

Viewpoint

# Lifting Hospital Electronic Health Record Data Treasures: Challenges and Opportunities

Alexander Maletzky<sup>1</sup>, PhD; Carl Böck<sup>2</sup>, PhD; Thomas Tschoellitsch<sup>3</sup>, MD; Theresa Roland<sup>4</sup>, PhD; Helga Ludwig<sup>4</sup>, MSc; Stefan Thumfart<sup>1</sup>, PhD; Michael Giretzlehner<sup>1</sup>, PhD; Sepp Hochreiter<sup>4</sup>, PhD; Jens Meier<sup>3</sup>, MD

<sup>1</sup>Research Department Medical Informatics, RISC Software GmbH, Hagenberg, Austria

<sup>2</sup>JKU LIT SAL eSPML Lab, Institute of Signal Processing, Johannes Kepler University, Linz, Austria

<sup>3</sup>Department of Anesthesiology and Critical Care Medicine, Kepler University Hospital GmbH, Johannes Kepler University, Linz, Austria

<sup>4</sup>ELLIS Unit Linz, LIT AI Lab, Institute for Machine Learning, Johannes Kepler University, Linz, Austria

**Corresponding Author:**

Alexander Maletzky, PhD

Research Department Medical Informatics

RISC Software GmbH

Softwarepark 32a

Hagenberg, 4232

Austria

Phone: 43 7236 93028406

Email: [alexander.maletzky@risc-software.at](mailto:alexander.maletzky@risc-software.at)

## Abstract

Electronic health records (EHRs) have been successfully used in data science and machine learning projects. However, most of these data are collected for clinical use rather than for retrospective analysis. This means that researchers typically face many different issues when attempting to access and prepare the data for secondary use. We aimed to investigate how raw EHRs can be accessed and prepared in retrospective data science projects in a disciplined, effective, and efficient way. We report our experience and findings from a large-scale data science project analyzing routinely acquired retrospective data from the Kepler University Hospital in Linz, Austria. The project involved data collection from more than 150,000 patients over a period of 10 years. It included diverse data modalities, such as static demographic data, irregularly acquired laboratory test results, regularly sampled vital signs, and high-frequency physiological waveform signals. Raw medical data can be corrupted in many unexpected ways that demand thorough manual inspection and highly individualized data cleaning solutions. We present a general data preparation workflow, which was shaped in the course of our project and consists of the following 7 steps: obtain a rough overview of the available EHR data, define clinically meaningful labels for supervised learning, extract relevant data from the hospital's data warehouses, match data extracted from different sources, deidentify them, detect errors and inconsistencies therein through a careful exploratory analysis, and implement a suitable data processing pipeline in actual code. Only few of the data preparation issues encountered in our project were addressed by generic medical data preprocessing tools that have been proposed recently. Instead, highly individualized solutions for the specific data used in one's own research seem inevitable. We believe that the proposed workflow can serve as a guidance for practitioners, helping them to identify and address potential problems early and avoid some common pitfalls.

(*JMIR Med Inform 2022;10(10):e38557*) doi: [10.2196/38557](https://doi.org/10.2196/38557)

**KEYWORDS**

electronic health record; medical data preparation; machine learning; retrospective data analysis

## Introduction

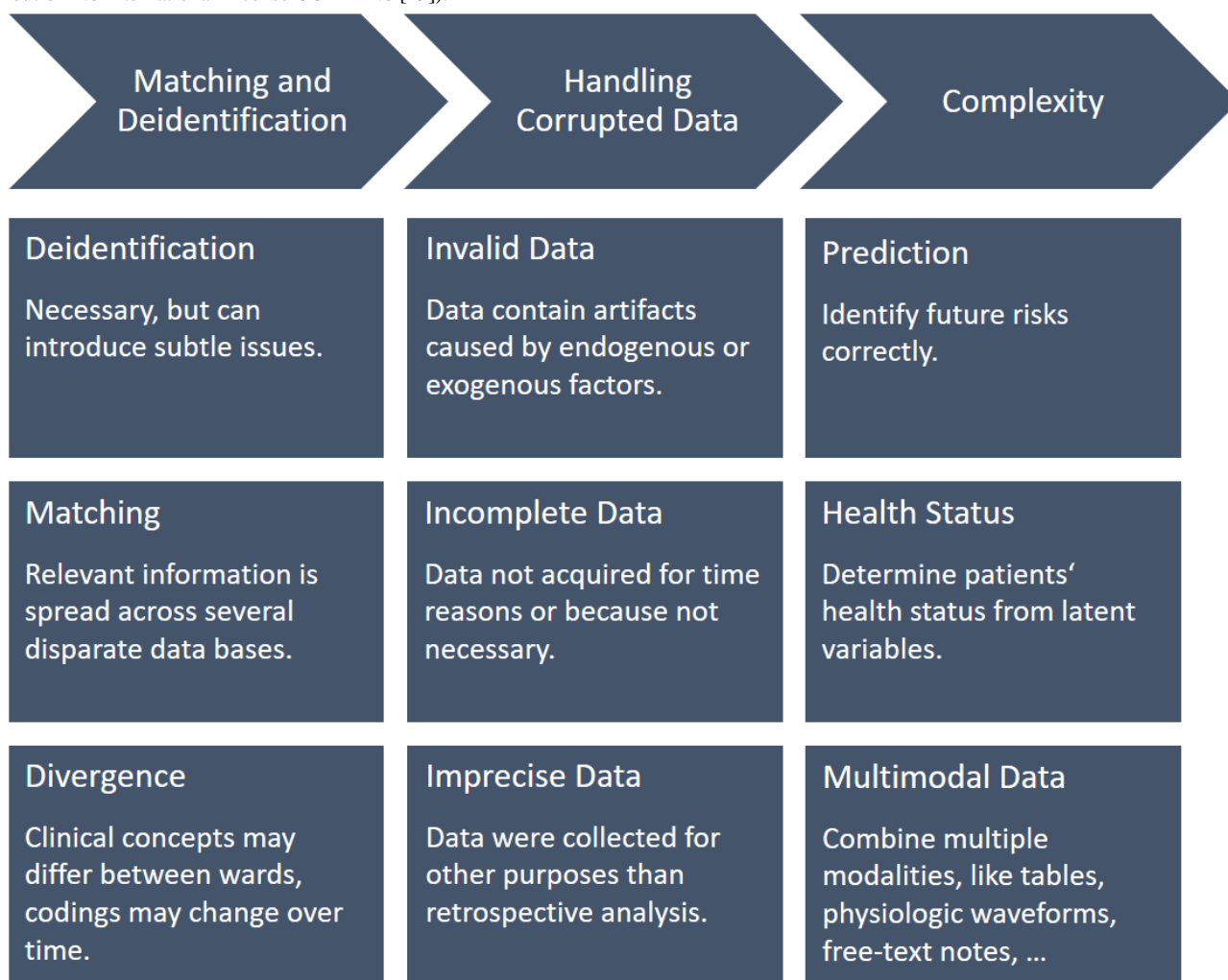
Electronic health records (EHRs) contain a vast amount of information about an individual's health status, including demographics, diagnoses, medication prescriptions, laboratory test results, high-frequency physiologic waveform signals, and others. Many prior studies have demonstrated how data science

and machine learning (ML) can be applied to large databases of EHRs to successfully train models to predict many different patient-related outcomes, including mortality risk [1-4], length of hospital or intensive care unit (ICU) stays [1-3], cardiovascular decompensation [3,5,6], postoperative complications [7], and, recently, COVID-19 diagnosis and pathogenesis [8-12]. Although data preparation requires

considerable time and effort [13,14], it is seldom represented in research outputs. One possible explanation could be that it is considered a “standard” task that always proceeds more or less the same and that can be automated to a large extent, thanks to readily applicable general purpose software tools [15-17]. In this paper, we illustrate through specific examples from a large-scale research project that this is not the case. Conducting a secondary (ML-based) analysis of raw EHRs from a hospital’s data warehouse is challenging in many respects due to several reasons. Above all, data were originally collected without any specific use case besides clinical application, and relevant information is usually distributed over multiple disparate

databases that often lack comprehensible documentation. If clinical concepts (variables, categorical values, units of measurement, etc) are represented differently across distinct sources or if the coding of clinical concepts changes over time, data harmonization can become a real issue. Moreover, incomplete or invalid data, although a well-known problem in principle, can occur in many (unexpected) forms and might only be noticed after careful manual inspection. Figure 1 summarizes the main challenges with EHR data that we encountered in our work and are ubiquitous in retrospective medical data analysis [18].

**Figure 1.** Primary challenges with retrospective medical data analysis (adapted from Johnson et al [18], which is published under Creative Commons Attribution 4.0 International License CC-BY 4.0 [19]).



Unlike many other papers about data preparation in a medical context, this work does not propose a novel generic data processing tool. Instead, we report the challenges that we faced and the lessons we have learned in a recent large-scale data science project. We present specific examples of messy and corrupted raw data to create awareness that (medical) data preparation is a nontrivial, labor-intensive endeavor, despite an ever-growing set of generic tools. Finally, we present a general data preparation workflow for similar research projects to help practitioners avoid the most common pitfalls.

The literature on medical data preparation for large-scale secondary (ML-based) analysis is scarce. Most studies have

focused on model development and the final predictive performance of the developed models and only mention a few fundamental aspects of the data preparation pipeline. This is particularly true for the work of Rajkomar et al [1], but for a good reason: deep neural networks are used to learn “good” representations of the data in an end-to-end fashion, relying on the networks’ ability to automatically handle messy data properly. The pipeline is based on Fast Healthcare Interoperability Resources [20], meaning that all data available in this format can be readily processed without further ado—no feature selection, harmonization, or cleaning is necessary. Although appealing at first glance, the proposed approach has

some limitations, as noted by the authors. Most importantly, deep neural networks often require massive amounts of data and computing resources to learn good representations. Second, the lack of data harmonization potentially impairs transferability across research locations; for example, for validation. Moreover, to train models in a supervised manner, one must provide labels, and depending on the use case, these labels may be difficult to extract, reintroducing the need for data preparation. It is also unclear whether the models developed in the aforementioned study [1] would have performed even better, had the data undergone more thorough manual inspection and curation.

Other studies have proposed generic data processing pipelines that can be applied off-the-shelf to well-known ICU benchmark databases such as Medical Information Mart for Intensive Care (MIMIC) [21-23] and the telehealth ICU collaborative research database (eICU-CRD) [24]. The most prominent examples being MIMIC-Extract [15], FIDDLE [16], cleaning and organization pipeline for EHR computational and analytic tasks [17], and Clairvoyance [25]. The authors of FIDDLE and Clairvoyance claim that their systems are sufficiently general to accommodate not only data extracted from MIMIC-III and eICU-CRD but also any EHR data available in a particular form. This may be true to a large extent, but we experienced that cleaning messy, raw data and bringing them into the required standardized form is at least as labor intensive (in terms of implementation effort) as the subsequent “generic” preprocessing steps that FIDDLE and Clairvoyance cover. Sculley et al [14] termed this phenomenon *glue code antipattern*. In general, MIMIC and eICU-CRD may be excellent benchmark databases, but we found that “real-world” data exported directly from a hospital’s IT infrastructure pose many challenges that are not present in these databases.

Shi et al [26] presented a medical data cleaning pipeline that explicitly addresses some of the issues that we also encountered in our research. They considered laboratory tests and similar measurements and proposed manually curated validation rules for numerical variables and an automatic strategy for harmonizing (misspelled) units of measurement through fuzzy search and variable-dependent conversion rules. The focus of Shi et al [26] is on improving the quality of data [27-29], whereas Wang et al [15], Tang et al [16], and Mandyam et al [17] are mainly concerned with transforming data into a form suitable for ML. A more detailed evaluation of FIDDLE, MIMIC-Extract, and cleaning and organization pipeline for EHR computational and analytical tasks and the approach to our data by Shi et al [26] can be found in [Multimedia Appendix 1](#) [15-17,26].

The extensive survey article by Johnson et al [18] summarizes the main issues of medical data analysis similar to that in this work. The authors also established a high-level categorization of these issues into *compartmentalization*, *corruption*, and *complexity* (Figure 1) and argued that data acquisition and preparation in the critical care context are particularly difficult because data are collected for a different purpose.

Sendak et al [30] arrived at similar conclusions, noting in particular that solutions developed for one site did not scale well across multiple sites because of redundant data validation

and normalization. The authors provided estimates for the expected cost of deploying a model to screen patients with chronic kidney disease in other hospitals. We refrain from extrapolating such estimates from our findings but agree that the costs for preprocessing data from other sites into a form suitable for existing prediction models will likely be significant.

## Methods

### Data Preparation

Raw EHRs stored in hospitals’ data warehouses cannot readily be used for developing clinical prediction models but must first be extracted, analyzed, and subjected to a series of preprocessing steps. These steps may differ between data modalities and sources but usually include some sort of *validation* (ensuring data accurately reflect reality), *harmonization* (establishing uniform representation of equivalent concepts), and *transformation* (bringing data into a form suitable for model development, eg, extracting useful information). Furthermore, it must be ensured that a sufficient number of data points are available in the first place and that clinically meaningful target labels can be extracted from them in the case of supervised learning. We demonstrate how this can be accomplished in a disciplined, effective, and efficient manner by referring to a specific data science project.

### Underlying Data Science Project

All results presented in this paper originate from a large-scale data science project for developing data-driven clinical prediction models. Specifically, the following 5 use cases were considered: (1) optimizing patient throughput in the ICU, (2) increasing the accuracy of treatment priorities in emergency medicine, (3) improving the selection of blood products, (4) predicting patient deterioration in the ICU to enable preventive interventions, and (5) predicting COVID-19 infections using routinely acquired laboratory tests [11]. All use cases were based on retrospective, routinely collected data from the Kepler University Hospital, a large university hospital in Linz, Austria. A wide variety of data modalities were used, including patient demographics, laboratory tests, diagnoses, vital signs, and even high-frequency physiological waveform signals. Information represented by natural-language text was mostly ignored (except for short free-text diagnoses), and imaging modalities were excluded altogether.

The amount of data varies among the 5 use cases; for instance, use cases 1 and 4 are naturally confined to patients admitted to the ICU, whereas for use case 2, only patients who visited the emergency department (ED) could be taken into account. The period covered by the data also depends on the use case. [Table 1](#) lists the particular period and the total number of patients for each of the 5 use cases. Altogether, the order of magnitude of the number of data items processed was  $10^9$  (excluding high-frequency waveform data) of which vital signs and laboratory tests constituted the vast majority.

The specific results of the 5 use cases were not the main focus of this paper. Instead, the use cases serve merely as illustrative examples throughout the remainder of this paper.

**Table 1.** Use cases considered in the research project<sup>a</sup>.

Use case	Short description	Period	Patients, n
1	Optimizing ICU <sup>b</sup> patient throughput	2010-2020	14,236
2	Increasing the accuracy of treatment priorities in ED <sup>c</sup>	2015-2020	77,972
3	Improving the selection of blood products	2016-2020	5855
4	Predicting patient deterioration in the ICU	2018-2020	3069
5	Predicting COVID-19 infections [11]	2019-2020	79,884

<sup>a</sup>Note that patient cohorts partly overlap.

<sup>b</sup>ICU: intensive care unit.

<sup>c</sup>ED: emergency department.

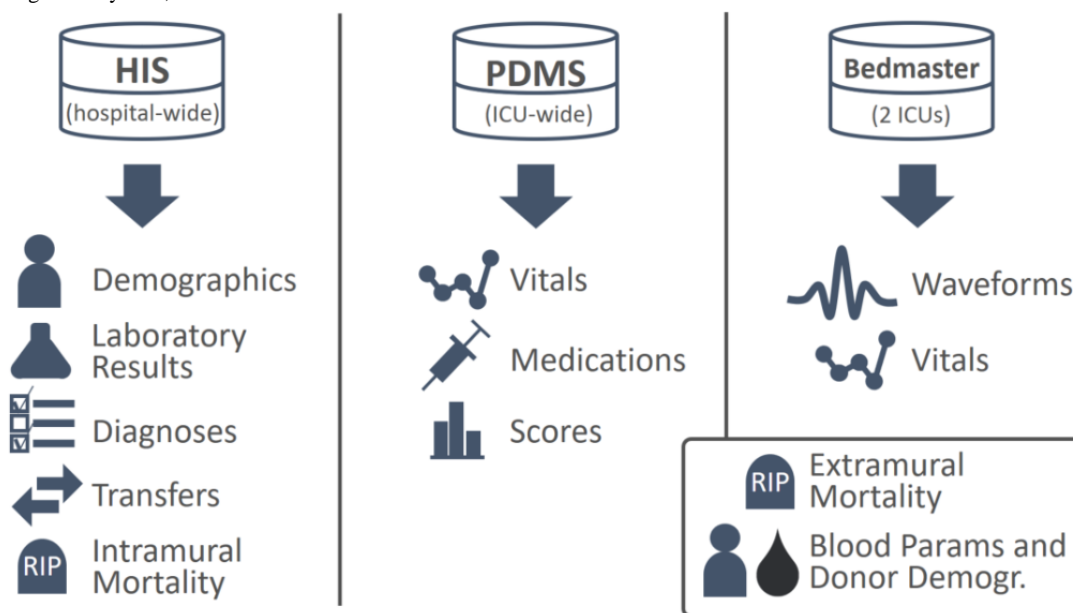
### Data Sources

Relevant data for the 5 mentioned use cases are contained in 3 central data management systems in the hospital’s IT infrastructure: the *hospital information system (HIS)*, *patient data management system (PDMS)*, and *Bedmaster system*. The HIS is a hospital-wide data warehouse that contains information on all patients admitted to the hospital. Among others, this includes demographics (date of birth, sex, etc), detailed information about in-hospital transfers, diagnoses, laboratory test results, and intramural mortality. PDMS is deployed in 5 ICUs associated with critical care in the hospital. Hence, it only contains information about patients admitted to the ICU during their hospital stay but complements the basic information found in the HIS with automatically recorded vital sign measurements (heart rate, blood pressure, body temperature, etc; up to 30 measurements per vital sign per hour), precise information about

administered medications, and manually recorded scores (eg, Glasgow Coma Scale). The Bedmaster system [31] can be connected to bedside monitoring devices and automatically stores the vital signs, physiological waveforms, and alarms produced by these devices. The temporal resolution of the acquired data far exceeds the resolution in PDMS, with vitals being recorded every 2 seconds and waveforms sampled at rates of 60 to 240 Hz. This system is only deployed in 2 of the 5 ICUs and was installed in March 2018. Hence, the number of patients covered by it is significantly smaller compared with HIS and PDMS.

In addition, information about the extramural mortality of patients after hospital discharge was obtained from the Austrian Federal Statistics Agency (use case 1), and information about blood products transfused in the hospital was obtained from a local blood bank (use case 3). Figure 2 summarizes all data sources and modalities used in the 5 cases.

**Figure 2.** Data sources and exported modalities in use cases 1 to 5. HIS, PDMS, and Bedmaster are data management systems deployed in the hospital, whereas information about extramural mortality and blood products had to be obtained from external sources. HIS: hospital information system; PDMS: patient data management system; ICU: intensive care unit.



### Ethics Approval

For each use case mentioned in this work, approval was obtained from the Ethics Committee of the Medical Faculty, Johannes

Kepler University, Linz, Austria. The corresponding study numbers are 1015/2021, 1233/2020, 1232/2020, 1014/2021, and 1104/2020.

## Results

### Data Overview

Before beginning to export raw data from their hospital-internal storage, it is imperative to obtain an overview of what kind and how much data are available. This might sound obvious but can be more intricate than it seems. For example, the number of patients or cases at one's disposal is not always indicative of the amount of suitable data. Specifically, in the common setting of supervised learning, only data points to which clinically meaningful target labels can be assigned are useful. In the blood transfusion use case 3, for instance, the types of transfusion-related complications we could consider were limited by the availability of sufficient pre- and posttransfusion laboratory measurements to identify the respective complications. Ultimately, sufficient labeled training samples could only be generated for predicting acute kidney injury and acute respiratory failure. Other organ systems, although interesting in principle, had to be excluded from the analysis. Acute respiratory failure also had to be excluded eventually because the class imbalance was found to be too strong.

Given the richness of information stored in EHRs, there are normally enough data that can be converted into features that clinical prediction models may attend to. However, one must be aware that information accumulates over time, meaning that more data about a patient are available toward the end of the hospital or ICU stay than toward the beginning. For us, this was especially relevant in use case 2, where treatment priorities and 30-day mortality of patients visiting the ED had to be predicted based on only a few pieces of information typically recorded in the ED.

### Defining Labels for Supervised Learning

Routinely collected retrospective EHR data do not always contain information about the outcomes that one wants to predict. Typical outcome parameters, apart from mortality or length of hospital stay, are often composites of several parameters that must be deduced from surrogate variables. Some authors, for instance, resort to hypotension as an indicator of cardiac instability [5,6], an approach we adopted in our use case 4 for predicting patient deterioration. Similarly, widely accepted criteria for organ system failure exist; for example, Kidney Disease: Improving Global Outcomes [32] for kidney disease and the Berlin definition for acute respiratory distress syndrome [33]. Both were used in use case 3.

Further problems can arise when trying to predict the effects of interventions. First, it might not always be possible to connect an observed outcome to a specific intervention, especially if multiple interventions occur within a short time. In the blood transfusion use case 3, in many cases, 2 or more blood products are transferred simultaneously, rendering it impossible to determine which of the administered transfusions are responsible for a posttransfusion complication. In such a situation, framing the prediction task as a *multiple instance learning problem* [34] might be the only remedy. Second, if the goal is to assess or improve existing clinical decision policies, one is confronted with questions such as "What would have happened to the patient if he/she had been treated differently?" Naturally, such

questions are difficult to answer based on retrospective data in which interventions and treatments are fixed, and counterfactual trajectories cannot be explored, although the literature on estimating counterfactual treatment outcomes through statistical analysis and ML exists [35]. In use case 1, where the primary goal was to predict the optimal time for discharging ICU patients back to a ward, we resorted to answering the proxy question of whether transferred patients should have better stayed longer in the ICU. We determined this by identifying patients who died or returned unexpectedly shortly after ICU discharge.

### Accessing and Extracting the Data

Hospital IT infrastructure is usually designed to provide easy access to the data of individual patients to deliver optimal care. Unfortunately, this does not imply that batches of data from distinct patients can be accessed, let alone extracted, easily. In particular, if the amount of manual interaction required for exporting data is too high, individual retrospective studies might be feasible, but the automatic real-time deployment of prediction models on live data may not be feasible. Data access can be challenging when there is only one source but even more so if there are multiple disparate data sources one must incorporate. In our project, we had to access 3 distinct databases: HIS, PDMS, and Bedmaster (Figure 2). HIS is a SAP-based system from which tables can be exported as CSV or Microsoft Excel files, and PDMS is a PostgreSQL relational database that allows exporting the results of queries in whatever table format is desired. In contrast, exporting data from the Bedmaster system turned out to be cumbersome because only XML and JSON exports are supported by default. Representing the massive amount of waveform and vital sign data in either of these verbose formats resulted in huge files that could not be processed efficiently; so, in the first step, we had to extract the relevant numerical values from the JSON files and store them in the more efficient HDF5 format. This process was considerably more intricate than anticipated because of inconsistencies in the exported data representation that are detailed in [Multimedia Appendix 2 \[36-39\]](#).

### Matching Data From Different Sources

Data exported from different sources must be matched to obtain coherent records of the patients or cases under consideration. Under normal circumstances, this is straightforward because of common identifiers. However, according to our experience, such identifiers do not always need to be present or change over time. Specifically, data exported from the Bedmaster system lack identifiers, such as patient or case IDs. Knowing only the ICU bed they stem from, as well as the precise timestamp of each single recorded value, we had to assign the corresponding IDs manually based on the information about which patient occupied which ICU bed at which time. This approach works but is cumbersome and adds extra complexity and is another potential source of mistakes. It is also more difficult to automate than simply joining tables on common ID columns.

Mappings between identifiers and the entities they refer to may change over time as experienced in our project with drug codes. Every drug has a unique code that is used to reference it in prescription tables, but for unknown reasons, the coding changes at certain points in time. The precise information when this

happens is stored in another table, so that drug names *can* be recovered from the provided codes and timestamps of prescriptions. Yet again, the whole process is not as straightforward as we would have hoped.

### Deidentification

Sensible personal information stored in EHRs can only be shared in a deidentified form. There are no universal rules *on how* data need to be deidentified, as long as identifying individual patients afterward becomes practically impossible. In our project, deidentification amounted to removing patient names and replacing hospital identifiers, such as patient IDs or case IDs, with project-internal identifiers that could be used to match corresponding data items across different tables. Furthermore, all timestamps were shifted by a random per-patient offset in the future to avoid reidentification of patients from knowing their exact admission or discharge times. Timestamps were shifted after matching data from different sources because some matching strategies depend on precise temporal information, as described earlier. The timestamps were shifted such that the time of day and day of week were preserved because both constitute potentially valuable information for downstream data analysis tasks. The same is true for seasonality, which was also roughly preserved. This deidentification policy is analogous to that used for MIMIC-III [21]. We remark that it is not as thorough as the policy implemented for releasing the more recent AmsterdamUMCdb [40]: there, theoretical concepts such as *k*-anonymity and *l*-diversity are considered to render reidentifying individual patients practically impossible under advanced threat models assuming “rogue researchers” and “rogue insurance companies” with access to the data. As, in our case, all data (even in deidentified form) are kept private and can only be accessed by project members, we did not deem such a thorough deidentification policy necessary.

Deidentification removes or replaces information that can otherwise be used to detect inconsistencies in the data, such as the same patient ID being accidentally assigned to multiple patients with different names. Therefore, it is crucial to ensure that any problems of this kind are detected and corrected either before or while deidentifying the data when the necessary information is still available. Specifically, we implemented extensive sanity checks that, for instance, ensure case and patient IDs are in a 1:n relationship (every case ID corresponds to a unique patient ID, but a patient ID can have multiple case IDs associated with it). All instances violating this principle are immediately reported to the human operator, allowing him or her to either overwrite one of the identifiers or discard the instances completely. Furthermore, missing patient IDs were automatically reconstructed from known case IDs whenever possible. The availability of patient IDs is essential because the random temporal offsets used for deidentifying timestamps are associated with patient IDs rather than case IDs. Finally, because hospital-assigned case IDs follow a clearly defined pattern that allows them to be distinguished from patient IDs, accidentally swapped case IDs and patient IDs are automatically exchanged before deidentification.

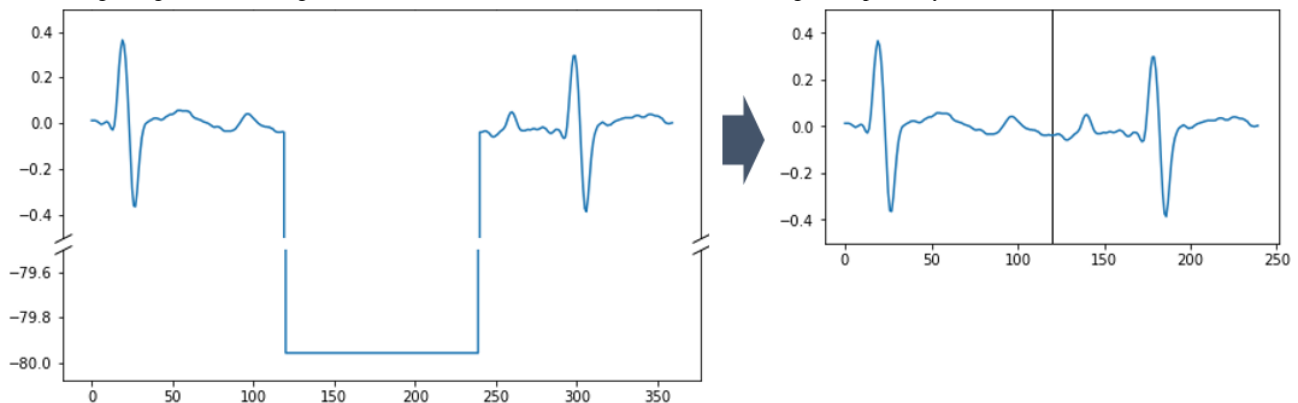
The kind of information that should be preserved by deidentification depends very much on the prediction task one wants to tackle. For example, in our approach, the temporal order of the data is preserved only within a patient but not across all patients. In particular, the total number of patients in the ICU at a certain point in time, a potentially relevant input for use case 1, can no longer be determined after deidentification. For the same reason, it is impossible to detect *domain shifts* in the deidentified data, which are systematic changes in the distribution of the data over time (domain shifts can be caused by many different factors such as new measurement equipment, laboratory test procedures, or changes in the prevalence of diseases in the patient population). Therefore, all relevant temporal features that could not be computed after deidentification had to be extracted and added to the data before deidentification.

### Inspection and Exploratory Analysis

Real-world data can be corrupted or otherwise ill-behaved in many unforeseeable ways, in addition to well-known issues related to missing values or invalid measurements, that a thorough inspection and exploratory analysis is inevitable. Indeed, in our experience, this is one of the most labor-intensive tasks in the entire data preparation pipeline. Owing to the nature of the problem, it is difficult to devise general rules for what one should pay attention to. Instead, we report one particularly subtle issue encountered in our work. It might be specific to our hospital but is meant to serve as an illustrative example of what can unexpectedly happen when working with EHRs. More examples can be found in [Multimedia Appendix 2](#).

In use case 4, we made heavy use of physiological waveform signals, such as electrocardiogram, arterial pressure, and oxygen saturation, to predict whether the condition of ICU patients will deteriorate within the next 15 minutes. Waveform signals are recorded by the Bedmaster system and can be exported as arrays of numerical values. It should be clear that because of the way in which these data are measured, there can be many types of measurement artifacts in the signals; that is, highly unusual waveform morphology caused by slipped sensors, or patient movements. This must be expected and addressed either explicitly by automatically detecting periods of invalid waveform data [36] or implicitly by relying on the subsequent ML algorithm’s ability to learn how to differentiate between normal and abnormal signals. An entirely unexpected issue is depicted in [Figure 3](#): occasionally, the signals assume constant low values for a short time. The natural guess of measurement errors (eg, caused by slipped sensors) is likely wrong because simply cutting out the constant low-value period leads to smooth curves in all inspected cases. Such situations may thus indicate data artifacts of unknown origin that must be removed to obtain coherent signals, but strangely they do not always occur in all simultaneously recorded waveforms at the same time. Therefore, we opted to refrain from cutting out fragments of the raw signals to avoid a possible temporal misalignment of different waveforms.

**Figure 3.** Short periods of constant low values in waveform signals might have to be cut out. Left: original signal with a 0.5-second period of constant low values. Right: signal after cutting out the low value; as can be seen, the 2 ends of the signal fit perfectly.



## Implementation

Eventually, the pipeline for extracting and preprocessing all relevant EHR data must be implemented in the actual code. This can be challenging in many respects. First, it is tempting to make extensive use of technologies aimed at rapid prototyping (eg, Jupyter notebooks) to quickly experiment with the data and preprocess them for a particular use case in a particular hard-coded way. This might work well in the short term; however, in the long term, a structured modular codebase that allows the exchange of individual components and adjustment (and logging!) of configuration settings is the better approach. In particular, the logging of configuration settings is of utmost importance to know exactly how data were preprocessed and how models were generated, thereby obtaining reproducible results.

Second, pipelines implemented to process a specific data modality for a particular use case should be reusable in other use cases that depend on the same data modality, at least to a certain extent. Even if the desired output format of the data after preprocessing differs between the 2 use cases, there are almost certainly some steps in the pipeline that are applicable to both. Reusing existing functionality rather than reimplementing it enables a consistent treatment of data across use cases and as a side effect may even help to abstract from the peculiarities of one use case and implement preprocessing functionality in a more general way. For example, we used laboratory test results in each of the 5 use cases either as features or for assigning labels (or both). In use case 4, the last 3 measurements of a fixed set of laboratory parameters relative to a given point in time are used as features, whereas in use case 3, the last measured value of a certain parameter before a blood transfusion is compared with measurements after the transfusion to determine whether it incurred a complication. Both are special cases of the more general principle of finding the last or first  $n$  measured values before or after a given point in time and could hence be implemented in one common function.

Finally, the inclusion of general-purpose third-party tools in the data preparation pipeline clearly has its benefits as well as potential downsides. On the one hand, functionality implemented therein does not have to be reimplemented (nor tested) from scratch, but on the other hand, Sculley et al [14] point out that it may lead to many glue code and pipeline jungles for bringing

data into the right shape. In our project, we restricted ourselves to well-established libraries from the Python ecosystem, including NumPy [41], Pandas [42], and scikit-learn [43] and deliberately avoided tools such as FIDDLE [16]. The former 3 are libraries of useful classes and functions that can be easily integrated into one's own pipeline. The latter implements a full medical data preparation pipeline itself, which, although being generic and customizable in principle, did not offer the amount of flexibility we would have required to accommodate our data.

More precisely, our data preparation pipeline consists of 3 main steps: harmonization, validation, and transformation. Harmonization, that is, ensuring that equivalent concepts are represented consistently, is very specific to each data modality and typically amounts to assigning unique names to equivalent variables and converting measured values into a common unit of measurement. Validation of recorded values happens with respect to manually specified, threshold-based rules. Analogous to Harutyunyan et al [3], we distinguished between invalid numerical values and extreme outliers. Each validation rule is characterized by 2 ranges  $r1 \subseteq r2$ , where everything inside  $r1$  is deemed admissible and everything outside  $r2$  is deemed an extreme outlier. Extreme outliers  $x \notin r2$  are deleted entirely, whereas the values  $x \in r2 \setminus r1$  are set to the nearest admissible value in  $r1$ . Finally, transformation also depends on the specific data modality under consideration but is often concerned with resampling EHR tables in an event-based entity-attribute-value format into a more ML-friendly wide table format. This proceeds by aggregating all observations within a given time window with respect to a fixed set of rules, such as taking the mean, sum, or temporally last of all the measured values. If a variable has not been measured at all in a time window, the "missing" recordings are imputed. As other authors have noted [44], clinical measurements are not missing at random; therefore, explicit *missingness masks* indicating whether a value has been imputed are added as extra features. In general, one must also be careful when imputing the mean or median of all observed values, as this could introduce bias. For example, if a variable is only measured if a patient has a certain condition, the measured values are not representative of the entire population.

## Discussion

### Principal Findings

The preceding sections illustrate that the preparation of EHRs for secondary analysis and the development of prediction models constitute a challenging endeavor. In addition to the well-known ubiquitous data problems for which generic off-the-shelf solutions exist (eg, imputation of missing values), we identified many issues in our raw data that had to be addressed individually. Even worse, none of these issues could be expected or popped up during the first quick scan of the data but instead were discovered only after a thorough exploratory analysis. Different kinds of patient identifiers being accidentally swapped is certainly something one would not expect at all, yet we found a few such cases in our data. The use of multiple codes or names for the same clinical concept is also not trivial to detect, especially if it is a mere artifact of the internal data representation that does not surface in clinical practice. If the mapping between codes and concepts changes over time, data harmonization becomes a true challenge. With regard to data validation, blindly discarding all nonnumeric values of a supposedly numeric variable fails to account for censored values such as “>120.0” (Multimedia Appendix 2) that do carry useful information. Finally, the subtle issues with waveform data reported above not only demand a thorough systematic analysis of timestamps and measured values but are also difficult to fix. Altogether, these observations support our claim that although generic tools such as FIDDLE [16] and Clairvoyance [25] doubtlessly do have their merits, one must be careful not to underestimate the additional effort of modality- and source-specific data analysis and preparation. In general, we believe that extensive libraries of well-documented, generic, and cleanly implemented functionalities focusing on the peculiarities of medical data preparation (harmonizing and validating physiological variables, resampling event-based entity-attribute-value tables into wide tables, etc) are more valuable than full-fledged end-to-end pipelines, regardless of how generic and configurable they are.

Extracting labels that indicate the outcome of interest from retrospective data can be more intricate than one might expect. Often, these outcomes (patient deterioration, organ system failure, optimal treatment policy, etc) are not explicitly recorded in EHRs and must therefore be approximated. The quality of such an approximation might influence not only the performance of the generated prediction models but also their applicability to clinical practice. Furthermore, if the definition of some label depends on scarcely recorded variables, only a few labeled samples may remain. In such a situation, methods based on *self-supervised and semisupervised learning* [45-47] might be the only remedy.

EHRs contain highly sensitive patient information that, for good reasons, must be deidentified before it can be shared with scientific partners in research projects. How and to what extent this needs to be carried out often not clearly defined, especially regarding the treatment of temporal information. Temporal data may contain highly relevant information depending on the concrete use case. On the one hand, knowing the time of day

and day of week of a particular event is necessary if the prediction task at hand has to take clinical routines into account; on the other hand, knowing the (rough) order of events across different patients enables detecting domain shifts in the underlying data distribution. Finally, if the use of a particular resource at any given point in time is of interest, this information must be extracted before deidentifying the timestamps, or timestamp deidentification must be avoided entirely. In our experience, it is good to first determine the kind of information one needs for a particular use case and then devise deidentification strategies that preserve as much of the previously determined information as possible while observing legal regulations and hospital-internal restrictions.

Finally, if the ultimate goal of developing prediction models is to deploy them in clinical practice, data access becomes a factor that must be considered. The more manual steps involved in exporting the data from the hospital IT infrastructure into the desired format, the more difficult real-time deployment will be. In our use case 4, automatically exporting the necessary data of all current ICU patients after every  $n$  minutes and then promptly processing them is challenging and currently work in progress. This mainly owes to the fact that the entire data warehousing system of the Kepler University Hospital was designed for clinical use rather than real-time analysis. However, alternatives exist; a sophisticated solution for efficient storage of and access to medical data for data science projects is presented in a study by McPadden et al [48].

### Workflow

The data preparation workflow we followed in our project is summarized in Figure 4, with rough estimates of the relative time and effort taken by the individual steps. We think that it generalizes to other data science projects with retrospective EHR data and hope that it can serve as guidance for other researchers to identify and address potential problems early and avoid some common pitfalls.

The presentation of the (linear) workflow in Figure 4 is simplified because in reality, there are many feedback loops. For instance, inspecting the data may reveal issues that can only be rectified if additional information is extracted from the system, and some issues might only surface after developing the first prediction models.

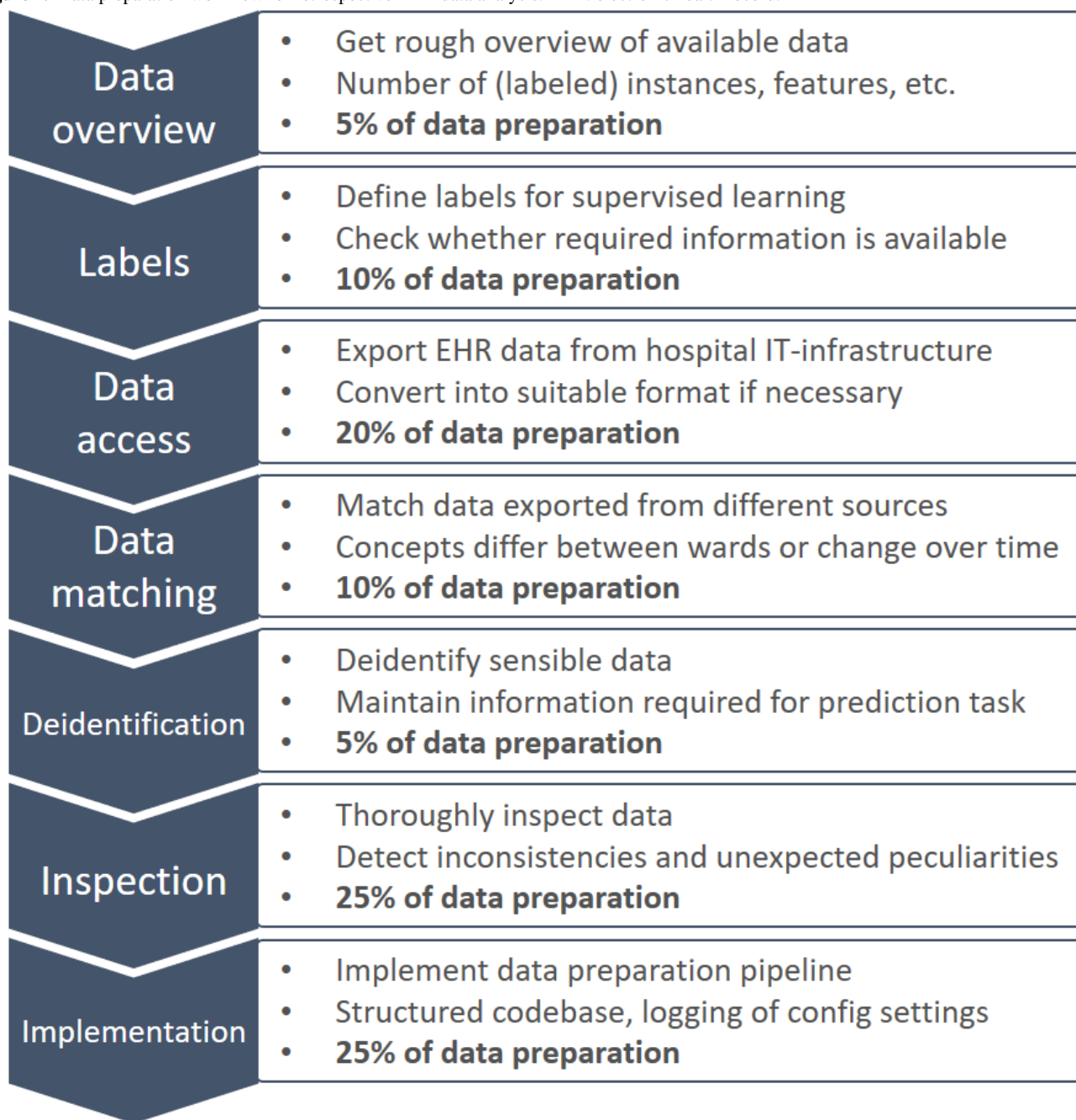
It is important to note that the results presented in this paper only refer to data preparation for subsequent model development but not to the development and validation of actual prediction models. We think these are “standard” tasks in data science and ML that are not specific to medical data. However, we do acknowledge that selecting the appropriate class of prediction models for a given task, optimizing hyperparameters, and training models in the right way are by no means trivial and require a lot of time and effort. This is also true for deploying models in clinical practice, where topics such as *handling domain shifts, detecting out-of-distribution data, and explaining model decisions in a manner comprehensible to patients* must be addressed. Things become even more difficult if existing models are to be deployed in other hospitals because most of the steps in the above workflow must be repeated. Only the definition of labels and (possibly) deidentification can be



skipped, and some parts of the existing pipeline implementation can perhaps be reused. According to our rough estimate, approximately 75% of the effort invested in the initial data preparation for developing prediction models must be reinvested

for each hospital that these models are deployed. As noted in a study by Sendak et al [30], this incurs significant additional costs.

**Figure 4.** Data preparation workflow for retrospective EHR data analysis. EHR: electronic health record.



**Conclusions**

Preparing raw medical data from productive environments for retrospective analysis and ML remains challenging and time consuming. Our findings suggest that real-world EHR data can be messy and corrupted in so many subtle ways that thorough

exploratory analysis and tailor-made preprocessing functionality for the data at hand are inevitable. We want to create awareness of this fact and hope that the sketched data preparation workflow becomes a valuable guidance for future large-scale data science projects involving routinely acquired medical data.

**Acknowledgments**

This research was funded by Medical Cognitive Computing Center (MC<sup>3</sup>) and supported by the strategic economic research program “Innovatives OÖ 2020” of the province of Upper Austria. This project was cofinanced by research subsidies granted by the government of Upper Austria and supported by the University SAL Labs initiative of Silicon Austria Labs (SAL) and its

Austrian partner universities for applied fundamental research for electronic based systems. RISC Software GmbH is a member of the Upper Austrian Research Innovation Network. The authors thank Philipp Moser for carefully proofreading a draft version of the manuscript.

## Conflicts of Interest

None declared.

## Multimedia Appendix 1

Brief evaluation and comparison of four EHR data preparation tools.

[\[DOCX File , 18 KB-Multimedia Appendix 1\]](#)

## Multimedia Appendix 2

Overview of the data modalities used in our research and specific issues encountered when processing them.

[\[DOCX File , 137 KB-Multimedia Appendix 2\]](#)

## References

1. Rajkumar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digit Med* 2018 May 8;1:18 [FREE Full text] [doi: [10.1038/s41746-018-0029-1](https://doi.org/10.1038/s41746-018-0029-1)] [Medline: [31304302](https://pubmed.ncbi.nlm.nih.gov/31304302/)]
2. Purushotham S, Meng C, Che Z, Liu Y. Benchmarking deep learning models on large healthcare datasets. *J Biomed Inform* 2018 Jul;83:112-134 [FREE Full text] [doi: [10.1016/j.jbi.2018.04.007](https://doi.org/10.1016/j.jbi.2018.04.007)] [Medline: [29879470](https://pubmed.ncbi.nlm.nih.gov/29879470/)]
3. Harutyunyan H, Khachatrian H, Kale DC, Ver Steeg G, Galstyan A. Multitask learning and benchmarking with clinical time series data. *Sci Data* 2019 Jun 17;6(1):96 [FREE Full text] [doi: [10.1038/s41597-019-0103-9](https://doi.org/10.1038/s41597-019-0103-9)] [Medline: [31209213](https://pubmed.ncbi.nlm.nih.gov/31209213/)]
4. Caicedo-Torres W, Gutierrez J. ISeeU: visually interpretable deep learning for mortality prediction inside the ICU. *J Biomed Inform* 2019 Oct;98:103269 [FREE Full text] [doi: [10.1016/j.jbi.2019.103269](https://doi.org/10.1016/j.jbi.2019.103269)] [Medline: [31430550](https://pubmed.ncbi.nlm.nih.gov/31430550/)]
5. Hatib F, Jian Z, Buddi S, Lee C, Settels J, Sibert K, et al. Machine-learning algorithm to predict hypotension based on high-fidelity arterial pressure waveform analysis. *Anesthesiology* 2018 Oct;129(4):663-674 [FREE Full text] [doi: [10.1097/ALN.0000000000002300](https://doi.org/10.1097/ALN.0000000000002300)] [Medline: [29894315](https://pubmed.ncbi.nlm.nih.gov/29894315/)]
6. Hyland SL, Faltys M, Hüser M, Lyu X, Gumbsch T, Esteban C, et al. Early prediction of circulatory failure in the intensive care unit using machine learning. *Nat Med* 2020 Mar;26(3):364-373. [doi: [10.1038/s41591-020-0789-4](https://doi.org/10.1038/s41591-020-0789-4)] [Medline: [32152583](https://pubmed.ncbi.nlm.nih.gov/32152583/)]
7. Corey KM, Kashyap S, Lorenzi E, Lagoo-Deenadayalan SA, Heller K, Whalen K, et al. Development and validation of machine learning models to identify high-risk surgical patients using automatically curated electronic health record data (Pythia): a retrospective, single-site study. *PLoS Med* 2018 Nov 27;15(11):e1002701 [FREE Full text] [doi: [10.1371/journal.pmed.1002701](https://doi.org/10.1371/journal.pmed.1002701)] [Medline: [30481172](https://pubmed.ncbi.nlm.nih.gov/30481172/)]
8. Bottino F, Tagliente E, Pasquini L, Napoli AD, Lucignani M, Figà-Talamanca L, et al. COVID mortality prediction with machine learning methods: a systematic review and critical appraisal. *J Pers Med* 2021 Sep 07;11(9):893 [FREE Full text] [doi: [10.3390/jpm11090893](https://doi.org/10.3390/jpm11090893)] [Medline: [34575670](https://pubmed.ncbi.nlm.nih.gov/34575670/)]
9. Tschoellitsch T, Dünser M, Böck C, Schwarzbauer K, Meier J. Machine learning prediction of SARS-CoV-2 polymerase chain reaction results with routine blood tests. *Lab Med* 2021 Mar 15;52(2):146-149 [FREE Full text] [doi: [10.1093/labmed/lmaa111](https://doi.org/10.1093/labmed/lmaa111)] [Medline: [33340312](https://pubmed.ncbi.nlm.nih.gov/33340312/)]
10. Yang D, Martinez C, Visuña L, Khandhar H, Bhatt C, Carretero J. Detection and analysis of COVID-19 in medical images using deep learning techniques. *Sci Rep* 2021 Oct 04;11(1):19638 [FREE Full text] [doi: [10.1038/s41598-021-99015-3](https://doi.org/10.1038/s41598-021-99015-3)] [Medline: [34608186](https://pubmed.ncbi.nlm.nih.gov/34608186/)]
11. Roland T, Böck C, Tschoellitsch T, Maletzky A, Hochreiter S, Meier J, et al. Domain shifts in machine learning based Covid-19 diagnosis from blood tests. *J Med Syst* 2022 Mar 29;46(5):23 [FREE Full text] [doi: [10.1007/s10916-022-01807-1](https://doi.org/10.1007/s10916-022-01807-1)] [Medline: [35348909](https://pubmed.ncbi.nlm.nih.gov/35348909/)]
12. Guleria P, Ahmed S, Alhumam A, Srinivasu PN. Empirical study on classifiers for earlier prediction of COVID-19 infection cure and death rate in the Indian states. *Healthcare (Basel)* 2022 Jan 02;10(1):85 [FREE Full text] [doi: [10.3390/healthcare10010085](https://doi.org/10.3390/healthcare10010085)] [Medline: [35052249](https://pubmed.ncbi.nlm.nih.gov/35052249/)]
13. 2016 Data Science Report. CrowdFlower. 2016. URL: [https://visit.figure-eight.com/rs/416-ZBE-142/images/CrowdFlower\\_DataScienceReport\\_2016.pdf](https://visit.figure-eight.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf); [accessed 2022-09-23]
14. Sculley D, Holt G, Golovin D, Davydov E, Phillips T, Ebner D, et al. Hidden technical debt in machine learning systems. In: *Proceedings of the 2015 Advances in Neural Information Processing Systems*. 2015 Presented at: NeurIPS '15; December 7-12, 2015; Montreal, Canada p. 2503-2511.
15. Wang S, McDermott MB, Chauhan G, Ghassemi M, Hughes MC, Naumann T. MIMIC-Extract: a data extraction, preprocessing, and representation pipeline for MIMIC-III. In: *Proceedings of the ACM Conference on Health, Inference, and Learning*. 2020 Apr Presented at: CHIL '20; April 2-4, 2020; Toronto, Canada p. 222-235. [doi: [10.1145/3368555.3384469](https://doi.org/10.1145/3368555.3384469)]

16. Tang S, Davarmanesh P, Song Y, Koutra D, Sjoding MW, Wiens J. Democratizing EHR analyses with FIDDLE: a flexible data-driven preprocessing pipeline for structured clinical data. *J Am Med Inform Assoc* 2020 Dec 09;27(12):1921-1934 [FREE Full text] [doi: [10.1093/jamia/ocaa139](https://doi.org/10.1093/jamia/ocaa139)] [Medline: [33040151](https://pubmed.ncbi.nlm.nih.gov/33040151/)]
17. Mandyam A, Yoo EC, Soules J, Laudanski K, Engelhardt BE. COP-E-CAT: cleaning and organization pipeline for EHR computational and analytic tasks. In: *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. 2021 Aug Presented at: BCB '21; August 1-4, 2021; Gainesville, FL, USA p. 1-9. [doi: [10.1145/3459930.3469536](https://doi.org/10.1145/3459930.3469536)]
18. Johnson AE, Ghassemi MM, Nemati S, Niehaus KE, Clifton DA, Clifford GD. Machine learning and decision support in critical care. *Proc IEEE Inst Electr Electron Eng* 2016 Feb;104(2):444-466 [FREE Full text] [doi: [10.1109/JPROC.2015.2501978](https://doi.org/10.1109/JPROC.2015.2501978)] [Medline: [27765959](https://pubmed.ncbi.nlm.nih.gov/27765959/)]
19. Attribution 4.0 International (CC BY 4.0). Creative Commons. URL: <https://creativecommons.org/licenses/by/4.0/> [accessed 2022-10-05]
20. Mandel JC, Kreda DA, Mandl KD, Kohane IS, Ramoni RB. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *J Am Med Inform Assoc* 2016 Sep;23(5):899-908 [FREE Full text] [doi: [10.1093/jamia/ocv189](https://doi.org/10.1093/jamia/ocv189)] [Medline: [26911829](https://pubmed.ncbi.nlm.nih.gov/26911829/)]
21. Johnson AE, Pollard TJ, Shen L, Lehman LW, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. *Sci Data* 2016 May 24;3:160035 [FREE Full text] [doi: [10.1038/sdata.2016.35](https://doi.org/10.1038/sdata.2016.35)] [Medline: [27219127](https://pubmed.ncbi.nlm.nih.gov/27219127/)]
22. Johnson AE, Stone DJ, Celi LA, Pollard TJ. The MIMIC code repository: enabling reproducibility in critical care research. *J Am Med Inform Assoc* 2018 Jan 01;25(1):32-39 [FREE Full text] [doi: [10.1093/jamia/ocx084](https://doi.org/10.1093/jamia/ocx084)] [Medline: [29036464](https://pubmed.ncbi.nlm.nih.gov/29036464/)]
23. Johnson A, Bulgarelli L, Pollard T, Horng S, Celi LA, Mark R. MIMIC-IV (version 2.0). *PhysioNet* 2022. [doi: [10.13026/7vcr-e114](https://doi.org/10.13026/7vcr-e114)]
24. Pollard TJ, Johnson AE, Raffa JD, Celi LA, Mark RG, Badawi O. The eICU collaborative research database, a freely available multi-center database for critical care research. *Sci Data* 2018 Sep 11;5:180178 [FREE Full text] [doi: [10.1038/sdata.2018.178](https://doi.org/10.1038/sdata.2018.178)] [Medline: [30204154](https://pubmed.ncbi.nlm.nih.gov/30204154/)]
25. Jarrett D, Yoon J, Bica I, Qian Z, Ercole A, van der Schaar M. Clairvoyance: a pipeline toolkit for medical time series. In: *Proceedings of the 8th International Conference on Learning Representations*. 2020 Presented at: ICLR '20; April 26-30, 2020; Addis Ababa, Ethiopia p. 1-16.
26. Shi X, Prins C, Van Pottelbergh G, Mamouris P, Vaes B, De Moor B. An automated data cleaning method for Electronic Health Records by incorporating clinical knowledge. *BMC Med Inform Decis Mak* 2021 Sep 17;21(1):267 [FREE Full text] [doi: [10.1186/s12911-021-01630-7](https://doi.org/10.1186/s12911-021-01630-7)] [Medline: [34535146](https://pubmed.ncbi.nlm.nih.gov/34535146/)]
27. Weiskopf NG, Bakken S, Hripcsak G, Weng C. A data quality assessment guideline for electronic health record data reuse. *EGEMS (Wash DC)* 2017 Sep 04;5(1):14 [FREE Full text] [doi: [10.5334/egems.218](https://doi.org/10.5334/egems.218)] [Medline: [29881734](https://pubmed.ncbi.nlm.nih.gov/29881734/)]
28. Mashoufi M, Ayatollahi H, Khorasani-Zavareh D. A review of data quality assessment in emergency medical services. *Open Med Inform J* 2018 May 31;12:19-32 [FREE Full text] [doi: [10.2174/1874431101812010019](https://doi.org/10.2174/1874431101812010019)] [Medline: [29997708](https://pubmed.ncbi.nlm.nih.gov/29997708/)]
29. Terry AL, Stewart M, Cejic S, Marshall JN, de Lusignan S, Chesworth BM, et al. A basic model for assessing primary health care electronic medical record data quality. *BMC Med Inform Decis Mak* 2019 Feb 12;19(1):30 [FREE Full text] [doi: [10.1186/s12911-019-0740-0](https://doi.org/10.1186/s12911-019-0740-0)] [Medline: [30755205](https://pubmed.ncbi.nlm.nih.gov/30755205/)]
30. Sendak MP, Balu S, Schulman KA. Barriers to achieving economies of scale in analysis of EHR data. A cautionary tale. *Appl Clin Inform* 2017 Aug 09;8(3):826-831 [FREE Full text] [doi: [10.4338/ACI-2017-03-CR-0046](https://doi.org/10.4338/ACI-2017-03-CR-0046)] [Medline: [28837212](https://pubmed.ncbi.nlm.nih.gov/28837212/)]
31. BedMasterEx. Data acquisition and infinite storage of medical device data. Anandic Medical Systems. URL: <https://www.bedmaster.net/en/products/bedmasterex>; [accessed 2022-09-23]
32. Ostermann M, Bellomo R, Burdmann EA, Doi K, Endre ZH, Goldstein SL, Conference Participants. Controversies in acute kidney injury: conclusions from a Kidney Disease: Improving Global Outcomes (KDIGO) Conference. *Kidney Int* 2020 Aug;98(2):294-309 [FREE Full text] [doi: [10.1016/j.kint.2020.04.020](https://doi.org/10.1016/j.kint.2020.04.020)] [Medline: [32709292](https://pubmed.ncbi.nlm.nih.gov/32709292/)]
33. ARDS Definition Task Force, Ranieri VM, Rubenfeld GD, Thompson BT, Ferguson ND, Caldwell E, et al. Acute respiratory distress syndrome: the Berlin definition. *JAMA* 2012 Jun 20;307(23):2526-2533. [doi: [10.1001/jama.2012.5669](https://doi.org/10.1001/jama.2012.5669)] [Medline: [22797452](https://pubmed.ncbi.nlm.nih.gov/22797452/)]
34. Cheplygina V, Tax DM, Loog M. Multiple instance learning with bag dissimilarities. *Pattern Recognit* 2015 Jan;48(1):264-275. [doi: [10.1016/j.patcog.2014.07.022](https://doi.org/10.1016/j.patcog.2014.07.022)]
35. Bica I, Alaa AM, Jordon J, van der Schaar M. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In: *Proceedings of the 8th International Conference on Learning Representations*. 2020 Presented at: ICLR '20; April 26-30, 2020; Addis Ababa, Ethiopia p. 1-26. [doi: [10.48550/arXiv.2002.04083](https://doi.org/10.48550/arXiv.2002.04083)]
36. De Ryck T, De Vos M, Bertrand A. Change point detection in time series data using autoencoders with a time-invariant representation. *IEEE Trans Signal Process* 2021;69:3513-3524. [doi: [10.1109/TSP.2021.3087031](https://doi.org/10.1109/TSP.2021.3087031)]
37. Phan HT, Borca F, Cable D, Batchelor J, Davies JH, Ennis S. Automated data cleaning of paediatric anthropometric data from longitudinal electronic health records: protocol and application to a large patient cohort. *Sci Rep* 2020 Jun 23;10(1):10164 [FREE Full text] [doi: [10.1038/s41598-020-66925-7](https://doi.org/10.1038/s41598-020-66925-7)] [Medline: [32576940](https://pubmed.ncbi.nlm.nih.gov/32576940/)]
38. International Statistical Classification of Diseases and Related Health Problems. 10th Revision. World Health Organization. 2019. URL: <https://icd.who.int/browse10/2019/en>; [accessed 2022-09-23]

39. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997 Nov 15;9(8):1735-1780. [doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735)] [Medline: [9377276](https://pubmed.ncbi.nlm.nih.gov/9377276/)]
40. Thorat PJ, Peppink JM, Driessen RH, Sijbrands EJ, Kompanje EJ, Kaplan L, Amsterdam University Medical Centers Database (AmsterdamUMCdb) Collaborators and the SCCM/ESICM Joint Data Science Task Force. Sharing ICU patient data responsibly under the society of critical care medicine/European Society of Intensive Care Medicine Joint Data Science Collaboration: the Amsterdam University Medical Centers Database (AmsterdamUMCdb) example. *Crit Care Med* 2021 Jun 01;49(6):e563-e577 [FREE Full text] [doi: [10.1097/CCM.0000000000004916](https://doi.org/10.1097/CCM.0000000000004916)] [Medline: [33625129](https://pubmed.ncbi.nlm.nih.gov/33625129/)]
41. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. *Nature* 2020 Sep;585(7825):357-362 [FREE Full text] [doi: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2)] [Medline: [32939066](https://pubmed.ncbi.nlm.nih.gov/32939066/)]
42. McKinney W. Data structures for statistical computing in Python. In: *Proceedings of the 9th Python in Science Conference*. 2010 Presented at: SciPy '10; June 28-July 3, 2010; Austin, TX, USA p. 56-61. [doi: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a)]
43. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. *J Mach Learn Res* 2011;12(2011):2825-2830.
44. Li J, Yan XS, Chaudhary D, Avula V, Mudiganti S, Husby H, et al. Imputation of missing values for electronic health record laboratory data. *NPJ Digit Med* 2021 Oct 11;4(1):147 [FREE Full text] [doi: [10.1038/s41746-021-00518-0](https://doi.org/10.1038/s41746-021-00518-0)] [Medline: [34635760](https://pubmed.ncbi.nlm.nih.gov/34635760/)]
45. Vincent P, Larochelle H, Bengio Y, Manzagol PA. Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th International Conference on Machine Learning*. 2008 Presented at: ICML '08; July 5-9, 2008; Helsinki, Finland p. 1096-1103. [doi: [10.1145/1390156.1390294](https://doi.org/10.1145/1390156.1390294)]
46. Yoo J, Zhang Y, Jordon J, van der Schaar M. VIME: extending the success of self- and semi-supervised learning to tabular domain. In: *Proceedings of the 2020 Advances in Neural Information Processing Systems*. 2020 Presented at: NeurIPS '20; December 6-12, 2020; Virtual p. 11033-11043.
47. Arik S, Pfister T. TabNet: attentive interpretable tabular learning. *Proc AAAI Conf Artif Intell* 2021 May 18;35(8):6679-6687. [doi: [10.1609/aaai.v35i8.16826](https://doi.org/10.1609/aaai.v35i8.16826)]
48. McPadden J, Durant TJ, Bunch DR, Coppi A, Price N, Rodgerson K, et al. Health care and precision medicine research: analysis of a scalable data science platform. *J Med Internet Res* 2019 Apr 09;21(4):e13043 [FREE Full text] [doi: [10.2196/13043](https://doi.org/10.2196/13043)] [Medline: [30964441](https://pubmed.ncbi.nlm.nih.gov/30964441/)]

## Abbreviations

- ED:** emergency department
- EHR:** electronic health record
- eICU-CRD:** telehealth ICU collaborative research database
- HIS:** hospital information system
- ICU:** intensive care unit
- MIMIC:** Medical Information Mart for Intensive Care
- ML:** machine learning
- PDMS:** patient data management system

*Edited by C Lovis; submitted 07.04.22; peer-reviewed by L Celi, FM Calisto, M Sendak; comments to author 09.07.22; revised version received 02.08.22; accepted 07.09.22; published 21.10.22*

*Please cite as:*

Maletzky A, Böck C, Tschoellitsch T, Roland T, Ludwig H, Thumfart S, Giretzlehner M, Hochreiter S, Meier J

*Lifting Hospital Electronic Health Record Data Treasures: Challenges and Opportunities*

*JMIR Med Inform* 2022;10(10):e38557

URL: <https://medinform.jmir.org/2022/10/e38557>

doi: [10.2196/38557](https://doi.org/10.2196/38557)

PMID:

©Alexander Maletzky, Carl Böck, Thomas Tschoellitsch, Theresa Roland, Helga Ludwig, Stefan Thumfart, Michael Giretzlehner, Sepp Hochreiter, Jens Meier. Originally published in *JMIR Medical Informatics* (<https://medinform.jmir.org>), 21.10.2022. This is an open-access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work, first published in *JMIR Medical Informatics*, is properly cited. The complete bibliographic information, a link to the original publication on <https://medinform.jmir.org/>, as well as this copyright and license information must be included.