical context, modeling the process and clearly expressing the goals. Data needs to be modeled; as well, it should be understood who and when data are collected, which is often related to the nature of missing data. Finally, having clearly in mind the deployment scenario is a key driver for designing XAI approaches. In this phase all actors involved in decision making should be involved, resorting to different instruments, from formal questionnaires to qualitative interviews. Several other development methodologies could be suitably adopted/extended/adapted when designing and implementing XAI systems in medicine, where the different stakeholders and the application domain are explicitly dealt with. As an example, well established methodologies as CommonKADS, supporting the design of knowledge-intensive systems coupled with UML notations, as well as methodologies dealing with the design of ontology- and/or data- based reasoning/-analytics systems could provide suitable techniques for domain understanding and modeling [31, 132, 161, 172].

As also reported by the EU white paper [64], AI systems and their decisions should be explained in a manner that is adapted to the appropriate stakeholder.

Among the specific features of medicine and health, we have to consider when designing XAI medical applications, we distinguish here:

- *The heterogeneous nature of medical data.* Medical data consists of images, movies, biosignals, and structured and unstructured alphanumeric data from electronic medical records. All of this data needs to be suitably integrated into and consistently and appropriately managed by XAI medical applications. Even though explainability has been considered for these different kinds of information systems (see, for example [98, 138]), further research efforts will have to deal with the elicitation of both visual and textual knowledge from such kinds of data, often left partially implicit by skilled physicians [82]. As an example, while radiology is mainly based on images, which are visually analyzed by radiologists even by the support of computerized devices, and related natural language reports, oncology deals mainly with knowledge represented in a textual way, often highly structured (as in the case of chemotherapy guidelines), while cardiology has a lot of information and related knowledge expressed through biosignals (e.g., the electrocardiogram) and movies (e.g., echcardiograms).

- *The presence of highly specialized knowledge in different clinical and healthcare domains.* Specific domains as cardiology, oncology, neurology, healthcare policy, and so on, have their own vocabulary, specific shared knowledge about diagnosis, treatments, and so on [146, 40]. XAI systems have, thus, to deal with jargon, abbreviations and terminological heterogeneity, idiosyncratic usage habits, and different kinds of knowledge, as previously stressed, especially when they have to support the exchange of shared information [150].

- *The presence of many different specialized processes, requiring the coordination of different stakeholders.* Explainability in medicine and health is often related to the results of prediction and/or classification tasks toward diagnosis and/or therapy effects and so on. Besides this "static" part, clinical tasks as monitoring, diagnosis, therapy, and prognosis are merged in a "dynamic" context, composed of complex medical or healthcare processes and pathways. In such processes, different healthcare actors, as clinicians, epidemiologists, nurses, and technicians, are involved with different roles. XAI systems cannot avoid facing these intertwined aspects, related to knowledge, information, processes, and actors, to suitably support specific clinical activities [50].

3.4.4 Can explainability draw us closer to wisdom?

Proposition: Explainability is a requirement to completing the data-information-knowledge-wisdom spectrum.

Understandability is an essential prerequisite for the transition from information to knowledge and provides a path to the realization of knowledge as wisdom. Explainability can, on the one hand, promote trust on the part of end users (compare with the previous section), and, on the other hand, promote understanding and, in turn, trust on the part of developers of algorithms, and finally also provide new insights. Trust is of eminent importance and is often underestimated and in order to bring AI into the real world, it must be trustworthy [84]. To be trustworthy, any AI must comply with applicable rules and regulations, adhere to ethical principles [131], follow legal issues [169] and be implemented in a secure and robust manner. This is particularly required by the EU High-Level Expert Group on AI [1].

In classical philosophy since ancient Greece, explanations have always been central, as the word philosophy itself means ”love of wisdom”. A good example is the deductive-nomological model of Hempel and Oppenheim (1948) [79] which is based on a formal structure of scientific explanation of a causal relationship using natural language. The model consists of two parts, the proposition to be explained (explanandum) and the explanation itself (explanans), which is composed of general law statements and (empirical) boundary conditions (antecedent statements) as premises. The preliminary work on this was already served by Karl Popper in his work ”Logic of Research” [144].

Colloquially, explanations differ in their completeness or degree of causality [139]. In his work, Tim Miller (2019) [126] combined insights from the social sciences with explanations in AI and divided explanatory questions into three classes: (1) what-questions, such as ”What event happened?”; (2) how-questions, such as ”How did this event happen?”; and (3) why-questions, such as ”Why did this event happen?”.

”Moving closer to wisdom” implies also that physicians and other clinical stakeholders receive some feedback on their own capabilities and attitudes towards explainability: do we need that AI systems have sophisticated explainability capabilities when it happens that physicians do not spend any effort to explain their choices? From one point of view, we could say that requirements about explainability have to be more strict for AI systems. Indeed, ”we may hold physicians responsible for their lack of explainability and potential mistakes,but we cannot do the same with AI” (from [105]). On the

[1] https://digital-strategy.ec.europa.eu/en/policies/expert-group-ai (access: February, 09, 2022)

other side, XAI systems can support clinicians in providing even more sound and founded decisions. In this direction, AI systems have to be considered as tools that require a specific and sound certification process also with regards to explainability. Similarly to what happens to marketed drugs, which need to follow strict certification processes before being approved, also XAI systems should be formally approved before used in real world clinical and healthcare contexts. Such kind of approach, followed by a continuous monitoring after the introduction of such tools in real clinical contexts, would help to clarify responsibilities both for physicians and for the producers of XAI systems.

3.4.5 Can an AI system that is not explainable be trustworthy?

Proposition: XAI is an integral component of trustworthy AI systems.

In 2019 the EU has published the Ethics Guidelines for trustworthy AI, which contains a general framework where explainability represents an important component[2]. These guidelines have been used as a basis for some of the sections of the proposal of the Artificial Intelligence Act released by the European Commission in April 2021. The guidelines correctly states that ”Trust in the development, deployment and use of AI systems concerns not only the technology's inherent properties, but also the qualities of the socio-technical systems involving AI applications ... it is not simply components of the AI system but the system in its overall context that may or may not engender trust.” To this end, AI systems should be lawful, i.e., complying with laws and regulations, ethical, i.e., being to ethical principles and robust, both from a technical and social perspective. The guidelines also provides seven requirements for implementation of AI trustworthy solutions, including:

- human agency and oversight
- technical robustness and safety
- privacy and data governance
- transparency
- diversity non-discrimination and fairness
- societal and environmental well-being
- accountability.

Explainability is considered as a component of transparency, together with traceability and communication. In our view explainability has an horizontal impact which is wider than what is stated in the guidelines. First of all, within transparency, it has a strong overlap with communication, which is related to

[2] https://www.aepd.es/sites/default/files/2019-12/ai-ethics-guidelines.pdf

understandability. Second, explainability is a key component of accountability, since it provides instrument to keep track of the decisions, going back to the "reasons-why" an AI tool, or a decision-maker empowered by AI solutions, has suggested the decision. Finally, it can be considered as a way to ensure technical robustness, providing explanations about change in decisions related to changes in the attribute values; this provides ways to control the performance of the algorithms and identify aberrant situations. Rather interestingly, trustworthiness allows to jointly consider two related concepts: explainability and reliability. "Reliability" is a component of robustness that indicates the degree of trust that we have on the prediction made by an ML model on a single example [133]. Coupled with local explainability ensures that local predictions can be used in a safety critical context as medicine is.

3.4.6 Is XAI in medicine always required?

Proposition: Explanations are not always required in order for an AI model to be useful. Functional specifications obtained from deep analysis of the problem domain and users should determine when explainability and interpretability are required.

While many recognize the necessity to incorporate explainability features in AI models, addressing user needs for understanding AI remains an open question. As the type of interpretability needed varies depending on the context, it is clear that XAI must take a human-centered approach. The same explanation may be more or less comprehensible to different users or even to the same user engaged in different roles and we should not confuse the different notions of interpretability because each kind serves a different purpose [177]. For instance, we cannot provide algorithm designers and end users with the same explanations. An ML expert might prefer an explanation that helps them debug the model and understand its inner-working [184]. In contrast, an end user might require a causal explanation of predictions to ensure that decisions informed by those predictions are fair [6].

The use of techniques to explain AI models has become central in human-centered systems. For example, visual analytics systems help users understand and interact with AI models by providing them with visualizations and tools that facilitate the exploration, analysis, interaction with AI models. To close the gap between XAI methods and user needs for transparency, the human-computer interaction community has called for interdisciplinary collaboration [1] and user-centered approaches to explainability [175]. The need to create effective explainability features in diverse medical applications led to novel ways to probe user needs. As an explanation can be seen as an answer to a question, Liao *et al.* represented user needs for explainability in terms of questions a user might ask about the AI model, thus creating a question bank, a list of prototypical user questions that XAI methods can address [112]. It is essential that model developers understand why an explanation is needed and what type of explanation is helpful for a given situation.

AI models do not need to be interpretable to be useful [81]. In this context, a blanket rejection of black-box methods in decision support systems may be hasty. For example, suppose an AI model yields accurate predictions that help clinicians better treat their patients. In that case, it may be useful even without a detailed explanation of how or why it works. Therefore, it is essential to identify biomedical applications in which black-box answers generated by AI models can have a useful role in decision support systems and thus can be safely used.

When an AI model produces the best results or yields accurate predictions that help clinicians better treat patients, it may be useful even without detailed explanations. For example, in reading medical images, trained AI systems enhance the performance of human radiologists in detecting cancers [11, 108]. That is not to say that AI interpretability is not valuable. In particular, when AI models are used in an automated fashion, laws and regulations should require a causal explanation of AI decisions to ensure that they are fair [14]. However, in situations when AI models do not lead to automated decision making, an explanation may not be needed and auditing [149] together with judicious testing of AI models via randomized control trials [152] might be sufficient.

Although the process used by AI models to generate predictions can be limited and biased, it is also different from human thought processes in ways that can reveal new connections. This creates a case for using black-box AI models as tools to guide human inquiry [75, 186]. For example, in a groundbreaking medical imaging study, a deep learning model was trained to diagnose diabetic retinopathy from retinal images [74]. The model achieved performance comparable to a committee of ophthalmologists. Further, the model accurately identified several characteristics that are not generally assessed with retinal images, including cardiological risk factors, age, and gender [143]. No one had previously noticed gender-based differences in human retinas, so the black-box observation inspired researchers to investigate how and why male and female retinas differ.

Moving to a final example, XAI needs to be declined in different ways in different contexts. Indeed, the explanation requirements regarding clinical medicine, for example, may have to deal with specialized physicians, who could have a knowledge in the specific domain that not requires XAI (but an AI system with certified good performances), while, considering, for example, the issue of pandemic management, requirements from epidemiology or national health policies and management could be extremely demanding, as possible relevant public decisions have to be suitably justified [35].

3.5 How temporalities and explainability are intertwined?

Until now, we did not consider explicitly the temporalities, which often are an essential aspect of the overall XAI, especially when focusing on domains from Medicine and Healthcare. Let us now focus on the different temporalities often

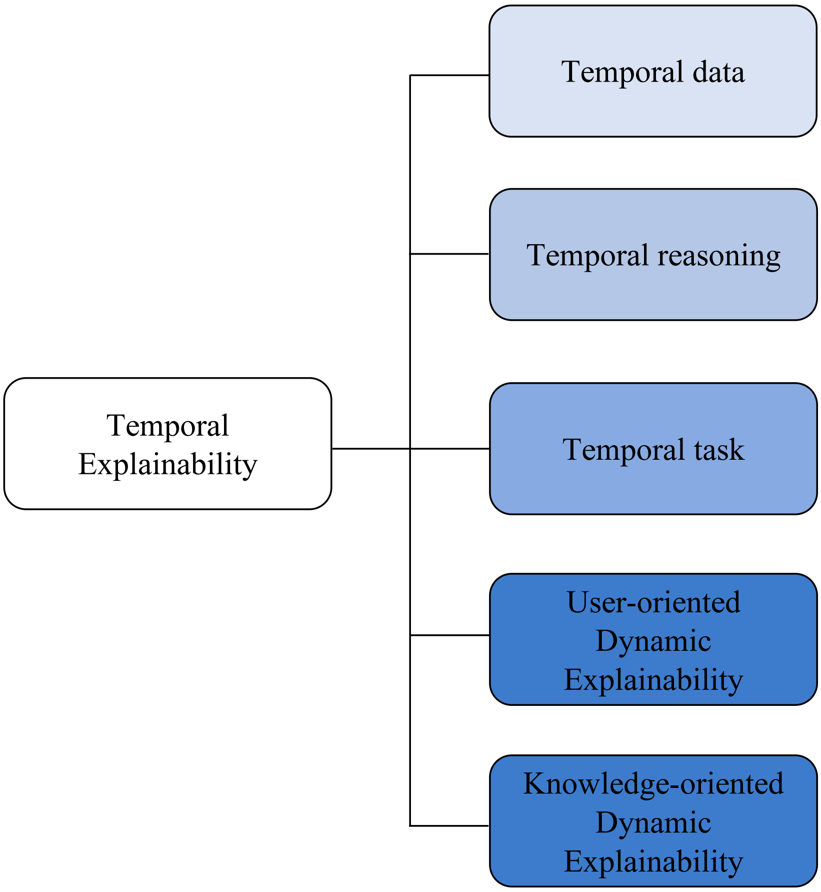

Temporal Explainability
Temporal data
Temporal reasoning
Temporal task
User-oriented Dynamic Explainability
Knowledge-oriented Dynamic Explainability

Figure 3-2: A classification for temporalities of XAI.

subtended by XAI in Medicine. The be on safe ground, we need to explicitly discuss what we mean by *temporality* in the context of AI and Computer Science. In general, both in philosophy, logic, and other branches of science, temporality refers to the perception of time and to the description of perceived phenomena having some time-related dimension.

In the computer science and AI areas, temporality has been declined in different aspects, which have been formally and technically addressed: from real-time systems [67], to temporal representation and reasoning [26], to temporal data modeling and querying [92, 93], to process modeling [51], to planning and scheduling [80, 135], to temporal constraints [145], to time and human-computer interaction [178], without any attempt to being complete.

Moving closer to XAI in Medicine and to the content of this thesis, temporality is an integral part of "explainability". Even though it is often implicitly considered, it deserves to be explicit addressed. Without any attempt to be complete, we introduce here some important issues, according which temporality can be declined when introducing XAI in contexts having explicit and unavoidable temporal aspects.

Thus, as shown in Figure 3-2, *temporal explainability* may be declined according to five different dimensions, not mutually exclusive:

(i) temporal data;

(ii) temporal reasoning;

(iii) temporal task;

(iv) user-oriented dynamic explainability;

(v) knowledge-oriented dynamic explainability.

Let us now discuss the different dimensions, also by introducing different examples from literature, which represent the different perspectives about temporal explainability.

Explainability and temporal data. When we talk about temporal explainability, we handle a problem based on temporal data. Suppose that the final goal is the classification, is important to explain to the user, how we obtained these results, which type of temporal aggregation is under the data, or how long is the history related to a specific entity in the database. In this context, temporal databases, time series analysis and temporal data warehouses provide a sound starting point for representing and managing data having temporal dimensions (e.g., valid time, transaction time [95, 94]), for the specification of temporal, possibly multidimensional queries [49], for the analysis of temporal behaviors [128]. Explainable approaches have to be able to explicitly show what is behind the modeling and the extraction of such temporal information.

Temporal reasoning. Temporal Reasoning and its application to Medicine, a process of inferring new knowledge from available facts, has long been considered an essential topic in AI research [52, 4, 26, 42, 68]. In the research area, many approaches and studies are proposed, as temporal logics, temporal constraint processing, temporal data mining, and others, which are inherently explainable, that could support the explainability also towards specific temporal reasoning mechanisms. In [77], for example, the authors focus on forecasting on temporal knowledge graphs (KGs), inferring future events based on past events. So that the users can understand and trust the predictions, as the predictions made by the learning models are interpretable. That is exactly why they propose an explainable reasoning approach for forecasting links on temporal knowledge graphs.

Another example is discussed in [113], where the authors propose TLogic, a novel symbolic framework based on temporal random walks in temporal knowledge graphs (tKGs). It is the first approach that directly learns temporal logical rules from tKGs and applies these rules to the link forecasting task. It is a rule-based link forecasting framework for tKGs, which starting from the extraction of temporal walks from the graph and then lifting these walks to a more abstract, semantic level, obtains temporal rules that can be generalized to new data.

The temporal task. Another aspect to investigate is the temporal (decision-based) task supported by a given explainable approach. As we already mentioned, explainability in medicine and health is often related to the results of prediction and/or classification tasks towards diagnosis and/or therapy effects and so on. Clinical tasks as monitoring, diagnosis, therapy, and prognosis are merged in a “dynamic” context where temporality is essential. One of the mainly investigated temporal task, in the context of explainability, is prediction. For example, the early prediction of mortality and risk of deterioration in COVID-19, can reduce mortality and increase the opportunity to have a better treatment. Because of the dynamism of the clinical indications about the disease and sometimes of a sudden deterioration in the condition of moderate-stage patients, it is crucial to develop an automated model that can preemptively predict which patients are at risk for ventilator support and mortality. Regarding this, in [13], the authors face the problem of the prediction of mechanical ventilation, showing how the use of ML algorithms can assist doctors to predict at-risk patients. Beyond the prediction, even the prevention in medicine is fundamental. By continuing to use the COVID-19 example, during the pandemic period, it was a requirement to study which factors were affecting the hospital admissions. In this respect, in [158], the authors propose a model that utilizes Formal Concept Analysis (FCA) to explain a machine learning technique called Long–short Term Memory (LSTM) on a dataset of hospital admissions due to COVID-19 in the United Kingdom. LSTM is employed to forecast the factors impacting admission due to COVID-19, while the FCA method, to develop rules or relationships between the attributes. The use of FCA to explain LSTM is a preliminary approach to an understandable and explainable AI.

User-oriented dynamic explainability. Another aspect to consider is the possibility to produce a dynamic explanation, which changes over the time, according to the user preferences. Recently, regarding the user-oriented dynamic explainability, many models have been designed to incorporate temporal information into recommender system to capture user dynamic preferences.

Considering again COVID-19, during the pandemic period when clinicians started to consider different therapeutic options, trying to cure patients without any benchmarks, it was essential to understand as soon as possible which was the most effective therapy. Over the time, they collected information about different responses and adverse events, having an idea on the possible therapies. Having an automated system to evaluate the multiple therapies for different types of patients according to the novel knowledge collected from clinical practice, would have been fundamental to support clinicians in their decisions. In general, this idea can be related to the recommendation systems, where the dynamic user preferences are used to define personal recommendations.

In [113], for example, the authors face the problem of simultaneously multimodel explanation generation and dynamic user preference modeling in the context of explainable dynamic personalized recommendations. The proposed model, Attentive Recurrent Neural Network (Ante-RNN), allows combining visual image information with text descriptions for better recommendation. They study how the short term preference modeling can capture user’s long-term interest dynamics. The learned attention weights can in turn help to provide reasonable interpretations of recommendation results. Then, a novel dynamic contextual attention mechanism is incorporated into Ante-RNN for modeling the complicated correlations among recent items and strengthening the user's short-term interests. Another example of recommendation system is discussed in [41]. The authors build a novel Dynamic Explainable Recommender (called DER) for more accurate user modeling and explanations. Specifically, with the desire to make a recommender system more interpretable, they design a time-aware gated recurrent unit (GRU) to model user dynamic preferences. Exploiting user reviews, they can provide adaptive recommendation explanations according to the user dynamic preference.

Knowledge-oriented dynamic explainability. Continuing to use COVID-19 as an example of clinical domain, in accordance with the previous temporal tasks, we have also to consider the dynamic aspects of medical knowledge. During the pandemic, indeed, the dynamism of the clinical indications or the clinical results reasoned that even the explainable AI became dynamic. Novel information may influence the results of an AI algorithm, which has the need to reanalyze the data to produce updated knowledge. Thus explainable approaches should also have the capability of explaining how and why some knowledge supporting/derived-from them changed over time and how such change influenced the obtained results.

3.6 Conclusions and Research Directions

The issue of explainability in AI is evolving at a rapid pace. As we have seen in this paper, there has been considerable research into XAI, but there is still much to be done. We note here five broad areas where more research is needed.

- **Bridging the gap between symbolic (ante hoc) and sub-symbolic (black-box) approaches.** Sub-symbolic ML approaches and symbolic ones are currently considered by two research communities, having often completely different perspectives and background. XAI requires that such dichotomy has to be overcame. Indeed, symbolic approaches, as the ones related to logics-based proposals, ontologies, query systems, Bayesian networks, and so on, would be grounded in order to use them in establishing explainability [154]. Research on the seamless proposal of "hybrid" systems, merging both sub-symbolic and symbolic approaches still requires a lot of joint efforts.

- **Engineering explainability into intelligent systems.** An important, even fundamental question is whether and how explainability can actually be engineered into AI. Even given our conceptual framework for thinking about XAI (Section 2), we still need to address the idiosyncrasies of individual intelligent systems as well as those of their users. We contend that more specialized research into the structural, functional, and behavioral characteristics of these systems and the environments in which they are situated should be the targets of rigorous mixed-methods research that encompasses the entire system ecology, from *in silico* to *in vivo* contexts.

- **Evaluating and improving the effects of explainable components and approaches.** The evaluation of intelligent systems, as a scientific and methodological discipline, is changing, yet there needs to be more systematic investigation and implementation of these methods. Too often, there is emphasis on the accuracy of a decision made by such systems, typical as a proportion- whether in terms of overall accuracy (percent ”correct”), or more nuanced indicators such as sensitivity, specificity, predictive values, or their derivatives such as the F-score or areas under the receiver operating characteristic curve or under the precision-recall curve. None of these well-used metrics indicate anything about the effects of the system on user beliefs, attitudes, or behavior. These are effects that require, again, deep mixed-methods research, this time applied to evaluating the effects of XAI (or its absence) on such issues as user acceptability, actions taken (or not) based on the results offered by the system, and overall impact on clinical or other workflows.

- **Determining when explainability is needed.** Is explainability always needed? This is a fair question, indeed. AI systems (or ”subsystems” that work in the background to provide some inference to assist another systems might not require real-time explainability. A feature selection algorithm as part of a knowledge discovery or decision making workflow is one example if such an AI. However, we would argue that in order for software developers who need to use such subsystems in their work, explainability to them is critically important. They have to know how that subsystem works and why. But to the end-user of that workflow, it might not be so important in real-time. Rather, an in-depth description of the entire workflow and its components should be provided so the end-user understands how the overall system works. This situation begs the question again: ”Is explainability always needed?”. The answer, we propose here, is yes, but titrated to the needs of the user at particular times or in response to specific events.

- **Investigating the design of user-centered and user-tailored explainability artifacts.** If there were ever a more urgent need for rigorous research into user-centered design, it is hard to see one that surpasses the field of XAI. Such design, as noted, must be sensitive to workflow contexts, certainly, but there are other equally important considerations. One of the most important of these is the involvement of users into the design process. Rapid prototype design paradigms should be used in order to keep users involved during all phases of the development and implementation of AI systems. We already do this to some extent in the field of knowledge engineering, although history is full of examples where gaps in knowledge acquisition and representation have led to system failures, some with catastrophic results

- **Empowering Explainability with temporal dimensions.** XAI in Medicine cannot avoid to consider temporality. In this direction, many specific concepts and techniques, inherently explainable, have been introduced in research areas, as temporal logics, temporal databases, temporal data mining and temporal information visualization, just to mention some of them. Specific research efforts are now required to extend/adapt/use these results to make XAI systems effective and sound.

We hope that our examination of the issues involved in developing XAI systems-our *manifesto*, if you will- will not be construed as the definitive work in this area. Rather, hope that the issues we considered here will stimulate further thought and hopefully fruitful research and development of XAI systems, particularly in medical contexts, but extending beyond to other contexts as well.

Chapter 4

Discovering Predictive Trend-Event Patterns

In this chapter, we present a new proposal, which exploits frequent temporal pattern mining and predictive pattern mining. Before the introduction of this technique, we briefly describe the fundamental notions of Support Vector Machine (SVM), a classification model we used for predicting patterns. Then we introduce the problem, proposing a new kind of predictive temporal pattern called PREDICTIVE TREND-EVENT PATTERN (*PTE*-P). It consists of multiple complex temporal features, where trends related to patient parameters and related medical events are suitably represented [117]. Considering the temporal specification of explainability presented in Section 3.5 of Chapter 3, this technique is related to temporal data and the temporal task, in this case the prediction, in fact, we exploit the temporal component of data in MIMIC-III database to predict the sepsis onset.

4.1 Introduction

The opportunity to store huge volumes of data offered by modern technologies gives the possibility to extract useful knowledge. Recently, many proposals have developed powerful tools that extract qualitative patterns from data to assess temporal relationships between such patterns [18, 88, 102]. Although there are many differences in their general approach, all these proposals focus on extracting/modeling temporal precedence relationships among patterns. No surprise that a core motivation for temporal data mining is to infer and verify hypotheses about the potential cause-effect interaction between medically-relevant events [156]. The increasing use and availability of longitudinal electronic data represent a significant opportunity to discover new knowledge from multivariate, time-oriented data. Often, in clinical domains, there is an intrinsic need to have a consistent intermediate interpretation layer, which explicitly identifies possible relationships between clinical patient data and events, such as drug intakes, and so on. The detection of specific temporal patterns, i.e., time intervals in which one or more time series assume a behavior of interest, plays a key role in such contexts. A research field that focuses on this direction is temporal data mining, working on structured and semi-structured data. Considering the current literature and the need for meaningful summary information for most of the medical domains, we wanted to combine complex temporal features/patterns[1], composed by interval-based data and instantaneous events, with a wrapper-based algorithm which extracts a compact and non-redundant set of *predictive patterns.*

The idea is to integrate frequent temporal pattern mining and predictive pattern mining, using statistical tests and a classification model in order to obtain a set of predictive *temporal* patterns for a class of interest (e.g., a given pathology). Before introducing the methodology, we briefly recall the basic notions of support vector machines, which are used in our framework. We propose an original mining methodology for discovering a compact set of highly predictive temporal patterns to ensure that every pattern in the result offers a significant predictive advantage to describe the class of interest. The new kind of predictive temporal pattern proposed is the PREDICTIVE TREND-EVENT PATTERN, namely *PTE*-P, which consists of multiple complex temporal features, where trends related to patient parameters and related medical events are suitably represented. We apply the proposed methodology to the prediction of sepsis, considering Medical Information Mart for Intensive Care (MIMIC)[100] medical record data.

4.2 Support Vector Machines

Support vector machines (SVM), introduced by Cortes and Vapnik [173], are the most famous and employed supervised discriminative classifier. Their success is given by the concept simplicity, their basis on strong mathematical foundations and statistical learning theory, and their accuracy. As a binary classifier, the central focus is finding a hyperplane that separates the samples of different outcomes. An SVM model represents samples as points in space, mapped in a way where samples of the two different categories are divided by a clear gap that is as wide as possible. When new samples are added, SVM maps them into that same space and predicts their category.

If the two classes are linearly separable, there exist many different hyperplanes that allow the perfect separation of the two classes. The simplest model of SVM is the so-called maximal margin classifier [59]. SVMs find an optimal hyperplane with a maximum distance to the closest point of the two classes. The instances which are closest to the optimal hyperplane are called support vectors.

We will start with the simplest case: linear machines trained on separable data. We label the training data x_i, y_i, $i = 1, ..., l$, $y_i \in \{-1, 1\}$. Suppose we have some hyperplane that separates the positive from the negative examples. The points x which lie on the hyperplane satisfy $w \cdot +b = 0$, where w is normal

[1] Hereinafter, we will use the expression *temporal features* instead of *temporal patterns*, to avoid any confusion between temporal patterns and predictive patterns, each of them being a possible collection of temporal patterns

to the hyperplane, $|b|/\|w\|$ is the perpendicular distance from the hyperplane to the origin, and $\|w\|$ is the Euclidean norm of w. Let $d_+(d_-)$ be the shortest distance from the separating hyperplane to the closest positive (negative) example. Define the "margin" of a separating hyperplane to be $d_+ + d_-$. For the linearly separable case, the support vector algorithm simply looks for the separating hyperplane with the largest margin. Suppose that all the training data satisfy the following constraints:

$$x_i \cdot w + b \geq +1 \quad for \quad y_i = +1$$
$$x_i \cdot w + b \leq -1 \quad for \quad y_i = -1$$

These can be combined into one set of inequalities:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall i$$

This results in a quadratic constrained optimization problem. To solve it: transform the constrained problem into an unconstrained one using Lagrange multipliers

$$L = \frac{1}{2} w^T \cdot w - \sum_{i=1}^{N} \alpha_i (y_i (x_i \cdot w^T + b) - 1)$$

The solution is obtained as a linear combination of the points of the training set.

$$\{w^{opt}\} = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$\alpha_i = 0$ means that the i-th point does not contribute to the solution. $\alpha_i \neq 0$ are called **support vectors**. Now the goal is to maximize the margin and minimize the "errors". We have to change the objective function to be minimized by assigning an extra cost for errors:

$$\arg\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i s.t. \quad y_i(x_i \cdot w + b) - 1 \geq 1 - \xi_i \qquad \xi_i \geq 0$$

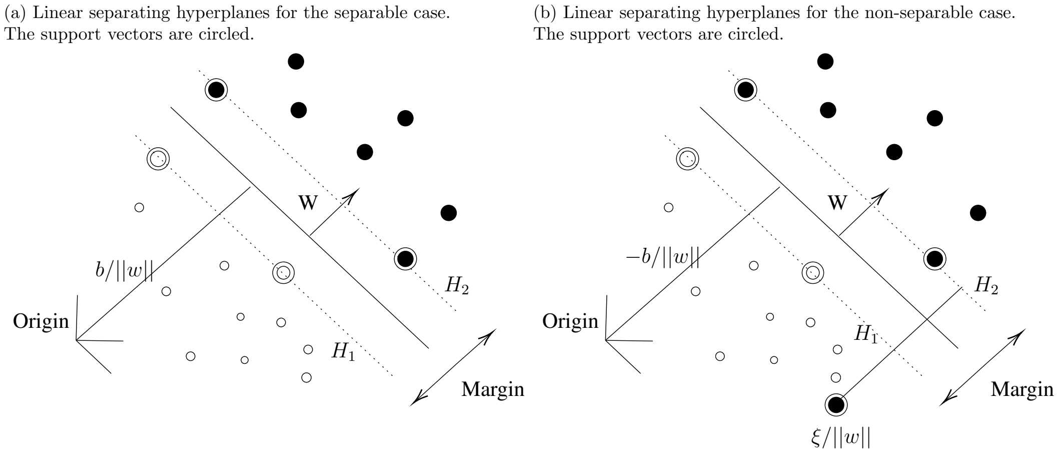
The solution is again given by:

$$\{w^{opt}\} = \sum_{i=1}^{N} \alpha_i y_i x_i$$

where N is the number of support vectors. Thus, the only difference from the optimal hyperplane case is that the α_i now has an upper bound of C.

$$L = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i - \sum_i \alpha_i \{y_i (x_i \cdot w^T + b) - 1 + \xi_i)\} - \sum_i \mu_i \xi_i$$

With real data sets, the hyperplane that would clearly separate the samples most often does not exist. To solve this problem, the solution is to relax the



(a) Linear separating hyperplanes for the separable case. The support vectors are circled.

(b) Linear separating hyperplanes for the non-separable case. The support vectors are circled.

Figure 4-1: Linear separating hyperplanes for separable and non-separable cases.

constraints by inserting in the formulation the so-called slack variables $\xi_i = 1, ..., l$, which admit that the hyperplane misclassifies some samples. In [58] the authors proposed the soft margin method, where the goal is to maximize the margin and minimize the "errors".

$$x_i \cdot w + b \geq +1 - \xi_i \quad for \quad y_i = +1$$

$$x_i \cdot w + b \leq -1 + \xi_i \quad for \quad y_i = -1$$

$$\xi_i \geq 0 \quad \forall_i$$

There are cases for which a hyperplane represents a too simplistic solution. The idea is to project the samples into a higher dimensional space where discrimination is easier. Finding the optimal hyperplane provides a linear classifier. Besides such linear kernels, SVMs are frequently used with non-linear kernels which in essence transform the original attribute space to a new, higher dimensional one in which the linear classifier is inferred. There exist different kernel functions, the most popular are polynomial, radial basis and sigmoid functions. The choice of the appropriate kernel should in principle depend on the properties of the data set and problem domain.

4.3 The temporal mining framework

In this section, we illustrate the new methodological framework and the basic steps needed to obtain a new kind of predictive temporal pattern, namely Predictive Trend-Event Patterns (*PTE*-Ps).

The framework aims at combining complex temporal features, composed of interval-based data (trends) and instantaneous events, with a wrapper-based algorithm that uses statistical tests and a classification model to extract a compact and non-redundant *predictive* set of patterns composed by such temporal features.

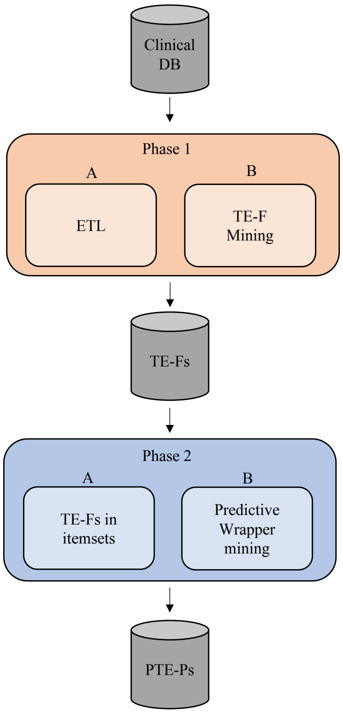


Figure 4-2: All the steps and data needed to mine *PTE*-Ps.

Figure 4-2 shows the main steps of the proposed methodology, consisting of two main phases:

- Phase 1: An Extract/ Transform/Load (ETL) process (Step 1.A) retrieves suitable data from a temporal database. Trend-Event Features are then mined as an intermediate result (Step 1.B).
- Phase 2: All the mined Trend-Event Features are transformed into corresponding itemsets, in order to evaluate their goodness using a classification model (Step 2.A). A Predictive Wrapper mining algorithm is then able to obtain a set of Predictive Trend-Event Patterns, by using a greedy approach to select and evaluate the patterns (Step 2.B).

The following subsections describe in detail these phases and the related concepts.

4.3.1 Trend-Event Features

Let us now introduce the specific complex temporal feature we will consider, named *Trend-Event Feature*(*TE*-F). Figure 4-3 provides a visual representation of the main concepts of *TE*-Fs.

Let us consider a time series of samples related to systolic blood pressure measurements and some possibly related events, e.g., a drug intake (Figure 4-3). Raw measure data related to systolic blood pressure measures are depicted as scattered points, where every point represents a tuple, and its position is given by its valid time VT and its measure. Event e, is represented by a vertical

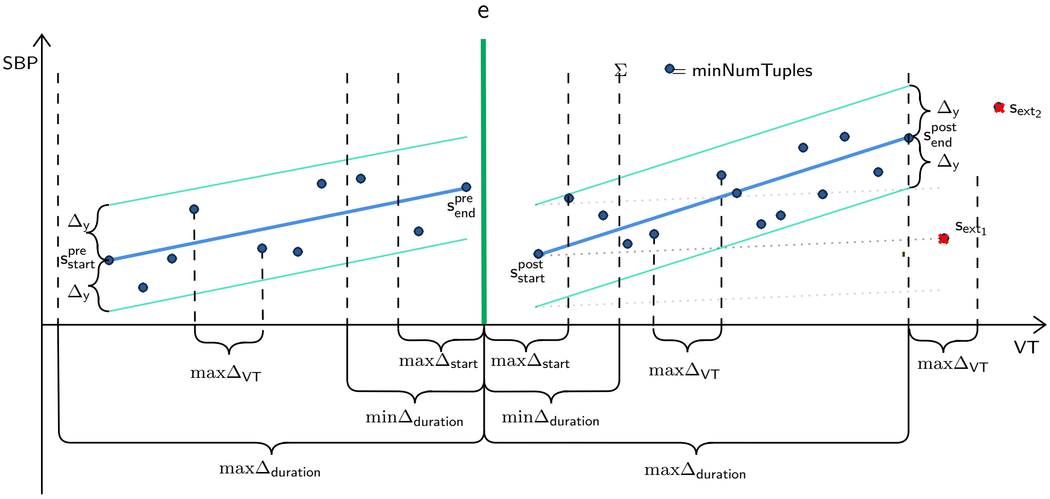


Figure 4-3: An example of *TE*-F. There are two trends, one that precedes event e (from s^{pre}_{start} to s^{pre}_{end}), and a second one right after E (from s^{post}_{start} to s^{post}_{end}). These are both valid trends because they satisfy every constraint. Moreover, s_{ext_1} and s_{ext_2} are external to these trends because they violate either Δ_y or $\max \Delta_{VT}$.

line, placed on its corresponding VT. It could be related to, for example, some drugs given to the patient.

Every tuple before the event could potentially be part of the trend before it ($trend^{pre}$), while every tuple after the event could be in $trend^{post}$. In both trends, the first tuple is denoted as s_{start}, while the last one is s_{end}. *TE*-Fs are based on the assumption that there can only be one single event for a VT. Thus, if many events occur at the same time, they will be merged in a single composite event. For example, let the event be a single drug administered to a patient: if at a certain moment, the patient receives two different drugs, the event will be composed of the conjunction of these two drugs, and it will differ from the events formed by these two drugs taken separately.

A trend is graphically represented by a segment line from s_{start} to s_{end}, and every other point of such trend should lay in this segment. However, this is an unlikely scenario with real data and it is necessary to allow a threshold to admit a point to be part of the trend. The sample s_{ext_1} represents the violation of the above constraint: a trend from s^{pre}_{start} to s_{ext_1} excludes many tuples in-between and consequently it is not a valid trend.

Let us now define these concepts more formally in a relational setting, by considering a classical temporal schema $R = U \cup \{VT\}$ where R is a set of *atemporal attributes* and VT is an attribute denoting the *valid time* of each tuple. We may safely assume $Dom(VT) = \mathbb{R}$. Given an attribute $M \in U$, we say that M is a *measure* if and only if $Dom(M)$ is a metric space. We assume that for all measures M it holds $Dom(M) = \mathbb{R}$. A *trend* is formed by tuples of some measurement M in U that satisfy the following criteria. First, tuples must share the same values for some non-empty *dimensions* $D \subseteq U$. Dimensions allow one to specify that measures are related to the same entity, e.g., dimensions could be used to consider only tuples related to the same patient or admission. Also, tuples must be temporally ordered using their VT. It means that, given an instance $\mathbf{r}$ of R, two tuples s, s' satisfy $s \rightarrow s'$ if and only if $s[D] = s'[D]$, $s[VT] < s'[VT]$, there is no tuple s'' in $\mathbf{r}$ with $s''[D] = s[D]$ s.t. $s[VT] < s''[VT] < s'[VT]$. In the following, we assume that there are no tuples with the same VT. Given a trend tr, we denote with s_{start} the first tuple of tr, and with s_{end} the last one. Moreover, tr is a non-empty set of tuples $tr = \{s_{start}, \ldots, s_{end}\}$ satisfying $s_{start} \rightarrow \ldots \rightarrow s_{end}$. Implicitly, it means that $\forall s_i \in tr$, $s_{start}[VT] < s_i[VT] < s_{end}[VT]$. Let e be the tuple that denotes the event, and let us assume that E is the attribute, whose values represent the possible occurring events (i.e., $e[E] \in \mathcal{E})$)). For each tuple $s_i \in tr^{pre}$, $s_i[VT] \leq e[VT]$; while for each tuple $s_i \in tr^{post}$, $s_i[VT] > e[VT]$.

There are some constraints necessary to determine a valid trend. Some of them are tuple-related and some are trend-related. As shown in Figure 4-3, a trend must satisfy all the following conditions on measures and on VT to hold:

- *minNumTuples* - It denotes the minimum number of tuples needed to form a trend. Even though just two tuples could build a trend, this parameter allows us to better define meaningful trends. In other words, there is a trend if:

$$\sum_{s_{start}}^{s_{end}} s_k \geq minNumTuples$$

- $\max \Delta_{VT}$ - It expresses the maximum time distance between two adjacent tuples to be part of the same trend. Given two tuples $s_m \in tr$ and s_{m+1}, $s_{m+1} \in tr$ if $t[VT]_{m+1} - t[VT]_m \leq \max \Delta_{VT}$. This is useful to avoid those measures that are taken too far away from the other, becoming irrelevant to the trend itself.

- $\max \Delta_{start}$ - It represents the maximum delay between a trend and an event e. A trend is insightful if the measurement closest to the event is measured within a negligible time distance, which is $\max \Delta_{start}$.

- $\min \Delta_{duration}$ - It represents the minimum span of a trend from an event. Controlling the time distance of a trend from an event is of central importance. This parameter is necessary to avoid those trends having most measures concentrated in a small time frame that is too close to the event.

- $\max \Delta_{duration}$ - It is complementary of $\min \Delta_{duration}$ and characterizes the maximum time span of a trend from an event. It is particularly useful when the goal is to extract trends for testing the short-term effects of a certain event. $\max \Delta_{duration}$ is different from $\max \Delta_{VT}$ because it is related to the whole trend, while $\max \Delta_{VT}$ considers couples of adjacent tuples within the trend.

- Δ_y - It represents the threshold where the difference between the measure and its position in the line is negligible, and must hold for all the tuples in *tr*. Given $s \in tr$, let $s_i[M]$ the projection of $t[M]$ on *tr*, namely:

  $$s_i[M] = (s[VT] - s_{start}[VT]) * m + s_{start}[M]$$

  where
  $$m = \frac{s_{end}[M] - s_{start}[M]}{s_{end}[VT] - s_{start}[VT]}$$

  there is a trend *tr* if $\forall s[M] \in \{s_{start}[M], \ldots, s_{end}[M]\}$ then $s_i[M]-\Delta_y\leq s[M] \leq s_i[M]+\Delta_y$.

If all the constraints depicted above hold, then we have a valid trend *tr*. The trend is then labeled using the alphabet $\Sigma_{trend} = \{$*increasing, steady, decreasing*$\}$. To label a trend correctly, we need to calculate its weighted rate of change with respect to the measure of the starting point, that is:

$$changeRate = \frac{s_{end}[M] - s_{start}[M]}{s_{end}[VT] - s_{start}[VT]} * \frac{100}{s_{start}[M]}$$

For the simplicity of the example, we assume that $s_{start}[M] \neq 0$. We define a new parameter *maxIncrease* as a threshold for *changeRate*, and we determine the label of a trend as follows:

- *increasing*: *changeRate* > *maxIncrease*
- *decreasing*: *changeRate* < *- (maxIncrease)*
- *steady*: otherwise

Through these parameters, clinicians can finely tune the trend extraction, to focus on those they consider appropriate for the kind of event and measurement they are analyzing.
Now we are ready to define *TE*-F:

Definition 13. *A Trend-Event Feature (TE-F) is a complex temporal feature with an expression of the form*

$$\textit{[measure; trend}^{pre} \in \{\textit{increasing, steady, decreasing}\}\textit{; ev} \in \mathcal{E}\textit{;}$$
$$\textit{trend}^{post} \in \{\textit{increasing, steady, decreasing}\}\textit{]}$$

For example, let *ev* be a tuple containing for attribute *adminDrug* attribute the value *propofol*. Complex temporal features like “A patient’s respiratory rate was *increasing* before the administration of *propofol*. After such administration, the respiratory rate was *steady*” would be represented through *TE*-F [*respRate*; *increasing*; *propofol*; *steady*].

Algorithm 1: Extraction of tr^{post} from the Trend-Event Feature mining algorithm

```
Input: measuresList, eventsList, currentEventIndex
Output: $trend^{post}(trendType)$
nextMIndex = eventsList[currentEventIndex](closestMeasureIndex)
/* Check $\max\Delta_{start}$ */
if (measuresList[nextMIndex](VT) - eventsList[currentEventIndex](VT) < $\max\Delta_{start}$) then
    $s^{post}_{start}$ = measuresList[nextMIndex]
    /* Discover of $s^{post}_{end}$ */
    index = nextMIndex + 1
    while index ¡ measuresList.size do
        $s^{post}_{previous}$ = measuresList[index - 1]
        $s^{post}_{current}$ = measuresList[index]
        /* Check $\max\Delta_{VT}$ and $\max\Delta_{duration}$ */
        if ($s^{post}_{current}$(VT) - $s^{post}_{previous}$(VT) > $\max\Delta_{VT}$ || $s^{post}_{current}$(VT) - eventsList[currentEventIndex](VT) > $\max\Delta_{duration}$) then
            break
        isOverDeltaY = false
        foreach ($s^{post}_{k} \in \{s^{post}_{start+1}, \ldots, s^{post}_{current}\}$) do
            /* $s^{post}_{proj}$ represents the projection of $s^{post}_{k}$ on the trend line. */
            $s^{post}_{proj}$(M) = $\frac{s^{post}_{current}(M)-s^{post}_{start}(M)}{s^{post}_{current}(VT)-s^{post}_{start}(VT)}$*($s^{post}_{ideal}$(VT) - $s^{post}_{start}$(VT)) + $s^{post}_{start}$(M);
            /* Check $\Delta_y$ (percentage) */
            if ($s^{post}_{k}(M) > s^{post}_{proj}(M) + \frac{s^{post}_{proj}(M)*\Delta_y}{100}$ or $s^{post}_{k}(M) < s^{post}_{proj}(M) - \frac{s^{post}_{proj}(M)*\Delta_y}{100}$) then
                isOverDeltaY = true
                break
        if isOverDeltaY = true then break;
        index++
    $s^{post}_{end}$ = measuresList[index - 1]
    /* Check minNumTuples and $\min\Delta_{duration}$ */
    if (index - 1 - nextMIndex > minNumTuples && $s^{post}_{end}$(VT) - eventsList[currentEventIndex](VT) > $\min\Delta_{duration}$) then
        /* [omitted calculation of the changeRate of $trend^{post}$] */
        /* Labeling */
        if ($trend^{post}$(changeRate) > maxIncrease) then return $trend^{post}$(trendType) = INCREASING;
        else if ($trend^{post}$(changeRate) < maxIncrease) then return $trend^{post}$(trendType) = DECREASING;
        else return $trend^{post}$(trendType) = STEADY;
    else
        return null
else
    return null
```

Tuple #	VT	PatientID	Event	Measure
1	12/10/2019 11:25	1	-	55
2	12/10/2019 11:55	1	-	57
3	12/10/2019 12:25	1	-	58
4	12/10/2019 12:55	1	-	62
5	12/10/2019 13:25	1	-	65
6	12/10/2019 13:30	1	$drug_1$	-
7	12/10/2019 13:55	1	-	65
8	12/10/2019 14:25	1	-	62
9	12/10/2019 14:55	1	-	67
10	12/10/2019 15:25	1	-	69
11	12/10/2019 15:55	1	-	62
12	12/10/2019 14:25	2	-	55
13	12/10/2019 14:55	2	-	57
14	12/10/2019 15:25	2	-	58
15	12/10/2019 15:55	2	-	62
16	12/10/2019 16:25	2	-	60
17	12/10/2019 16:55	2	-	61
18	12/10/2019 17:25	2	-	62
19	12/10/2019 17:30	2	$drug_2$	-
20	12/10/2019 17:55	2	-	67
21	12/10/2019 18:25	2	-	69
22	12/10/2019 18:55	2	-	71

Table 4.1: Excerpt of a sample table needed to obtain Trend-Event features.

4.3.2 Mining Trend-Event Features

The first step needed to obtain *PTE*-Ps is through the retrieval of Trend-Event Features (*TE*-Fs) from a traditional relational clinical database containing time-stamped data. Trend-Event Features (*TE*-Fs) are complex features composed by an event and the two trends immediately before/after such event.

First of all, it is necessary to perform an Extract/Transform/Load (ETL) process in order to model data coming from different tables and reduce them to a simplified common structure. Each different kind of measure considered needs its own table, with attributes *VT*, *PatientID*, *Event*, and *Measure*, that represent the valid time of the tuple, a unique identifier for a patient (or an admission), the medically relevant event, and the measure we are interested in, respectively.

Table 4.1 represents a small example of data we may have after our ETL process. To extract *TE*-Fs, we need to partition the table grouping data by *PatientID.* That is, in the example of Table 4.1, it extracts a *TE*-F from tuple 1 to 11, and another one from 12 to 22.

The first goal is to discover every possible event and then possibly derive a trend before and after it. To speed up this process, we create two different lists:

- `measuresList` - containing only the measures and VT of such partition, ordered by their VT
- `eventsList` - containing all the events of such partition, their VT, and the index of the closest measure that is after them

These lists avoid checking sequentially whether any tuple represents either an event or a measure.

In the following, we describe the Algorithm 1 that derives the trend that succeeded an event (i.e., $trend^{post}$). The derivation of the trend that preceded the event is specular to this one, and thus, it is omitted. The algorithm is performed for each event. If we are not able to retrieve either a valid $trend^{pre}$ or a valid $trend^{post}$ for an event, the current event is ignored, and the whole process restarts for the next one. The pseudocode in Algorithm 1 helps to better understand the following steps.

1. Check for $\max \Delta_{start}$. Given the event and the first measure after it (i.e., s^{post}_{start}), we have to check that the difference of their VT is lower than $\max \Delta_{start}$.
2. Discover s^{post}_{end}. A loop through `measuresList` finds the final point of the trend, called s^{post}_{end}. Initially, it is the measure right before s^{post}_{start}, and for each iteration it becomes the next tuple in `measuresList` if all the measures from s^{post}_{start} to s^{post}_{end} satisfy all the constraints. The steps performed within a single loop are thus:
    (a) Check for $\max \Delta_{VT}$ and $\max \Delta_{duration}$. It is necessary to check if the temporal distance between the current tuple and the previous one is less than $\max \Delta_{VT}$, and the temporal distance between the current tuple and the event is less than $\max \Delta_{duration}$. If these constraints are not satisfied, we exit the loop.
    (b) Check for Δ_y. As all the measures of a trend must lay within a vertical threshold $\Delta_y$from the line from s^{pre}_{start} to s^{pre}_{end}, we have to check that every tuple s^{post}_k satisfies this constraint. If there is some violation of it, we cannot have a valid trend with the current s^{post}_{end}.
3. Check *minNumTuples* and $\min \Delta_{duration}$. After s^{post}_{end} is finally retrieved, we verify that the number of measures in the current trend is greater than *minNumTuples*. Now, we can also check if the length of the trend satisfies the $\min \Delta_{duration}$ constraint.
4. Labeling. Finally, we label our $trend^{post}$ according to its changeRate and the constraint *maxIncrease*.

Finally, we need to check whether an event influenced them or not. To verify this condition, we consider a new line from s^{pre}_{start} to s^{post}_{end}. If all the tuples in-between satisfy the constraint with respect to Δ_y, then we could build a single trend from s^{pre}_{start} to s^{post}_{end}. In this case, the event is not influencing the trends related to the considered measure. Thus, we do not derive any *TE*-F. After all these steps, we finally obtain a *TE*-F.

4.3.3 From *TE*-Fs to Predictive Trend-Event Patterns

TE-Fs represent all the temporal complex features that hold for each patient. Here, we want to use all these features together and verify if other patients experienced them, perhaps related to some common pathological state.

Our goal is to obtain a small set of predictive and non-redundant patterns. To achieve that, we developed an algorithm that relies upon the minimum predictive pattern (MPP) mining approach proposed in [16] and its extension presented in [118].

Before the detailed explanation of the algorithm, it is necessary to further extend and define the properties of a pattern in our temporal context.

Definition 14. *Let* $\Sigma = \{i_1, i_2, \ldots, i_k\}$ *denote a set of* k items. Σ *is also called the* alphabet*.*

Assume an item $I = (\mathit{tf}, \mathit{bval})$, where *tf* is a (possibly temporal) feature and *bval* is a boolean value.

Definition 15. *An itemset pattern is a conjunction of items:* $P = i_1\wedge, \ldots, \wedge i_k$.

If a pattern contains k items, we will call it a *k-pattern* (an item is a 1-pattern).

Thus, for features being *TE*-Fs, we define a 1-Predictive Trend-Event Pattern (1-*PTE*-P) as a 1-pattern with an expression of the form

$$[m \ \& \ \mathit{trend}^{pre} \ \& \ ev \ \& \ \mathit{trend}^{post} = \{\,\text{true} \mid \text{false}\,\}]$$

where m is the name of a measure, trend^{pre} and $\mathit{trend}^{post} \in$ {increasing, steady, decreasing} and $ev \in \mathcal{E}$. An instance of such 1-*PTE*-P could be written as

$$[\mathit{Resp.\ Rate} \ \& \ \mathit{increasing} \ \& \ \mathit{propofol} \ \& \ \mathit{steady} = \text{true}]$$

A k-Predictive Trend-Event Pattern (k-*PTE*-P) is a k-pattern with an expression of the form

$$\begin{gathered}\{[m_1 \ \& \ tr_1^{pre} \ \& \ ev_1 \ \& \ tr_1^{post} = \{\,\mathit{true} \mid \mathit{false}\,\}] \\ [m_2 \ \& \ tr_2^{pre} \ \& \ ev_2 \ \& \ tr_2^{post} = \{\,\mathit{true} \mid \mathit{false}\,\}], ..., \\ [m_k \ \& \ tr_k^{pre} \ \& \ ev_k \ \& \ tr_k^{post} = \{\,\mathit{true} \mid \mathit{false}\,\}]\}\end{gathered}$$

where m_i stands for a name of a measure, tr_i^{pre} and $\mathit{trend}_i^{post} \in$ {increasing, steady, decreasing} and $ev_i \in \mathcal{E}$, for $i = 1, 2, k$.

Let $D = \{t_1, t_2, \ldots, t_n\}$ be a set of n *transactions*. Each transaction $t \in D$ is unique (i.e., has a unique transaction ID), and contains a subset of items in Σ. More specifically, in our context, a transaction may contain, for example, all the *TE*-Fs of a patient's hospital admission.

The instances that contain pattern P define a group $D_P = \{t_j | P \in t_j\}$. We say that pattern P is a subpattern of pattern P', denoted as $P \subset P'$, if every item in P is contained in P'. Note that the empty pattern Φ defines the entire population. If P is a subpattern of P' ($P \subset P'$), then D_P is a supergroup of $D_{P'}$ ($D_P \supseteq D_{P'}$).

In the new dataset built upon *TE*-Fs, to derive *PTE*-Ps, a transaction represents the overall set of *TE*-Fs related to a single patient admission, represented as 1-*PTE*-Ps.

Extracting predictive patterns

We are interested in mining patterns that are predictive of class c, e.g., a patient pathological state. For pattern P, we can define a predictive pattern (or a rule) R: $P \Rightarrow c$ with respect to class label c. The confidence of R is the precision (or posterior probability of c in the group D_P). Note that confidence of $\Phi \Rightarrow c$ is the prior probability of c. We say that rule R': $P' \Rightarrow c'$ is a subrule of rule R: $P \Rightarrow c$ if $c' = c$ and $P' \subset P$.

Let $\Omega = \{P_1, ..., P_m\}$ be a set of patterns predictive of c.

The support of pattern P in D, denoted as *sup(P,D)*, is the fraction of transactions t in D that contains P.

$$supp(P) = \frac{|\{t|t \in D \wedge P \subseteq t\}|}{|D|} = \frac{|D_P|}{|D|}$$

Given a user-specified minimum support threshold σ, we say that P is a frequent pattern if *supp(P, D)* $\geq \sigma$.

Frequent pattern mining represents our first step in the direction of extracting predictive patterns. We adopt the Apriori algorithm to eliminate all the patterns below a minimum support threshold.

Then, we filter the remaining patterns through the binomial test, to keep only the ones that significantly improve the confidence of simpler patterns (i.e., all of its subpatterns), thus obtaining a set of MPPs. Using SVM as a classification model with respect to the given pathological state for patients, we perform an additional selection over the MPPs, to further eliminate non-essential and overlapping patterns, thus selecting a minimal set. The minimal set is evaluated according to how the Area Under the Precision-Recall Curve (AUPRC) improves for the considered classification.

Even though the Area Under the Receiver Operating Characteristics curve (AUROC) is currently considered the standard method to assess the accuracy of predictive distribution models, Davis and Goadrich [60] suggest that AUPRC is more informative than AUROC when dealing with highly unbalanced datasets. Given the fact that precision is directly influenced by class imbalance, the Precision-Recall curves are more suitable to highlight differences between models derived from highly imbalanced data sets. Indeed, the AUROC curves plot False Positive Rate (FPR) vs. True Positive Rate (TPR), and thus the effect of true negatives for unbalanced sets could bias the final result. For this reason, to judge and compare the quality of different sets of predictive patterns, we use the AUPRC.

Building compact sets of predictive patterns efficiently

To avoid the full pattern subset search, it is necessary to adopt a greedy approach that generates, examines, and selects the patterns level-wise, where

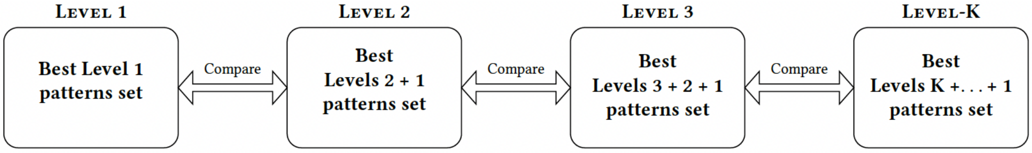

Level 1
Level 2
Level 3
Level-K
Best Level 1 patterns set
Compare
Best Levels 2 + 1 patterns set
Compare
Best Levels 3 + 2 + 1 patterns set
Compare
Best Levels K +. . . + 1 patterns set

Figure 4-4: The level-wise wrapper method used to extract PTE-Ps from [117].

a level k covers all k-patterns. For each level, we retrieve the set of MPPs; then, we use them to construct the final set of patterns greedily. We move from level 1 pattern sets to level k pattern sets. At each level k, we follow a bottom-up approach: we create the subset of patterns Ω starting from the most specific patterns (level k of patterns) and greedily adding to the set more general patterns of lower complexity (i.e., patterns of previous levels, from $k-1$ to 1). The reason for using the bottom-up greedy search process is that it tends to retain a greater number of the more specific patterns.

Figure 4-4 summarizes this level-wise approach: when all the k-level patterns have been tested, we start testing $\{k-1,\ldots,1\}$-level patterns in combination with the pattern set Ω. The greedy search algorithm on level j, with $k > j \geq 1$, works by first testing each j-pattern in combination with those in the current pattern set Ω. Each of these combinations is ranked in terms of the AUPRC score. This ranking defines the order in which all the j-level patterns are sequentially tested. If they successfully improve the AUPRC score, they are individually added to the resulting set of patterns Ω. When all the 1-patterns have been tested, Ω represents the best set of predictive patterns for level k. We compare its score against the one for level $k-1$: if it improves the overall score, we start to retrieve Ω for $k+1$-level patterns and we stop this process only if the score is not improving over the previous level, or if it is not possible to add new patterns at the current level.

The evaluation of patterns is performed as follows: the dataset D is split into the training and test sets (70% and 30%, respectively). All the pattern selection and learning are always done on the training set, using the test set only for the final evaluation. We use multiple internal validation splits of the training data to make the comparison. Tuples are reshuffled 10 times, and for every reshuffle 80% of the data is used as the internal training set while the remaining 20% is the internal validation set. The goodness for a specific set of patterns Ω is then estimated by averaging the AUPRC score for all internal splits obtained through reshuffling.

4.4 Experimental results: Discovering predictive trend-event patterns

4.4.1 System Configuration

To mine PTE-Ps on the MIMIC-III dataset, we develop a running prototype for mining analysis, named $PTEPminer$ (Predictive Trend-Event Pattern miner).

PTEPminer is a Java-based system that extracts *PTE*-Ps and allows the user to configure multiple parameters, such as: Δ_y, $\max\Delta_{start}$, $\max\Delta_{VT}$, $\min\Delta_{duration}$, $\max\Delta_{duration}$, *minNumTuples*, max increase (defined as max hourly increase), and the name of attributes corresponding to patient ID, *VT*, trend parameters, and events. Moreover, it allows us to define the percentage of the training set, as well as the number of internal reshuffling for the training set. PTEPminer is tested on a machine with an Intel Core i7-6700 and 32 GB of RAM, equipped with Windows 10 64-bit, Java 13, and PostgreSQL 12.

4.4.2 Dataset and Data Transformation

Our methodology has been applied in the clinical domain of the Intensive Care Unit (ICU), to mine MIMIC-III data [100], aiming at identifying *PTE*-Ps predictive of sepsis.

An ETL process was necessary to transform the MIMIC-III raw data into a form that we could use to mine *PTE*-Ps. To obtain *TE*-Fs, we use five different tables.

Chartevents, contains all the vital signs of the admitted patients, and we used it to extract measurements of heart rate, diastolic and systolic blood pressure, white blood cells, body temperature, and also the ID of the admitted patient associated to them.

Table Inputevents_MV contains information about any fluid administrated to the patients, and we used it to extract some of them. In particular, we decided to analyze some drugs usually administered to patients with sepsis [29], such as: phenylephrine, norepinephrine, vancomycin, pantoprazole, piperacillin, dopamine, ciprofloxacin, epinephrine, and vasopressin. Moreover, we also extracted more generic medications, such as metoprolol, potassium chloride, and omeprazole.

Table D_Items allows one to label every event or measure related to a patient correctly.

Finally, the fourth table Patients is needed to extract only *adult* patients, while the fifth one, Admissions, is used to verify that every measure and event happened within the admission period.

These tables are joined together, grouping measures and events for each patient and ordering them by *VT*. If multiple drugs are given to a patient at the same time (i.e., same *VT*), they are grouped together and not treated as separate events. For example, if a patient received ibuprofen and atenolol at the same *VT*, their influence on vital signs are analyzed together even if the medications are stored in two different records.

To summarize, there is a new table for each vital sign considered, and all the tables contain four attributes: *PatientID*, *VT*, *Event*, and *Measure*.

From these tables, we obtain *TE*-Fs, which need some further transformations to become *PTE*-Ps. Predictive trend-event patterns are retrieved from a single new table that needs to be created. In this table, each row represents a single patient admission, while the attributes are made by all the discovered *TE*-Fs. Finally, the class to predict must be added as another attribute. To this extent, we extract from table DIAGNOSES_ICD diagnoses related to sepsis and severe sepsis cases.

4.4.3 Results

For our test, we tune PTEPminer to extract *TE*-Fs with the parameters shown in Table 4.2 and 4.3. The first table contains values used independently from the vital signs considered, while the second one focuses on the maximum hourly increase, in percentage, that we used to label a trend as increasing, steady, or decreasing, respectively.

max Δ_{start}	**max** Δ_{VT}	**min** $\Delta_{duration}$	**max** $\Delta_{duration}$	Δ_y (%)	**min** ***num Tuples***
2 h	6 h	2 h	12 h	10	6

Table 4.2: Running parameters for PTEPminer.

	HR	**DBP**	**SBP**	**WBC**	**Temp**
Max Hourly Increase (%)	6.0	2.5	2.5	6.0	0.5

Table 4.3: Values used to define the max hourly increase for each vital sign considered.

We were able to retrieve 129 534 *TE*-Fs related to 2109 different patients' admissions. We identified 1433 different kinds of *TE*-Fsand used them to create the corresponding *PTE*-Ps. We then used PTEPminer to extract only the subset of *PTE*-Ps that maximize the prediction of sepsis. Let us note that, in our data, there were 603 diagnoses of sepsis or severe sepsis over 2109 admissions. Overall, 46 patterns were extracted in the most predictive set of *PTE*-Ps, with a score of 0.7690 obtained on the test data. This set was composed of 29 *1-patterns* and 17 *2-patterns*. Table 4.4 shows the top-ten patterns w.r.t. their absolute weight, which gives information about their relevance for the discrimination of the classes through SVM, along with the support, and precision of each pattern assigned by the final classification model.

In the set, we can identify 33 patterns (out of 46) that match exactly sepsis-related symptoms and/or treatments, and 11 more patterns related to Pantoprazole. Pantoprazole is a proton pump inhibitor (PPI) and, even though it is not used to treat sepsis, PPIs are used to treat stress-related mucosal damage (SRMD). SRMD is an erosive gastritis of unclear pathophysiology, which can occur rapidly after a severe insult such as trauma, surgery, *sepsis*, or burns [32]. In other words, it is still reasonable to mine patterns with Pantoprazole, because it is weakly related to sepsis. Finally, there are only three patterns that include one item that refers to an event we would consider weakly related to sepsis.

Pattern	Weight	Support	Precision
HR & Steady & Norepinephrine & Steady = true	0.6174	0.3381	0.5691
DBP& Steady & Pantoprazole & Steady = true ∧ DBP & Decreasing & Norepinephrine & Steady = true	0.4377	0.0278	0.7317
DBP& Steady & Vasopressin & Steady = true	0.3289	0.0237	0.5143
HR & Steady & Vasopressin & Steady = true∧ HR & Steady & Ciprofloxacin & Steady = true	0.3223	0.0501	0.7568
Systolic Blood Pressure & Increasing & Piperacillin & Steady = true	0.2403	0.0576	0.3882
HR & Steady & Norepinephrine & Steady = true ∧ DBP & Steady & Pantoprazole & Increasing = true	0.2277	0.0264	0.7179
HR & Steady & Norepinephrine & Steady = true ∧ DBP & Steady & Pantoprazole & Steady = true	0.2246	0.0556	0.6463
HR & Steady & 2xNorepinephrine & Steady = true	0.2231	0.1280	0.6772
DBP & Decreasing & Norepinephrine & Steady = true	0.2144	0.0976	0.5556
SBP & Steady & 3xPhenylephrine & Steady = true	0.2073	0.0386	0.5614
SBP & Increasing & 3xNorepinephrine & Steady = true	0.1949	0.0257	0.5789
HR & Steady & Vasopressin & Steady = true ∧ HR & Steady & 2xNorepinephrine & Steady = true	0.1865	0.0589	0.7701
SBP & Decreasing & Piperacillin & Steady = true ∧ HR & Steady & Norepinephrine & Steady = true	0.1849	0.0278	0.7317
HR & Steady & Piperacillin & Steady = ttrue	0.1841	0.2920	0.4362
HR & Steady & Norepinephrine & Steady = true ∧ DBP& Decreasing & Vancomycin & Steady = true	0.1784	0.0427	0.6984
DBP & Increasing & Vancomycin & Increasing = true	0.1689	0.0257	0.4737
SBP & Steady & Norepinephrine & Decreasing = true	0.1671	0.0860	0.6535
HR & Steady & Vancomycin & Steady = true	0.1640	0.5982	0.3681
SBP & Increasing & Piperacillin & Decreasing = true	0.1530	0.0332	0.4082
HR & Steady & Vancomycin—Piperacillin & Steady = true	0.1387	0.0420	0.4032
DBP& Increasing & 3xPhenylephrine & Steady = true	0.1364	0.0264	0.4103
HR & Steady & Pantoprazole & Steady = true ∧ DBP & Steady & Piperacillin & Decreasing = true	0.1305	0.0305	0.5778
DBP & Steady & Ciprofloxacin & Increasing = true	0.1292	0.0339	0.4400
HR & Steady & Pantoprazole & Steady = true ∧ DBP & Increasing & Phenylephrine & Decreasing = true	0.1237	0.0237	0.6286
SBP & Increasing & Phenylephrine & Steady = true	0.1179	0.0738	0.4128
SBP & Increasing & Phenylephrine & Steady = true ∧ HR & Steady & Piperacillin & Steady = true	0.1139	0.0312	0.6087
DBP & Decreasing & Piperacillin & Increasing = true	0.0919	0.0230	0.4118
DBP & Increasing & Vancomycin & Steady = true	0.0888	0.0732	0.3981
DBP & Steady & Piperacillin & Decreasing = true	0.0886	0.0508	0.4667
DBP & Steady & Piperacillin & Steady = true	0.0865	0.1016	0.4533
HR & Steady & Pantoprazole & Steady = true ∧ DBP & Steady & Potassium Chloride & Increasing = true	0.0852	0.0583	0.4651
HR & Decreasing & Norepinephrine & Steady = true	0.0796	0.0935	0.6014
DBP & Decreasing & Piperacillin & Steady = true	0.0722	0.0562	0.4819
HR & Steady & Vancomycin & Steady = true∧ HR & Steady & Potassium Chloride & Increasing = true	0.0657	0.0705	0.4615
HR & Steady & Pantoprazole & Steady = true∧ HR & Steady & Ciprofloxacin & Steady = true	0.0654	0.1152	0.4765
DBP& Steady & Potassium Chloride & Increasing = true	0.0612	0.1159	0.3509
SBP & Decreasing & Norepinephrine & Increasing = true	0.0574	0.0583	0.6047
SBP & Steady & Pantoprazole & Steady = true ∧ DBP & Decreasing & Norepinephrine & Steady = true	0.0489	0.0257	0.7105
DBP & Increasing & Vancomycin & Decreasing = true	0.0459	0.0318	0.3830
HR & Steady & Vancomycin & Steady = true∧ HR & Steady & Pantoprazole & Steady = true	0.0323	0.2541	0.4533
SBP & Decreasing & Norepinephrine & Decreasing = true	0.0235	0.0495	0.6164
HR & Steady & Vancomycin & Steady = true ∧ DBP & Steady & Pantoprazole & Steady = true	0.0227	0.0745	0.5091
HR & Steady & Ciprofloxacin & Steady = true	0.0197	0.2378	0.3875
SBP & Steady & Ciprofloxacin & Increasing = true	0.0072	0.0271	0.4250
HR & Steady & Pantoprazole & Steady = true	0.0057	0.3516	0.3815
HR & Steady & Vancomycin & Steady = true ∧ DBP & Increasing & Phenylephrine & Decreasing = true	0.0049	0.0359	0.5472

Table 4.4: The final set containing the most predictive trend-even patterns with their absolute weight, support and precision.

We compare results about our mined PTE-Ps with those using the wrapper predictive patterns (W-PPs) proposed in [118]. Even though we use AUPRC for evaluating the quality of the predictive rule set, we report also the AUROC values, in order to give a comprehensive view of the results.

Figure 4-5 summarizes the obtained results. We tested W-PPs on the same dataset, and we obtained an AUPRC of 0.5184 over 24 429 patients' admissions. We also test W-PPs using only those admissions with at least one valid TE-F, to compare W-PPs and PTE-Ps on the same set of admissions (W-PPs SA). In this case, AUPRC of W-PPs SA increases to 0.6718 but it is still lower than the one obtained for PTE-Ps. As opposed to PTE-Ps, W-PPs treat separately vital signs and drug administrations (i.e., an item refers only to a vital sign, or to a drug administration). Moreover, W-PPs are completely atemporal, thus ignoring any temporal evolution of the underlying data. It is fundamental to notice that the focus on this particular comparison should not be only on the overall score, but also on the intrinsic meaning that these two different kinds of patterns are offering.

Finally, we test a combination of PTE-Ps and W-PPs: in this environment,

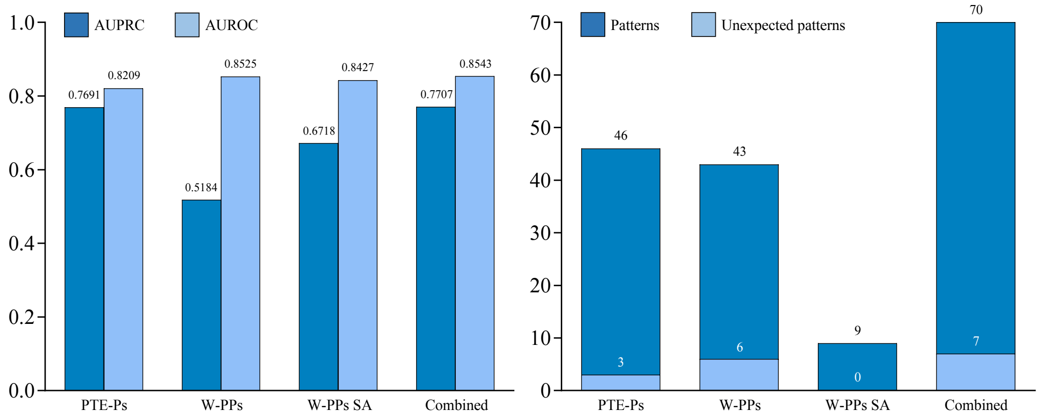


Figure 4-5: On the left, AUPRC and AUROC values for *PTE*-Ps, *W*-PPs and the combination of them. On the right, patterns and unexpected patterns: *PTE*-Ps, *W*-PPs and the combination of them.

PTEPminer decides what kind of predictive patterns to select. Compared to the best set of *PTE*-Ps, this setting slightly increases the overall AUPRC score to 0.7707, but it also increases the number of discovered patterns (70) as well as the ones weakly related to sepsis (7). The final set of these combined patterns is composed of 106 items, 65 are *PTE*-Ps and 41 *W*-PPs.

4.5 Conclusions

We introduce a new type of predictive temporal patterns, the PTEPs. They address the problem of extracting a compact set of complex predictive patterns, composed of interval-based data (i.e., trends) and instantaneous events, through a classification model. The new *temporal* knowledge associated to these predictive patterns determines a huge difference from the atemporal patterns discovered with previous approaches, and it makes difficult any kind of direct comparison between the two sets. Nonetheless, with these preliminary experiments, we show that it is also possible to use both kinds of patterns combined together. Anyhow, it is key to focus on the meaning of the mined sets of patterns and the importance of temporal knowledge, rather than a mere score comparison. Our results, although preliminary, show that *PTEPminer* could be an interesting tool for mining clinical *temporal* data. The findings presented here have the potential to be effective predictive patterns for sepsis. The results of the analyses represent a first step, which still needs to be validated in a more detailed and clinically-oriented way by clinicians. As a next step, we will explore with clinicians if there are particularly interesting patterns among the expected and the unexpected ones, and especially which importance they assign to them, according to their specific medical knowledge.

Chapter 5

A framework for the discovery of predictive temporal functional dependencies

In this chapter, 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. We then discuss the computational aspects of APFDs, the reliability of the derived APFDs, and report some results in deriving APFDs from real clinical data using MIMIC-III and MIMIC-IV datasets, related to patients from Intensive Care Units. Considering the temporal specification of explainability presented in Section 3.5 of Chapter 3, APFDs considers temporal data and temporal task, in this case, the prediction of the onset of Acute Kidney Injury (AKI). In addition, they focus also on temporal reasoning, supporting the explainability also towards specific temporal reasoning mechanisms.

Let us notice that in this work, we are aiming in mining APFDs, building a new framework which includes the formal definitions of the dependencies and the mining part, motivating and illustrating a 3-window model. We decided to focus on the proposal of the methodological framework and its novelties with respect to the use of TFDs to support the prediction task. As the framework has many specific aspects, we preferred to present the formal relational calculus-based definitions of the single general concepts together with a pointwise exemplification. As concerns, the theoretical aspects should be considered when introducing a new class of approximate TFDs, for example satisfiability, the logical implication and inference rules, and finally, the axiomatizability, were not covered. They could be interesting research directions, but out of the scope of this thesis.

5.1 Introduction

Data mining has been receiving considerable attention. 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. 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 [130, 157].

Knowledge from databases may be expressed by discovering patterns and data dependencies. Database dependencies express relevant characteristics of datasets, thereby enabling various critical analyses of data. Functional dependencies (FDs) are among the most important data dependencies for (relational) databases. In literature, there are different extensions of functional dependencies, from the temporal functional dependencies, which deal with data temporalities [180, 174, 97, 176, 47], to the approximate functional dependencies that hold on *most* tuples of a database [107, 90, 91, 114].

In a context where current systems enable us to store huge quantities of data, another important role of data mining is towards supporting 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., precision and recall are widely accepted in the information retrieval domain [12, 34], while in medicine the stakeholders prefer the ROC curve [57]. Often it is not possible to understand why machine learning algorithms are proposing specific predictions and such already known black-box problem [39] interferes with the communication between data scientists and domain experts, as the need for explainability is not fully supported. A workaround to overcome this problem is to exploit explainable temporal data mining techniques, able to possibly reveal intrinsic data dependencies. For example, focusing on the medical domain without loss of generality, through such techniques, clinical data sources would enable us to rapidly generate prediction models for thousands of clinical problems, support clinical decision-making, speed up medical processes, prevent and stratify risks, predict mortality, and improve the patient quality of life [129, 22, 19]. In this regard, discovering temporal patterns represents an explainable way of studying hidden data dependencies, supporting clinicians to focus on the most interesting and relevant discovered temporal data associations.

Moving to the database context, in the last decade Functional Dependencies (FDs), a well-known concept allowing the representation of dependencies between attribute sets in a database, received renewed attention [37, 157, 46, 24, 109] from different points of view and for different goals. Indeed, 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) [122, 159]. Indeed, data quality is becoming an urgent topic in the current context, where the huge amount of data are processed on a daily basis, often without any verifiable data quality process [65, 148]. 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 [38, 24, 107, 70, 109].

Temporal Functional Dependencies (TFDs) received some interest since the nineties, initially as a way for specifying constraints on temporal data [181, 47, 25], and, more recently, as a mining approach in their approximate version, looking for hidden temporal patterns inside data [46, 157, 54].

To the best of our knowledge, TFDs have not yet been considered for the prediction task. Such decision-support task is mainly devoted to the prediction of some (future) event based on a (past) data history. Thus, as time is an inherent feature of this task, TFDs are interesting candidates as a formal tool, for discovering the predictivity of the stored data. Within this context, in this chapter 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, to deal with the required approximation. The main novelty can be summarized in the formalization of a new framework to exploit the predictive aspect of the APFDs, according to the following specific aspects:

- We introduce a new temporal framework based on three temporal windows: observation window (OW), waiting window (WW), and prediction window (PW). The waiting window is explicitly introduced to create a time span before the prediction for being able to (possibly) manage the predicted event.

- We define and exemplify the entire framework for the approximate predictive functional dependencies (APFDs) in a formal way through declarative formulas in relational calculus. It allows the representation of temporal patterns (made by attribute values) related to a set of observed entities (e.g., patients) and characterizes their predictivity, with respect to a target attribute (e.g., a disease).

- We introduce two new error measures: H_3, focused on the number of entities we accept to lose, in order to deal with entities that create "noise" in the dataset; J_3, focused on the number of tuples for each entity we accept to lose, in order to satisfy the APFD, avoiding the situation where an entity remains with few tuples, so with without enough information to evaluate the prediction on it.

A preliminary and simplified proposal of this framework has been sketched in [10], in the context of medical data mining.

The chapter unfolds as follows. Section 4.2 contains the related work; in Section 5.2 we discuss a motivating scenario from the clinical field; in Section 5.3 we introduce the 3-window-based framework for prediction and the formalization of the PFDs; in Section 5.6 we discuss and evaluate the *goodness* of the predictivity of APFDs; finally, in Section 5.9 we draw some conclusions and sketch out possible directions.

5.2 A motivating scenario from Clinical Medicine

Nowadays, technology allows us to collect automatically huge amounts of medical information. A key point to be considered is the temporal component, necessary for a correct representation of the information within computer-based systems, for querying information, for reasoning about time, for the design process of analysis tools for prediction, for personalized medicine and finally for therapy support. To illustrate the relevance and the potential meaning of our approach, we consider a real-world example from the domain of Intensive Care Unit (ICU) focusing on patients suffering from Acute Kidney Injury (AKI) [171], used as a reference throughout the chapter.

Intensive care units provide critical care and life support for most severely ill and injured patients in the hospital. Patients are constantly monitored and subjected to laboratory tests, in order to detect the deterioration of conditions or the occurrence of various adverse events influencing the already fragile state. Clinicians record parameters such as the administered drugs, levels of different indicators for example blood urea nitrogen, calcium, chloride, creatinine, hemoglobin, platelet, potassium, prothrombin time, partial thromboplastin time, and white blood count, and finally different measures such as arterial blood pressure, heart rate, systolic blood pressure, diastolic blood pressure, respiratory rate, temperature, oxygen saturation and glucose.

In ICU, Acute Kidney Injury is a frequent clinical problem, characterized by a sudden loss of the ability of the kidneys to excrete wastes, concentrate urine, store electrolytes, and maintain fluid balance [162].

In 2012, KDIGO (Kidney Disease: Improving Global Outcomes) published specific guidelines [106] for the definition of AKI, where a patient receives the diagnosis if one of the following criteria is satisfied: (i) an increase in serum creatinine by ≥ 0.3 mg/dl (≥ 26.5 $\mu mol/l$) within 48 h, (ii) an increase in serum creatinine to ≥ 1.5 times baseline within the previous 7 days and (iii) a urine volume ≤ 0.5 ml/kg/h for 6 hours.

Figure 5-2 represents a simplified view of a real-world clinical database, containing ICU patients' data. Relations `VitalSigns`, `SpO`$_2$, `VitalSigns`, `Drug`, and `AKIDiag`, represent examples for clinical data related to vital sign monitoring, blood tests, drug therapies, and diagnosed AKI, respectively. Let us assume that the considered database, named after `IcuDB`, is a *temporal database*, i.e., a database composed by 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 [96].

As we are interested in discovering whether some clinical data features allow the early identification of AKI patients, let us assume that we derive through a suitable query the (possibly materialized) view `PatientHistory`. It represents different ordered states of patients, according to the different valid times, we would like to associate to a final state, specifying whether the patient has AKI. For each patient, `PatientHistory` stores the patient's name, the heart rate, the oxygen saturation, the administered drug– associated

Tuple #	**Patient**	**HR**	**VT**$_{HR}$	**SpO**$_2$	**VT**$_{SPO_2}$	**Drug**	**VT**$_{Drug}$	**AKI**	**VT**$_{AKI}$
1	Daisy	High	9	Low	11	Aspirin	13	False	18
2	Daisy	Low	2	High	4	Aspirin	6	False	18
3	Daisy	Low	2	High	4	Aspirin	6	False	12
4	Daisy	Medium	5	Medium	7	Indapamide	9	False	18
5	Luke	Low	7	High	8	Ibuprofen	12	True	17
6	Luke	Low	7	High	8	Ibuprofen	12	True	21
7	Luke	Medium	9	High	13	Sulindac	14	True	17
8	Luke	Medium	9	High	13	Sulindac	14	True	21
9	Stevie	High	1	Low	2	Aspirin	5	True	10
10	Stevie	High	1	Low	2	Aspirin	5	False	12
11	Stevie	High	1	Low	2	Aspirin	5	False	9
12	Stevie	High	1	Low	2	Indapamide	7	False	9
...	Stevie	...	...	...	...	...	...	...	...
36	Stevie	Medium	4	Medium	7	Metolazone	8	False	12

Figure 5-1: View `PatientHistory` storing data of a temporal query on database `IcuDB`

with the three considered states, respectively–, the diagnosis of AKI, and the different valid times, by means of attributes *Patient, HR, 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 time domain). Figure 5-1 (partially) shows a possible instance of `PatientHistory` describing a clinical history of three patients, Daisy, Luke, and Stevie, 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`. Moreover, some constraints have been specified, as we will discuss in the following, with respect to the allowed temporal distances between such ordered states and between them and the following AKI diagnosis. 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?*

- *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 health.

- *Which kinds of error thresholds are we allowed to specify for deriving reliable predictions?* Indeed, it could be that, after discovering that

`VitalSigns`

Tuple #	Patient	HR	BP	VT
1	Daisy	Low	Normo	2
2	Daisy	Medium	Normo	5
3	Daisy	High	Hypo	9
4	Luke	Low	Hyper	7
5	Luke	Medium	Hyper	9
6	Luke	Medium	Normo	15
7	Stevie	High	Hyper	1
8	Stevie	High	Normo	3
9	Stevie	Medium	Hyper	4
10	Stevie	High	Hyper	5

`SpO`$_2$

Tuple #	Patient	SpO_2	VT
1	Daisy	High	4
2	Daisy	Medium	7
3	Daisy	Low	11
4	Luke	High	8
5	Luke	High	13
6	Luke	Medium	18
7	Stevie	Low	2
8	Stevie	Medium	5
9	Stevie	Medium	6
10	Stevie	Medium	7

`Drug`

Tuple #	Patient	Drug	VT
1	Daisy	Aspirin	6
2	Daisy	Indapamide	9
3	Daisy	Aspirin	13
4	Luke	Ibuprofen	12
5	Luke	Sulindac	14
6	Luke	Indapamide	20
7	Stevie	Aspirin	5
8	Stevie	Indapamide	7
9	Stevie	Metolazone	8
10	Stevie	Indapamide	9

`AkiDiag`

Tuple #	Patient	AKI	VT
1	Daisy	False	9
2	Daisy	False	12
3	Daisy	False	18
4	Luke	True	14
5	Luke	True	17
6	Luke	True	21
7	Stevie	True	6
8	Stevie	False	9
9	Stevie	True	10
10	Stevie	False	12

Figure 5-2: Four (simplified) relations of temporal database `IcuDB`

some 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 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 [44].

In the following, we will use `IcuDB` data, depicted in Figure 5-2, to exemplify the different aspects of our proposal. Different examples will also consider some fragments of the overall database. 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.

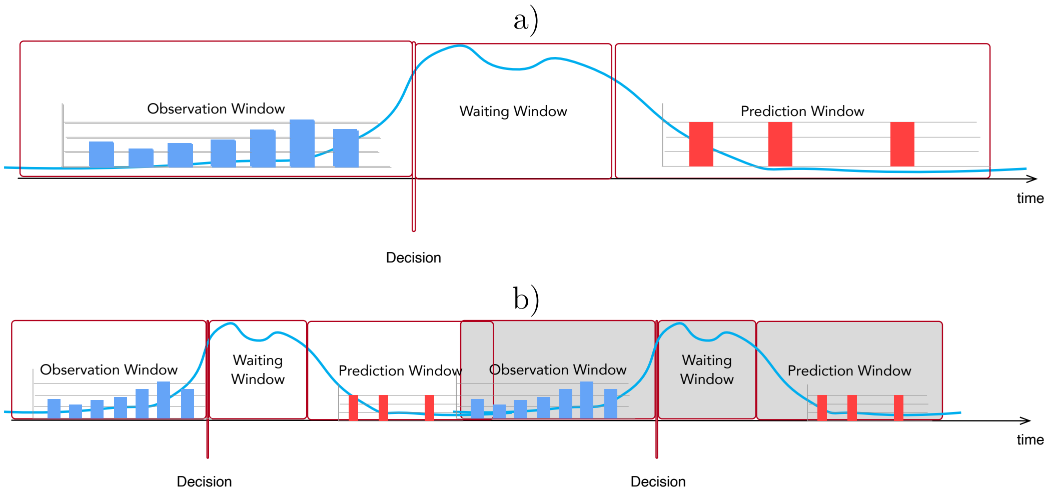


Figure 5-3: The time windows of the proposed framework: (a) the anchored and (b) the unanchored –sliding window– case.

5.3 The predictive aspects of functional dependencies

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

In general, the prediction models exploit the use of two-time windows, namely (i) a data collection (or observation) window, and (ii) a prediction window. Even though there are approaches [69, 142] which consider a third temporal window, to the best of our knowledge, a general and formal prediction framework considering three different time windows has not yet been considered in the data mining literature.

According to this view, depicted in Figure 5-3,

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

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

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

A first distinction is between (i) *anchored* and (ii) *unanchored* time windows. Indeed, with anchored time windows, we are able to represent specific periods of the considered time axis. An example of anchored time windows for the motivating scenario could consist of specifying OW as the first 4 hours from the admission to the ICU, the following 2 hours as WW (i.e., the fifth and sixth hour after the ICU admission), ending with the PW from the seventh to the tenth hour after the ICU admission. Figure 5-3 a) depicts the three anchored windows, the time-point corresponding to the decision moment, and a possible temporal evolution of some observed quantitative parameter, having some exponential behavior. On the other side, unanchored time windows represent windows that "move" through the time axis, constraining only the distance between the considered data. An example of such kind of windows for our scenario could consist in specifying again 4 hours, 2 hours, and 4 hours for OW, WW, and PW, respectively, but not anchored to any point of the time axis. Figure 5-3 b) represents two partially overlapping views, representing unanchored time windows. In this case, we may consider a possibly infinite number of unanchored (sliding) windows, even by specifying the width of the step size of sliding.

A second subtle distinction, which may provide different results for prediction and is orthogonal concerning the distinction between anchored and unanchored time windows, is between (i) *fixed-length* and (ii) *variable-length* time windows. Indeed, OW, and consequently the following WW and PW, could be either of fixed length, without any further constraint related to the temporal position of data inside it, or of variable length, and thus ending with the last time point associated to the data to consider in the window. In this case, the WW would start even before the maximum span allowed to OW. For example, we may require that valid times for HR, SPO$_2$, and Drug have to be within an OW of 6 hours, with a WW of two hours. Such OW could be of fixed length and thus we would consider AKI valid times after 8 hours from the beginning of OW. However, we could also require such valid times to be within the same maximum time span, but with an OW ending with the valid time of drug data. In this case, WW would possibly start even before the delay of 6 hours from the valid time of HR and consequently, the AKI valid time could be before 8 hours from the beginning of OW. In the following, unless differently specified, we will mainly consider fixed-length windows.

5.4 Defining Predictive FDs

Let us start by briefly recalling the concept of FD in the context of relational databases. 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 $\forall t, t' \in r(t[X] = t'[X], \rightarrow t[Y] = t'[Y])$.

As we will discuss in the following, the main idea here is to propose, step by step, a general framework allowing the definition of "specialized" functional dependencies having the antecedent composed by a set of attributes, called *predictive attributes*, ordered according to the corresponding valid times, and the consequent defined as the predicted boolean attribute. Let us consider a database DB as a set of temporal relational schemata $\{R_1, ..., R_n\}$ and a set of corresponding relations $\{r_1, ..., r_n\}$. Any schema R_i has attributes $ZU_i \cup \{VT\}$, where $\forall R_i, R_j$ with $i \neq j$ it holds $U_i \cap U_j = \emptyset$. U_i is a set of attributes representing properties of an entity, which is identified by attributes Z (hereinafter the entity attributes). VT is the attribute representing the temporal dimension of the tuples.

Definition 16 (State expression (SE))**.** *Given a set of relation schemata $R_1, ...R_n$ of database DB, a **State expression** with schema $ZU \cup \{VT\}$ where $U \subseteq U_iU_j...U_m$ is defined as:*

$$\begin{gathered}R^{SE} \equiv \{t \mid \exists t_i, ..., t_j, ...t_m(R_i(t_i)... \wedge R_j(t_j)... \wedge R_m(t_m) \wedge \\ t[Z \cup (U_i \cap U) \cup \{VT\}] = t_i[Z \cup (U_i \cap U) \cup \{VT\}]... \wedge \\ t[Z \cup (U_j \cap U) \cup \{VT\}] = t_j[Z \cup (U_j \cap U) \cup \{VT\}] \wedge \\ t[Z \cup (U_m \cap U) \cup \{VT\}] = t_m[Z \cup (U_m \cap U) \cup \{VT\}])\}\end{gathered}$$

Example 1. *As depicted in Figure 5-1, relation* `PatientHistory` *considers three different states for patients, where different attributes are considered for each state. The SE allowing us to focus on attribute HR is defined as*

$$HR^{SE} \equiv \{t : \{Patient, HR, VT\} \mid \exists t_1(\texttt{VitalSigns}(t_1) \wedge t = t_1[Patient, HR])\}$$

Let us now move to the definition of relational expression ables to represent the evolution of information through different temporal states.

Definition 17 (K-State evolution expression (KSE))**.** *Given a set of State Expressions $\mathcal{SE}$, a **K-State evolution expression** with schema $Z\overline{U}^0\overline{U}^1..\overline{U}^k \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k\}$ is defined as:*

$$R^{KSE} \equiv \Theta(R_0^{SE}, R_1^{SE}, ...R_k^{SE})$$

where $R_0^{SE}, R_1^{SE}, ...R_k^{SE} \in \mathcal{SE}$ and any R_i^{SE} has schema $ZU^i \cup \{VT^i\}$ with $0 \leq i \leq k$.

For any R^{KSE} the following formula holds

$$\begin{gathered}\forall t(R^{KSE}(t) \rightarrow (t[\overline{VT}^0] < t[\overline{VT}^1] \wedge .. \wedge t[\overline{VT}^{k-1}] < t[\overline{VT}^k] \wedge \\ \exists t_0(R_0^{SE}(t_0) \wedge t[Z\overline{U}^0 \cup \{\overline{VT}^0\}] = t_0[ZU^0 \cup \{VT^0\}]) \wedge \\ \exists t_1(R_1^{SE}(t_1) \wedge t[Z\overline{U}^1 \cup \{\overline{VT}^1\}] = t_1[ZU^1 \cup \{VT^1\}]) \wedge ... \wedge \\ \exists t_k(R_k^{SE}(t_k) \wedge t[Z\overline{U}^k \cup \{\overline{VT}^k\}] = t_k[ZU^k \cup \{VT^k\}]))\end{gathered}$$

Expression Θ allows the specification of different evolutions of State Expressions. While the definition forces some basic "structural" properties, the specification of how to compose such data evolution, which is heavily dependent on the considered application domain, is left to the designer of the KSE.

Tuple #	Patient	$\overline{HR}^0$	$\overline{VT}^0$	$\overline{SpO_2}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{VT}^2$
1	Stevie	High	1	Low	2	Indapamide	9
2	Stevie	High	1	Low	2	Metolamide	8
3	Stevie	High	1	Low	2	Indapamide	7
4	Stevie	High	1	Low	2	Aspirin	5
5	Stevie	Medium	4	Medium	5	Indapamide	9
..	...	...	...	...	...	...	...
34	Daisy	Low	2	High	4	Aspirin	13
35	Daisy	Low	2	High	4	Indapamide	9
..	...	...	...	...	...	...	...
53	Luke	Medium	15	Medium	18	Indapamide	20

Figure 5-4: An excerpt of KSE specified in Example 2 and evaluated on the temporal database `IcuDB` depicted in Figure 5-2

The basic properties we called structural are mainly related to (i) the association of different SEs according to the entity attributes, (ii) to the temporal order between SEs, whose attributes are (iii) suitably renamed according to the given order.

Among the different possible KSEs we distinguish here

1. simple KSEs;
2. metric KSEs, where constraints on the temporal distances between SEs are specified;
3. KSEs with "next" SEs;
4. KSEs with "next" SEs and metric constraints.

The previous definition of K-state evolution expression is the most simple one, allowing to join k ordered states of entities. In general, we may think of more complex expressions, still retaining the idea of representing k-state evolutions of entities but with more strict temporal requirements as those introduced by [47] for next views, where states are required to be one the next of the other (i.e., without any state in between).

Example 2. *Figure 5-4 depicts an excerpt of the instance of a simple KSE, composed by 3 SEs HR^{SE}, SpO_2^{SE}, $Drug^{SE}$, where HR^{SE} has been defined in Example 1, $SpO_2^{SE} \equiv$ `SpO2`, $Drug^{SE} \equiv$ `Drug`, with respect to the database* `IcuDB` *depicted in Figure 5-1. The KSE expression $\Theta(HR^{SE}, SpO_2^{SE}, Drug^{SE})$ corresponds to the simplest query, mainly requiring only the temporal order between the different states considered.*

$$
\begin{gathered}
SimPatEv^{KSE} \equiv \{t \mid t[\overline{VT}^0] < t[\overline{VT}^1] \wedge t[\overline{VT}^1] < t[\overline{VT}^2] \wedge \\
\exists t_0(HR^{SE}(t_0) \wedge t[Patient, \overline{HR}^0, \overline{VT}^0] = t_0[Patient, HR, VT]) \wedge \\
\exists t_1(SpO_2^{SE}(t_1) \wedge t[Patient, \overline{SpO_2}^1, \overline{VT}^1] = t_1[Patient, SpO_2, VT]) \wedge \\
\exists t_2(Drug^{SE}(t_2) \wedge t[Patient, \overline{Drug}^2, \overline{VT}^2] = t_2[Patient, Drug, VT])\}
\end{gathered}
$$

Tuple #	Patient	$\overline{HR}^0$	$\overline{VT}^0$	$\overline{SpO_2}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{VT}^2$
1	Stevie	High	1	Low	2	Aspirin	5
2	Stevie	High	3	Medium	6	Indapamide	7
3	Stevie	Medium	4	Medium	6	Indapamide	7
4	Stevie	High	5	Medium	6	Indapamide	7
5	Stevie	High	1	Medium	5	Indapamide	7
..	...	...	...	...	...	...	...
26	Daisy	Low	2	High	4	Aspirin	6
27	Daisy	Medium	5	Medium	7	Indapamide	9
..	...	...	...	...	...	...	...
31	Luke	Medium	15	Medium	18	Indapamide	20

Figure 5-5: An excerpt of KSE specified in Example 3 and evaluated on the temporal database `IcuDB` depicted in Figure 5-2

Tuple #	Patient	$\overline{HR}^0$	$\overline{VT}^0$	$\overline{SpO_2}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{VT}^2$
1	Stevie	High	1	Low	2	Aspirin	5
2	Stevie	Medium	4	Medium	5	Indapamide	9
..	...	...	...	...	...	...	...
35	Daisy	Low	2	High	4	Indapamide	9
..	...	...	...	...	...	...	...

Figure 5-6: An excerpt of KSE specified in Example 4 and evaluated on the temporal database `IcuDB` depicted in Figure 5-2

Example 3. *Figure 5-5 depicts an excerpt of the instance of a metric KSE composed by 3 SEs HR^{SE}, SpO_2^{SE}, $Drug^{SE}$, where HR^{SE} has been defined in Example 1, $SpO_2^{SE} \equiv \texttt{SpO}_2$, $Drug^{SE} \equiv \texttt{Drug}$, and $t[\overline{VT}^k] - t[\overline{VT}^{k-1}] < 3$ for $k = 1, 2$, for any tuple t of the KSE. The metric KSE expression $\Theta(HR^{SE}, SpO_2^{SE}, Drug^{SE})$ requires that the different states are within 4 time units from the previous one and corresponds to the following query*

$$MetPatEv^{KSE} \equiv \{t \mid t[\overline{VT}^1] - t[\overline{VT}^0] \leq 4 \wedge t[\overline{VT}^2] - t[\overline{VT}^1] \leq 4 \wedge$$
$$\exists t_0 \in HR^{SE}(t[Patient, \overline{HR}^0, \overline{VT}^0] = t_0[Patient, HR, VT]) \wedge$$
$$\exists t_1 \in SpO_2^{SE}(t[Patient, \overline{SpO_2}^1, \overline{VT}^1] = t_1[Patient, SpO_2, VT]) \wedge$$
$$\exists t_2 \in Drug^{SE}(t[Patient, \overline{Drug}^2, \overline{VT}^2] = t_1[Patient, Drug, VT])\}$$

Example 4. *Figure 5-6 depicts an excerpt of the instance of a KSE composed by 3 "next" SEs HR^{SE}, SpO_2^{SE}, $Drug^{SE}$, where HR^{SE} has been defined in Example 1, $SpO_2^{SE} \equiv \texttt{SpO}_2$, $Drug^{SE} \equiv \texttt{Drug}$, with respect to the database `IcuDB` depicted in Figure 5-1. The figures show only tuples of Figure 5-4 which satisfy also the further next condition. The KSE expression $\Theta(HR^{SE}, SpO_2^{SE}, Drug^{SE})$ corresponds to a query, mainly requiring the temporal order between the different states considered and the absence of other* *intermediate SEs.*

$$
\begin{gathered}
NextPatEv^{KSE} \equiv \{t \mid t[\overline{VT}^0] < t[\overline{VT}^1] \wedge t[\overline{VT}^1] < t[\overline{VT}^2] \wedge \\
\exists t_0, t_1, t_2(HR^{SE}(t_0) \wedge t[Patient, \overline{HR}^0, \overline{VT}^0] = t_0[Patient, HR, VT] \wedge \\
SpO_2^{SE}(t_1) \wedge t[Patient, \overline{SpO_2}^1, \overline{VT}^1] = t_1[Patient, SpO_2, VT] \wedge \\
Drug^{SE}(t_2) \wedge t[Patient, \overline{Drug}^2, \overline{VT}^2] = t_2[Patient, Drug, VT] \wedge \\
\neg \exists t_1', t_2'(SpO_2^{SE}(t_1') \wedge Drug^{SE}(t_2') \wedge \\
t[Patient, \overline{SpO_2}^1] = t_1'[Patient, SpO_2] \wedge \\
t[Patient, \overline{Drug}^2] = t_2'[Patient, Drug] \wedge \\
t[\overline{VT}^0] < t_1'[VT] \wedge t_1'[VT] < t[\overline{VT}^1] \wedge \\
t[\overline{VT}^1] < t_2'[VT] \wedge t_2'[VT] < t[\overline{VT}^2])))\}
\end{gathered}
$$

Once considered the predictive attributes, we turn now our attention to the predicted attribute. For sake of simplicity, we focus on a (single) boolean attribute, which will be used as the value representing the status to be predicted. The extension of our approach to a multiclass prediction is straightforward.

Definition 18 (Target expression (TE))**.** *Given a relation schema $R \in DB$ containing some boolean attribute B, a* ***Target expression*** *is defined as:* $R^{TE} \equiv \{t : Z\dot{B} \cup \{\dot{VT}\} \mid \exists t'(R(t') \wedge t[Z\dot{B} \cup \{\dot{VT}\}] = t'[ZB \cup \{VT\}])\}$

Mainly, with this definition, we are renaming and projecting the relation schema and the attribute, we will focus on as for prediction. It is worth observing that such renaming, together with that of KSEs, is required to possibly use the same relation schemata with different roles in the following definition of prediction expressions.

To observe the data for prediction, we use the 3-window framework, either through anchored windows, as in Definition 19, by setting the starting point of OW to a specific time point, or through sliding windows, as in Definition 20, by sliding the windows through the entire database time span.

Accordingly, we will consider two different kinds of *K-State Prediction Expressions* (KSPEs).

Definition 19 (K-State Prediction Expression with anchored windows)**.** *Given a K-State Evolution Expression R^{KSE} with schema $Z\overline{U}^0\overline{U}^1..\overline{U}^k \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k\}$, a target expression R^{TE}, with attributes $Z\dot{B} \cup \{\dot{VT}\}$, and three anchored time windows (i.e., intervals) OW, WW, PW (i.e., the observation window, the waiting window, and the prediction window, respectively), an anchored K-State Prediction Expression R^{aKSPE} is defined as:*

$$
\begin{gathered}
R^{aKSPE} \equiv \{t \mid R^{KSE}(t[Z\overline{U}^0\overline{U}^1..\overline{U}^k \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k\}]) \wedge \\
R^{TE}(t[Z\dot{B} \cup \{\dot{VT}\}]) \;\wedge t[\overline{VT}^0] \geq OW_s \wedge t[\overline{VT}^k] \leq OW_s + OW \wedge \\
t[\dot{VT}] > OW_s + OW + WW \wedge t[\dot{VT}] < OW_s + OW + WW + PW\}
\end{gathered}
$$

where OW_s denotes the anchor timepoint (i.e., the lower bound) of the observation window.

Definition 20 (K-State Prediction Expression with unanchored windows)**.** *Given a K-State Evolution Expression R^{KSE} with schema $Z\overline{U}^0\overline{U}^1..\overline{U}^k \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k\}$ and a target expression R^{TE}, with attributes $Z\dot{B} \cup \{\dot{VT}\}$, and three anchored time windows (i.e., intervals) OW, WW, PW (i.e., the observation window, the waiting window, and the prediction window, respectively), an unanchored K-State Prediction Expression R^{mKSPE} is defined as:*

$$\begin{gathered}R^{uKSPE} \equiv \{t \mid R^{KSE}(t[Z\overline{U}^0\overline{U}^1..\overline{U}^k \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k\}]) \wedge \\ R^{TE}(t[Z\dot{B} \cup \{\dot{VT}\}]) \ \wedge t[\overline{VT}^k] - t[\overline{VT}^0] \leq OW \wedge \\ t[\dot{VT}] - t[\overline{VT}^0] > OW + WW \wedge t[\dot{VT}] - t[\overline{VT}^0] < OW + WW + PW\}\end{gathered}$$

As we already underlined, without loss of generality, we are considered both anchored and unanchored fixed-length time windows. The versions of such prediction expressions with variable-length time windows would be obtained by replacing the conjunctions of Definitions 19 and 20

$$\begin{gathered}.. \wedge t[\dot{VT}] > OW_s + OW + WW \wedge t[\dot{VT}] < OW_s + OW + WW + PW.. \\ \textit{and} \\ .. \wedge t[\dot{VT}] - t[\overline{VT}^0] > OW + WW \wedge t[\dot{VT}] - t[\overline{VT}^0] < OW + WW + PW..\end{gathered}$$

$$\begin{gathered}\textit{with} \\ .. \wedge t[\dot{VT}] > OW_s + t[\overline{VT}^k] + WW \wedge t[\dot{VT}] < OW_s + t[\overline{VT}^k] + WW + PW.. \\ \textit{and} \\ .. \wedge t[\dot{VT}] - t[\overline{VT}^k] > WW \wedge t[\dot{VT}] - t[\overline{VT}^k] < WW + PW.., \textit{respectively}\end{gathered}$$

Moreover, such prediction expressions could be easily extended to consider only some specific target tuples within PW, e.g., the first one and/or the last one, and so on. It is worth pointing out that, in the definition of both metric KSEs and of KSPEs, we make use of a nonstandard (arithmetic) selection condition, containing for example, $(t[\overline{VT}^k] - t[\overline{VT}^0])$ or $(OW + WW + PW)$. However, such can be turned into a standard relational algebra/calculus expression, as shown in detail in [48].

Example 5. *Assuming a length of 6 hours for the OW, a length of 2 hours for the WW, and a length of 10 hours for the PW, we may specify a suitable KSPE for the prediction of AKI diagnosis, by considering relations of database* `IcuDB`*, SE HR^{SE}, TE $AkiDiag^{TE}$, simply obtained by suitably renaming attributes of the database relation AkiDiag, and KSE $MetPatEv^{KSE}$, specified in Examples 1 and 3, respectively.*

Tuple #	Patient	$\overline{HR}^0$	$\overline{VT}^0$	$\overline{SpO_2}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{VT}^2$	$\dot{AKI}$	$\dot{VT}$
1	Daisy	High	9	Low	11	Aspirin	13	False	18
2	Daisy	Low	2	High	4	Aspirin	6	False	12
3	Daisy	Low	2	High	4	Aspirin	6	False	18
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	21
8	Stevie	High	1	Low	2	Aspirin	5	True	10
9	Stevie	High	1	Low	2	Aspirin	5	False	12
...	...	...	...	...	...	...	...	...	...
19	Stevie	High	3	Medium	7	Metolazone	8	False	12

Figure 5-7: An excerpt of the instance of KSPE specified in Example 5, evaluated on the temporal database `IcuDB` depicted in Figure 5-2

$$
\begin{gathered}
AKIPred^{mKSPE} \equiv \\
\{t \mid MetPatEv^{KSE}(t[\{Pat, \overline{HR}^0 \overline{SpO_2}^1, \overline{Drug}^2, \overline{VT}^0, \overline{VT}^1, \overline{VT}^2\}]) \wedge \\
AkiDiag^{TE}(t[\{Pat, \dot{AKI}, \dot{VT}\}]) \wedge t[\overline{VT}^2] - t[\overline{VT}^0] \leq 6 \wedge \\
t[\dot{VT}] - t[\overline{VT}^0] > 6 + 2 \quad \wedge \quad t[\dot{VT}] - t[\overline{VT}^0] < 6 + 2 + 10\}
\end{gathered}
$$

Figure 5-7 depicts an excerpt of the evaluation of such KSPE on the database `IcuDB`*, using the SEs and KSEs specified in the previous examples.*

Example 6. *Let us consider the simple KSE $SimPatEv^{KSE}$ specified in Example 2. We may build a KSPE for AKI prediction, by specifying an OW of 6 hours a WW of 1 hour and a PW of 10 hours. The evaluation of such KSPE on* `IcuDB` *would return the relation (partially) depicted in Figure 5-1, with the corresponding attributes suitably renamed.*

5.4.1 Extending the target relation into an interval-based relation

Considering the predicted attribute, it could be interesting to observe some temporal properties over time intervals. For example, we can observe some temporal patterns which are valid within a period of interest.

Following, we define a new target relation, based on intervals, an extended version of K-State prediction Expression with anchored windows 22, and finally an extended version of K-State prediction Expression with unanchored windows 23.

Definition 21 (Interval-based Target expression ($\tilde{TE}$))**.** *Given a relation schema $R \in DB$ containing an attribute B (i.e. a pattern), an Interval-based*

Target expression $\tilde{TE}$ is defined as:

$$R^{\tilde{TE}} \equiv \{t : Z\dot{B} \cup \{\dot{VT}_{start}, \dot{VT}_{end}\} \mid \exists t'(R(t') \wedge t[Z\dot{B} \cup \{\dot{VT}_{start}, \dot{VT}_{end}\}] = t'[ZB \cup \{VT_{start}, VT_{end}\}])\}$$

Definition 22 (Extended K-State Prediction Expression with anchored windows)**.** *Given a K-State Evolution Expression R^{KSE} with schema $Z\overline{U}^0\overline{U}^1..\overline{U}^k \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k\}$, an Interval-based Target expression $R^{\tilde{TE}}$, with attributes $Z\dot{B} \cup \{\dot{VT}_{start}, \dot{VT}_{end}\}$, and three anchored time windows (i.e., intervals) OW, WW, PW (i.e., the observation window, the waiting window, and the prediction window, respectively), an extended anchored K-State Prediction Expression R^{aKSPE} is defined as:*

$$\begin{gathered}
R^{aKSPE} \equiv \{t \mid R^{KSE}(t[Z\overline{U}^0\overline{U}^1..\overline{U}^k \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k\}]) \wedge \\
R^{\tilde{TE}}(t[Z\dot{B} \cup \{\dot{VT}_{start}, \dot{VT}_{end}\}]) \wedge \\
t[\overline{VT}^0] \geq OW_s \wedge t[\overline{VT}^k] \leq OW_s + OW \wedge \\
t[\dot{VT}_{start}] > OW_s + OW + WW \wedge \\
t[\dot{VT}_{end}] < OW_s + OW + WW + PW\}
\end{gathered}$$

where OW_s denotes the anchor timepoint (i.e., the lower bound) of the observation window.

Definition 23 (Extended K-State Prediction Expression with unanchored windows)**.** *Given a K-State Evolution Expression R^{KSE} with schema $Z\overline{U}^0\overline{U}^1..\overline{U}^k \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k\}$ and an Interval-based Target expression R^{TE}, with attributes $Z\dot{B} \cup \{\dot{VT}_{start}, \dot{VT}_{end}\}$, , and three anchored time windows (i.e., intervals) OW, WW, PW (i.e., the observation window, the waiting window, and the prediction window, respectively), an extended unanchored K-State Prediction Expression R^{mKSPE} is defined as:*

$$\begin{gathered}
R^{uKSPE} \equiv \{t \mid R^{KSE}(t[Z\overline{U}^0\overline{U}^1..\overline{U}^k \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k\}]) \wedge \\
R^{\tilde{TE}}(t[Z\dot{B} \cup \{\dot{VT}_{start}, \dot{VT}_{end}\}]) \wedge \\
t[\overline{VT}^k] - t[\overline{VT}^0] \leq OW \wedge \\
t[\dot{VT}_{start}] - t[\overline{VT}^0] > OW + WW \wedge \\
t[\dot{VT}_{end}] - t[\overline{VT}^0] < OW + WW + PW\}
\end{gathered}$$

Example 7. *Assuming to use the same framework as in Example 5, a length of 6 hours for the OW, a length of 2 hours for the WW, and a length of 10 hours for the PW, we may specify a suitable KSPE for the prediction of AKI diagnosis, by considering relations of database* `IcuDB`*, SE HR^{SE}, $\tilde{TE}$ $AkiDiagIntervals^{\tilde{TE}}$ in Figure 5-8, simply obtained by suitably renaming attributes of the database relation $AkiDiagIntervals$, and KSE $MetPatEv^{KSE}$,*

Tuple #	Patient	$\dot{AKI}$	$\dot{VT}_{start}$	$\dot{VT}_{end}$
1	Daisy	FTFF	9	12
2	Daisy	TFFT	13	16
3	Daisy	FFTF	17	20
4	Luke	FTFF	14	17
5	Luke	FTFF	18	21
6	Luke	FFFT	22	25
7	Stevie	FFFT	6	9
8	Stevie	TTTF	10	13
9	Stevie	FFTF	14	17
10	Stevie	FTTF	18	21

Figure 5-8: A relation `AkiDiagIntervals` that represents different pattern diagnoses of four hours, on the temporal database `IcuDB`

Tuple #	Patient	$\overline{HR}^0$	$\overline{VT}^0$	$\overline{SpO_2}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{VT}^2$	$\dot{AKI}$	$\dot{VT}_{start}$	$\dot{VT}_{end}$
1	Daisy	High	9	Low	11	Aspirin	13	FFTF	17	20
2	Daisy	Low	2	High	4	Aspirin	6	FTFF	9	12
3	Daisy	Low	2	High	4	Aspirin	6	TFFT	13	16
4	Daisy	Medium	5	Medium	7	Indapamide	9	TFFT	13	16
5	Daisy	Medium	5	Medium	7	Indapamide	9	FFTF	17	20
5	Luke	Low	7	High	8	Ibuprofen	12	FTFF	14	17
6	Luke	Low	7	High	8	Ibuprofen	12	FTFF	18	21
7	Luke	Medium	9	High	13	Sulindac	14	TFFF	18	21
7	Luke	Medium	9	High	13	Sulindac	14	FFFT	22	25
8	Stevie	High	1	Low	2	Aspirin	5	FFTF	14	17

Figure 5-9: An excerpt of the instance of KSPE specified in Example 7, evaluated on the temporal database `IcuDB` depicted in Figure 5-2

specified in Examples 1 and 3, respectively.

$$
\begin{gathered}
AkiPredPattern^{mKSPE} \equiv \\
\{t \mid MetPatEv^{KSE}(t[\{Pat, \overline{HR}^0 \overline{SpO_2}^1, \overline{Drug}^2, \overline{VT}^0, \overline{VT}^1, \overline{VT}^2\}]) \wedge \\
AkiDiagIntervals^{\tilde{TE}}(t[\{Pat, \dot{AKI}, \dot{VT}_{start}, \dot{VT}_{end}\}]) \wedge \\
t[\overline{VT}^2] - t[\overline{VT}^0] \leq 6 \wedge t[\dot{VT}_{start}] - t[\overline{VT}^0] > 6 + 2 \wedge \\
t[\dot{VT}_{end}] - t[\overline{VT}^0] < 6 + 2 + 10\}
\end{gathered}
$$

Figure 5-9 depicts an excerpt of the evaluation of such KSPE on the database `IcuDB`, using the SEs and KSEs specified in the previous examples.

After having built such kind of expressions, allowing the association, for a given entity, of ordered state values and a final target attribute value, representing the state we want to predict according to the given time windows, we can, finally, introduce some special FDs, where the consequent corresponds to the target attribute and the antecedent is composed by a (sub)set of attributes representing different evolving states of the given entity. In other words, given

Tuple #	Patient	$\overline{HR}^0$	$\overline{VT}^0$	$\overline{SpO_2}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{VT}^2$	$\dot{AKI}$	$\dot{VT}$
1	Daisy	High	9	Low	11	Aspirin	13	False	18
2	Daisy	Low	2	High	4	Aspirin	6	False	18
3	Daisy	Low	2	High	4	Aspirin	6	False	12
4	Daisy	Medium	5	Medium	7	Indapamide	9	False	18
7	Luke	Medium	9	High	13	Sulindac	14	True	17
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	9
12	Stevie	High	1	Low	2	Indapamide	7	False	9
36	Stevie	Medium	4	Medium	7	Metolazone	8	False	12

Figure 5-10: A KSPE instance, subset of view `PatientHistory`, depicted in Figure 5-1 (with the attributes suitably renamed).

a KSPE, we introduce the definition of *Predictive Functional Dependency* as follows:

Definition 24 (Predictive Functional Dependency (PFD))**.** *Given a KSPE with schema* $Z\overline{U}^0\overline{U}^1..\overline{U}^k\dot{B}\cup\{\overline{VT}^0,\overline{VT}^1,..,\overline{VT}^k,\dot{VT}\}$*, or with schema* $Z\overline{U}^0\overline{U}^1..\overline{U}^k\dot{B}\cup\{\overline{VT}^0,\overline{VT}^1,..,\overline{VT}^k,\dot{VT}_{start},\dot{VT}_{end}\}$*, a Predictive Functional Dependency is an FD of the following form:*

$$\overline{X}^h\,\overline{S}^i\,...\overline{W}^j \rightarrow \dot{B}$$

$$with\ \ 0 \leq h < i < ... < j \leq k$$

$$where\ \overline{X}^h \subseteq \overline{U}^h, \overline{S}^i \subseteq \overline{U}^i, ..., \overline{W}^j \subseteq \overline{U}^j$$

and $\dot{B}$ *is the predicted attribute (boolean or pattern values)or the interval-based attribute.*

Example 8. *Let us consider the relation depicted in Figure 5-10, which contains a subset of tuples of view* `PatientHistory` *discussed in Section 5.2, with suitably renamed attributes. It is straightforward to observe that the PFDs* $\overline{HR}^0, \overline{SpO_2}^1 \rightarrow \dot{AKI}$ *and* $\overline{Drug}^2 \rightarrow \dot{AKI}$ *hold. On the other side, PFDs* $\overline{HR}^0 \rightarrow \dot{AKI}$ *and* $\overline{SpO_2}^1 \rightarrow \dot{AKI}$ *do not hold.*

5.5 Discovering Approximate PFDs

To mine PFDs, we need to deal with some kind of approximation, as it could happen that some PFDs hold on a subset of the given KSPE and we have to evaluate whether considering such subset is acceptable with respect to the prediction task supported by the considered PFDs. In other words, we require a PFD f to be satisfied by most tuples of a KSPE instance w. A very small portion of tuples of w is allowed to violate the dependency. Violating tuples are used to calculate an error measure g(f, w). If it is less than or equal to a given threshold ε, f is approximately satisfied on w. A number of methods have been proposed to compute the error measure [71]. In the context of predictive functional dependencies, we consider one of the measures proposed in [107] and introduce two error measures, specifically tailored to the predictive purpose of approximate PFDs.

Considering a KSPE instance w over a schema $Z\overline{U}^0\overline{U}^1..\overline{U}^k\dot{U}_p \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k, \dot{VT}\}$ and any set $s \subseteq w$, the first error measure G_3 considers the minimum number of tuples in w to be deleted to obtain a relation where the FD holds [107]. In our context, it is expressed as:

Definition 25 (Error measure G_3)**.** *Given a PFD $\overline{X}^h\overline{S}^i...\overline{W}^j \to \dot{B}$, a KPSE instance w with schema $Z\overline{U}^0\overline{U}^1..\overline{U}^k\dot{B} \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k, \dot{VT}\}$, where $\overline{X}^h \subseteq \overline{U}^h, \overline{S}^i \subseteq \overline{U}^i, \overline{W}^j \subseteq \overline{U}^j$, and any relation $s \subseteq w$, such that $s \models \overline{X}^h\overline{S}^i...\overline{W}^j \to \dot{B}$, the error measure $G_3$is expressed as:*

$$G_3(w, s) = |w| - |s|$$

The related *scaled measurement* g_3 is defined as:

$$g_3(w, s) = \frac{G_3(w, s)}{|w|}$$

Example 9. *Considering the KSPE instance (fragment) in Figure 5-7, the predictive functional dependency $\overline{HR}^0, \overline{SpO_2}^1 \to A\dot{K}I$ is not satisfied because of tuples 1, 8 and 9, where values "High" and "Low" for $\overline{HR}^0$ and $\overline{SpO_2}^1$ are associated both to "True" and "False" values of target attribute $A\dot{K}I$; and because of tuples 2,3, 5, and 6, where values "Low" and "High" for $\overline{HR}^0$ and $\overline{SpO_2}^1$ are associated both to "True" and "False". Looking at the minimum number of tuples to be deleted to obtain a relation where the PFD holds, we may delete tuples 2,3, and 8. In this case, g_3 is equal to 0.3.*

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

Definition 26 (Error measure H_3)**.** *Given a PFD $\overline{X}^h\overline{S}^i...\overline{W}^j \to \dot{B}$, a KPSE instance w with schema $Z\overline{U}^0\overline{U}^1..\overline{U}^k\dot{B} \cup \{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k, \dot{VT}\}$, where $\overline{X}^h \subseteq \overline{U}^h, \overline{S}^i \subseteq \overline{U}^i, \overline{W}^j \subseteq \overline{U}^j$, and any relation $s \subseteq w$, such that $s \models \overline{X}^h\overline{S}^i...\overline{W}^j \to \dot{B}$, the error measure H_3 is expressed as:*

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

The related *scaled measurement* h_3 is defined as:

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

Tuple #	Patient	$\overline{HR}^0$	$\overline{VT}^0$	$\overline{SpO_2}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{VT}^2$	$\dot{AKI}$	$\dot{VT}$
1	Daisy	High	9	Low	11	Aspirin	13	False	18
2	Daisy	Low	2	High	4	Aspirin	6	False	12
3	Daisy	Low	2	High	4	Aspirin	6	False	18
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
8	Stevie	High	1	Low	2	Aspirin	5	True	10
9	Stevie	High	1	Low	2	Aspirin	5	False	12
19	Stevie	High	3	Medium	7	Metolazone	8	False	12

Figure 5-11: An instance of KSPE, subset of the instance depicted in Figure 5-7.

Example 10. *Considering the KSPE instance in Figure 5-11, the PFD* $\overline{HR}^0$ *,* $\overline{SpO_2}^1 \rightarrow \dot{AKI}$*, is not satisfied because of tuples 1, 8, and 9, and of tuples 2,3,5, and 6. A first option for having such PFD satisfied would be to delete tuples 2,3, and 8. In this case, all the entities, i.e., patients, of the KSPE instance would be still represented, and, thus,* $h_3 = 0$*. A second option would consist in deleting tuples 5,6, and 8. In this case, Luke's tuples would disappear completely, and thus* h_3 *would be* $1/3$*.*

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

Definition 27 (Error measure J_3). *Given a PFD* $\overline{X}^h\,\overline{S}^i\,...\overline{W}^j \rightarrow \dot{B}$*, a KPSE instance* w *with schema* $Z\overline{U}^0\overline{U}^1..\overline{U}^k\dot{B}\cup\{\overline{VT}^0,\overline{VT}^1,..,\overline{VT}^k,\dot{VT}\}$*, where* $\overline{X}^h \subseteq \overline{U}^h, \overline{S}^i \subseteq \overline{U}^i, \overline{W}^j \subseteq \overline{U}^j$*, and any relation* $s \subseteq w$*, such that* $s \models \overline{X}^h\,\overline{S}^i\,...\overline{W}^j \rightarrow \dot{B}$*, the error measure* J_3 *is expressed as:*

$$J_3(w,s) = \max_{(v\in\{t[Z]|t\in s\})}\{|\{t[Z]|t\in w \wedge t[Z]=v\}| - |\{t[Z]|t\in s \wedge t[Z]=v\}|\}$$

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

$$j_3(w,s) = \max_{(v\in\{t[Z]|t\in s\})}\{\frac{|\{t[Z]|t\in w \wedge t[Z]=v\}| - |\{t[Z]|t\in s \wedge t[Z]=v\}|}{|\{t[Z]|t\in w \wedge t[Z]=v\}|}\}$$

Example 11. *Considering the instance of KSPE in Figure 5-12, the PFD* $\overline{HR}^0, \overline{SpO_2}^1, \rightarrow \dot{AKI}$ *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.*

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

Tuple #	Patient	$\overline{HR}^0$	$\overline{VT}^0$	$\overline{SpO_2}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{VT}^2$	$\dot{AKI}$	$\dot{VT}$
1	Daisy	High	9	Low	11	Aspirin	13	False	18
2	Daisy	Low	2	High	4	Aspirin	6	False	18
3	Daisy	Low	2	High	4	Aspirin	6	False	12
4	Daisy	Medium	5	Medium	7	Indapamide	9	False	18
5	Luke	Low	7	High	8	Ibuprofen	12	True	17
6	Luke	Low	7	High	8	Ibuprofen	12	True	21
7	Luke	Medium	9	High	13	Sulindac	14	True	17
8	Luke	Medium	9	High	13	Sulindac	14	True	21
9	Stevie	High	1	Low	2	Aspirin	5	True	10
10	Stevie	High	1	Low	2	Aspirin	5	False	12
11	Stevie	High	1	Low	2	Aspirin	5	False	9
12	Stevie	High	1	Low	2	Indapamide	7	False	9
36	Stevie	Medium	4	Medium	7	Metolazone	8	False	12

Figure 5-12: An instance of KSPE, corresponding to data depicted in Figure 5-1 for `PatientHistory`.

Definition 28 (Approximate Predictive Functional Dependency (APFD))**.** *Given a KPSE instance w with schema $Z\overline{U}^0\overline{U}^1..\overline{U}^k\dot{B}\cup\{\overline{VT}^0, \overline{VT}^1, .., \overline{VT}^k, \dot{VT}\}$, w fulfills the APFD $\overline{X}^h\,\overline{S}^i\,...\overline{W}^j \overset{\varepsilon}{\rightarrow} \dot{B}$ (written as $w \models \overline{X}^h\,\overline{S}^i\,...\overline{W}^j \overset{\varepsilon}{\rightarrow} \dot{B}$), where $\varepsilon =< \varepsilon_g, \varepsilon_h, \varepsilon_j >$ and $\overline{X}^h \subseteq \overline{U}^h, \overline{S}^i \subseteq \overline{U}^i, \overline{W}^j \subseteq \overline{U}^j$, if a relation $s \subseteq w$ exists such that $s \models \overline{X}^h\,\overline{S}^i\,...\overline{W}^j \rightarrow \dot{B}$ with $g_3 \leq \varepsilon_g \wedge h_3 \leq \varepsilon_h \wedge j_3 \leq \varepsilon_j$. In other words, $\varepsilon_g, \varepsilon_h, \varepsilon_j$ are the maximum acceptable errors defined by the user for g_3, h_3, and j_3, respectively.*

Example 12. *Suppose that our final goal is to preserve at least the 65% of the tuples ($\varepsilon_g = 0.35$), the 80% of the patients ($\varepsilon_h = 0.2$), and the 50% of the tuples for each patient ($\varepsilon_j = 0.5$). In Figure 5-13, the PFD $\overline{HR}^0, \overline{SpO_2}^1 \rightarrow \dot{AKI}$ is satisfied by considering a (sub)instance s by deleting tuples 2, 3, 9, 13, and 15. Thus, in this case, $g_3 = 5/16$, $h_3 = 1/3$, as tuples for Daisy would disappear; and $j_3 = 3/10$ as we delete tuples of Stevie, besides those of Daisy. It is easy to see that $g_3 < \varepsilon_g$, $j_3 < \varepsilon_j$, while $h_3 > \varepsilon_h$. On the other side, if we consider the instance s', by deleting tuples 5, 6, 9, 13, and 15, we would observe that the PFD is still satisfied, while $g_3 = 5/16$, $h_3 = 0$, and $j_3 = 2/4$. In this case, all the errors are below or equal to the given thresholds. Thus, we can say that $w \models \overline{HR}^0, \overline{SpO_2}^1 \overset{\epsilon}{\rightarrow} \dot{AKI}$ with $\epsilon \equiv< 0.35, 0.2, 0.5 >$.*

If we set the error thresholds as $\varepsilon_g = 0.35$, $\varepsilon_h = 0.4$, and $\varepsilon_j = 0.3$ (mainly we accept to discard some more patients, but we increase the number of tuples per patient we want to preserve), we can observe that $s \models \overline{HR}^0, \overline{SpO_2}^1 \rightarrow \dot{AKI}$, while $s' \not\models \overline{HR}^0, \overline{SpO_2}^1 \rightarrow \dot{AKI}$. Thus, $w \models \overline{HR}^0, \overline{SpO_2}^1 \overset{\epsilon}{\rightarrow} \dot{AKI}$ also with $\epsilon \equiv< 0.35, 0.4, 0.4 >$.

It is easy to prove that if $w \models \overline{X}^h\,\overline{S}^i\,...\overline{W}^j \overset{\varepsilon}{\rightarrow} \dot{B}$, it will also hold $w \models \overline{\overline{X}}^h\,\overline{\overline{S}}^i\,...\overline{V}^{i1}\,...\overline{\overline{W}}^j\,...\overline{Z}^{j1} \overset{\varepsilon}{\rightarrow} \dot{B}$, where $\overline{X}^h\,\overline{S}^i\,...\overline{W}^j \subset \overline{\overline{X}}^h\,\overline{\overline{S}}^i\,...\overline{V}^{i1}\,...\overline{\overline{W}}^j\,...\overline{Z}^{j1}$.

As an example, as $w \models \overline{HR}^0, \overline{SpO_2}^1 \overset{\varepsilon}{\rightarrow} \dot{AKI}$ for the KPSE instance w depicted in Figure 5-13, it is also the case that $w \models \overline{HR}^0, \overline{SpO_2}^1, \overline{Drug}^2 \overset{\varepsilon}{\rightarrow} \dot{AKI}$.

Tuple #	Patient	$\overline{HR}^0$	$\overline{VT}^0$	$\overline{SpO_2}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{VT}^2$	$\dot{AKI}$	$\dot{VT}$
2	Daisy	Low	2	High	4	Aspirin	6	False	18
3	Daisy	Low	2	High	4	Aspirin	6	False	12
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	17
8	Luke	Medium	9	High	13	Sulindac	14	True	21
9	Stevie	High	1	Low	2	Aspirin	5	True	10
10	Stevie	High	1	Low	2	Aspirin	5	False	12
11	Stevie	High	1	Low	2	Aspirin	5	False	9
12	Stevie	High	1	Low	2	Indapamide	7	False	9
13	Stevie	High	1	Low	2	Indapamide	7	True	10
14	Stevie	High	1	Low	2	Indapamide	7	False	12
15	Stevie	High	1	Medium	5	Indapamide	7	False	9
17	Stevie	High	1	Medium	5	Indapamide	7	True	10
18	Stevie	High	1	Medium	6	Indapamide	7	True	10
36	Stevie	Medium	4	Medium	7	Metolazone	8	False	12

Figure 5-13: An instance of KSPE, where $\overline{HR}^0, \overline{SpO_2}^1 \xrightarrow{\epsilon} \dot{AKI}$ holds with $\varepsilon_g = 0.35$, $\varepsilon_h = 0.4$, $\varepsilon_j = 0.4$ and with $\varepsilon_g = 0.35$, $\varepsilon_h = 0.2$, $\varepsilon_j = 0.5$.

After adding the new attribute $\overline{Drug}^2$ in the antecedent, nothing changes for KSPE instance $s \subseteq w$, for which $\overline{HR}^0, \overline{SpO_2}^1 \rightarrow \dot{AKI}$ holds.

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

Definition 29 (Minimal APFD). *An APFD $\overline{X}^h\,\overline{S}^i ...\overline{W}^j \xrightarrow{\varepsilon} \dot{B}$ is minimal for w, if $w \models \overline{X}^h\,\overline{S}^i ...\overline{W}^j \xrightarrow{\varepsilon} \dot{B}$ and $\forall \overline{V} \subset \overline{X}^h\,\overline{S}^i ...\overline{W}^j$ we have that $w \not\models \overline{V} \xrightarrow{\varepsilon} \dot{B}$.*

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

Example 13. *Considering the KSPE w depicted in Figure 5-13, it is straightforward to observe that the following two APFDs hold for $\epsilon \equiv < 0.35, 0.4, 0.4 >$ and are minimal.*

$$w \models \overline{HR}^0, \overline{SpO_2}^1 \xrightarrow{\varepsilon} \dot{AKI}$$
$$w \models \overline{Drug}^2 \xrightarrow{\varepsilon} \dot{AKI}$$

As for the minimality of the first APFD, both $\overline{SpO_2}^1 \xrightarrow{\varepsilon} \dot{AKI}$ and $\overline{HR}^0 \xrightarrow{\varepsilon} \dot{AKI}$ cannot satisfy the first threshold, i.e., $g_3 \leq 0.35$

5.6 The computational aspects of APFDs

In this section, we discuss computational aspects related to the extraction of APFDs.

5.6.1 Computing APFDs

The computational approaches for deriving AFDs focus on error g_3 for the considered relation w, and thus, they mainly derive a maximal (sub) relation $s \subseteq w$ that satisfies the related FDs [107, 109]. In literature, there are several algorithms for the discovery of functional dependencies, among which there is TANE [91]. As for the first experimental evaluations, we adopted a suboptimal solution, on top of the well-known TANE [91] algorithm, a popular approximate functional dependency detection algorithm, customizing it to mine only approximate functional dependencies with a fixed consequent, the predicted attribute $\dot{B}$.

To find all minimal non-trivial dependencies, TANE works as follows. It starts the search from singleton sets of attributes and works its way to larger attribute sets through the set containment lattice level by level. When the algorithm is processing a set X, it tests dependencies of the form $X \backslash A \rightarrow A$, where $A \in X$. This guarantees that only non-trivial dependencies are considered.

In our proposal, we compute all the Approximate Predictive Functional Dependencies, considering the three errors, g_3, h_3, j_3.

Given a KSPE instance w and the predicted attribute $\dot{B}$, our approach is mainly based on the following steps:

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

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

It is well known the complexity of deriving AFDs is exponential in the number of attributes [91, 109], while the complexity of checking a single dependency is linear in the number of tuples (data complexity). In our proposal, as the "maximality" of s is related to a composite error threshold $\varepsilon {=} < \varepsilon_g, \varepsilon_h, \varepsilon_j >$ and many possible relations s would be derived to evaluate a single APFD, the data complexity needs to be deeply studied and evaluated. We will specifically deal with this issue in the next Section 5.6.2.

5.6.2 The (data) complexity of deriving an APFD

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

This section is devoted to the theoretical analysis of the complexity to derive a relation $s \subseteq w$ considering the error thresholds G_3 and H_3. We reduced the problem in hand to a general $3SAT$ problem, showing that checking an APFD considering all the three thresholds belongs to the class NP.

Before starting with the theoretical analysis let us recall that an instance of SAT problem is a logical formula formed by a conjunction of disjunctive clauses. Namely, each clause is a disjunction of literals, and the general formula is a conjunction of disjunctive clauses. Therefore, an instance of SAT is a conjunction of clauses, each of them representable as a set of literals. In the specific case of $3SAT$, each clause has exactly 3 literals [137].

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

Definition 30. *Given a relation $w \subset \mathbb{N}^3$, a natural number $0 \leq k < |w|$, and a natural number $0 \leq h < |\pi_Z(w)|$ we say that w admits a* conflict resolution *of order (k, h) if there exists a subset $w^- \subseteq w$ such that:*

1. *$|w^-| \leq k$*
2. *for every pair of triplets $(a, \dot{b}, z), (a', \dot{b}', z') \in w \setminus w^-$ if $a = a'$ then $\dot{b} = \dot{b}'$;*
3. *$|\pi_Z(w)| - |\pi_Z(w \setminus w^-)| \leq h$.*

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

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

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

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

For the sake of brevity, given a clause $\{a_1^i, a_2^i, a_3^i\}$ in $C = \{\{a_1^1, a_2^1, a_3^1\}, \ldots, \{a_1^n, a_2^n, a_3^n\}\}$ and an assignment $\sigma : \mathcal{A} \rightarrow \{0, 1\}$ we say that $\{a_1^i, a_2^i, a_3^i\}$ is homogeneous w.r.t σ, or simply *homogeneous* when σ is clear from the context,

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

if and only if $\sigma(a_1^i) = \sigma(a_2^i) = \sigma(a_3^i)$. Then, Problem 2 may be equivalently redefined as: given a set of clauses $C = \{\{a_1^1, a_2^1, a_3^1\}, \ldots, \{a_1^n, a_2^n, a_3^n\}\}$ deciding whether or not there exists an assignment σ for the variables in C that makes C satisfied and at most p clauses of C homogeneous w.r.t σ.

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

Theorem 1. *Problem 2 is NP-Complete [137].*

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

Theorem 2. *Problem 1 is NP-Hard.*

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

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

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

For the left-to-right direction, let us assume that $C = \{\{a_1^1, a_2^1, a_3^1\}, \ldots, \{a_1^n, a_2^n, a_3^n\}\}$ and p is a positive instance of Problem 2. Let $\mathcal{A}$ the set of all and only variables which appear in C. Thus, there exists an assignment $\sigma : \mathcal{A} \to \{0, 1\}$ and at most p distinct indexes $i_1, \ldots i_p$ such that $\sigma(a_1^{i_k}) = \sigma(a_2^{i_k}) = \sigma(a_3^{i_k})$ for each $1 \leq k \leq p$. Let $w_C^- = \{(a_j^i, 1, 2i) : \sigma(a_j^i) = 0\} \cup \{(a_j^i, 0, 2i+1) : \sigma(a_j^i) = 1\}$. Let us observe that $w_C^- \subseteq w_C$. For proving that w_C^- satisfies the three conditions of Definition 30 for the pair $(|w_C|, p)$ we need to prove the following useful property:

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

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

Now we are ready to prove that conditions 1., 2., and 3. of Definition 30 are satisfied by the pair $|w_C|$ and p and thus $(w_C, |w_C|, p)$ is a positive instance of Problem 1. Condition 1. of Definition 30 imposes that $|w_C^-| \leq |w_C|$ which is trivially satisfied since $w_C^- \subseteq w_C$. Condition 2. of Definition 30 imposes that for every pair of triplets $(a^i_j, \dot{b}, z), (a^{i'}_j, \dot{b}', z') \in w_C \setminus w_C^-$ if $a^i_j = a^{i'}_{j'}$, i.e., they represent the occurrence of the same variable possibly in two distinct clauses we have $\dot{b} = \dot{b}'$. Let us assume by contradiction that this is not the case, then there exists $(a^i_j, 0, z), (a^{i'}_{j'}, 1, z') \in w_C \setminus w_C^-$ for some $z, z' \in \{2, \ldots, 2n+1\}$ with $a^i_j = a^{i'}_{j'}$. By definition of w_C^- the fact that $(a^i_j, 0, z) \in w_C \setminus w_C^-$ means that $\sigma(a^i_j) = 0$ while $(a^{i'}_{j'}, 1, z') \in w_C \setminus w_C^-$ means that $\sigma(a^{i'}_{j'}) = 1$ since $a^i_j = a^{i'}_{j'}$ we have a contradiction.

Condition 3. of Definition 30 imposes that $|\pi_Z(w_C)| - |\pi_Z(w_C \setminus w_C^-)| \leq p$. Let us assume by contradiction that there exist $p+1$ distinct indexes $2 \leq i_1 < \ldots < i_{p+1} \leq 2n+1$ such that $i_j \notin \pi_Z(w_C \setminus w_C^-)$ for every $1 \leq j \leq p+1$. This means that for every $1 \leq j \leq p+1$ if i_j is even (resp., odd) then $(a^i_q, 1, i_j) \in w_C^-$ (resp., $(a^i_q, 0, i_j) \in w_C^-$) for each $1 \leq q \leq 3$ and thus by definition of w_C^- we have $\sigma(a^i_q) = 0$ for each $1 \leq q \leq 3$, thus the clause $i_j/2$ (resp., $(i_j - 1)/2$) is homogeneous w.r.t to σ.

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

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

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

Let us assume w.l.o.g. that w_C^- is minimal, that is for every $(a, \dot{b}) \in \pi_{A\dot{B}}(w_C^-)$ we have that there exists $(a, \dot{b}') \in \pi_{A\dot{B}}(w_C \setminus w_C^-)$ with $\dot{b} \neq \dot{b}'$. In other words, any tuple in $\pi_{A\dot{B}}(w_C^-)$ "conflicts" with at least one tuple in $\pi_{A\dot{B}}(w_C \setminus w_C^-)$. Under this assumption, we may easily prove that $\mathcal{A}' = \mathcal{A}$. Let us assume by contradiction that $\mathcal{A}' \subset \mathcal{A}$. Thus, there exists $a \in \mathcal{A} \setminus \mathcal{A}'$ such that $(a, 0), (a, 1) \in \pi_{A\dot{B}}(w_C^-)$. If we take $w_C^{=} = w_C^- \setminus \{(a, 0, z) : (a, 0, z) \in w_C^-\}$ we have that $w_C \setminus w_C^{=}$ admits a $(|w_C|, p')$ conflict resolution with $p' \leq p$ since, informally speaking, we are possibly "reducing" the size of w_C^-. By construction, we have that $\{(a, 0, z) : (a, 0, z) \in w_C^-\} \neq \emptyset$ because since $a \in \mathcal{A}$ we have that there exists at least one clause $\{a^i_1, a^i_2, a^i_3\}$ in C for which $a^i_j = a$ for some $j \in \{1, 2, 3\}$ and thus $(a, 0, 2i+1) \in w_C$. Thus, we can conclude that w_C^- is not minimal (contradiction). By having $\mathcal{A}' = \mathcal{A}$ we can now claim that f is also a completely defined assignment for C. Let us prove that f is an assignment that makes at most p clauses in C homogeneous. Let us assume by contradiction that f makes at least $p+1$ distinct clauses homogeneous and let $i_1 < \ldots < i_{p+1}$ be the indexes of such clauses. By construction and by minimality of w_C^-, let us assume that for every $1 \leq h \leq p+1$ either $(a_j^{i_h}, 0, 2i+1) \in w_C \setminus w_C^-$ for every $j \in \{1, 2, 3\}$ – in such a case $f(a_1^{i_h}) = f(a_2^{i_h}) = f(a_3^{i_h}) = 0$–, or $(a_1^{i_h}, 0, 2i) \in w_C \setminus w_C^-$ for every $j \in \{1, 2, 3\}$ – in such a case $f(a_1^{i_h}) = f(a_2^{i_h}) = f(a_3^{i_h}) = 1$. This means that for each $1 \leq h \leq p+1$, if $f(a_1^{i_h}) = f(a_2^{i_h}) = f(a_3^{i_h}) = 1$, we have $2i_h \in \pi_Z(w_C \setminus w_C^-)$ and $2i_h + 1 \notin \pi_Z(w_C \setminus w_C^-)$. Symmetrically, for each $1 \leq h \leq p+1$ if $f(a_1^{i_h}) = f(a_2^{i_h}) = f(a_3^{i_h}) = 0$ we have $2i_h \notin \pi_Z(w_C \setminus w_C^-)$ and $2i_h + 1 \in \pi_Z(w_C \setminus w_C^-)$. Let $U = \{2i_1, 2i_2 + 1, \ldots, 2i_{p+1}, 2i_{p+1} + 1\}$. We can conclude that $\pi_Z(w_C \setminus w_C^-) \cap U$ and $\pi_Z(w_C^-) \cap U$ is a bi-partition of U with $|\pi_Z(w_C \setminus w_C^-) \cap U| = |\pi_Z(w_C^-) \cap U| = p+1$. Since we have $(\pi_Z(w_C^-) \cap U) \cap \pi_Z(w_C \setminus w_C^-) = \emptyset$ and trivially $\pi_Z(w_C^-) \cap U \subseteq \pi_Z(w_C)$, we have that $(\pi_Z(w_C^-) \cap U) \subseteq (\pi_Z(w_C) \setminus \pi_Z(w_C \setminus w_C^-))$ and, thus, $|\pi_Z(w_C^-) \cap U| = p+1$ $\leq |\pi_Z(w_C)| - |\pi_Z(w_C \setminus w_C^-)|$. Thus $|\pi_Z(w_C)| - |\pi_Z(w_C \setminus w_C^-)| \geq p+1$ (contradiction).

□

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

```
Algorithm 2: ApproximateDependencyCheck
Input: an instance $w$ of relation $W$, and three real numbers $\epsilon_{g_3}$, $\epsilon_{h_3}$, and
       $\epsilon_{j_3}$ in $[0,1]$
Output: a relation $s \subseteq w$ s.t. $s \models A \to \dot{B}$, $g_3(w,s) \geq 1 - \epsilon_{g_3}$,
        $h_3(w,s) \geq 1 - \epsilon_{h_3}$, $j_3(w,s) \geq 1 - \epsilon_{j_3}$
begin
    guess $s \subseteq w$
    for $v \in \pi_A(s)$ do                                   ▷ Check if $s \models A \to \dot{B}$
        if $|\pi_{\dot{B}}(\sigma_{A=v}(s))| \geq 2$ then
            fail
    if $\frac{|s|}{|w|} < 1 - \epsilon_{g_3}$ then                          ▷ Check $g_3(w,s)$
        fail
    if $\frac{|\pi_Z(s)|}{|\pi_Z(w)|} < 1 - \epsilon_{h_3}$ then                   ▷ Check $h_3(w,s)$
        fail
    for $z \in \pi_Z(s)$: do                                  ▷ Check $j_3(w,s)$
        if $\frac{|\sigma_{Z=z}(s)|}{|\sigma_{Z=z}(w)|} < 1 - \epsilon_{j_3}$ then
            fail
    return $s$
```

Algorithm 3: DeterministicADC

```
Input: an instance $w$ of the relation $W$, and three real numbers $\epsilon_{g_3}$, $\epsilon_{h_3}$,
       and $\epsilon_{j_3}$ in $[0,1]$
Output: a relation $s \subseteq w$ s.t. $s \models A \rightarrow \dot{B}$, $g_3(w,s) \geq 1-\epsilon_{g_3}$,
        $h_3(w,s) \geq 1-\epsilon_{h_3}$, $j_3(w,s) \geq 1-\epsilon_{j_3}$
▷ Prepare data for initial call according to epsilons
begin
    $del \leftarrow \lfloor \epsilon_{g_3} |w| \rfloor$
    $count \leftarrow \epsilon_{h_3} \lfloor |\pi_Z(w)| \rfloor$
    for $z \in \pi_Z(w)$: do
        $thresholds[z] \leftarrow \lfloor \epsilon_{j_3} |\sigma_{Z=z}(w)| \rfloor$
    return RecADC($w, del, count, thresholds$)
Function RecADC($w, del, count, thresholds$):
    ▷ This is the last recursive call before success
    if $w = \emptyset$ then
        return $\emptyset$
    let $a \in \pi_A(w)$
    ▷ For each value of B
    for $boolean\_val \in \{0,1\}$ do
        ▷ $del\_tuples$: tuples removed according to selection
        $del\_tuples \leftarrow \sigma_{A=a \wedge \dot{B}=boolean\_val}(w)$
        $s \leftarrow \sigma_{A=a \wedge \dot{B}=\neg boolean\_val}(w)$
        $out \leftarrow \{\}$
        for $z \in \pi_Z(del\_tuples)$: do
            $thresholds'[z] \leftarrow thresholds[z] - |\sigma_{Z=z}(del\_tuples)|$
            if $thresholds'[z] < 0 \leq thresholds[z]$ then
                $out \leftarrow out \cup \{z\}$
        ▷ $out$: the z groups that must disappear, since their
        tuples passed below the threshold $\epsilon_{j_3}$ in the current
        state
        if $count - |out| \geq 0$ then
            ▷ $count'$: represent the z groups still to be
            considered
            $count' \leftarrow count - |out|$
            $del\_tuples \leftarrow del\_tuples \cup \sigma_{Z=z:z \in out}(w)$
            if $del - |del\_tuples| \geq 0$ then
                ▷ If the final test succeeds, we proceed with the
                recursive call on the updated values
                $del' \leftarrow del - |del\_tuples|$
                $w' \leftarrow w \setminus (del\_tuples \cup s)$
                $s' \leftarrow$ RecADC($w', del', count', thresholds'$)
                if $s' \neq$ fail then
                    return $s \cup s'$
    return fail
```

Proved that the Problem 1 is *NP-Hard*, it is now necessary to find a deterministic algorithm that could stop the analysis of a relation, as soon as it verifies that the relation cannot satisfy the given APFD. Algorithm 3 provides the pseudo-code of such algorithm. The general idea of this algorithm is searching for a solution considering one tuple at a time, until it is possible to generate a solution, which satisfies the selected thresholds. Throughout the code, w is the entire relation. $del, count, thresholds$ represent the counters that control the errors. del counts the number of remaining tuples, $count$ controls the number of remaining entities, and $thresholds$ verifies the number of remaining tuples for each entity. After a trivial check about the (non) emptiness of relation w, for each value $a \in \pi_A(w)$, we try one boolean value and verify the dependency, if it fails, we try the second boolean value and verify the dependency. If both choices failed, then the algorithm fails. If one of the boolean values satisfies the thresholds, we update the counters, building at every step an intermediate relation s', as long as the thresholds are satisfied.

5.7 Towards the quality of APFD: coverage and reliability

We face the issue of the "predictivity" of the discovered APFDs, discussing about different ways of evaluating the "quality" of our dataset and of the derived APFDs.

5.7.1 Dealing with unbalanced datasets

First of all, it is interesting to understand whether the approximation is consistent according to the predicted class (e.g., the patients diagnosed with AKI). As the data could be heavily unbalanced, it could of interest to understand whether the approximation derived a relation s which is similar to the original relation w, as for the number of tuples with true/false values for the predicted attribute.

We could calculate the ratio of "true" tuples in w, i.e., $p_1 = \frac{|\{t|t \in w \land t[\dot{B}]\}|}{|w|}$, then calculate the ratio of "true" tuples in s, i.e., $p_2 = \frac{|\{t|t \in s \land t[\dot{B}]\}|}{|s|}$, and define p as the ratio between these two values. p could be a useful indicator to check if the ratio of "true" tuples in s did not heavily change with respect to the "true" tuples in w.

Figures 5-14 and 5-15 report the combinations of values for HR and SPO_2 and the related number of "true" and "false" tuples with respect to AKI, for KSPEs discussed in Examples 5 and 6, respectively. In both cases we have five possible combinations of values; the ratio between tuples related to AKI and the total number of tuples is 8/36 for relation of Figure 5-15, and 5/19 for that in Figure 5-14. As it is usual in medical domains, both datasets are unbalanced. However, it is important to observe as this unbalancedness may vary (from 22% to 26%) even according to slightly different evolution expressions for the same database. For example, in our case, from the same small database examples

HR	SPO_2	AKI-true	AKI-false
High	Low	1	2
High	Medium	1	9
Low	High	2	2
Medium	High	1	0
Medium	Medium	0	1

Figure 5-14: Attributes values for HR and SPO_2 with the associate number of true and false tuples for AKI, related to the KSPE discussed in Example 6

HR	SPO_2	AKI-true	AKI-false
High	Low	2	5
High	Medium	2	12
Low	High	2	2
Medium	High	2	0
Medium	Medium	0	9

Figure 5-15: Attributes values for HR and SPO_2 with the associate number of true and false tuples for AKI, related to the KSPE discussed in Example 5

and with two different KSEs, we obtain two different KSPE instances, where in the first case the value combination *Medium–Medium* for attributes HR and SPO_2 is present in 9 tuples (Figure 5-14), while in the second case is appearing in just one tuple (Figure 5-15).

5.7.2 Dealing with reliability

Moreover, we have to pay attention both to the number of distinct attribute value combinations of tuples of s and to the number of tuples corresponding to such value combinations. Indeed, if there is a reasonable number of tuples for each attribute value combination, we may say that the prediction is “reliable”. Instead, if there are some value combinations having a very low number of corresponding tuples in s, we obtain a prediction that is focused on specific combinations, possibly not completely reliable.

On the other side, if s has a restricted number of predictive attribute value combinations, it may be that the prediction is not providing any hint for too many possible missing attribute value combinations. In the examples provided in Figures 5-14 and 5-15 we may obtain either a relation s by taking only one value combination (over 5) for AKI patients or a relation with two different value combinations for AKI patients. Even though the original database is small, it shows that we need to deeply analyze also the data underlying a given APFD.

These two aspects together lead to consider the entropy [166] of both s and w. The general formula for the entropy is

$$H = -\sum p_i \, log \, p_i$$

where in our case i is the number of combinations.

In general the entropy of s is higher than the entropy of w, because deleting tuples that do not hold for the set of APFDs , we loose combinations that arise the entropy. Going into details, the value of entropy could give an idea of the combinations distribution. A high entropy means having different value combinations where the distribution of occurrences is very similar. A low entropy means having cases where the number of occurrences is very high, and cases where the number of occurrences is very low. Even in this case, analyzing the ratio between the entropy of w and the entropy of s can be a further way of describing our dataset, in order to obtain an intuition of the lost combinations and the probability distribution.

For example:

Example 14. *For the KSPE instance represented in Figure 5-14 the entropy of w is 2.3, while that of s is 2.001; for the KPSE instance represented in Figure 5-15 the entropy of w is 2.4, while that of s is 2.05.*

5.8 Experimental results: Discovering Approximate predictive functional dependencies (APFDs)

5.8.1 Data preparation

Especially in medicine, the new generation systems collect a considerable amount of data. These systems monitor the patient status, continuously providing new data to clinicians, which have to be analyzed and understood.

We evaluated the use of APFDs in the context of Intensive Care Unit (ICU) with Acute Kidney Injury (AKI). AKI is described as sudden loss of kidney function characterized by an increasing in serum creatinine levels and a reduced urinary output, involving not only the kidney excretory function, but also tissue injury [104]. As data source for the analyses, we use two different databases: Medical Information Mart for Intensive Care III (MIMIC-III) and Medical Information Mart for Intensive Care IV (MIMIC-IV). MIMIC-III [100] contains data associated with 53.423 distinct hospital admissions for adult patients who were hospitalized to the critical care units of the Beth Israel Deaconess Medical Center from 2001 to 2012. It is a collection of 26 tables, regarding demographic data, vital sign measurements made at the bedside, laboratory test results, procedures, medications, observations and notes charted by care providers, procedure and diagnostic codes, imaging reports, and mortality. MIMIC-IV dataset [99] is a public database of patients admitted to the Beth Israel Deaconess Medical Center (BIDMC) in Boston, USA. It contains de-identified data of patients admitted to ICU or the emergency department (ED) between 2008 and 2019.

During this preliminary phase, it could be necessary to implement a procedure useful to prepare the data for the next steps. Within this procedure, we may have to deal with issues such as missing values, temporal aggregation (if needed), granularity definition, categorization of numerical values, and last but not least, create an evolving diagnosis over the time span. Not all the issues are always present, and if present, may have different shades according to the problem in hand, especially for the evolving diagnosis. Following, we detail the different issues and possible solutions.

Missing values

The task of temporal modeling in electronic health record (EHR) data is very challenging because the data is multivariate and the time series for clinical variables are acquired asynchronously, which means they are measured at different time moments and are irregularly sampled in time.

In general a variable can be regarded as missing if the value of the variable (outcome or covariate) for the patient is not observed. In contexts as ICU, where patients are constantly monitored, not all values are recorded.

Considering the diagnosis of AKI, creatinine and urine are the primary measures in order to diagnose the onset of the illness [106]. Creatinine values are obtained from laboratory exams, therefor if in the database there is a missing value, probably is a real missing information, maybe because clinicians did not prescribe a laboratory test. Instead, regarding urine, a missing value may depend on a record error in the database. In ICU patients are catheterized, and the monitoring is continuous, so we can assume to have all the values.

To overcome this problem, we use a function presented in [49], in which we consider the calculation of missing measures using a linear function, although our approach can be generalized for arbitrary functions, according to the matter in hand. By using a simple linear monotonic function, we assume that the measure increases linearly over time. Given a measure *M*, its value *v*, an interval-based coordinate *in*, and an interval *l* , the *Calc* function is defined as follows:

$$Calc(M,\ v,\ in,\ l) = \frac{v \cdot (end(\cap(l, in)) - start(\cap(l, in)) + 1)}{end(in) - start(in) + 1}$$

This function allows us to have all the urine values, suitable to evaluate the status of the patient.

Value aggregation and categorization

In ICU, multiple monitoring systems generate a large quantities of measurements. To evaluate a temporal evolution that involves a long time span, it could be useful to aggregate values according to maximum, minimum, or average functions every suitably long interval, predefined according to the problem in hand. This technique reduces the amount of data to deal with, without missing information from the constant patient's monitoring.

Another considerable aspect is the categorization for the numerical values. In a dataset, variables may be classified into two main categories: categorical and numeric, described as discrete or continuous. The categorization of numerical values, according to clinical literature, reducing the variability, and homogenize the data, thus simplifying the discovery of recurrent temporal patterns. Again there is no loss of information, as we bring the data to a more generic level, inn order to facilitate the analysis.

Defining the granularity for the data

In large databases, it is often likely to deal with different granularities. In this framework it is important to deal with a single granularity which guarantees that the temporal evolution expressions are all based on the same time span.

In our case, a useful approach is to fix a shared granularity to all measurements, for example one granule could be represented by an interval of one hour, and we calculate the valid times as distances from a fixed baseline. The valid time of each measure is calculated as a difference from the admission to ICU, in terms of one-hour interval. This allows us to perform temporal differences because measures are all based on the same unit of time.

Evolving diagnosis

Depending on the disease's nature, observing the diagnosis only once at a specific time point may not be representative of the health status of the patient. For all diseases that can have an evolution over time, because they may be reversible or only a health condition, the instant diagnosis is not sufficient to provide a prediction. For example, monitoring the diagnosis over time, means having the possibility to observe how vital signs change according the health state, or a drug's influence over the time span.

Another important aspect to consider is the possibility to anticipate a sudden change in the patient's health, to preserve the subject's condition. Having a continuous monitoring of the diagnosis, enables to improve the medical condition, modify therapies, formulate a more precise prognosis and possibly anticipate complications.

Since our framework is based on building temporal evolutions that predict a future event, we think that is appropriate to study also the evolution over the time of the future event.

5.8.2 Experiments using MIMIC-III

System Configuration

To mine APFDs on the MIMIC-III dataset, we test the entire framework, from the ETL prcedures to the KSPEs on a server with 16 core, 12 GB of RAM, 1TB disk, equipped with Ubuntu 18.04 and Postgres 12. We mine the APFDs from the generated KSPEs on a machine with a 2,3 GHz Intel Core i9 8 core, 16 GB of RAM, equipped with macOS Catalina 10.15.7, Python 2.7. During the ETL procedure, to build a K-State evolution Expression, the computational time to obtain a table increases, increasing the number of temporal states, from seconds to hours (with seven states). While mining the APFDs, according to the K-State prediction Expression dimension, usually the algorithm takes few minutes.

Data extraction

To generate the cohort of patients, we perform an ETL process from the raw data of MIMIC -III, through an implementation via PostgreSQL and PL/pgSQL.

Initially, considering *PATIENTS* and *ICUSTAYS* tables, we extract the 50.711 patients admitted to ICU with age between 14 and 89, recorded the cohort into *TOT_ICUSTAYS* table.

Features extraction: aggregation, categorization

We extract the most relevant features to predict the risk of the AKI onset, and to understand which are the combinations of values that can be possibly significant in order to characterize the sick and healthy patients.

The main MIMIC-III tables used during the features extraction phase are: *ADMISSIONS, PATIENTS, LABEVENTS, CHARTEVENTS, OUTPUTEVENTS, D_ICD_DIAGNOSES, PRESCRIPTIONS, D_LABITEMS* and *D_ITEMS*. We consider the features in different groups:

- Demographics: Gender, age and ethnicity;
- Medications: Drugs administered to the patients during the ICU hospitalization. Four categories were considered: Diuretics, Non-Steroidal Anti-Inflammatory Drugs (NSAID), Angiotensin and Radiocontrast agents;
- Comorbidities: Presence of more than one disease or condition in the same patient at the same time such as congestive heart failure, peripheral vascular, hypertension, diabetes, myocardial infarction (MI), coronary artery disease (CAD), liver disease, cirrhosis, jaundice and sepsis;
- Chart-events: Vital signs measured at the bedside like diastolic blood pressure, glucose, heart rate, mean arterial blood pressure, respiration rate, SpO_2, systolic blood pressure and temperature;
- Lab-events: laboratory test result such as bicarbonate, blood urea nitrogen (BUN), calcium, chloride, creatinine, hemoglobin, international normalized ratio (INR), platelet, potassium, prothrombin time (PT), partial thromboplastin time (PTT) and white blood count (WBC).

The used itemIDs or labels for MIMIC-III dataset are reported in Tables 5.1 and 5.2.

First, by joining the cohort with *CHARTEVENTS* or *LABEVENTS* tables, we obtain the table which contains all the measurements of the considered features, recorded during the ICU stay of the patient. After that, we perform a categorization into 'low', 'medium', 'high', according to the clinical literature,

CATEGORY	FEATURES	LABEL IN MIMIC-III
Medications	Diuretics	”Hydrochlorothiazide”, ”Atenolol -Chlorthalidone” ”Chlorthalidone”, ”Eplerenone”, ”benazepril-hydrochlorothiazide”, ”olmesartan/hydrochlorothiazide”, ”Indapamide”, ”Triamterene-Hydrochlorothiazide”, ”hydrochlorothiazide”, ”Metolazone”, ”Torsemide ”, ”Furosemide”, ”Furosemide (Elixir)”,”Triamterene”, ”NEO* PO*Furosemide(10mg/1ml)”, ”NEO*IV*Furosemide”, ”Bumetanide”,”Ethacrynic acid”, ”Amiloride HCl”, ”Spironolactone”, ”NEO*PO* Spironolactone”, ”Eplerenone(INSPRA)”, ”HCTZ/Triamterene (Maxzide)”, ”triamterenehydrochlorothiazid”
	NSAID	”Aspirin”, ”Aspirin EC”, ”Aspirin Desensitization (AERD)”, ”aspirin”, ”Aspirin Desensitization (Angioedema)”, ”Aspirin (Rectal)”, ”Aspirin Desens”, ”Dipyridamole-Aspirin”, ”Butalbital-Aspirin-Caffeine”, ”Aspirin 325 mg or placebo ”, ”Aspirin 81 mg/ Placebo”, ”Aspirin 81 mg /Placebo”, ”Aspirin Childrens”, ”Aspirin (Buffered)”, ”Aspirin-Caffeine-Butalbital”, ”Aspirin Desensitization”, ”Aspirin 325mg or placebo ”, ”Aspirin 81 mg /placebo”, ”Aspirin 325mg/ placebo”, ”Excedrin Aspirin Free”, ”Aspirin 325 mg or Placebo”, ”Aspirin 81 mg or placebo ” ”Celecoxib”, ”celecoxib”, ”Celebrex”, ”diclofenac sodium”, ”Diclofenac Sodium DR”,”Voltaren-XR”, ”Diflunisal”, ”Etodolac”, ”Ibuprofen”, ”Naproxen”, ”Aleve”, ”Ibuprofen Suspension”, ”Indomethacin Sodium”, ”NEO*IV*Indomethacin Sodium”, ”Indomethacin”, ”Indomethacin XR”, ”Ketorolac”, ”Ketorolac Tromethamine”, ”ketorolac”, ”Ketorolac TroMETHamine”, ”Nabumetone”, ”nabumetone”, ”Piroxicam”, ”Salsalate”, ”Sulindac”
	Radio	”Ethiodol”
	Angiotensin	”amlodipine-benazepril”, ”Benazepril”, ”benazepril”, ”Benazepril HCl”, ”benazepril-hydrochlorothiazide”,”Lotensin”, ”Captopril”,”NEO*PO*Captopril”, ”Enalaprilat”, ”Enalapril Maleate”, ”Fosinopril”, ”Fosinopril Sodium”, ”Lisinopril”, ”Zestril”, ”Moexipril HCl”, ”Moexipril”, ”Quinapril”, ”Ramipril”, ”Altace”, ”Trandolapril”, ”Candesartan”, ”Candesartan Cilexetil”, Cilexetil”, ”candesartan”, ”Atacand”,”Atacand HCT”, ”Irbesartan”,”irbesartan”, ”*NF* Irbesartan”,”Avapro”, ”Losartan Potassium”, ”Losartan”, ”Hyzaar(Losartan/HCTZ)”, ”olmesartan/hydrochlorothiazide”, ”Olmesartan”,”olmesartan”, ”Benicar”, ”Benicar HCT”, ”Micardis HCT”, ”Micardis”, ”Valsartan”,”*NF*Valsartan”, ”Diovan”, ”Diovan/Hydrochlorthiazide”, ”Diovan HCT”, ”tabletsCandesartan”

Table 5.1: Labels in MIMIC-III for Medications

CATEGORY	FEATURES	ITEMID/ICD9_CODE IN MIMIC III
Comorbidity	Congestive heart failure	4280
	Peripheral vascular Hypertension	44389- 4439 - 74769 - l9972 - 74760 4019 - 4011 - 3482 - 36504 - 40591 - 40501 4010 - 4160 - 64212 - 64292 - 64291 - 40509 64211 - 64213 - 64214 - 40599 - 40511 - 40519 45930 - 64290 - 64293 - 64294 - 64210
	Diabetes	25083 - 25073 - 25082 - 25043 - 25063 - 25053 25093 - 25072 - 2535 - 25003 - 25062 - 25052 25042 - 25092 - 25002 - 24971 -25013 - 25081 24961 - 24951 - 24981 - 24941 - 24991 - 25071 25023 - 25011 - 25031 - 25033 - 25041 - 25012 25080 - 24911 - 24960 - 24950 - 24950 - 24990 25021 - 25061 - 25051 - 25091 - 25001 - 25070 25022 - 25020 - 25010 - 25060 - 25030 - 25050 25032 - 25040 - 25090 - 25000 - 24970 - 24920 24921 - 24910 - 24930 - 24931 - 24980 - 24940 24900 - 24901
	Liver disease	5718 - 5719 - 5728
	Coronary Artery Disease (CAD)	41412 - 41406 - 74685 - 4142 - 41401
	Cirrhosis	5716 - 5715 - 5712
	Jaundice	7824 - 7745 - 77431 - 7741 - 77439 - 77430 7744 - 7746 - 7742 - l7740
	Myocardial Infarction	41081 - 41091 - 4110 - 41082 - 41092 41080- 41090 - 41032 - 41030 - 412 41011 - 41001 - 41041 - 41021 - 41031 41052 - 41050 - 41181 - 41012 - 41010 41002 - 41000 - 41042 - 41040 - 41022 41020 - 41051 - 42979
	Sepsis	99591 - 99592 - 67020 - 67024 - 67022
Charted events	Heart rate	211, 220045
	Systolic blood pressure (SysBP)	5 - 442 - 455 - 6701 - 220179 - 220050
	Diastolic blood pressure (DiasBP)	8368 - 844 - 8555 - 220180 - 22005
	Mean arterial blood pressure (MeanBP)	456 - 52 - 6702 - 443 - 220052 220181 - 225312
	Respiration rate (Resprate)	618 - 615 - 220210 - 224690
	Temperature	223762 - 676 - 223761 - 678
	Sp02	646 - 2220277
	Glucose	807 - 811 - 1529 - 3745 - 3744 225664 - 220621 - 226537
Laboratory events	Bicarbonate	50882
	Blood Urea Nitrogen (BUN)	51006
	Chloride	50902 - 50806
	Creatinine	50912
	Hemoglobin	51221 - 50810
	International Normalized Ratio (INR)	51237
	Platelet	51265
	Potassium	50971 - 50822
	Prothrombin Time (PT)	51274
	Partial Thromboplastin Time (PTT)	51275
	White Blood Count (WBC)	51301 - 51300
	Calcium	50893
	Urine	40055 - 43175 - 40069 - 40094 - 40715 40473 - 40085 - 40057 - 40056 - 40405 40428 - 40086 - 40096 - 40651

Table 5.2: ItemIDs in MIMIC-III for comorbidity, charted events, and laboratory events.

Patient ID	Creatinine Value	Valid time	Category
38272437	2	2	high
38036146	2	264	high
39084108	1	23	high
32928326	1	41	normal
30673979	2	4	high
31695091	1	104	normal
30866972	2	180	high
34381316	6	81	high
30571521	2	44	high
34995591	1	151	high

Table 5.3: Creatinine table after the categorization.

Feature	Low	Medium	High
Bicarbonate	<23	23-28	>28
Blood Urea Nitrogen (BUN)	<5	5-20	>20
Calcium	<8.5	8.5-10.5	>10.5
Cloride	<96	96-106	>106
Creatinine	<0.6 (M), <0.5 (W)	0.6-1.2(M), 0.5-1.1 (W)	>1.2(M), >1.1 (W)
Hemoglobin	<14 (M), <12 (W)	14-18(M), 12-16 (W)	>18 (M), >16 (W)
Hematocrit	<40 (M), <36 (W)	40-54(M), 36-48 (W)	>54 (M), >48 (W)
International Normalized Ratio (INR)	<0.8	0.8-1.2	>1.2
Platelet	<150	150-400	>400
Potassium	<3.5	3.5-5.5	>5.5
Prothrombin Time (PT)	<10	10-13	>13
Partial Thromboplastin Time (PTT)	<25	25-35	>35
White Blood Count (WBC)	<4.5	4.5-11	>11

Table 5.4: Lab-events categorization according to clinical literature.

reducing the variability, thus simplifying the discovery of recurrent temporal patterns. For the creatinine and hematocrit measures, we categorize also according to the sex of the patient. The results are stored into a new column attribute, named as the feature considered. You can see an example in Table 5.3. Tables 5.5 and 5.4 display the thresholds used for the categorization of laboratory test and charted observations.

To assign a valid time to each measure, we calculate each valid time as a difference from the charttime of the measure and the admission to ICU, using intervals of 6 hours as granule. We divide the time span until the discharge from the ICU, or the death of the patient. Namely, we give the value 1, if the difference is from 0 to 5, the value 2 if the difference is from 6 to 11 and so on. For each icustay id, we compute the average, the minimum and the maximum of the values in each interval. We label the entire cohort of patients according to the KDIGO criteria, reported in Chapter 2.1 in two different ways. The first one is a punctual diagnosis, when the patient complies with at least one of the three criteria, we label this subject as AKI (1), otherwise NOT AKI (0). The second one is a persistence diagnosis, from the admission to the discharge from the ICU , we evaluate the conditions of the patient every specified time interval, and define the subject as AKI (1), or NOT AKI (0).

Feature	**Low**	**Medium**	**High**
Diastolic Blood Pressure (DiasBP)	<60	60-80	>80
Glucose	<72	70-108	>108
Heart Rate	<60	60-100	>100
Mean arterial Blood Pressure (MeanBP)	<70	70-100	>100
Respiration Rate	<12	12-20	>20
SpO_2	<96	96-100	>100
Systolic Blood Pressure (SysBP)	<90	90-120	>120
Temperature	<36.1 (celsius)	36.1-37.5 (celsius)	>37.5 (celsius)

Table 5.5: Chart-events categorization according to clinical literature.

Patient labeling

In this experimental set, we perform two different types of labelling, both related to the general KDIGO criteria. The first one is performed considering a single event for the diagnosis of the patient, the second one considering successive events of diagnosis, for the entire ICU hospitalization.

Single event labeling
Considering the first criterion (Increase in serum creatinine by $\geq$ 0.3 mg/dl ($\geq$ 26.5 $\mu mol/l$) within 48 hours) for identifying AKI patients, we use the cohort of patients recorded in *TOT_ICUSTAYS* and *LABEVENTS* table in order to obtain the *CREATININES* table, which collects all the serum creatinine values recorded during the hospitalization in ICU of each patient. Then, we create the *CREATININE_PATIENTS* table which contains for each *ICUSTAY_ID* the AKI flag (0,1), and the AKI_TIME attribute that corresponds to the CHARTTIME of the measurement during which there is a serum creatine increase. This increase is computed for every interval of 48 hours, starting from the admission to ICU until the end of stay. The number of patients for which there is an increase greater than or equal to 0.3 mg/dl within 48 hours is 11.184.

Afterwards, according to the second criterion (Increase in serum creatinine to$\geq$ 1.5 times baseline, within the previous 7 days), from the initial cohort in *TOT_ICUSTAYS* and *LABEVENTS* table, we create the *CREATININE_BASELINE* table. For each ICUSTAY_ID, we find the corresponding creatinine baseline which is equal to the mean of the serum creatinine values within 7 days prior the admission to the ICU. The execution of the queries in Figure 5-16 retrieved the AKI patients. The result is stored in the *AKI_BASELINE* table which includes 3.966 AKI patients.

At the end, we evaluate the third criterion (Urine volume <0.5 ml/kg/h for 6 hours). Firstly, we create the *ICUSTAY_WEIGHT* table with the average weight per ICU stay for each patient, joining the *CHARTEVENTS* and *TOT_ICUSTAYS* tables.

Then, by joining the *OUTPUTEVENTS, TOT_ICUSTAYS* and *ICUSTAY_WEIGHT* tables, we compute for each patient all the urine values with their corresponding CHARTTIME. The result, combined with the corresponding weight for each patient, is stored in the *URINE_MEASURES* table. Starting with the first urine measurement in the ICU stay, we calculate the urine

```
CREATE TABLE creatinine_baseline AS
SELECT I. icustay_id ,
ROUND ( cast (avg(C. valuenum ) AS numeric ) ,2) AS baseline
FROM mimic . labevents C, tot_icustays I
WHERE itemid =50912
AND C. hadm_id =I. hadm_id
AND C. charttime >= I. intime - interval '7 day '
AND C. charttime < I. intime
AND I. age between 14 AND 89
GROUP BY I. icustay_id ;

CREATE TABLE aki_baseline AS
SELECT I. icustay_id , min ( charttime ) AS aki_time
FROM mimic . labevents l, creatinine_baseline c, tot_icustays I
WHERE l. hadm_id = I. hadm_id
AND c. icustay_id = I. icustay_id
AND l. itemid = 50912
AND l. valuenum is not NULL
AND l. charttime >= I. intime
AND l. charttime <= I. outtime
AND l. valuenum >= 1.5* c. baseline
AND I. age BETWEEN 14 AND 89
GROUP BY I. icustay_id
ORDER BY I. icustay_id;
```

Figure 5-16: Queries to calculate the second criterion.

rate for each interval of 6 hours. We create the *URINE_PATIENTS* table which contains 21.887 AKI patients. We record for each ICUSTAY_ID the corresponding AKI flag (0 ,1) and the related AKI_TIME, namely the time of the last urine measurement in the 6 hours interval as AKI time. After retrieving all the AKI patients according to different KDIGO criteria, we combine the three obtained tables *CREATININE_PATIENTS, AKI_BASELINE and URINE_PATIENTS* into a single result table *TOT_PATIENTS* that contains 25.740 patients diagnosed with AKI. Here, for each ICUSTAY_ID, we store the INTIME and OUTTIME attributes that respectively represent the time when the patient is admitted to the ICU and when the patient is discharged from the ICU, the corresponding AKI_TIME and AKI values.

The left outer join between *TOT_ICUSTAYS, TOT_PATIENTS* and *ADMISSIONS* tables is carried out to find the ALL_PATIENTS table containing all the ICUSTAY_IDs with their corresponding diagnosis and some information about the patients, like HADM_ID, INTIME, OUTTIME, AGE, GENDER, ETHNICITY, AKI and AKI_TIME. The value of the AKI_TIME for the patients affected by AKI corresponds to the date and time during which the AKI was diagnosed, which otherwise is equal to NULL if the patient did not get AKI. We compute the AKI_TIME as a new value evaluated as a difference from the admission to ICU and the AKI_TIME, with a six hours interval as granule. For example, if the diagnosis is in the first six hours, the related VT is equal to 1.

Multiple events labeling

In the second part of the work we recompute the diagnosis, reproducing the persistence of the diagnosis according to the three KDIGO criteria. For the first and third criteria, we implement the queries using PL/pgSQL, whereas the second criterion is calculated using PostgreSQL.

For patients who die during the permanence in ICU, we consider only the values from the admission to ICU to the death time. For patients who are discharged from the ICU alive, we consider measures from the admission to the discharge from the ICU. We start considering the first criterion. For the patients hospitalized in ICU for a period of time less than 48 hours, we apply the criteria during this interval. Instead, for the patients who stay in ICU for a period of time longer than 48 hours, we calculate the increase of the serum creatinine every 48 hours, starting from the INTIME (date and time of the patient admission to the ICU), and moving this sliding window every 24 hours. If in the considered interval, there are less than 2 measurements, the increase could not be calculated, therefore the corresponding diagnosis is "NOT DEFINED". Otherwise, if there are at least two values, we calculate the criteria, defining the patients as "TRUE" if there is an increase in serum creatinine $\geq$ 0.3 mg/dl within 48 hours, otherwise "FALSE". Thus, for each ICUSTAY_ID, we store the 48 hours overlapped diagnoses in the ILL attribute which could take values "AKI", "NOT AKI", or "NOT DEFINED" with the corresponding START_INTERVAL and END_INTERVAL during which the increase is computed. The diagnosis occurs at the end of the interval, so in correspondence of the END_INTERVAL value.

For the second criterion, we calculate the BASELINE table containing for each ICUSTAY_ID the corresponding START_INTERVAL and END_INTERVAL, related respectively to the seventh day before the admission to the ICU and the INTIME, and the attribute ILL which took "TRUE" or "FALSE" value.

To find AKI patients according to the third criterion, we consider only the ICUSTAY_IDs that were hospitalized for a period of time greater than 6 hours, due to the definition of the criterion. Starting from the INTIME, we calculate the urine rate for each 6 hours interval, moving the sliding window every 3 hours. If the number of measurements inside the considered interval is at least two, then we assign the diagnosis, otherwise we apply the *Calc* function 5.8.1.

To solve the problem of overlapped intervals, we use the smallest granule (in our case 3 hours according to the third criterion), to define the diagnosis as follow (Table 5.6):

- if there is at least one "AKI" value, the final diagnosis will be "AKI";
- if there is a "NOT AKI" in an ambiguous situation, the final diagnosis will be "NOT AKI";

Diagnosis 1	**Diagnosis 2**	**Final diagnosis**
AKI	AKI	AKI
AKI	NOT AKI	AKI
AKI	NOT DEFINED	AKI
NOT AKI	NOT AKI	NOT AKI
NOT AKI	NOT DEFINED	NOT AKI
NOT DEFINED	NOT DEFINED	NOT DEFINED

Table 5.6: Definition of diagnoses overlap.

- if there is a double ambiguous situation, we will have the label "NOT DEFINED".

Discovering APFDs from MIMIC-III

For this experimental set, we consider the four categories of medications: diuretics, Non-Steroidal Anti-Inflammatory Drugs (NSAID), angiotensin and radiocontrast agents. For the *chartevents* we consider heart rate, SpO_2, and body temperature. For the *labevents*, we consider laboratory test of serum creatinine, urine and white blood cell count (WBC).

We choose a 3-window moving framework base on: an *observation window* of 36 hours, where we collect all the measures related to each patient, a *waiting window* of 12 hours where we do not consider any event, and then a *prediction window* of 72 hours, where there is the onset of the illness according to one of the KDIGO criteria, or the discharge from the ward when any criteria satisfied.

We generated four different KSPEs with unanchored widow, with three different Θ expressions. The first and the second KSPEs regard a single event as diagnosis (as we showed in section 5.4), instead the other two are related to the diagnosis expressed as an attribute diagnosis pattern (as we showed in section 5.4.1). Specifically the attribute is represented by the concatenation of the last three changes in the diagnosis before the discharge from the ICU or the death in ICU. Whereby:

- A KSPE with six temporal states, where each temporal state is composed of one measure, temporally ordered, where $\overline{VT}^k = \overline{VT}^{k-1} + 1$ for $k = 1, .., 5$. This KSPE involves 456 patients, and the following features: drugs, creatinine,heart rate, SpO_2, WBC, body temperture; (KSPE 1)

- A KSPE with three temporal states, each one composed of two measures recorded at the same valid time, temporally ordered, i.e., $\overline{VT}^0 < \overline{VT}^1 < \overline{VT}^2$. This KSPE involves 1.913 patients, and body temperture-SpO_2, heart rate - WBC, drugs-creatinine (KSPE 2);

- A KSPE with three temporal states, each one composed of two measures recorded at the same valid time, temporally ordered, i.e., $\overline{VT}^0 < \overline{VT}^1 < \overline{VT}^2$ This KSPE involves 319 patients, and body temperture-SpO_2, heart rate - WBC, drugs-creatinine. (KSPE 3)

Patient ID	$\overline{Drug}^0$	$\overline{VT^0}$	$\overline{Creatinine}^1$	$\overline{VT^1}$	$\overline{Heartrate}^2$	$\overline{VT^2}$	$\overline{SpO_2}^3$	$\overline{VT^3}$	$\overline{WBC}^4$	$\overline{VT^4}$	$\overline{Temperature}^5$	$\overline{VT^5}$
200069	diuretics	4	medium	5	medium	6	medium	7	medium	8	medium	9
200143	diuretics	45	high	46	medium	47	low	48	medium	49	medium	50
200147	NSAID	1	medium	2	high	3	medium	4	high	5	medium	6
200147	diuretics	1	medium	2	high	3	medium	4	high	5	medium	6
200206	diuretics	7	medium	8	medium	9	medium	10	low	11	high	12
200206	diuretics	3	medium	4	medium	5	low	6	low	7	medium	8
200231	NSAID	6	high	7	medium	8	low	9	medium	10	medium	11
200349	diuretics	2	high	3	low	4	medium	5	high	6	medium	7
200375	diuretics	85	low	86	high	87	medium	88	medium	89	medium	90
200387	diuretics	2	high	3	low	4	medium	5	medium	6	high	7

Table 5.7: KSE with $\overline{VT}^k = \overline{VT}^{k-1} + 1$ for $k = 1, .., 5$

Patient ID	$\overline{Drug}^0$	$\overline{VT^0}$	$\overline{Creatinine}^1$	$\overline{VT^1}$	$\overline{Heartrate}^2$	$\overline{VT^2}$	$\overline{SpO_2}^3$	$\overline{VT^3}$	$\overline{WBC}^4$	$\overline{VT^4}$	$\overline{Temperature}^5$	$\overline{VT^5}$	$\dot{AKI}$	$\dot{VT}$
200147	diuretics	1	medium	2	high	3	medium	4	high	5	medium	6	1	10
200147	NSAID	1	medium	2	high	3	medium	4	high	5	medium	6	1	10
2001231	NSAID	6	high	7	medium	8	low	9	medium	10	medium	11	1	15
200349	diuretics	2	high	3	low	4	medium	5	high	6	medium	7	0	14
200711	NSAID	1	medium	2	high	3	low	4	high	5	high	6	0	11
200206	diuretics	1	medium	2	medium	3	medium	4	medium	5	medium	6	1	13
200231	NSAID	1	high	2	medium	3	medium	4	medium	5	medium	6	0	13
200349	NSAID	2	high	3	medium	4	medium	5	high	6	medium	7	0	14
200375	diuretics	6	low	7	medium	8	low	9	high	10	medium	11	0	21
200387	angiotensin	1	high	2	medium	3	medium	4	high	5	low	6	0	12

Table 5.8: KSPE 1 built from the KSE in Table 5.7

- A KSPE with six temporal states, where each temporal state is composed of one measure, temporally ordered, where $\overline{VT}^k < \overline{VT}^{k-1} + 3$ for $k = 1, .., 5$. This KSPE involves 192 patients,and the following features: drugs, creatinine,heart rate, SpO$_2$, WBC, body temperture; (KSPE 4)

Starting to analyze the first KSPE, we build a K-state evolution expression of six temporal states obtaining 1705 rows and 1165 distinct patients. In Table 5.7, we show an extract from the related table.

Applying a 3-window framework of 36 hours for the observation window, 12 hours for the waiting window, and 72 hours for the prediction window, under the following condition:

$$(KSE.VT_2 - KSE.VT_0 < 7) \text{ and } (PAT.VT_{AKI} - KSE.VT_0 > 8) \text{ and}$$

$$(PAT.VT_{AKI} - KSE.VT_0 \leq 20)$$

we obtain a KSPE containing 587 rows and 456 distinct patients, 150 cases and 306 controls. In the following Table 5.8, we report an extract of this KSPE.

Regarding the second KSPE, we build a K-state evolution expression of three temporal states obtaining 2.471.277 rows and 3.071 distinct patients. In Table 5.9, we show an extract from the related table.

We obtain a KSPE containing 7.020 rows and 1913 distinct patients, 539 cases and 1374 controls. In the following Table 5.10, we report an extract of this KSPE.

Regarding the third KSPE, we use the previous K-state evolution expression as in KSPE 3 obtaining a KSPE with 1171 row, related to 234 patient with "false true false", and 85 with "true false true" patterns. In this case, true is equivalent to AKI diagnosis, and false is equivalent to NOT AKI diagnosis.

Patient ID	$\overline{Bodytemperature}^0$	$\overline{SpO_2}^0$	$\overline{VT^0}$	$\overline{Heartrate}^1$	$\overline{WBC}^1$	$\overline{VT^1}$	$\overline{Drug}^2$	$\overline{Creatinine}^2$	$\overline{VT^2}$
278886	medium	medium	14	low	medium	16	nsaid	high	20
215169	high	low	9	medium	medium	11	nsaid	medium	15
243028	high	medium	13	medium	high	14	angiotensin	medium	18
259387	high	medium	3	high	medium	7	diuretics	high	9
278228	high	low	4	medium	medium	7	diuretics	medium	10
210104	medium	medium	5	high	high	6	diuretics	medium	11
228264	medium	medium	5	medium	high	6	diuretics	high	9
249613	medium	medium	4	medium	high	6	nsaid	medium	10
206666	medium	low	4	high	medium	5	diuretics	medium	9
208435	high	medium	1	medium	high	2	diuretics	medium	6

Table 5.9: KSE with $\overline{VT}^0 < \overline{VT}^1 < \overline{VT}^2 < \overline{VT}^3$

Patient ID	$\overline{Bodytemperature}^0$	$\overline{SpO_2}^0$	$\overline{VT^0}$	$\overline{Heartrate}^1$	$\overline{WBC}^1$	$\overline{VT^1}$	$\overline{Drug}^2$	$\overline{Creatinine}^2$	$\overline{VT^2}$	$\dot{AKI}$	$\dot{VT}$
200069	high	low	3	high	high	5	NSAID	medium	8	0	12
200069	high	medium	3	high	high	5	NSAID	medium	8	0	12
200087	medium	medium	9	high	medium	11	diuretics	low	15	0	18
206767	medium	medium	6	medium	high	9	NSAID	medium	12	1	18
200159	high	medium	1	high	medium	3	diuretics	medium	5	1	12
200159	medium	medium	1	high	medium	2	diuretics	medium	5	1	12
237104	medium	medium	6	medium	high	8	diuretics	medium	12	0	19
205719	high	low	6	medium	high	7	diuretics	high	11	0	18
200168	low	medium	9	high	high	10	NSAID	medium	14	0	25
269692	medium	medium	2	medium	high	3	fiuretics	medium	6	1	14

Table 5.10: KSPE 2 built from the KSE in Table 5.9

In Table 5.11, we show an extract.

Regarding the last KSPE, we build a K-state evolution expression of three temporal states obtaining 102.011 rows and 5.300 distinct patients. In Table 5.12, we show an extract from the related table.

We obtained a KSPE containing 699 rows and 192 distinct patients, 142 ”false true false” and 50 ”true false true” patterns. In this case, true is equivalent to AKI diagnosis, and false is equivalent to NOT AKI diagnosis. In the following Table 5.13, we report an extract of this KSPE.

In Table 5.14, we report some of the obtained APFDs, with the corresponding error thresholds, through the use of our algorithm inspired by TANE. We select some APFDs in order to give an idea of the data under this type of functional dependency. Starting from the first KSPE, we analyze the APFD $\overline{Drug}^1, \overline{Creatinine}^2$, $\overline{HR}^3, \overline{SpO_2}^4, \overline{WBC}^5 \rightarrow \dot{AKI}$. In Table 5.15, we report the more common or peculiar value combinations for the AKI patients. From this, we can deduce that these value combinations for this temporal evolution, could delineate a profile for a AKI patient.

On the other hand, in Table 5.16 we report some value combinations for the same APFD, which have the same occurrences in the two classes, so probably they are not very meaningful to predict the onset of the illness.

In Table 5.17, we show some value combinations common to both classes, but with a discrete difference between the two groups. In this case, each combination could be evaluated to be descriptive for one group with respect to the other one.

Anther way to visualize the results, is the use of bar plots. In this case, we choose to visualize the results for $\overline{SpO_2}^1, \overline{HR}^2, \overline{WBC}^2, \overline{Drug}^3, \overline{Creatinine}^3$

Patient ID	$\overline{Bodytemperature}^0$	$\overline{SpO_2}^0$	$\overline{VT}^0$	$\overline{Heartrate}^1$	$\overline{WBC}^1$	$\overline{VT}^1$	$\overline{Drug}^2$	$\overline{Creatinine}^2$	$\overline{VT}^2$	$\dot{AKI}$
240724	medium	medium	5	medium	high	7	nsaid	medium	11	false true false
200300	high	medium	3	medium	high	5	diuretics	medium	9	true false true
284078	medium	medium	1	medium	medium	2	diuretics	medium	6	true false true
224518	high	medium	2	medium	high	4	angiotensin	low	8	false true false
244689	medium	medium	10	medium	medium	11	diuretics	high	12	true false true
209456	medium	medium	5	low	medium	7	nsaid	high	11	false true false
263777	medium	medium	5	medium	medium	6	nsaid	medium	10	false true false
216553	medium	medium	2	medium	medium	4	diuretics	high	6	true false true
259999	low	medium	1	medium	medium	2	nsaid	medium	6	false true false
200300	high	medium	3	medium	high	5	nsaid	medium	9	true false true

Table 5.11: KSPE 3 built from the KSE in Table 5.9

Patient ID	$\overline{Drug}^0$	$\overline{VT}^0$	$\overline{Creatinine}^1$	$\overline{VT}^1$	$\overline{Heartrate}^2$	$\overline{VT}^2$	$\overline{SpO_2}^3$	$\overline{VT}^3$	$\overline{WBC}^4$	$\overline{VT}^4$	$\overline{Temperature}^5$	$\overline{VT}^5$
252871	nsaid	2	medium	3	medium	4	medium	5	high	6	medium	7
247632	nsaid	4	low	5	medium	6	low	7	medium	9	medium	10
269227	diuretics	2	medium	3	medium	5	low	6	high	7	high	8
291422	nsaid	5	medium	6	medium	7	medium	9	medium	10	high	11
235055	diuretics	6	medium	7	medium	8	low	10	medium	11	medium	12
200099	diuretics	1	medium	2	medium	3	low	4	high	5	low	6
259127	diuretics	1	medium	2	medium	4	medium	5	high	6	medium	7
257861	nsaid	6	high	7	medium	8	low	10	medium	11	medium	12
287402	diuretics	5	medium	6	low	7	low	9	medium	10	medium	11
218006	diuretics	1	medium	2	medium	4	medium	5	medium	6	medium	7

Table 5.12: KSE with $\overline{VT}^k < \overline{VT}^{k-1} + 3$ for $k = 1, .., 5$

$\rightarrow \dot{AKI}$ from KSPE 2. In Figure 5-17, we visualize the value combinations most common in NOT AKI patients, but also present in the other class.

Instead, in Figure 5-18, we visualize all the combinations, which probably are not meaningful for any class in study, because of the comparable number of occurrences.

Analyzing the APFD $\overline{HR}^2, \overline{WBC}^2, \overline{Drug}^3 \rightarrow \dot{AKI}$ of the KSPE 3, we can observe the distribution of value combinations in the two different trends of diagnoses. These trends are composed of three diagnoses, that correspond to the last three diagnoses before the end of hospitalization in ICU.

In Table 5.18, it is possible to observe that there are some combinations under the APFD which characterize only one trend of diagnoses. They are the most interesting ones, because they probably identify this trend of diagnoses. There are other combinations which delineate both classes. In this case, when the difference is small, this combination could not be meaningful for neither of the classes.

Figures 5-19 and 5-20 are referred to the last KSPE. In the first one, we examine the APFD $\overline{Drug}^1, \overline{Creatinine}^2 \rightarrow \dot{AKI}$. We have a different distribution of occurrences in the majority of combinations, with only one peculiar combination for the "false true false" trend, and one combination probably meaningless, because the occurrences are the same for both trends. In the second one, we examine the APFD $\overline{Creatinine}^2, \overline{BodyTemperature}^6 \rightarrow \dot{AKI}$. In this case, we have different combinations which depict one trend of diagnoses, except for one combination that is peculiar of the other trend.

Patient ID	$\overline{Drug}^0$	$\overline{VT^0}$	$\overline{Creatinine}^1$	$\overline{VT^1}$	$\overline{Heartrate}^2$	$\overline{VT^2}$	$\overline{SpO_2}^3$	$\overline{VT^3}$	$\overline{WBC}^4$	$\overline{VT^4}$	$\overline{Temperature}^5$	$\overline{VT^5}$	$\dot{AKI}$
272370	angiotensin	5	medium	6	medium	7	medium	9	medium	10	medium	11	true false true
252028	diuretics	6	medium	7	high	8	low	10	medium	11	low	12	false true false
226798	diuretics	2	high	3	medium	4	medium	6	high	7	high	8	false true false
200684	diuretics	6	medium	7	high	8	low	10	medium	11	medium	12	true false true
221263	diuretics	1	medium	2	medium	3	medium	4	medium	6	medium	7	false true false
216535	diuretics	1	medium	2	high	3	low	4	medium	5	high	6	true false true
200081	diuretics	6	medium	7	medium	8	low	9	medium	11	high	12	false true false
220938	diuretics	7	medium	8	medium	9	medium	10	medium	11	medium	12	false true false
226822	diuretics	6	high	7	medium	8	low	9	medium	10	low	12	false true false
239043	diuretics	1	medium	2	medium	3	medium	4	high	6	medium	7	true false true

Table 5.13: KSPE 4 built from the KSE in Table 5.12

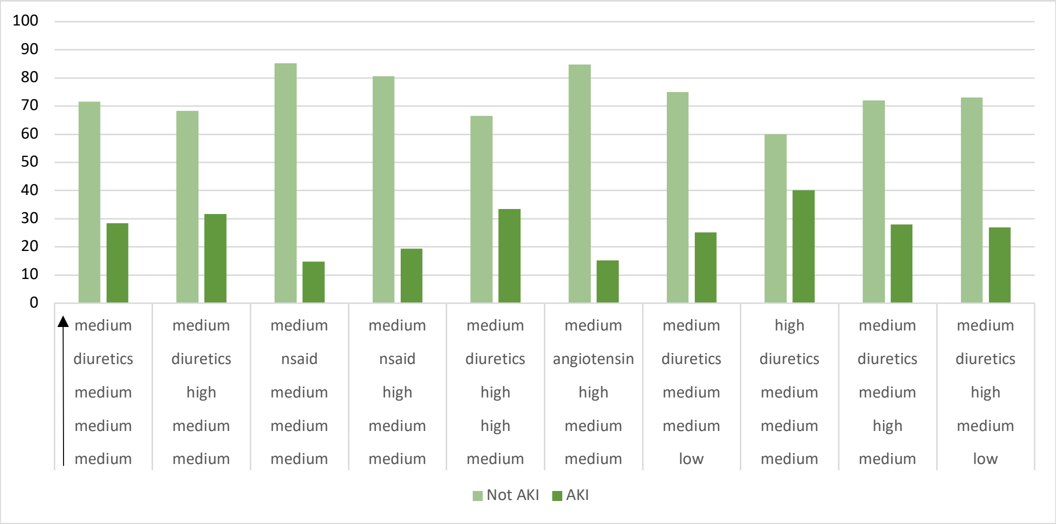


Figure 5-17: Value combinations for the not AKI patients, under $\overline{SpO_2}^1$, $\overline{HR}^2$, $\overline{WBC}^2$, $\overline{Drug}^3$, $\overline{Creatinine}^3 \rightarrow \dot{AKI}$ from KSPE 2.

5.8.3 Experiments using MIMIC-IV

System Configuration

To mine APFDs on the MIMIC-IV dataset, we test the entire framework, from the ETL prcedures to the KSPEs on a server with 16 shared core, 12 GB of RAM, 1TB disk, equipped with Ubuntu 18.04 and Postgres 12. We mine the APFDs from the generated KSPEs on a machine with a 2,3 GHz Intel Core i9 8 core, 16 GB of RAM, equipped with macOS Catalina 10.15.7, Python 2.7. During the ETL procedure, to build a K-State evolution Expression, the computational time to obtain a table increases, increasing the number of temporal states, from seconds to hours (with seven states). While mining the APFDs, according to the K-State prediction Expression dimension, usually the algorithm takes few minutes.

Data extraction

To transform the MIMIC-IV raw data in a form useful for mining the APFDs, we apply an ETL process, again using PostgreSQL and PL/pgSQL. Considering tables PATIENTS AND ICUSTAYS, we select patients from 18 to 90, obtaining 73.729 subjects.

APFD	ε_g	ε_h	ε_j	KSPE
$\overline{Creatinine}^2, \overline{HR}^3, \overline{SpO_2}^4, \overline{WBC}^5, \overline{BodyTemperature}^6 \to A\dot{K}I$	29%	30%	30%	KSPE #1
$\overline{Drug}^1, \overline{Creatinine}^2, \overline{HR}^3, \overline{SpO_2}^4, \overline{WBC}^5, \overline{BodyTemperature}^6 \to A\dot{K}I$	29%	28%	50%	KSPE #1
$\overline{Drug}^1, \overline{Creatinine}^2, \overline{HR}^3, \overline{SpO_2}^4, \overline{WBC}^5 \to A\dot{K}I$	29%	28%	70%	KSPE #1
$\overline{Drug}^1, \overline{Creatinine}^2, \overline{SpO_2}^4, \overline{WBC}^5, \overline{BodyTemperature}^6 \to A\dot{K}I$	29%	28%	70%	KSPE #1
$\overline{BodyTemperature}^1, \overline{SpO_2}^1, \overline{HR}^2, \overline{WBC}^2, \overline{Drug}^3, \overline{Creatinine}^3 \to A\dot{K}I$	20%	0%	0%	KSPE #2
$\overline{BodyTemperature}^1, \overline{HR}^2, \overline{WBC}^2, \overline{Drug}^3, \overline{Creatinine}^3 \to A\dot{K}I$	21%	0%	0%	KSPE #2
$\overline{BodyTemperature}^1, \overline{WBC}^2, \overline{Drug}^3, \overline{Creatinine}^3 \to A\dot{K}I$	22%	0%	0%	KSPE #2
$\overline{BodyTemperature}^1, \overline{SpO_2}^1, \overline{HR}^2, \overline{Drug}^3, \overline{Creatinine}^3 \to A\dot{K}I$	22%	0%	0%	KSPE #2
$\overline{SpO_2}^1, \overline{HR}^2, \overline{WBC}^2, \overline{Drug}^3, \overline{Creatinine}^3 \to A\dot{K}I$	22%	0%	0%	KSPE #2
$\overline{BodyTemperature}^1, \overline{SpO_2}^1, \overline{HR}^2, \overline{WBC}^2, \overline{Creatinine}^3 \to A\dot{K}I$	22%	0%	0%	KSPE #2
$\overline{BodyTemperature}^1, \overline{SpO_2}^1, \overline{HR}^2, \overline{WBC}^2, \overline{Drug}^3 \to A\dot{K}I$	22%	0%	0%	KSPE #2
$\overline{BodyTemperature}^1, \overline{SpO_2}^1, \overline{Drug}^3, \overline{Creatinine}^3 \to A\dot{K}I$	28.72%	35%	35%	KSPE #3
$\overline{HR}^2, \overline{WBC}^2, \overline{Drug}^3 \to A\dot{K}I$	28.72%	35%	35%	KSPE #3
$\overline{Drug}^1, \overline{Creatinine}^2, \overline{HR}^3, \overline{SpO_2}^4, \overline{WBC}^5, \overline{BodyTemperature}^6 \to A\dot{K}I$	15%	0%	0%	KSPE #4
$\overline{Drug}^1, \overline{Creatinine}^2, \overline{HR}^3, \overline{WBC}^5, \overline{BodyTemperature}^6 \to A\dot{K}I$	18%	0%	0%	KSPE #4
$\overline{Drug}^1, \overline{HR}^3, \overline{SpO_2}^4, \overline{WBC}^5, \overline{BodyTemperature}^6 \to A\dot{K}I$	18%	0%	0%	KSPE #4
$\overline{Creatinine}^2, \overline{HR}^3, \overline{SpO_2}^4, , \overline{WBC}^5, \overline{BodyTemperature}^6 \to A\dot{K}I$	18%	0%	0%	KSPE #4
$\overline{SpO_2}^4, \overline{BodyTemperature}^6 \to A\dot{K}I$	23%	0%	0%	KSPE #4
$\overline{Drug}^1, \overline{Creatinine}^2 \to A\dot{K}I$	23%	0%	0%	KSPE #4
$\overline{WBC}^5, \overline{BodyTemperature}^6 \to A\dot{K}I$	23%	0%	0%	KSPE #4
$\overline{Drug}^1, \overline{BodyTemperature}^6 \to A\dot{K}I$	23%	0%	0%	KSPE #4
$\overline{Creatinine}^2, \overline{BodyTemperature}^6 \to A\dot{K}I$	23%	0%	0%	KSPE #4
$\overline{HR}^3, \overline{BodyTemperature}^6 \to A\dot{K}I$	23%	0%	0%	KSPE #4
$\overline{Drug}^1, \overline{WBC}^5 \to A\dot{K}I$	23%	0%	0%	KSPE #4
$\overline{Creatinine}^2, \overline{HR}^3 \to A\dot{K}I$	23%	0%	0%	KSPE #4
$\overline{Creatinine}^2, \overline{WBC}^5 \to A\dot{K}I$	23%	0%	0%	KSPE #4

Table 5.14: A list of APFDs valid on one of the four KSPEs, with different error thresholds.

Feature extraction: aggregation, categorization

As we explained in Chapter 2.1, MIMIC-IV is divided into different *modules.*

For this experimental set, from the *Core* module, we use the *patients* and *admissions* tables. The *admissions* table contains information regarding the patient's admision to the hospital, for example the time of admission and discharge, demographic information, and the source of admission.

From the *hosp* module, we consider *prescriptions, labevents, d_labitems* tables. *Prescription* table contains information about the prescribed medications. We consider three different classes of drugs:

- Diuretic drugs: nebivolol, moexipril, sotalol, lisinopril, carvedilol, Methyldopa, propranolol, benazepril, ambrisentan, clonidine, triamterene, pindolol, furosemide, bosentan, minoxidil, hydrochlorothiazide, spironolactone, tolvaptan, irbesartan, chlorothiazide, prazosin, bumetanide, quinapril, labetalol, amiloride, doxazosin, atenolol, diazoxide, metoprolol, esmolol, candesartan, nadolol, losartan, captopril, valsartan, trandolapril, acebutolol, hydralazine, metolazone, eplerenone, ramipril, aliskiren, macitentan, guanfacine.
- Nephrotoxic drugs: gentamicin, vancomycin, salsalate, nabumetone, to-

$\overline{Drug}^1$	$\overline{Creatinine}^2$	$\overline{HR}^3$	$\overline{SpO_2}^4$	$\overline{WBC}^5$	Cases	Controls
diuretics	medium	medium	low	high	12	10
nsaid	medium	medium	low	medium	6	2
diuretics	medium	high	low	medium	6	4
nsaid	high	high	medium	medium	5	1
angiotensin	medium	medium	low	medium	3	2
diuretics	low	medium	medium	high	3	2
angiotensin	low	medium	low	high	2	0
nsaid	high	low	medium	medium	2	1
angiotensin	medium	low	medium	medium	1	0
diuretics	high	high	low	high	1	0
angiotensin	medium	low	low	medium	1	0
diuretics	low	medium	low	medium	1	0
nsaid	high	high	low	high	1	0
angiotensin	medium	high	medium	low	1	0
angiotensin	medium	medium	low	low	1	0

Table 5.15: Value combinations of $\overline{Drug}^1, \overline{Creatinine}^2, \overline{HR}^3, \overline{SpO_2}^4, \overline{WBC}^5 \rightarrow A\dot{K}I$, common for the cases from KSPE 1.

$\overline{Drug}^1$	$\overline{Creatinine}^2$	$\overline{HR}^3$	$\overline{SpO_2}^4$	$\overline{WBC}^5$
nsaid	high	high	low	medium
diuretics	high	low	medium	high
diuretics	high	medium	low	low
angiotensin	medium	high	low	medium
angiotensin	medium	high	medium	high
nsaid	low	medium	low	high
nsaid	low	medium	medium	high

Table 5.16: Value combinations less meaningful related to$\overline{Drug}^1$, $\overline{Creatinine}^2$, $\overline{HR}^3$,$\overline{SpO_2}^4$,$\overline{WBC}^5 \rightarrow A\dot{K}I$ from KSPE 1.

bramycin, sulindac, ketorolac, amikaci, naproxen, ibuprofen.

- Chemotherapy drugs: pemetrexed, eribulin, ibrutinib, cabozantinib, asparaginase, everolimus,pegaspargase, mitomycin, gemcitabine, gemtuzumab ozogamicin, bendamustine, carfilzomib, erlotinib, lenvatinib, dabrafenib, bleomycin, imatinib, nivolumab, ruxolitinib, oxaliplatin, decitabine, fluorouracil, temsirolimus, dactinomycin, dasatinib, etoposide, capecitabine, osimertinib, irinotecan, mitoxantrone, idarubicin, pertuzumab, arsenic trioxide, trametinib, bevacizumab, venetoclax, anagrelide, vincristine, dacarbazine, vemurafenib, brentuximab vedotin, ixazomib, pentostatin, rituximab, methotrexate, sunitinib, alemtuzumab, cetuximab, crizotinib, axitinib,bortezomib, temozolomide, cytarabine, cisplatin, panitumumab, cladribine, azacitidine, paclitaxel, ifosfamide, enasidenib, tretinoin, midostaurin, lapatinib, pralatrexate, carboplatin, cyclophosphamide, lorla-

$\overline{Drug}^1$	$\overline{Creatinine}^2$	$\overline{HR}^3$	$\overline{SpO_2}^4$	$\overline{WBC}^5$
diuretics	medium	medium	medium	medium
diuretics	medium	medium	medium	high
diuretics	high	medium	medium	medium
angiotensin	medium	medium	medium	medium
nsaid	medium	medium	medium	medium

Table 5.17: Value combinations common to both classes, related to $\overline{Drug}^1$, $\overline{Creatinine}^2$, $\overline{HR}^3$,$\overline{SpO_2}^4$,$\overline{WBC}^5$ → $A\dot{K}I$ from KSPE 1.

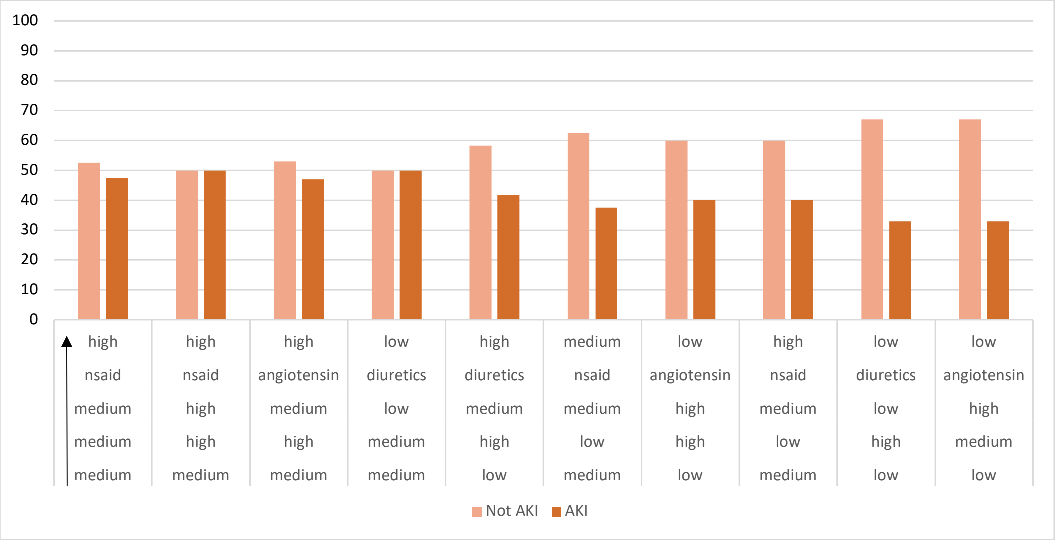


Figure 5-18: Value combinations under $\overline{SpO_2}^1$, $\overline{HR}^2$, $\overline{WBC}^2$, $\overline{Drug}^3$, $\overline{Creatinine}^3$ → $A\dot{K}I$ from KSPE 2.

tinib, trastuzumab, sorafenib, busulfan, mercaptopurine, celecoxib, daunorubicin, porfimer sodium, melphalan, pembrolizumab, clofarabine, doxorubicin.

The *labevents* table contains results of all laboratory measurements made for a single patient. In our experiment, we considered the creatinines and urine values. The *d_labitems* table is the dictionary for the hospital laboratory database.

The last one is the *ICU* module, where we consider *d_items, chartevents, icustays, outputevents*. The *d_items* table is a table that described all itemid. The *chartevents* table contains all charted items occurring during the ICU stay. We consider the heart rate (item id = 220047), SpO$_2$ (item id=220277), potassium (item id= 52452, 50822, 52610, 50971), and calcium (item id= 221456, 227525, 228317, 229618, 229640). The *icustays* tables considers all the logistic information regarding the hospitalization in ICU.

We consider all the values recorded in ICU, so the story of the patient is delineated from the admission to ICU, to the discharge from the ICU, or the time of death, if the patient dies before the discharge.

$\overline{HR}^2$	$\overline{WBC}^2$	$\overline{Drug}^3$	false true false	true false true
low	medium	diuretics	6	0
low	medium	nsaid	2	0
low	high	diuretics	3	0
high	low	diuretics	8	0
low	high	nsaid	2	0
low	high	angiotensin	1	0
high	medium	angiotensin	12	1
medium	low	angiotensin	8	2
medium	low	nsaid	1	3
medium	medium	angiotensin	34	6
medium	low	diuretics	20	7
high	medium	diuretics	33	9
high	high	nsaid	16	11
high	medium	nsaid	6	11
high	high	diuretics	25	11
medium	high	angiotensin	42	17
medium	high	nsaid	183	53
medium	high	diuretics	196	63

Table 5.18: Value combinations of $\overline{HR}^2$,$\overline{WBC}^2$,$\overline{Drug}^3 \rightarrow A\dot{K}I$ from KSPE 3.

Table 5.19: Chart-events categorization according to clinical literature.

Feature	**Low**	**Medium**	**High**
Hear rate	$\leq$60	60-100	$\geq$100
SpO_2	$\leq$95	95-98	$\geq$98
Potassium	$\leq$3.5	3.5-5	$\geq$5
Calcium	$\leq$2.2	2.2-2.6	$\geq$2.6

In this set of experiments, we decide to use all recorded values of the features that we considered, namely heart rate, SpO_2, potassium, calcium, and blood pressure. Therefore, we do not perform any type of aggregation. Instead, we decide to perform a categorization, following the thresholds reported in Table 5.19, into 'low', 'medium', 'high', according to the clinical literature. Also the choice of which drugs to consider, is based on the clinical literature. It is well known that Nephrotoxic drugs contribute to AKI in hospitalized patients, and chemotherapeutics and antimicrobials have direct chemical nephrotoxicity. These drugs, cleared via the kidneys (vancomycin), induce the kidney dysfunction leading the accumulation of the drug and its metabolites [104].

Finally for each feature, we consider as valid time the distance in hours from the admission to ICU to the valid time of the measure. We use 1 hour interval as granule.

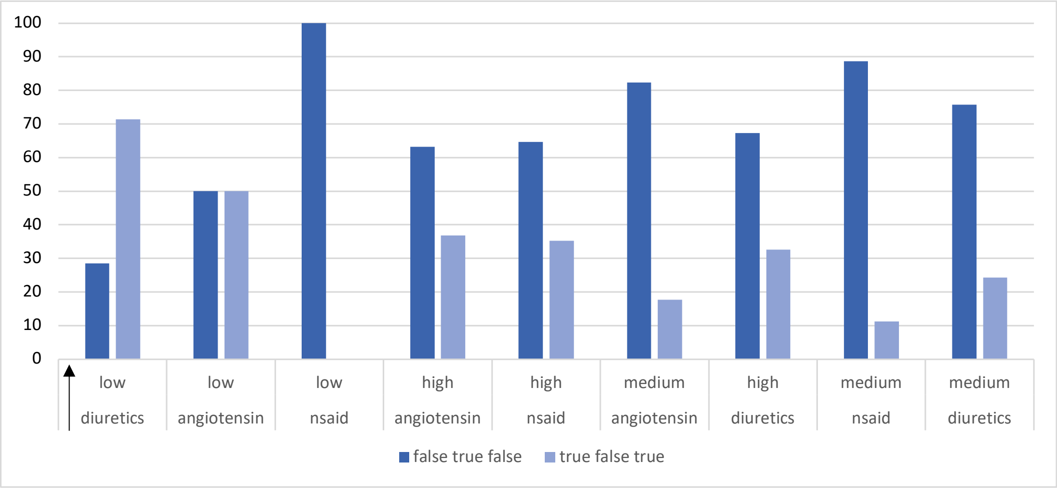


Figure 5-19: A general overview of value combinations under $\overline{Drug}^1, \overline{Creatinine}^2 \rightarrow A\dot{K}I$ from KSPE 4.

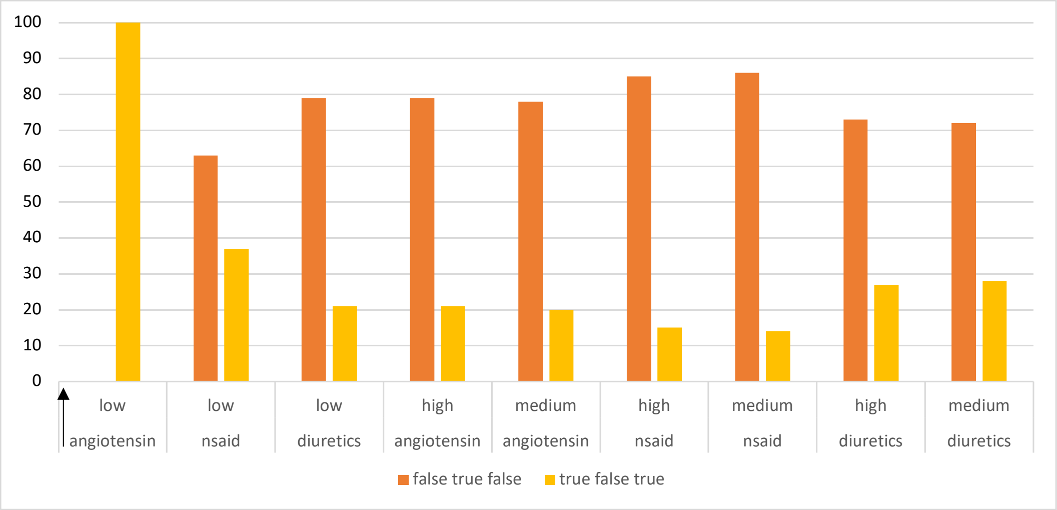


Figure 5-20: Value combinations under $\overline{Drug}^1, \overline{WBC}^5 \rightarrow A\dot{K}I$ from KSPE 4.

Patient labeling

We label the patients according to the different severity stages reported in Chapter 2.1. If a patient is healthy, we label it with a 0, otherwise with 1,2, or 3 according to the severity stages. We evaluate each patient from the admission to ICU until the discharge or time of death.

Regarding the first important measure, the serum creatinine (item id=50912), we evaluate the criteria by applying a moving window of 48 hours, according to the KDIGO general guidelines, and moving this window 24 hours forward every time. Starting from the first recorded value, we scan all the measurements, from the admission to ICU, to the last available time window. More precisely: for the baseline analysis, we consider the mean of the values of seven days before the admission to ICU. We label the patients considering different stages, which are the following:

- Stage 1: 1.5-1.9 times the baseline OR $\geq$ 0.3 mg/dl within 48 hours;
- Stage 2: 2.0-2.9 times the baseline;
- Stage 3: 3.0 times the baseline OR $\geq$ 4.0 mg/dl within 48 hours.

Regarding the second important measure, the urine output, we evaluate the first stage with a 6 hours window, the second one with 12 hours window, and the third one with 24 hours window, by moving these windows 1 hours forward every time. Even in this case, to evaluate the urine output for the diagnosis, we need to consider the weight of the patient. More precisely, we consider the following criteria to define the stage of AKI diagnosis:

- Stage 1: <0.5 m/kg/h/h for 6 hours;
- Stage 2: <0.5 m/kg for 12 interval of times;
- Stage 3: <0.3 ml/kg/h for $\geq$ 24 hours, or Anuria for $\geq$ 12 hours

Since urine is a value constantly measured in ICU, but not always recorded, we apply the *Calc* function reported in the previous Section 5.8.2, to calculate those measurements not explicitly recorded in the database. We apply this function, in other to deal with missing values, obtaining a diagnosis response for each time interval. For each patient, we scan the ICU stay with different time intervals according to the considered criteria. Regarding the first criterion, we evaluate 6-hours intervals, moving forward 1 hour every time. Regarding the second criterion, we scan the story of the patient from the admission divided it in 12-hours intervals, moving forward 1 hour every time. For the third criterion, we consider 24-hours intervals moving forward 12 hours every time.

We check continuously the clinical situation defined by an absence of urine values for 12 hours. In this case, we declare an "Anuria" situation. At the end of the labeling phase, we compare the results from different criteria , taking the most severe stage, every three hours.

Discovering APFDs from MIMIC-IV

Here, we report some of the results obtained with the following 3-window moving framework: an observation window of 48 hours, where we collect all the measures related to each patient, a waiting window of 12 hours where we do not consider any event, and then a prediction window of 72 hours. We generated six different KSPEs with two different Θ expressions. Three of them refer to single event diagnosis (as we showed in section 5.4), the others refer to the pattern attribute (as we showed in section 5.4.1), where a pattern is composed of 3 diagnoses, one every three hours. Whereby:

- A KSPE with three temporal states, temporally ordered, i.e., $\overline{VT}^k < \overline{VT}^{k-1}+5$ for $k = 1, .., 3$. This KSPE involves 3618 patients and considers the following features: creatinine, diuretics, and creatinine, and a single event for the diagnosis; (KSPE 1)

Patient ID	$\overline{Creatinine}^0$	$\overline{VT}^0$	$\overline{Diuretics}^1$	$\overline{VT}^1$	$\overline{Creatinine}^2$	$\overline{VT}^2$
39971339	high	87	furosemide	91	high	95
37401376	high	30	bumetanide	31	high	32
37886887	normal	49	furosemide	52	normal	55
32097635	high	83	bumetanide	87	high	90
36601636	high	0	furosemide	2	high	3
34185841	high	119	furosemide	120	high	123
31441718	high	70	furosemide	72	high	74
38576850	high	30	furosemide	33	high	36
34304021	high	375	captopril	379	high	383
32266080	normal	117	hydralazine	120	normal	124

Table 5.20: KSE with $\overline{VT}^k < \overline{VT}^{k-1} + 5$ for $k = 1, .., 3$ related to the two first KSPE

- A KSPE with three temporal states, temporally ordered, i.e., $\overline{VT}^k < \overline{VT}^{k-1} + 5$ for $k = 1, .., 3$. This KSPE involves 3143 patients and considers the following features: creatinine, diuretics, and creatinine, and multiple events for the diagnosis; (KSPE 2)

- A KSPE with four temporal states, temporally ordered, i.e., $\overline{VT}^k < \overline{VT}^{k-1} + 5$ for $k = 1, .., 4$. This KSPE involves 1240 patients and considers the following features: creatinine, nephrotoxic drugs, potassium, and creatinine, and a single event for the diagnosis; (KSPE 3)

- A KSPE with four temporal states, temporally ordered, i.e., $\overline{VT}^k < \overline{VT}^{k-1} + 5$ for $k = 1, .., 4$.. This KSPE involves 1108 patients and considers the following features: creatinine, nephrotoxic drugs, potassium, and creatinine, and multiple events for the diagnosis; (KSPE 4)

- A KSPE with four temporal states, temporally ordered, i.e., $\overline{VT}^0 < \overline{VT}^1 < \overline{VT}^2 < \overline{VT}^3$. It involves 19 patients and considers the following features: chemotherapeutics, diuretics, nephrotoxic drugs, and creatinine, and a single event for the diagnosis; (KSPE 5)

- A KSPE with four temporal states, temporally ordered, i.e., $\overline{VT}^0 < \overline{VT}^1 < \overline{VT}^2 < \overline{VT}^3$. This KSPE involved 19 patients and considers the following features: chemotherapeutics, diuretics, nephrotoxic drugs, and creatinine, and multiple events for the diagnosis (KSPE 6).

Starting the analyze from the first two KSPEs, which derived from the same KSE, in Table 5.20, we report and extract from this KSE.

In Tables 5.21 and 5.22, we report an extract from the KSPE 1 and 2.

Patient ID	$\overline{Creatinine}^0$	$\overline{VT}^0$	$\overline{Diuretics}^1$	$\overline{VT}^1$	$\overline{Creatinine}^2$	$\overline{VT}^2$	$\dot{AKI}$	$\dot{VT}$
37584582	high	11	furosemide	15	high	17	3	91
38020942	high	291	furosemide	292	high	296	0	357
32686570	high	21	furosemide	25	high	28	3	97
31435957	high	141	furosemide	143	high	145	0	229
32310637	normal	287	furosemide	288	normal	290	0	358
38957578	high	73	furosemide	76	high	78	3	194
34144864	normal	135	hydralazine	136	normal	139	3	249
30773739	high	126	furosemide	129	high	132	0	208
33958927	high	30	furosemide	33	high	34	3	132
39902669	high	35	furosemide	37	high	41	1	164

Table 5.21: Extract from KSPE 1

Patient ID	$\overline{Creatinine}^0$	$\overline{VT}^0$	$\overline{Diuretics}^1$	$\overline{VT}^1$	$\overline{Creatinine}^2$	$\overline{VT}^2$	$\dot{AKI}$	$\dot{VT}$
39845441	high	61	hydralazine	65	high	68	111	163
39986775	high	67	furosemide	69	high	71	111	190
31272962	high	129	labetalol	132	high	136	333	226
39659993	normal	127	furosemide	130	normal	132	333	226
34567413	high	0	furosemide	2	high	5	333	73
34353119	low	127	hydralazine	131	low	134	110	217
34294044	high	8	bumetanide	11	high	14	333	82
31452403	normal	1	furosemide	4	normal	7	333	91
30463812	high	18	furosemide	20	high	21	333	109
34185841	high	70	furosemide	72	high	75	000	145

Table 5.22: Extract from KSPE 2

Patient ID	$\overline{Creatinine}^0$	$\overline{VT}^0$	$\overline{Nephrotoxic}^1$	$\overline{VT}^1$	$\overline{Potassium}^2$	$\overline{VT}^2$	$\overline{Creatinine}^3$	$\overline{VT}^3$
34058245	high	0	vancomycin	3	normal	4	high	8
30515259	normal	16	vancomycin	17	normal	19	normal	20
39554302	high	681	vancomycin	682	low	686	high	688
38400237	high	128	vancomycin	132	normal	136	high	139
38118363	high	5	vancomycin	7	normal	9	high	11
32307723	normal	0	vancomycin	2	normal	5	normal	9
36838455	normal	12	vancomycin	13	high	17	high	19
33434203	normal	0	vancomycin	1	normal	4	normal	8
31043947	normal	8	vancomycin	9	normal	12	high	16
31299423	high	6	vancomycin	7	high	10	high	11

Table 5.23: KSE with $\overline{VT}^k < \overline{VT}^{k-1} + 5$ for $k = 1,..,4$, related to the second couple of KSPEs

The second couple of KSPEs derive from the same KSE. In Table 5.23, we report and extract from this KSE. In Tables 5.24 and 5.25 we report an extract from the KSPE 3 and 4.

Patient ID	$\overline{Creatinine}^0$	$\overline{VT}^0$	$\overline{Nephrotoxic}^1$	$\overline{VT}^1$	$\overline{Potassium}^2$	$\overline{VT}^2$	$\overline{Creatinine}^3$	$\overline{VT}^3$	$\dot{AKI}$	$\dot{VT}$
35645463	high	1	vancomycin	4	high	8	high	12	3	72
36415396	high	11	vancomycin	14	normal	16	high	20	3	133
38889793	high	8	vancomycin	11	high	15	high	17	3	117
36886134	low	6	vancomycin	9	normal	12	low	16	1	66
38722664	high	7	vancomycin	9	high	13	high	16	3	109
34230481	high	0	vancomycin	4	high	6	high	8	0	110
39786399	high	6	vancomycin	7	low	10	high	12	3	109
35157183	normal	1	vancomycin	3	normal	6	normal	7	3	62
34913252	high	10	vancomycin	12	low	13	high	17	0	101
37278580	low	53	ketorolac	54	normal	57	low	61	0	138

Table 5.24: Extract from KSPE 3

Patient ID	$\overline{Creatinine}^0$	$\overline{VT}^0$	$\overline{Nephrotoxic}^1$	$\overline{VT}^1$	$\overline{Potassium}^2$	$\overline{VT}^2$	$\overline{Creatinine}^3$	$\overline{VT}^3$	$\dot{AKI}$	$\dot{VT}$
39939179	high	75	vancomycin	78	normal	81	high	82	222	172
36814961	high	55	vancomycin	57	normal	58	high	60	333	154
32025098	high	107	vancomycin	111	normal	113	high	117	333	208
31573495	normal	1	vancomycin	4	low	8	normal	12	300	109
36779265	high	21	vancomycin	23	normal	25	high	27	333	91
35645463	high	1	vancomycin	4	high	8	high	12	333	100
30849103	high	6	vancomycin	9	high	13	high	16	333	82
34294167	low	17	vancomycin	19	normal	21	normal	24	333	145
34702715	high	0	vancomycin	2	normal	5	high	8	000	73
32385234	high	150	vancomycin	151	high	152	high	154	333	235

Table 5.25: Extract from KSPE 4

Patient ID	$\overline{Chemoterapic}^0$	$\overline{VT}^0$	$\overline{Diuretics}^1$	$\overline{VT}^1$	$\overline{Nephrotoxic}^2$	$\overline{VT}^2$	$\overline{Creatinine}^3$	$\overline{VT}^3$
30281768	rituximab	118	furosemide	358	vancomycin	474	low	482
37270064	etoposide	82	labetalol	196	vancomycin	400	low	775
37270064	rituximab	138	esmolol	233	vancomycin	306	high	388
32075128	rituximab	161	furosemide	210	vancomycin	346	normal	351
35611327	cisplatin	76	furosemide	175	vancomycin	289	high	304
37270064	etoposide	82	labetalol	237	vancomycin	320	low	601
37270064	etoposide	161	furosemide	187	vancomycin	400	low	769
37270064	etoposide	228	clonidine	281	vancomycin	306	low	829
37270064	rituximab	161	labetalol	266	vancomycin	320	low	650
37270064	rituximab	123	labetalol	204	vancomycin	306	low	619

Table 5.26: KSE with $\overline{VT}^0 < \overline{VT}^1 < \overline{VT}^2 < \overline{VT}^3$, related to the last couple of KSPEs

Patient ID	$\overline{Chemoterapic}^0$	$\overline{VT}^0$	$\overline{Diuretics}^1$	$\overline{VT}^1$	$\overline{Nephrotoxic}^2$	$\overline{VT}^2$	$\overline{Creatinine}^3$	$\overline{VT}^3$	$\dot{AKI}$	$\dot{VT}$
33992578	cisplatin	48	furosemide	63	vancomycin	87	high	197	3	147
32616999	cyclophosphamide	178	labetalol	181	vancomycin	198	high	212	3	243
38685284	cyclophosphamide	146	furosemide	165	vancomycin	167	high	293	3	237
33992578	cisplatin	48	furosemide	63	vancomycin	87	low	341	3	129
35759009	rituximab	237	metolazone	262	vancomycin	263	high	524	1	333
36770877	bortezomib	218	furosemide	238	vancomycin	259	normal	346	3	302
36214523	cisplatin	81	furosemide	94	vancomycin	99	high	159	2	174
33992578	cisplatin	48	furosemide	63	vancomycin	87	high	391	1	119
39659993	venetoclax	50	furosemide	49	vancomycin	91	normal	203	0	164
33992578	cisplatin	48	furosemide	63	vancomycin	88	high	284	2	114

Table 5.27: Extract from KSPE 5

Patient ID	$\overline{Chemoterapic}^0$	$\overline{VT}^0$	$\overline{Diuretics}^1$	$\overline{VT}^1$	$\overline{Nephrotoxic}^2$	$\overline{VT}^2$	$\overline{Creatinine}^3$	$\overline{VT}^3$	$\dot{AKI}$	$\dot{VT}$
30281768	rituximab	441	furosemide	434	vancomycin	474	low	491	333	523
30281768	rituximab	441	furosemide	465	vancomycin	474	low	590	333	532
32616999	cyclophosphamide	178	labetalol	172	vancomycin	198	high	302	333	271
33020731	etoposide	164	furosemide	172	vancomycin	189	high	269	111	289
33992578	cisplatin	48	furosemide	63	vancomycin	87	high	197	333	136
37054128	crizotinib	52	furosemide	61	vancomycin	72	high	281	333	181
32616999	cyclophosphamide	178	labetalol	172	vancomycin	196	high	249	333	253
33992578	cisplatin	48	furosemide	63	vancomycin	70	high	225	221	118
32492710	venetoclax	27	furosemide	9	vancomycin	14	high	19	333	46
36038690	cyclophosphamide	274	furosemide	282	vancomycin	317	low	386	333	379

Table 5.28: Extract from KSPE 6

APFD	ε_g	ε_h	ε_j	KSPE
$\overline{Diuretics}^1, \overline{Creatinine}^2 \rightarrow \dot{AKI}$	48.1%	5%	98%	KSPE #1
$\overline{Creatinine}^0, \overline{Diuretics}^1 \rightarrow \dot{AKI}$	48.1%	5%	98%	KSPE #1
$\overline{Creatinine}^0, \overline{Diuretics}^1, \overline{Creatinine}^2 \rightarrow \dot{AKI}$	51.1%	0%	30%	KSPE #2
$\overline{Creatinine}^0, \overline{Creatinine}^2 \rightarrow \dot{AKI}$	51.3%	0%	30%	KSPE #2
$\overline{Diuretics}^1 \rightarrow \dot{AKI}$	51.3%	0%	30%	KSPE #2
$\overline{Creatinine}^0, \overline{Nephrotoxic}^1, \overline{Potassium}^2, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	40%	5%	98%	KSPE #3
$\overline{Creatinine}^0, \overline{Nephrotoxic}^1 \rightarrow \dot{AKI}$	40.5%	5%	98%	KSPE #3
$\overline{Nephrotoxic}^1, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	40.5%	5%	98%	KSPE #3
$\overline{Nephrotoxic}^1, \overline{Potassium}^2 \rightarrow \dot{AKI}$	40.5%	5%	98%	KSPE #3
$\overline{Creatinine}^0, \overline{Potassium}^2, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	40.5%	5%	98%	KSPE #3
$\overline{Creatinine}^0, \overline{Nephrotoxic}^1 \rightarrow \dot{AKI}$	43%	0%	17%	KSPE #4
$\overline{Nephrotoxic}^1, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	43%	0%	17%	KSPE #4
$\overline{Nephrotoxic}^1, \overline{Potassium}^2 \rightarrow \dot{AKI}$	43%	0%	17%	KSPE #4
$\overline{Creatinine}^0, \overline{Potassium}^2, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	43%	0%	17%	KSPE #4
$\overline{Chemoterapy}^0, \overline{Diuretics}^1 \rightarrow \dot{AKI}$	51.3%	0%	30%	KSPE #5
$\overline{Chemoterapy}^0, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	51.3%	0%	30%	KSPE #5
$\overline{Chemoterapy}^0, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	40%	0%	30%	KSPE #5
$\overline{Chemoterapy}^0, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	40%	0%	25%	KSPE #5
$\overline{Chemoterapy}^0, \overline{Diuretics}^1 \rightarrow \dot{AKI}$	45%	0%	20%	KSPE #5
$\overline{Chemoterapy}^0, \overline{Diuretics}^1 \rightarrow \dot{AKI}$	45%	0%	25%	KSPE #5
$\overline{Chemoterapy}^0, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	45%	0%	25%	KSPE #5
$\overline{Chemoterapy}^0, \overline{Creatinine}^3 \rightarrow \dot{AKI}$	55%	0%	30%	KSPE #6
$\overline{Chemoterapy}^0, \overline{Diuretics}^1 \rightarrow \dot{AKI}$	55%	0%	30%	KSPE #6

Table 5.29: A list of APFDs valid on one of the six KSPEs, with different error thresholds.

Finally, in Table 5.26, we report an extract from the KSE which generate the last couple of KSPEs. In Table 5.27, we report an extract of the KSPE 5.

In Table 5.28, we report an extract of the KSPE 6.

In Table 5.29, we report some of the obtained APFDs with the corresponding error thresholds. Taking as example the APFD, $\overline{Creatinine}^0, \overline{Diuretics}^1 \rightarrow \dot{AKI}$ related to the first KPSE , in Table 5.30 we report the combinations peculiar to the highest level of severity, and on the contrary, the combinations of the healthy patient. Instead, the other combinations, are less meaningful, because of the occurrence of the same combination in the healthy and the severe patients.

$\overline{Creatinine}^0$	$\overline{Diuretics}^1$	Healthy	Stage 1	Stage 2	Stage 3
high	propranolol	0	0	0	42
low	spironolactone	0	0	0	12
high	moexipril	37	0	0	0
low	metolazone	72	0	0	0
high	prazosin	9	0	0	63
low	hydrochlorothiazide	16	0	0	52
normal	doxazosin	32	0	0	20
normal	propranolol	160	0	0	128
normal	ramipril	41	0	0	31

Table 5.30: A list of value combinations reletad to $\overline{Creatinine}^0, \overline{Diuretics}^1 \rightarrow A\dot{K}I$ from KSPE 1.

$\overline{Nephrotoxic}^1$	$\overline{Creatinine}^3$	000	012	233	333
amikacin	normal	6	0	0	15
amikacin	high	0	0	0	45
gentamicin	normal	60	0	0	50
gentamicin	high	30	0	0	110
ibuprofen	normal	0	0	0	28
ketorolac	low	3	0	0	27
ketorolac	high	51	0	0	87
ketorolac	normal	51	0	3	102
vancomycin	high	2625	3	246	16164
vancomycin	normal	2034	6	81	3945
vancomycin	low	387	0	21	711

Table 5.31: A list of value combinations reletad to $\overline{Nephrotoxic}^1$ $\overline{Creatinine}^3 \rightarrow A\dot{K}I$ from KSPE 4.

In Table 5.31, we collect different value combinations for four type of patterns. Let us notice that there are some combinations common between different patterns, and then we have combinations peculiar for any diagnoses trend. In this case it is possible to delineate some desiderable profiles which distinguish the different temporal patterns.

Another example is reported In Table 5.32. We collect different value combinations for the same type of patterns. Let us notice that these are 12 combinations common in patterns of healthy patients, and severe ill patients. Most of them, demonstrate an improvement in the creatinine values.

In Table 5.33, we find interesting value combinations from the last KSPE with the dependency $\overline{Chemoterapy}^0, \overline{Creatinine}^3 \rightarrow A\dot{K}I$. They are combinations which recognize only severe ill patients.

$\overline{Creatinine}^0$	$\overline{Potassium}^2$	$\overline{Creatinine}^3$	000	012	233	333
low	normal	high	27	0	0	15
high	low	normal	36	0	0	120
high	normal	normal	171	0	0	147
high	high	low	3	0	0	9
high	low	low	48	0	0	24
normal	high	low	9	0	0	9
high	high	normal	27	0	0	192
low	normal	normal	39	0	0	48
low	high	low	39	0	0	24
normal	normal	low	39	0	0	51
normal	low	high	18	0	0	72
normal	normal	high	54	0	0	384

Table 5.32: A list of value combinations reletad to $\overline{Creatinine}^0$, $\overline{Potassium}^2$, $\overline{Creatinine}^3 \rightarrow A\dot{K}I$ from KSPE 4.

$\overline{Chemoterapy}^0$	$\overline{Creatinine}^3$	000	012	233	333
crizotinib	high	0	0	0	528
gemcitabine	high	0	0	0	42
gemcitabine	normal	0	0	0	63
idarubicin	high	0	0	0	273
methotrexate	normal	0	0	0	27
carboplatin	high	0	0	0	0
rituximab	high	0	0	0	324

Table 5.33: A list of value combinations reletad to $\overline{Chemoterapy}^0$, $\overline{Creatinine}^3 \rightarrow A\dot{K}I$ from KSPE 6.

5.9 Conclusions

In this Chapter, we introduced a 3-window framework for the specification and evaluation of APFDs, dealing with the capability of exploiting data dependencies for the prediction task. The declarative framework, which we represented through relational calculus queries and formulas, allows one to consider different kinds of anchored and unanchored time windows, as well as different evolution expressions. Evolution expressions are able to represent different kinds of temporal history, related to domain-depending entities (e.g, patients), which may be derived from the stored temporal data. Approximate predictive functional dependencies have then been introduced to mine data with respect to the prediction task. Such APFDs have been specified with respect to three different kinds of errors, not only related to the number of tuples to be deleted for having the corresponding FD holding, but also to the number of entities not having tuples considered by the given FD, and by the percentage of tuples, we admit to discard for any entity. Many examples have been provided to support the different aspects we discussed in the chapter. We also discussed some critical issues we have to deal with when considering real-world databases, as those related to the computational aspects, or to the presence of unbalanced data, and so on. We finally applied our approach to real clinical data, specifically to MIMIC III dataset, obtaining results that demonstrate the applicability of this new type of temporal pattern mining in medicine, but also in other contexts where the core of the problem is finding temporal patterns in the past to predict a future event. As a future work, we will face the issues discussed in Section 5.6, both related to the computational aspects and to the analysis of the reliability of APFDs with respect to data features. Moreover, we plan to consider different types of predicted consequent, as temporal predictive patterns, composed by a suitable history of events we need to control and manage through prediction.

Chapter 6

Conclusions

In this thesis, we addressed the problem of explainability, discussing just some of the most significant challenges that need to be addressed with scientific and engineering rigor in a variety of biomedical domains. In literature, there is still a lack of different topics such as bridging the gap between symbolic (ante hoc) and sub-symbolic (black-box) approaches, engineering explainability into intelligent systems, evaluating and improving the effects of explainable components and approaches, determining when explainability is needed. All of these could be new future research directions.

Among the different challenges, we addressed the concept of predictivity in the context of temporal data mining in the clinical domain, analyzing how temporality and explainability are intertwined. The first research direction was focused on the introduction a new kind of Predictive Temporal Pattern, called Predictive Trend-Event Patterns (PTE-Ps). The new temporal knowledge associated with these predictive patterns determined a notable difference from the atemporal patterns. With preliminary results, we showed that it is possible to find effective predictive patterns for certain illness.

As a second research direction, we proposed a methodology for deriving a new kind of approximate temporal functional dependencies, called Approximate Predictive Functional Dependencies (APFDs), based on a new three-window framework base on the Observation Window (OW), the Waiting Window (WW) and the Prediction Window (PW). The declarative framework, which we represented through relational calculus queries and formulas, allows one to consider different kinds of anchored and unanchored time windows, as well as different evolution expressions. Evolution expressions are able to represent a different kind of temporal histories, related to domain-depending entities (e.g, patients), which may be derived from the stored temporal data. APFDs have been specified with respect to three different kinds of errors, not only related to the number of tuples to be deleted for having the corresponding FD holding, but also to the number of entities not having tuples considered by the given FD, and by the percentage of tuples, we admit to discard for any entity. So we preferred to present the formal relational calculus-based definitions of the single general concepts together with a pointwise exemplification. As a future work, we plan to consider different types of the predicted consequent, as temporal predictive patterns, composed by a suitable history of events we need to control and manage through prediction. For the theoretical aspects, other interesting research directions could be the satisfiability, the logical implication and inference rules, and the axiomatizability.

Chapter 7

List of pubblications

In this list of publications, there are the research publication reported in the different chapters of the thesis, and in addition there are other publication written during this PhD.

- *A manifesto on explainability for artificial intelligence in medicine*, Carlo Combi, Beatrice Amico, Riccardo Bellazzi, Andreas Holzinger, Jason H Moore, Marinka Zitnik, John H Holmes, Artificial Intelligence in Medicine, Vol. 133, 102423, Elsevier 2022

- *A 3-Window Framework for the Discovery and Interpretation of Predictive Temporal Functional Dependencies*, Beatrice Amico and Carlo Combi, Artificial Intelligence in Medicine - 20th International Conference on Artificial Intelligence in Medicine, AIME 2022, Vol. 13263 of LNCS, pages 299–309, Springer 2022, **Best Student Paper Award Winner**, doi:10.1007/978-3-031-09342-5_29

- *A Reproducible ETL Approach for Window-based Prediction of Acute Kidney Injury in Critical Care Unit and Some Preliminary Results with Support Vector Machines*, Isabela A. Chiorean, Beatrice Amico, Carlo Combi, John H. Holmes, IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2021, pages 3532–3539, IEEE 2021, doi:10.1109/BIBM52615.2021.9669143

- *Discovering predictive trend-event patterns in temporal clinical data*, Matteo Mantovani, Beatrice Amico, Carlo Combi, SAC 2021: The 36th ACM/SIGAPP Symposium on Applied Computing, pages 570–579, ACM 2021, doi: 10.1145/3412841.3441937

- *Data Mining in Medicine*, Invited Chapter for Machine Learning for Data Science Handbook:Data Mining & Knowledge Discovery Handbook, Beatrice Amico, Carlo Combi, Yuval Shahar, Springer Nature, Accepted for pubblication 20 July 2022

- *Latent class trajectory modeling of 2-component Disease Activity Score in 28 joints identifies multiple rheumatoid arthritis phenotypes of response*

to biologic disease-modifying antirheumatic drugs, Arianna Dagliati, Darren Plant, Nisha Nair, Meghna Jani, Beatrice Amico, Niels Peek, and Ann W Morgan, John Isaacs, Anthony G Wilson, Kimme L Hyrich, and others, Arthritis & Rheumatology, Vol.72-10, pages=1632–1642, Wiley Online Library 2020

Bibliography

[1] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In *Proceedings of the International Conference on Human Computer Interaction, CHI*, pages 1–18, 2018.

[2] Ziawasch Abedjan, Cuneyt G Akcora, Mourad Ouzzani, Paolo Papotti, and Michael Stonebraker. Temporal rules discovery for web data cleaning. *Proceedings of the VLDB Endowment*, 9(4):336–347, 2015.

[3] Ajaya Adhikari, David MJ Tax, Riccardo Satta, and Matthias Faeth. LEAFAGE: Example-based and Feature importance-based Explanations for Black-box ML models. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–7. IEEE, 2019.

[4] Klaus-Peter Adlassnig, Carlo Combi, Amar K. Das, Elpida T. Keravnou, and Giuseppe Pozzi. Temporal representation and reasoning in medicine: Research directions and challenges. *Artif. Intell. Medicine*, 38(2):101–113, 2006.

[5] J Adler-Milstein, JH Chen, and G Dhaliwal. Next-generation Artificial Intelligence for Diagnosis: From Predicting Diagnostic Labels to "Wayfinding". *JAMA*, 2021.

[6] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. Towards a Unified Framework for Fair and Stable Graph Representation Learning. *Proceedings of Conference on Uncertainty in Artificial Intelligence, UAI*, 2021.

[7] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499. Citeseer, 1994.

[8] Seyoung Ahn, Jeehyeong Kim, Soo Young Park, and Sunghyun Cho. Explaining Deep Learning-Based Traffic Classification Using a Genetic Algorithm. *IEEE Access*, 9:4738–4751, 2020.

[9] James F Allen. Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154, 1984.

[10] Beatrice Amico and Carlo Combi. A 3-window framework for the Discovery and Interpretation of Predictive Temporal Functional Dependencies. In *International Conference on Artificial Intelligence in Medicine*, pages 299–309. Springer, 2022.

[11] Diego Ardila, Atilla P Kiraly, Sujeeth Bharadwaj, Bokyung Choi, Joshua J Reicher, Lily Peng, Daniel Tse, Mozziyar Etemadi, Wenxing Ye, Greg Corrado, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature Medicine*, 25(6):954–961, 2019.

[12] Monika Arora, Uma Kanjilal, and Dinesh Varshney. Evaluation of information retrieval: precision and recall. *International Journal of Indian Culture and Business Management*, 12(2):224–236, 2016.

[13] Nida Aslam. Explainable Artificial Intelligence Approach for the Early Prediction of Ventilator Support and Mortality in COVID-19 Patients. *Computation*, 10(3):36, 2022.

[14] Boris Babic, Sara Gerke, Theodoros Evgeniou, and I Glenn Cohen. Beware explanations from AI in health care. *Science*, 373(6552):284–286, 2021.

[15] Amie J Barda, Christopher M Horvat, and Harry Hochheiser. A qualitative research framework for the design of user-centered displays of explanations for machine learning model predictions in healthcare. *BMC medical informatics and decision making*, 20(1):1–16, 2020.

[16] Iyad Batal and Milos Hauskrecht. Constructing classification features using minimal predictive patterns. In *Proceedings of the international conference on Information and knowledge management (CIKM)*, 2010.

[17] Iyad Batal, Hamed Valizadegan, Gregory F Cooper, and Milos Hauskrecht. A temporal pattern mining approach for classifying electronic health record data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4):1–22, 2013.

[18] Riccardo Bellazzi, Cristiana Larizza, Paolo Magni, and Roberto Bellazzi. Temporal data mining for the quality assessment of hemodialysis services. *Artificial Intelligence in Medicine*, 34(1):25 – 39, 2005.

[19] Riccardo Bellazzi, Cristiana Larizza, Paolo Magni, and Roberto Bellazzi. Temporal data mining for the quality assessment of hemodialysis services. *Artificial intelligence in medicine*, 34(1):25–39, 2005.

[20] Riccardo Bellazzi, Cristiana Larizza, and Alberto Riva. Temporal abstractions for interpreting diabetic patients monitoring data. *Intelligent Data Analysis*, 2(1-4):97–122, 1998.

[21] Riccardo Bellazzi and Blaz Zupan. Predictive data mining in clinical medicine: current issues and guidelines. *International Journal of Medical Informatics*, 77(2):81–97, 2008.

[22] Riccardo Bellazzi and Blaz Zupan. Predictive data mining in clinical medicine: current issues and guidelines. *International journal of medical informatics*, 77(2):81–97, 2008.

[23] Rinaldo Bellomo, Claudio Ronco, John A Kellum, Ravindra L Mehta, and Paul Palevsky. Acute renal failure–definition, outcome measures, animal models, fluid therapy and information technology needs: the second international consensus conference of the acute dialysis quality initiative (ADQI) group. *Critical care*, 8(4):1–9, 2004.

[24] Laure Berti-Équille, Hazar Harmouch, Felix Naumann, Noël Novelli, and Saravanan Thirumuruganathan. Discovery of Genuine Functional Dependencies from Relational Data with Missing Values. *Proc. VLDB Endow.*, 11(8):880–892, 2018.

[25] Claudio Bettini, Sushil Jajodia, and Sean Wang. *Time granularities in databases, data mining, and temporal reasoning*. Springer Science & Business Media, 2000.

[26] Claudio Bettini and Angelo Montanari. Temporal representation and reasoning. *Data Knowl. Eng.*, 44(2):139–141, 2003.

[27] Philip Bohannon, Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis. Conditional functional dependencies for data cleaning. In *2007 IEEE 23rd international conference on data engineering*, pages 746–755. IEEE, 2006.

[28] Roger C Bone, Robert A Balk, Frank B Cerra, R Phillip Dellinger, Alan M Fein, William A Knaus, Roland MH Schein, and William J Sibbald. Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis. *Chest*, 101(6):1644–1655, 1992.

[29] Roger C Bone, Robert A Balk, Frank B Cerra, R Phillip Dellinger, Alan M Fein, William A Knaus, Roland MH Schein, and William J Sibbald. Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis. *Chest*, 101(6):1644–1655, 1992.

[30] P. Bozzola, G. Bortolan, C. Combi, F. Pinciroli, and C. Brohet. A hybrid neuro-fuzzy system for ECG classification of myocardial infarction. In *IEEE Conference on Computers in Cardiology*, pages 241–244, 1996. cited By 23.

[31] Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. Elsevier, 2004.

[32] Stephen Brett. Science review: the use of proton pump inhibitors for gastric acid suppression in critical illness. *Critical care*, 9(1):45, 2004.

[33] John Brooke. Sus: A retrospective. *Journal of Usability Studies*, 8(2):29–40, 2003.

[34] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

[35] Longbing Cao. AI in combating the COVID-19 pandemic. *IEEE Intell. Syst.*, 37(2):3–13, 2022.

[36] Loredana Caruccio, Vincenzo Deufemia, Felix Naumann, and Giuseppe Polese. Discovering relaxed functional dependencies based on multi-attribute dominance. *IEEE Transactions on Knowledge and Data Engineering*, 33(9):3212–3228, 2020.

[37] Loredana Caruccio, Vincenzo Deufemia, Felix Naumann, and Giuseppe Polese. Discovering Relaxed Functional Dependencies Based on Multi-Attribute Dominance. *IEEE Trans. Knowl. Data Eng.*, 33(9):3212–3228, 2021.

[38] Loredana Caruccio, Vincenzo Deufemia, and Giuseppe Polese. Relaxed functional dependencies—a survey of approaches. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):147–165, 2015.

[39] Davide Castelvecchi. Can we open the black box of AI? *Nature News*, 538(7623):20, 2016.

[40] Jinying Chen, Emily Druhl, Balaji Polepalli Ramesh, Thomas K Houston, Cynthia A Brandt, Donna M Zulman, Varsha G Vimalananda, Samir Malkani, and Hong Yu. A Natural Language Processing System That Links Medical Terms in Electronic Health Record Notes to Lay Definitions: System Development Using Physician Reviews. *J Med Internet Res*, 20(1):e26, Jan 2018.

[41] Xu Chen, Yongfeng Zhang, and Zheng Qin. Dynamic explainable recommendation based on neural attentive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 53–60, 2019.

[42] Luca Chittaro and Angelo Montanari. Temporal representation and reasoning in artificial intelligence: Issues and approaches. *Ann. Math. Artif. Intell.*, 28(1-4):47–106, 2000.

[43] E. F. Codd. Normalized data base structure: A brief tutorial. In *Proceedings of the 1971 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control*, SIGFIDET '71, page 1–17, New York, NY, USA, 1971. Association for Computing Machinery.

[44] Carlo Combi, Beatrice Amico, Riccardo Bellazzi, Andreas Holzinger, Jason H Moore, Marinka Zitnik, and John H Holmes. A manifesto on explainability for artificial intelligence in medicine. *Artificial Intelligence in Medicine*, page 102423, 2022.

[45] Carlo Combi, Massimo Franceschet, and Adriano Peron. Representing and reasoning about temporal granularities. *Journal of Logic and Computation*, 14(1):51–77, 2004.

[46] Carlo Combi, Matteo Mantovani, Alberto Sabaini, Pietro Sala, Francesco Amaddeo, Ugo Moretti, and Giuseppe Pozzi. Mining approximate temporal functional dependencies with pure temporal grouping in clinical databases. *Computers in biology and medicine*, 62:306–324, 2015.

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

[48] Carlo Combi, Angelo Montanari, and Pietro Sala. A uniform framework for temporal functional dependencies with multiple granularities. Technical Report RR 81/2011, Department of Computer Science, University of Verona, Verona, Italy, 2011.

[49] Carlo Combi, Barbara Oliboni, Giuseppe Pozzi, Alberto Sabaini, and Esteban Zimányi. Enabling instant-and interval-based semantics in multidimensional data models: the T+ MultiDim Model. *Information Sciences*, 518:413–435, 2020.

[50] Carlo Combi, Barbara Oliboni, Alessandro Zardini, and Francesca Zerbato. A Methodological Framework for the Integrated Design of Decision-Intensive Care Pathways - an Application to the Management of COPD Patients. *J. Heal. Informatics Res.*, 1(2):157–217, 2017.

[51] Carlo Combi, Barbara Oliboni, and Francesca Zerbato. A modular approach to the specification and management of time duration constraints in BPMN. *Inf. Syst.*, 84:111–144, 2019.

[52] Carlo Combi and Giuseppe Pozzi. Temporal representation and reasoning in medicine. *Artif. Intell. Medicine*, 38(2):97–100, 2006.

[53] Carlo Combi and Alberto Sabaini. Extraction, analysis, and visualization of temporal association rules from interval-based clinical data. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 238–247. Springer, 2013.

[54] Carlo Combi and Pietro Sala. Mining approximate interval-based temporal dependencies. *Acta Informatica*, 53(6-8):547–585, 2016.

[55] Stefano Concaro, Lucia Sacchi, Carlo Cerra, Pietro Fratino, and Riccardo Bellazzi. Mining healthcare data with temporal association rules: Improvements and assessment for a practical use. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 16–25. Springer, 2009.

[56] Roberto Confalonieri, Tillman Weyde, Tarek R Besold, and Fermín Moscoso del Prado Martín. Trepan Reloaded: A Knowledge-driven Approach to Explaining Artificial Neural Networks. *arXiv:1906.08362*, 2019.

[57] Nancy R Cook. Comments on'evaluating the added predictive ability of a new marker: From area under the ROC curve to reclassification and beyond'by mj pencina et al., statistics in medicine. *Statistics in medicine*, 27(2):191–195, 2008.

[58] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[59] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[60] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[61] R Phillip Dellinger, Mitchell M Levy, Andrew Rhodes, Djillali Annane, Herwig Gerlach, Steven M Opal, Jonathan E Sevransky, Charles L Sprung, Ivor S Douglas, Roman Jaeschke, et al. Surviving Sepsis Campaign: international guidelines for management of severe sepsis and septic shock, 2012. *Intensive care medicine*, 39(2):165–228, 2013.

[62] Mauro Dragoni, Ivan Donadello, and Claudio Eccher. Explainable AI meets persuasiveness: Translating reasoning results into behavioral change advice. *Artificial Intelligence in Medicine*, 105:101840, 2020.

[63] Shaker El-Sappagh, José M Alonso, Farman Ali, Amjad Ali, Jun-Hyeog Jang, and Kyung-Sup Kwak. An ontology-based interpretable fuzzy decision support system for diabetes diagnosis. *IEEE Access*, 6:37371–37394, 2018.

[64] European Commission. White Paper on Artificial Intelligence: a European approach to excellence and trust. White Paper COM(2020) 65 final, European Commission, Brussels, February 2020.

[65] Wenfei Fan, Floris Geerts, and Xibei Jia. Semandaq: a data quality system based on conditional functional dependencies. *Proc. VLDB Endow.*, 1(2):1460–1463, 2008.

[66] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

[67] Thomas Ferrère, Oded Maler, Dejan Nickovic, and Amir Pnueli. From real-time logic to timed automata. *J. ACM*, 66(3):19:1–19:31, 2019.

[68] Michael Fisher. Temporal representation and reasoning. In Frank van Harmelen, Vladimir Lifschitz, and Bruce W. Porter, editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 513–550. Elsevier, 2008.

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

[70] Chris Giannella and Edward Robertson. On approximation measures for functional dependencies. *Inf. Syst.*, 29(6):483–507, August 2004.

[71] Chris Giannella and Edward Robertson. On Approximation Measures for Functional Dependencies. *Inf. Syst.*, 29(6):483–507, August 2004.

[72] Dongxiao Gu, Kaixiang Su, and Huimin Zhao. A case-based ensemble learning system for explainable breast cancer recurrence prediction. *Artificial Intelligence in Medicine*, 107:101858, 2020.

[73] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.*, 51(5):93:1–93:42, 2019.

[74] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.

[75] Deisy Morselli Gysi, Ítalo Do Valle, Marinka Zitnik, Asher Ameli, Xiao Gan, Onur Varol, Susan Dina Ghiassian, JJ Patten, Robert A Davey, Joseph Loscalzo, et al. Network medicine framework for identifying drug-repurposing opportunities for COVID-19. *Proceedings of the National Academy of Sciences*, 118(19), 2021.

[76] Ira J Haimowitz and Isaac S Kohane. Managing temporal worlds for medical trend diagnosis. *Artificial Intelligence in Medicine*, 8(3):299–321, 1996.

[77] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*, 2020.

[78] Omer David Harel and Robert Moskovitch. Complete closed time intervals-related patterns mining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4098–4105, 2021.

[79] Carl G. Hempel and Paul Oppenheim. Studies in the Logic of Explanation. *Philosophy of science*, 15(2):135–175, 1948.

[80] Andreas G. Hofmann and Brian C. Williams. Temporally and spatially flexible plan execution for dynamic hybrid systems. *Artif. Intell.*, 247:266–294, 2017.

[81] Elizabeth A Holm. In defense of the black box. *Science*, 364(6435):26–27, 2019.

[82] Andreas Holzinger. Explainable AI and Multi-Modal Causability in Medicine. *i-com*, 19(3):171–179, 2021.

[83] Andreas Holzinger, André Carrington, and Heimo Müller. Measuring the quality of explanations: the system causability scale (SCS). *KI-Künstliche Intelligenz*, pages 1–6, 2020.

[84] Andreas Holzinger, Matthias Dehmer, Frank Emmert-Streib, Rita Cucchiara, Isabelle Augenstein, Javier Del Ser, Wojciech Samek, Igor Jurisica, and Natalia Díaz-Rodríguez. Information fusion as an integrative cross-cutting enabler to achieve robust, explainable, and trustworthy medical artificial intelligence. *Information Fusion*, 79(3):263–278, 2021.

[85] Andreas Holzinger, Georg Langs, Helmut Denk, Kurt Zatloukal, and Heimo Müller. Causability and explainability of artificial intelligence in medicine. *WIREs Data Mining Knowl. Discov.*, 9(4), 2019.

[86] Andreas Holzinger, Bernd Malle, Anna Saranti, and Bastian Pfeifer. Towards Multi-Modal Causability with Graph Neural Networks enabling Information Fusion for explainable AI. *Information Fusion*, 71(7):28–37, 2021.

[87] Andreas Holzinger and Heimo Mueller. Toward Human-AI Interfaces to Support Explainability and Causability in Medical AI. *IEEE COMPUTER*, 54(10):78–86, 2021.

[88] Frank Höppner and Frank Klawonn. *Finding Informative Rules in Interval Sequences*, pages 125–134. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[89] Miroslav Hudec, Erika Minarikova, Radko Mesiar, Anna Saranti, and Andreas Holzinger. Classification by ordinal sums of conjunctive and disjunctive functions for explainable AI and interpretable machine learning solutions. *Knowledge Based Systems*, 220:106916, 2021.

[90] Ykä Huhtala, Juha Karkkainen, Pasi Porkka, and Hannu Toivonen. Efficient discovery of functional and approximate dependencies using partitions. In *Proceedings 14th International Conference on Data Engineering*, pages 392–401. IEEE, 1998.

[91] Yka Huhtala, Juha Kärkkäinen, Pasi Porkka, and Hannu Toivonen. TANE: an efficient algorithm for discovering functional and approximate dependencies. *The computer journal*, 42(2):100–111, 1999.

[92] Christian S. Jensen and Richard T. Snodgrass. Temporal database. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[93] Christian S. Jensen and Richard T. Snodgrass. Temporal query languages. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[94] Christian S. Jensen and Richard T. Snodgrass. Transaction time. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[95] Christian S. Jensen and Richard T. Snodgrass. Valid time. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[96] Christian S. Jensen and Richard T. Snodgrass. Valid time. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

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

[98] Weina Jin, Xiaoxiao Li, and Ghassan Hamarneh. Evaluating explainable AI on a Multi-Modal Medical Imaging Task: Can Existing Algorithms Fulfill Clinical Requirements? In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11945–11953. AAAI Press, 2022.

[99] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv. *version 0.4). PhysioNet. https://doi. org/10.13026/a3wn-hq05*, 2020.

[100] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[101] Neesha Jothi, Wahidah Husain, et al. Data mining in healthcare–a review. *Procedia Computer Science*, 72:306–313, 2015.

[102] Po-shan Kam and Ada Wai-Chee Fu. Discovering Temporal Patterns for Interval-Based Events. In *Data Warehousing and Knowledge Discovery, Second International Conference, DaWaK 2000*, pages 317–326, 2000.

[103] Ramisetty Kavya, Jabez Christopher, Subhrakanta Panda, and Y Bakthasingh Lazarus. Machine learning and XAI approaches for Allergy Diagnosis. *Biomedical Signal Processing and Control*, 69:102681, 2021.

[104] John A Kellum, Paola Romagnani, Gloria Ashuntantang, Claudio Ronco, Alexander Zarbock, and Hans-Joachim Anders. Acute kidney injury. *Nature reviews Disease primers*, 7(1):1–17, 2021.

[105] Hendrik Kempt, Jan-Christoph Heilinger, and Saskia K. Nagel. Relative explainability and double standards in medical decision-making. *Ethics Inf. Technol.*, 24(2):20, 2022.

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

[107] Jyrki Kivinen and Heikki Mannila. Approximate inference of functional dependencies from relations. *Theoretical Computer Science*, 149(1):129–149, 1995.

[108] Andreas Kleppe, Ole-Johan Skrede, Sepp De Raedt, Knut Liestøl, David J Kerr, and Håvard E Danielsen. Designing deep learning studies in cancer diagnostics. *Nature Reviews Cancer*, 21(3):199–211, 2021.

[109] Sebastian Kruse and Felix Naumann. Efficient Discovery of Approximate Dependencies. *Proc. VLDB Endow.*, 11(7):759–772, 2018.

[110] Thomas K Landauer. *The trouble with computers: Usefulness, usability, and productivity.* MIT press, 1995.

[111] Markus Langer, Daniel Oster, Timo Speith, Holger Hermanns, Lena Kästner, Eva Schmidt, Andreas Sesing, and Kevin 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.*, 296:103473, 2021.

[112] Q Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the AI: informing design practices for explainable AI user experiences. In *Proceedings of the International Conference on Human Computer Interaction, CHI*, pages 1–15, 2020.

[113] Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4120–4127, 2022.

[114] Stéphane Lopes, Jean-Marc Petit, and Lotfi Lakhal. Functional and approximate dependency mining: database and FCA points of view. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2-3):93–114, 2002.

[115] Michael J Maher. Constrained dependencies. *Theoretical Computer Science*, 173(1):113–149, 1997.

[116] Konstantinos Makris and Loukia Spanou. Acute kidney injury: definition, pathophysiology and clinical phenotypes. *The clinical biochemist reviews*, 37(2):85, 2016.

[117] Matteo Mantovani, Beatrice Amico, and Carlo Combi. Discovering predictive trend-event patterns in temporal clinical data. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 570–579, 2021.

[118] Matteo Mantovani, Carlo Combi, and Milos Hauskrecht. Mining Compact Predictive Pattern Sets Using Classification Model. In *Artificial Intelligence in Medicine - 17th Conference on Artificial Intelligence in Medicine*, pages 386–396, 2019.

[119] Matteo Mantovani, Carlo Combi, and Matteo Zeggiotti. Discovering and analyzing trend-event patterns on clinical data. *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–10, 2019.

[120] Aniek F Markus, Jan A Kors, and Peter R Rijnbeek. The role of explainability in creating trustworthy artificial intelligence for health care: a comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, page 103655, 2020.

[121] Barbara Mukami Maweu, Sagnik Dakshit, Rittika Shamsuddin, and Balakrishnan Prabhakaran. Cefes: A CNN explainable framework for ECG signals. *Artificial Intelligence in Medicine*, 115:102059, 2021.

[122] Mirjana Mazuran, Elisa Quintarelli, Letizia Tanca, and Ugolini Stefania. Semi-automatic support for evolving functional dependencies. In *19th International Conference on Extending Database Technology (EDBT 2016)*, pages 293–304. Springer Verlag, 2016.

[123] Ravindra L Mehta, John A Kellum, Sudhir V Shah, Bruce A Molitoris, Claudio Ronco, David G Warnock, and Adeera Levin. Acute kidney injury network: report of an initiative to improve outcomes in acute kidney injury. *Critical care*, 11(2):1–8, 2007.

[124] Corrado Mencar and José M Alonso. Paving the way to explainable artificial intelligence with fuzzy modeling. In *International Workshop on Fuzzy Logic and Applications*, pages 215–227. Springer, 2018.

[125] Martino Mensio, Emanuele Bastianelli, Ilaria Tiddi, and Giuseppe Rizzo. Mitigating bias in deep nets with knowledge bases: The case of natural language understanding for robots. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1)*, pages 1–9, 2020.

[126] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[127] Gregoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73(2):1–15, 2018.

[128] Robert Moskovitch. Multivariate temporal data analysis-a review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(1):e1430, 2022.

[129] Robert Moskovitch, Fernanda Polubriaginof, Aviram Weiss, Patrick Ryan, and Nicholas Tatonetti. Procedure prediction from symbolic Electronic Health Records via time intervals analytics. *Journal of biomedical informatics*, 75:70–82, 2017.

[130] Robert Moskovitch and Yuval Shahar. Fast time intervals mining using the transitivity of temporal relations. *Knowledge and Information Systems*, 42(1):21–48, 2015.

[131] Heimo Mueller, Michaela T. Mayrhofer, Evert-Ben Van Veen, and Andreas Holzinger. The Ten Commandments of Ethical Medical AI. *IEEE COMPUTER*, 54(7):119–123, 2021.

[132] Hamid R. Nemati, David M. Steiger, Lakshmi S. Iyer, and Richard T. Herschel. Knowledge warehouse: an architectural integration of knowledge management, decision support, artificial intelligence and data warehousing. *Decis. Support Syst.*, 33(2):143–161, 2002.

[133] G Nicora, M Rios, A Abu-Hanna, and R Bellazzi. Evaluating pointwise reliability of machine learning prediction. *Journal of Biomedical Informatics*, 2022.

[134] Pavel Novitski, Cheli Melzer Cohen, Avraham Karasik, Varda Shalev, Gabriel Hodik, and Robert Moskovitch. All-Cause Mortality Prediction in T2D Patients. In *International Conference on Artificial Intelligence in Medicine*, pages 3–13. Springer, 2020.

[135] Angelo Oddi, Riccardo Rasconi, and Amedeo Cesta. Project scheduling as a disjunctive temporal problem. In Helder Coelho, Rudi Studer, and Michael J. Wooldridge, editors, *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 967–968. IOS Press, 2010.

[136] Kalia Orphanou, Arianna Dagliati, Lucia Sacchi, Athena Stassopoulou, Elpida Keravnou, and Riccardo Bellazzi. Incorporating repeating temporal association rules in naïve bayes classifiers for coronary heart disease diagnosis. *Journal of biomedical informatics*, 81:74–82, 2018.

[137] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[138] Seyedeh Neelufar Payrovnaziri, Zhaoyi Chen, Pablo Rengifo-Moreno, Tim Miller, Jiang Bian, Jonathan H. Chen, Xiuwen Liu, and Zhe He. Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review. *J. Am. Medical Informatics Assoc.*, 27(7):1173–1185, 2020.

[139] Judea Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60, 2019.

[140] Matteo Pennisi, Isaak Kavasidis, Concetto Spampinato, Vincenzo Schinina, Simone Palazzo, Federica Proietto Salanitri, Giovanni Bellitto, Francesco Rundo, Marco Aldinucci, Massimo Cristofaro, et al. An explainable AI system for automated COVID-19 assessment and lesion categorization from CT-scans. *Artificial intelligence in medicine*, 118:102114, 2021.

[141] Dragutin Petkovic, Russ Altman, Mike Wong, and Arthur Vigil. Improving the explainability of random forest classifier–user centered approach. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium*, pages 204–215. World Scientific, 2018.

[142] Parivash Pirasteh, Slawomir Nowaczyk, Sepideh Pashami, Magnus Löwenadler, Klas Thunberg, Henrik Ydreskog, and Peter Berck. Interactive feature extraction for diagnostic trouble codes in predictive maintenance: A case study from automotive domain. In *Proceedings of the Workshop on Interactive Data Mining*, pages 1–10, 2019.

[143] Ryan Poplin, Avinash V Varadarajan, Katy Blumer, Yun Liu, Michael V McConnell, Greg S Corrado, Lily Peng, and Dale R Webster. Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning. *Nature Biomedical Engineering*, 2(3):158–164, 2018.

[144] Karl Popper. *Die Logik der Forschung. Zur Erkenntnistheorie der modernen Naturwissenschaft.* Springer-Verlag, Wien, 1935.

[145] Roberto Posenato and Carlo Combi. Adding flexibility to uncertainty: Flexible simple temporal networks with uncertainty (FTNU). *Inf. Sci.*, 584:784–807, 2022.

[146] Seth M. Powsner, José Costa, and Robert J. Homer. Clinicians Are From Mars and Pathologists Are From Venus: Clinician Interpretation of Pathology Reports. *Archives of Pathology & Laboratory Medicine*, 124(7):1040–1046, 06 2000.

[147] Abdulhakim Qahtan, Nan Tang, Mourad Ouzzani, Yang Cao, and Michael Stonebraker. Pattern functional dependencies for data cleaning. 2020.

[148] Abdulhakim Ali Qahtan, Nan Tang, Mourad Ouzzani, Yang Cao, and Michael Stonebraker. Pattern functional dependencies for data cleaning. *Proceedings of the VLDB Endowment*, 13:684 – 697, 2020.

[149] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the International Conference on Fairness, Accountability, and Transparency*, pages 33–44, 2020.

[150] Nicole M. Rau, Mir A. Basir, and Kathryn E. Flynn. Parental understanding of crucial medical jargon used in prenatal prematurity counseling. *BMC Medical Informatics Decis. Mak.*, 20(1):169, 2020.

[151] Mauricio Reyes, Raphael Meier, Sérgio Pereira, Carlos A. Silva, Fried-Michael Dahlweid, Hendrik von Tengg-Kobligk, Ronald M. Summers, and Roland Wiest. On the Interpretability of Artificial Intelligence in Radiology: Challenges and Opportunities. *Radiology: Artificial Intelligence*, 2(3):e190043, 2020. PMID: 32510054.

[152] Samantha Cruz Rivera, Xiaoxuan Liu, An-Wen Chan, Alastair K Denniston, and Melanie J Calvert. Guidelines for clinical trial protocols for interventions involving artificial intelligence: the SPIRIT-AI extension. *BMJ*, 370, 2020.

[153] Claudio Ronco, Rinaldo Bellomo, and John A Kellum. Acute kidney injury. *The Lancet*, 394(10212):1949–1964, 2019.

[154] Jeffrey D. Rudie, Andreas M. Rauschecker, Long Xie, Jiancong Wang, Michael Tran Duong, Emmanuel J. Botzolakis, Asha Kovalovich, John M. Egan, Tessa Cook, R. Nick Bryan, Ilya M. Nasrallah, Suyash Mohan, and James C. Gee. Subspecialty-Level Deep Gray Matter Differential Diagnoses with Deep Learning and Bayesian Networks on Clinical Brain MRI: A Pilot Study. *Radiology: Artificial Intelligence*, 2(5):e190146, 2020. PMID: 33937838.

[155] Lucia Sacchi, Cristiana Larizza, Carlo Combi, and Riccardo Bellazzi. Data mining with temporal abstractions: learning rules from time series. *Data Mining and Knowledge Discovery*, 15(2):217–247, 2007.

[156] Lucia Sacchi, Cristiana Larizza, Carlo Combi, and Riccardo Bellazzi. Data mining with Temporal Abstractions: learning rules from time series. *Data Min. Knowl. Discov.*, 15(2):217–247, 2007.

[157] Pietro Sala, Carlo Combi, Matteo Mantovani, and Romeo Rizzi. Discovering evolving temporal information: Theory and application to clinical databases. *SN Comput. Sci.*, 1(3):153, 2020.

[158] Nurul Izrin Md Saleh, Hadhrami Ab Ghani, and Zairul Jilani. Defining factors in hospital admissions during COVID-19 using LSTM-FCA explainable model. *Artificial Intelligence in Medicine*, 132:102394, 2022.

[159] Philipp Schirmer, Thorsten Papenbrock, Sebastian Kruse, Felix Naumann, Dennis Hempfing, Torben Mayer, and Daniel Neuschäfer-Rube. DynFD: Functional Dependency Discovery in Dynamic Datasets. In *EDBT*, pages 253–264, 2019.

[160] Tjeerd AJ Schoonderwoerd, Wiard Jorritsma, Mark A Neerincx, and Karel Van Den Bosch. Human-centered XAI: Developing design patterns for explanations of clinical decision support systems. *International Journal of Human-Computer Studies*, 154:102684, 2021.

[161] Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Robert de Hoog, Nigel Shadbolt, Walter Van de Velde, and Bob Wielinga. *Knowledge Engineering and Management: The CommonKADS Methodology.* MIT Press, Cambridge, MA, 2000.

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

[163] Alberto Segura-Delgado, María José Gacto, Rafael Alcalá, and Jesús Alcalá-Fdez. Temporal association rule mining: An overview considering the time variable as an integral or implied component. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(4):e1367, 2020.

[164] Yuval Shahar. A framework for knowledge-based temporal abstraction. *Artificial intelligence*, 90(1-2):79–133, 1997.

[165] Yuval Shahar and Mark A Musen. Knowledge-based temporal abstraction in clinical domains. *Artificial intelligence in medicine*, 8(3):267–298, 1996.

[166] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[167] Alexander Shknevsky, Yuval Shahar, and Robert Moskovitch. Consistent discovery of frequent interval-based temporal patterns in chronic patients' data. *Journal of biomedical informatics*, 75:83–95, 2017.

[168] Dan A Simovici, Dana Cristofor, and Laurentiu Cristofor. Impurity measures in databases. *Acta Informatica*, 38(5):307–324, 2002.

[169] Karl Stoeger, David Schneeberger, and Andreas Holzinger. Medical Artificial Intelligence: The European Legal Perspective. *Communications of the ACM*, 64(11):34–36, 2021.

[170] Erico Tjoa and Cuntai Guan. A survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Trans. Neural Networks Learn. Syst.*, 32(11):4793–4813, 2021.

[171] Shigehiko Uchino, Rinaldo Bellomo, Donna Goldsmith, Samantha Bates, and Claudio Ronco. An assessment of the RIFLE criteria for acute renal failure in hospitalized patients. *Critical care medicine*, 34(7):1913–1917, 2006.

[172] Alejandro A. Vaisman and Esteban Zimányi. *Data Warehouse Systems - Design and Implementation, Second Edition*. Data-Centric Systems and Applications. Springer, 2022.

[173] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[174] Victor Vianu. Dynamic functional dependencies and database aging. *Journal of the ACM (JACM)*, 34(1):28–59, 1987.

[175] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. Designing theory-driven user-centric explainable AI. In *Proceedings of the International Conference on Human Computer Interaction, CHI*, pages 1–15, 2019.

[176] X Sean Wang, Claudio Bettini, Alexander Brodsky, and Sushil Jajodia. Logical design for temporal databases with multiple granularities. *ACM Transactions on Database Systems (TODS)*, 22(2):115–170, 1997.

[177] Adrian Weller. Transparency: Motivations and challenges. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 23–40. Springer, 2019.

[178] Mikael Wiberg and Erik Stolterman. Time and temporality in HCI research. *Interact. Comput.*, 33(3):250–270, 2021.

[179] Jef Wijsen. Reasoning about qualitative trends in databases. *Information Systems*, 23(7):463–487, 1998.

[180] Jef Wijsen. Temporal fds on complex objects. *ACM Transactions on Database Systems (TODS)*, 24(1):127–176, 1999.

[181] Jef Wijsen. *Temporal Dependencies*, pages 3955–3961. Springer New York, New York, NY, 2018.

[182] Jozef Wijsen. Design of temporal relational databases based on dynamic and temporal functional dependencies. In *Recent Advances in Temporal Databases*, pages 61–76. Springer, 1995.

[183] Dacosta Yeboah, Louis Steinmeister, Daniel B Hier, Bassam Hadi, Donald C Wunsch, Gayla R Olbricht, and Tayo Obafemi-Ajayi. An explainable and statistically validated ensemble clustering model applied to the identification of traumatic brain injury subgroups. *IEEE Access*, 8:180690–180705, 2020.

[184] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNexplainer: Generating explanations for graph neural networks. *Advances in Neural Information Processing Systems, NeurIPS*, 32:9240, 2019.

[185] Jianlong Zhou, Amir H. Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics. *Electronics*, 10(5):593, 2021.

[186] Marinka Zitnik, Marcus W Feldman, Jure Leskovec, et al. Evolution of resilience in protein interactomes across the tree of life. *Proceedings of the National Academy of Sciences*, 116(10):4426–4433, 2019.