

Event Log Generation in MIMIC-IV Research Paper

Jonas Cremerius¹, Luise Pufahl², Finn Klessascheck¹, and Mathias Weske¹

¹ Hasso Plattner Institute, University of Potsdam, Germany

{Jonas.Cremerius, Mathias.Weske}@hpi.de, Finn.Klessascheck@student.hpi.de

² Software & Business Engineering, Technische Universitaet Berlin, Germany

Luise.Pufahl@tu-berlin.de

Abstract. Public event logs are valuable for process mining research to evaluate process mining artifacts and identify new and promising research directions. Initiatives like the BPI Challenges have provided a series of real-world event logs, including healthcare processes, and have significantly stimulated process mining research. However, the healthcare related logs provide only excerpts of patient visits in hospitals. The Medical Information Mart for Intensive Care (MIMIC)-IV database is a public available relational database that includes data on patient treatment in a tertiary academic medical center in Boston, USA. It provides complex care processes in a hospital from end-to-end. To facilitate the use of MIMIC-IV in process mining and to increase the reproducibility of research with MIMIC, this paper provides a framework consisting of a method, an event hierarchy, and a log extraction tool for extracting useful event logs from the MIMIC-IV database. We demonstrate the framework on a heart failure treatment process, show how logs on different abstraction levels can be generated, and provide configuration files to generate event logs of previous process mining works with MIMIC.

Keywords: Event Log Generation · Process Mining · Healthcare · MIMIC

1 Introduction

Process mining methods and techniques are experiencing a tremendous uptake in a broad range of organizations. These techniques help to make the real-world execution of business processes more transparent and support an evidenced-based process analysis and redesign [23]. Therefore, process mining receives increased attention in the healthcare domain, where traditionally manifold data is logged due to quality control and billing purposes [8, 22].

Public available event logs, in which the process data is stored as an ordered list, are essential for developing new process mining techniques and methods and evaluating their impact and limitations. In recent years, different initiatives, such as the BPI Challenges running since 2011, e.g., [7], or the conformance checking challenge [20], and additional research including [17] have provided publicly accessible event logs.

Public data sets are relevant to stimulate research in healthcare as well. For example, MIMIC (Medical Information Mart for Intensive Care) is a large, de-identified relational database including patients that received critical care in the Beth Israel Deaconess Medical Center [13] in Boston, USA. Whereas MIMIC-III contains data about activities in the intensive care unit (ICU) between 2001-2012, MIMIC-IV provides data on the complete hospital stay between 2008-2019, including procedures performed, medications given, laboratory values taken, triage information, and more. It provides the opportunity to develop and evaluate process mining techniques on patient care processes, such as in [2, 15]. However, the uptake of this rich and data-intensive database is limited in the process mining community so far.

The relational database’s complexity, the data’s richness, and the need to flatten the data in a meaningful way in an event log has hampered the uptake of MIMIC-IV by the process mining community. Additionally, access to MIMIC-IV requires a data use agreement, including a training provided by the Collaborative Institutional Training Initiative (CITI) about collecting, using and disclosing health information. However, the process to gain access is clearly defined and usually does not take more than a few days.

This paper aims at simplifying the event log extraction from the MIMIC-IV database and its reusability. In particular, it provides a framework including an extraction method, an event hierarchy for MIMIC-IV, and a Python log extraction tool to ease the log extraction from MIMIC-IV. The remainder of this paper is organized as follows. In the next section, background on MIMIC is given in Sect. 2, and related work is discussed in Sect. 3. The event extraction framework for MIMIC-IV is presented in Sect. 4, followed by an evaluation in Sect. 5. The paper concludes in Sect. 6.

2 MIMIC-IV Database

MIMIC-IV is a publicly available dataset provided by the Laboratory for Computational Physiology (LCP) at the Massachusetts Institute of Technology (MIT). It comprises de-identified health data associated with thousands of hospital admissions. The project was launched in the early 2000s with MIMIC-I. It is still ongoing with the recent release of MIMIC-IV, including data from 2008-2019.

The data is derived from a hospital-wide Electronic Health Record (EHR) and an Intensive Care Unit (ICU) specific system, such as MetaVison [13]. So far, MIMIC-IV contains data from a single hospital. The ultimate goal is to incorporate data from multiple institutions capable of supporting research on cohorts of critically ill patients worldwide. To ensure the data represents a real-world healthcare dataset, data cleaning steps were not performed [13].

The MIMIC-IV relational database consists of 35 tables separated into four modules consisting of emergency department (**ed**), hospital (**hosp**), intensive care unit (**icu**), and **core**. Fig. 1 illustrates a simplified data model of the database with its modules. In **core**, demographic information, such as age and marital status, transfers between departments, and admission information including their

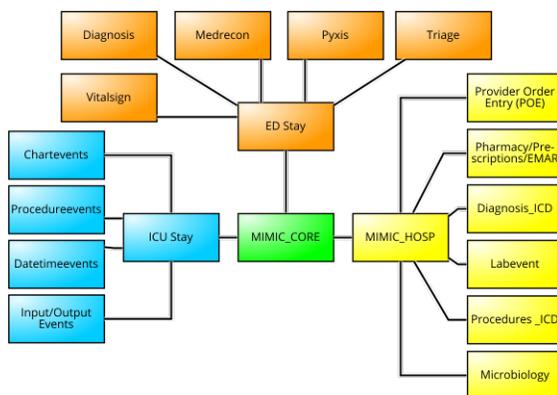


Fig. 1: MIMIC-IV 1.0 simplified data model. The colours represent the respective modules: Green: Core, Yellow: Hosp, Blue: ICU, Orange: ED

admission location is stored. The `hosp` module provides all data acquired from the hospital-wide electronic health record, including laboratory measurements, microbiology, medication administration, billed diagnoses/procedures, and orders made by providers. The `ed` module adds information about patients' first contact with the hospital in the emergency department, including data about triage, suspected diagnosis, and measurements made. Lastly, the `icu` module contains precise information obtained from an ICU visit, including machine recordings and procedures performed. This schema is conforming with MIMIC-IV 1.0. In June 2022, MIMIC-IV 2.0 was released, which transferred the tables from `core` to `hosp`, which is a minor change, as it modifies the high-level schema and not the relations between the tables. However, the documentation is still structured as shown in Fig. 1. The provided method including the log extraction tool is conform with both versions.

To ensure patient confidentiality, all dates in MIMIC-IV have been shifted randomly. Thus, process mining techniques, such as bottleneck analysis, are not possible to apply. However, dates are internally consistent with respect to each patient, so the actual time between events is preserved.

3 Related Work

In this section, we want to review research works on event log extraction, and on applying process mining to data from MIMIC.

Event Log Extraction. An event log serves as the basis for process mining techniques. However, the preparation of an event log is often not trivial as business processes might be executed with the help of multiple IT systems and the

Table 1: Research works on applying process mining to MIMIC-III

| Ref. | Year | Goal | Patient Cohort | Notion | Events |
|------|------|------------------------------|---|---------------|-----------------------------------|
| [2] | 2017 | reduce variation in pathways | congestive heart failure patients | hospital adm. | admission, lab, prescription, ICU |
| [15] | 2018 | assess the data quality | cancer patients | hospital adm. | admission |
| [14] | 2018 | analyse cancer pathways | cancer patients | hospital adm. | admission and icustays |
| [16] | 2020 | detect disease trajectories | ≤ 16 years, btw. 2001-2012, ≤ 2 stays | subject | admission and diagnosis |
| [18] | 2020 | compare ICU treatment | cancer patients | hospital adm. | ICU procedures |

data is often stored not in the structure of an event log, but often in relational databases [6]. For the interested reader, Diba et al. [6] provide a structured literature review on techniques for event data extraction, correlation, and abstraction to prepare an event log. Remy et al. [22] present challenges in the event log abstraction from a data warehouse of a large U.S. health system. Jans and Soffers describe in [11] relevant decisions that need to be made to create an event log from a relational database: related (1) to the process as a whole, such as “*which process should be selected and its exact scope?*”, (2) to the selection of the process instance, such as “*what is the notion of an instance*” and to the event level, such as “*what type of events and attributes to include*”. In a later research work, the authors [12] provide a nine-step procedure to create an event log from a relational database, starting with stating a goal over identifying key tables and relationships until defining the case notion, and selecting event types and their attributes. This procedure will serve as a basis to create a method for extracting event logs from MIMIC-IV.

In the last years, event log extraction approaches and tools were developed to support practitioners in extracting event logs from their databases, such as *on-prom* [4] using ontologies for the extraction, *eddytools* [9] for a case notion recommendation, and *RDB2Log* [3] for a quality-informed log extraction. Still, we observed that these tools could not be easily applied for MIMIC-IV. Reasons include the need to merge tables for obtaining complete information about events. Additionally, a patient cohort definition is necessary to deal with the complexity of healthcare processes. Thus, they are not used in this work.

Process Mining with MIMIC. This part presents research papers that used process mining to analyze the MIMIC database. The identified research works used the MIMIC-III database because MIMIC-IV has been published recently. We analyzed their goals, their used patient cohort, their used case notion, and selected event types for the event log preparation, summarized in Table 1.

Alharbi et al. [2], and Kurniati et al. [15] target methodological goals for the analysis of the healthcare data, such as reducing the variation in clinical pathway data and assessing the data quality. The other three research works follow

medical analysis goals, such as analyzing cancer pathways, comparing the treatment of different cancer types at the ICU, and detecting disease trajectories. It can be observed that, on the one hand, patient cohorts with a specific diagnosis were selected, such as cancer and congestive heart failure patients. However, on the other hand, a broader patient cohort was also selected in a specific age range and a certain length of stay. As a notion of the process instance, two applied solutions can be observed: The subject (i.e., the patient with their *subject.id*) or the hospital stay (i.e., *hadm.id*) is selected. Whereas the subject covers all events that happen to a specific patient, including possibly several hospital admissions, the hospital admission comprises only events related to one admission. If a patient had several admissions for a specific diagnosis, it is represented as different traces for this patient. Finally, the research works applying process mining to the MIMIC data using different event data are presented. The high-level `admission` events of the `core` including information on the time of admission, discharge, etc. were used [2, 14–16]. Kurniati et al. [14] select additionally high-level information on the ICU stay, such as `ICU intime`, whereas Marazza et al. [18] chose detail procedure events of the ICU stay. As Kusuma et al. [16] aim at detecting disease trajectories, they select additionally to the admission events also the diagnosis as an event. The diagnosis has no own timestamp, and the authors decided to use the time of admission. Alharbi et al. [2] select for their analysis a broad range of events, also lab, prescriptions, and ICU events.

It can be observed, that current research works on MIMIC use case-dependent SQL scripts that cannot be easily adapted for other use cases. This makes it difficult to reproduce the event log extraction and hinders researchers inexperienced with MIMIC to use this data source. In this research work, we want to provide an event log extraction tool to ease the access to MIMIC for the process mining community.

4 Event Log Extraction Framework for MIMIC-IV

This section presents the event log extraction framework for MIMIC-IV. It results from an analysis of related work and the MIMIC database and its documentation. Based on the event log preparation procedure by [12], we propose a method to derive event logs from MIMIC-IV including an event hierarchy in Section 4.1. In Section 4.2, the Python tool for event log extraction from MIMIC-IV is introduced.

4.1 Method and Event Hierarchy

The method to extract event logs from MIMIC-IV consists of six steps from goal definition and patient cohort definition, over selecting the case notion and attributes until the selection of event types and their enrichment, as shown in Fig. 2a. For each step, we describe the goal and activities, its mapping to the event preparation procedure by [12], and possibilities for configurations.

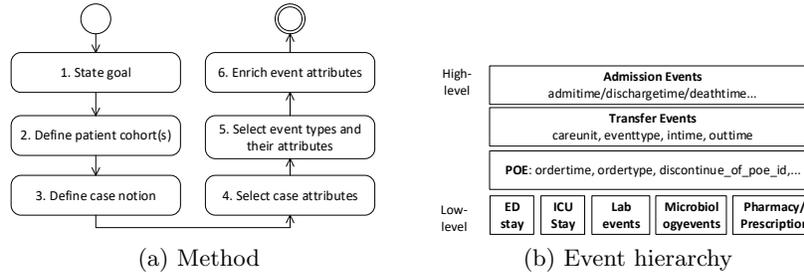


Fig. 2: Method for event log extraction from MIMIC-IV and its event hierarchy.

1) *State goal*. As described by [12] in step P1, for a useful event log preparation, the goal of the process mining project needs to be defined. The need for the goal definition also applies to the event log extraction from the MIMIC-IV database. Possible medical analysis goals are the process variant exploration of clinical pathways, disease trajectory modeling, conformance analysis to clinical guidelines, etc. [21]. It can also be a methodological goal, such as analyzing the data quality.

2) *Define patient cohort(s)*. As suggested by [12] in step P2, the boundaries of a process have to be defined. In healthcare, the scope of a process is usually defined by selecting a particular patient cohort, e.g., congestive heart failure patients [2] or cancer patients [15]. Patient cohorts are often selected via the diagnosis of the `hosp` stay with the help of the International Statistical Classification of Diseases and Related Health Problems (ICD) codes³—a global system to label medical diagnosis consistently. Another possibility are Diagnostic Related Groups (DRGs), a code system that is used for determining the costs or the reimbursement rate of a case. It is based on diagnoses, procedures, age, sex, discharge status, and the presence of complications or co-morbidities. Additionally, an age range or the length of stay could be used to focus on specific patient cohorts.

3) *Define case notion*. As given by [12] in step P5, an attribute has to be selected that determines the process instance (i.e., the case id of an event log). By analyzing the MIMIC-IV database and the related work, we identified two possible notions of cases, the subject identifier (the patient with its `subject_id`) or the hospital administration identifier (`hadm_id`). With `subject_id` the complete patient history, including several admissions, can be analyzed. With `hadm_id`, each patient admission is represented as an individual trace in the event log. Further, each hospital admission consists of stays in different departments, such as the ICU or ED stay, on which the focus could also be during the analysis. The instance granularity needs to be selected (step P6 [12]) and its parent and child activities. This is well-supported in MIMIC-IV: The main identifier, the `subject_id` and `hadm_id`, is available in all tables as a foreign key. Only `hadm_id`

³ <https://www.icd10data.com/ICD10CM/Codes>

is not available in the `ed` module, but the `stay_id` stored in `ed` tables can be mapped to an `hadm_id`.

4) *Select case attributes.* After the patient cohorts and the case notion have been selected, in the next step, additional attributes of cases, the traces in an event log, need to be selected as also suggested by [12] in step P8. Case attributes can be used to filter and cluster in the process mining project. Here available patient data, such as their *gender* or *age*, diagnosis data, such as the *ICD_code*, or admission data, such as *discharge_location* or *insurance* could be selected based on the selected case notion.

5) *Select event types and their attributes.* When the instances and their attributes are selected, the event types as also suggested by [12] in step P7 and event attributes in step P9 can be selected. Therefore, key tables (step P3 [12]) and their relationships (step P4) need to be identified. By analyzing MIMIC-IV and related work, we developed a hierarchy including possibly relevant event types for MIMIC-IV, as shown in Fig. 2b. The top shows the most high-level events, whereas the bottom shows low-level events. In the following, we present the different types of events in more detail, starting from the top:

Admission events, such as *admittime*, *dischargetime* etc., can be all together found in the `admissions` table of the `core` module. They provide high-level information about the patients' stays (e.g., when was the admission to the hospital or the discharge). Almost all related works have used this event type, either alone or with other event types, such as ICU stay information. If the admission events are requested, then all the "time"-events are provided including *admittime/dischargetime/deathtime* etc.

On the next level, the **transfer events** of the `transfers` table, also in the `core`, provide insights about which departments/care units a patient has visited during the hospital stay. These events can be used to analyze the path of a patient through the hospital. Each table entry represents one transfer event for which the *intime* or *outtime* can be selected to be used as a timestamp. The other attributes of this table are provided as event attributes.

The next level of detail is the **provider order entry (POE) events** that provide insights into ordered treatments and procedures for a patient. The `POE` table is part of the `hosp` module. These events do not represent the activities that have been finally executed, but they represent what has been planned and ordered for a patient. Additionally, the attributes *discontinue_of_poe_id* and *discontinued_by_poe* provide insights whether the order was cancelled. Each entry of the `POE` table represents one order for a patient of a specific hospital admission, and as timestamp, the *ordertime* can be used. The additional attributes of the `POE` table are added as event attributes. Some `POE` events, such as lab or medication events, can also be enriched with details about the activity execution from other tables. For instance, details on laboratory or microbiology examinations can be found in the `labevents` or `microbiologyevents` tables. The `pharmacy`, the `prescriptions` and the `EMAR` table provide details on the

medications that a patient has received⁴.

Finally, also low-level details on specific aspects of the hospital stay of a patient can be deduced from specific tables, such as events of the ED stay, ICU stay or the `labevents`. We allow deriving event data from any combination of low-level tables. For instance, medications prescribed (`prescription`) can be analysed in combination with procedures performed (`procedures_icd`).

6) *Enrich event attributes* Optionally, events can be enriched by additional event attributes from any other table in MIMIC-IV if events have multiple timestamps. For example, the `transfers` table includes the times when patients entered and left the respective hospital department, or the `pharmacy` table includes the times when a medication was started and ended to be given. As shown in [5], events from the `transfers` table can be enhanced by aggregated laboratory values, such that for each department visit, the average laboratory value is known and can be analyzed. We allow adding aggregated information from any table in MIMIC-IV, so that not only laboratory values but also medication or procedure information can be added.

4.2 Event Log Extraction Tool

The event log extraction tool that forms an integral part of the framework presented in this paper has been implemented using Python 3.8 and is available as an open-source tool on GitHub⁵. It implements the method for event log extraction from MIMIC-IV (cf. Fig. 2a, Sec. 4.1). For this, access to and credentials for a MIMIC-IV instance running on PostgreSQL are required. The tool provides two ways of extracting logs: Either a user is guided interactively through the method, being prompted for input along with the six steps, starting at the second, as stating the goal is not supported by us.

Or, a user can provide a configuration file⁶, which contains definitions and selections for one or more of the separate steps, as well as additional parameter configurations, such as the required database credentials. Then, the user is only asked to provide input for those steps that have not been configured using the configuration file. Thus, while logs that have been extracted out of MIMIC-IV cannot be shared due to the data use agreement, a configuration file defining the application of the extraction method on the MIMIC-IV database can be shared instead. We provide configuration files for the event logs presented in Section 5.

Besides that, it is possible to extract event logs either as a log file conforming to the XES standard (cf. [1]), or as a .csv file, depending on the desired format

⁴ The reason for having three tables is that medications are prescribed first and then given to a patient. The prescription is stored in `pharmacy` with detailed information in `prescription`. Administration details can be found in `EMAR`, where nurses scan a barcode at the patient’s bedside at the moment when the medication is given. The tables are connected through a common identifier `pharmacy_id`.

⁵ <https://github.com/bptlab/mimic-log-extraction>

⁶ Example configuration files and an explanation of what they can configure can be found in the tool’s GitHub repository.

and the tooling that is to be applied afterwards on the event log. For more in-depth information on how to install, configure, and run the tool, we refer the reader to the corresponding GitHub repository.

5 Evaluation

In the following, we evaluate the presented MIMIC log extraction framework in a twofold manner. First, we show how far we could replicate the event logs generated by other research works on process mining with MIMIC. Second, we apply the method to an example use case and demonstrate findings and research challenges.

Replicating Event Logs of research works in MIMIC. We were able to provide configuration files for almost all the related work presented in Sect. 3. One exception is [16], as they manually attached a timestamp to the `diagnoses_icd` table. It should be noted, that we could not generate the final event logs for all works, as some applied post-processing, such as event abstraction. However, we could replicate the cohort, case notion, case attribute, event, and event attribute selection of them, which is the goal of this tool so far. The configuration files can be found in the GitHub repository.

Demonstration on Heart Failure Treatment. We demonstrate the event log extraction method for MIMIC-IV and present one level of the event hierarchy in detail for the heart failure treatment case⁷.

The **goal (1)** of this demonstration is to discover the hospital treatment process of patients having heart failure and to identify, if common treatment practices are applied. The **cohort (2)** consists of heart failure patients. Heart failure is the leading cause of hospitalizations in the U.S. and represents one of the biggest cohorts in MIMIC-IV besides newborns, with 7,232 admissions [10]. It was chosen based on ICD codes and DRG codes⁸ related to heart failure. We have selected the hospital admission as the **case notion (3)**, because we want to focus on the steps taken specifically for patients with heart failure instead of analyzing the complete patient history. The chosen **case attributes (4)** are related to the hospital admission, such as *admittime*, *admission_location* and the list of diagnosis (from the `diagnosis_icd` table).

Regarding the chosen **event type (5)**, we will only present the POE level due to space limitations. The POE level provides a good overview of main activities of treating heart failure patients. The results for the other hierarchy levels can be found in the following report⁹.

⁷ The detailed event log descriptions with their configuration files can be found in a GitHub repository: https://github.com/bptlab/mimic-log-extraction/tree/main/sample.config_files.

⁸ The selected ICD and DRG codes can be found in the configuration files.

⁹ <https://github.com/bptlab/mimic-log-extraction/blob/main/EventLogGenerationReport.pdf>

The process model in Fig. 3 shows the sequence and frequency of heart failure related treatments and procedures ordered for the patients. We filtered manually for events that are typical activities performed for patients with heart failure [19]. We displayed frequency and case coverage in brackets for each activity. This process represents typical characteristics of healthcare processes, including highly repetitive tasks and flexible order of activities. It can be observed that monitoring is highly relevant for heart failure patients, especially telemetry is common for patients suffering from cardiac conditions, as well as X-rays or CT scans for the diagnosis. Additionally, activities for managing heart failure can be observed, such as oxygen therapy, renal replacement therapy in the form of hemodialysis, or palliative care [19].

Repetitive events, such as Vitals/Monitoring make it almost impossible to observe a process order, especially in directly follows graphs, as these events have a high amount of ingoing and outgoing arcs. Identifying these automatically and dealing with them can be an interesting way of making process models more readable. Additionally, one could think about methods and visualizations to analyse discontinued orders (*discontinue_of_poe_id* and *discontinued_by_poe*). As the POE level contains a high amount of different events, one could also think about methods supporting process analysts and domain experts to find events of interest.

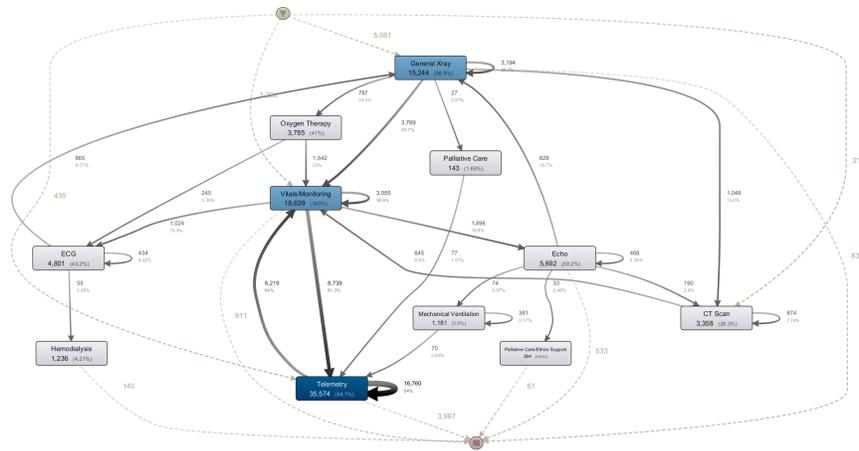


Fig. 3: POE events, showing treatments and procedures ordered at the hospital. Activity filter: Manually selected events given in a guideline [19] (10% of all with 100% case coverage), Paths filter: 45%

We see, that the POE level comes with interesting challenges for process mining

in healthcare. Also, the other identified event abstraction levels demonstrated relevant research challenges, which are discussed in the above-mentioned report. As there is a need for healthcare tailored frameworks in process mining, MIMIC could provide a necessary data source to research innovative solutions working on real-world data [21].

6 Conclusion

This paper presented a method, an event hierarchy, and a tool to extract event logs from MIMIC-IV, an anonymized database on hospitals stays, in a structured manner. The rich database of interacting healthcare processes including a high amount of additional event data offers process mining research for healthcare a relevant source of event logs for developing and evaluating new process mining artifacts. We demonstrated for a heart failure use case how event logs can be created and presented challenges coming along with healthcare processes.

The presented MIMIC-IV log extraction tool focuses on event log extraction only, and does not provide functionality for further processing, which could be extended in the future. Additionally, the tool extracts currently one event of a medical activity with a selected timestamp and stores the other timestamps as event attributes. The XES standard allows having multiple events of an activity representing its lifecycle changes. In future, our framework could be extended, such that multiple events of a medical activity, such as the ordering, its start and end can be captured as individual events. In the use case demonstration, we have, on the lower abstraction level, manually filtered for relevant events after the event log extraction. This could be improved in the future by supporting the event selection based on user preferences.

Event logs from this database cannot be directly shared because of a data use agreement. With our tool, configuration files for the event log extraction can be easily shared supporting reproducibility and extensibility of research. As a result of this work, the configuration files of process mining research works on MIMIC-III/IV have been provided.

References

1. IEEE standard for extensible event stream (xes) for achieving interoperability in event logs and event streams. IEEE Std 1849-2016 pp. 1–50 (2016)
2. Alharbi, A., Bulpitt, A., Johnson, O.: Improving pattern detection in healthcare process mining using an interval-based event selection method. In: Business Process Management Forum 2017. pp. 88–105. Springer (2017)
3. Andrews, R., van Dun, C.G., Wynn, M.T., Kratsch, W., Röglinger, M., ter Hofstede, A.H.: Quality-informed semi-automated event log generation for process mining. *Decision Support Systems* **132**, 113265 (2020)
4. Calvanese, D., Kalayci, T.E., Montali, M., Santoso, A.: The onprom toolchain for extracting business process logs using ontology-based data access. In: Proc. of the BPM Demo Track and BPM Dissertation Award, co-located with BPM 2017. vol. 1920. CEUR-WS.org (2017)

5. Cremerius, J., Weske, M.: Supporting domain data selection in data-enhanced process models. In: *Wirtschaftsinformatik 2022 Proc.* 3 (2022)
6. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **10**(3), e1346 (2020)
7. van Dongen, B.: BPI challenge 2020 (Mar 2020), https://data.4tu.nl/collections/BPI-Challenge_2020/5065541/1
8. Erdogan, T.G., Tarhan, A.: Systematic mapping of process mining studies in healthcare. *IEEE Access* **6**, 24543–24567 (2018)
9. Gonzalez Lopez de Murillas, E.: Process mining on databases: extracting event data from real-life data sources. Ph.D. thesis, Mathematics and Computer Science (Feb 2019), proefschrift
10. Jackson, S.L., Tong, X., King, R.J., Loustalot, F., Hong, Y., Ritchey, M.D.: National Burden of Heart Failure Events in the United States, 2006 to 2014. *Circ Heart Fail* **11**(12), e004873 (12 2018)
11. Jans, M., Soffer, P.: From relational database to event log: decisions with quality impact. In: *BPM Workshops*. pp. 588–599. Springer (2017)
12. Jans, M., Soffer, P., Jouck, T.: Building a valuable event log for process mining: an experimental exploration of a guided process. *Enterprise Information Systems* **13**(5), 601–630 (2019)
13. Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L.A., Mark, R.: Mimic-iv (2020), <https://physionet.org/content/mimiciv/1.0/>
14. Kurniati, A.P., Hall, G., Hogg, D., Johnson, O.: Process mining in oncology using the mimic-iii dataset. In: *Journal of Physics: Conference Series*. vol. 971, p. 012008. IOP Publishing (2018)
15. Kurniati, A.P., Rojas, E., Hogg, D., Hall, G., Johnson, O.A.: The assessment of data quality issues for process mining in healthcare using medical information mart for intensive care iii, a freely available e-health record database. *Health informatics journal* **25**(4), 1878–1893 (2019)
16. Kusuma, G., Kurniati, A., McInerney, C.D., Hall, M., Gale, C.P., Johnson, O.: Process mining of disease trajectories in mimic-iii: A case study. In: *Process Mining Workshops*. pp. 305–316. Springer (2020)
17. Mannhardt, F.: Sepsis cases - event log (Dec 2016), https://data.4tu.nl/articles/dataset/Sepsis_Cases_-_Event_Log/12707639/1
18. Marazza, F., Bukhsh, F.A., Geerdink, J., Vijlbrief, O., Pathak, S., Keulen, M.v., Seifert, C.: Automatic process comparison for subpopulations: Application in cancer care. *International journal of environmental research and public health* **17**(16), 5707 (2020)
19. McDonagh, T.A., et al.: 2021 ESC Guidelines for the diagnosis and treatment of acute and chronic heart failure. *European Heart Journal* **42**(36), 3599–3726 (08 2021)
20. Munoz-Gama, J., de la Fuente, R.R., Sepúlveda, M.M., Fuentes, R.R.: Conformance checking challenge 2019 (ccc19) (Feb 2019), https://data.4tu.nl/articles/dataset/Conformance_Checking_Challenge_2019_CCC19_/12714932/1
21. Munoz-Gama, J., et al.: Process mining for healthcare: Characteristics and challenges. *Journal of Biomedical Informatics* **127**, 103994 (2022)
22. Remy, S., Pufahl, L., Sachs, J.P., Böttinger, E., Weske, M.: Event log generation in a health system: A case study. In: *International Conference on Business Process Management*. pp. 505–522. Springer (2020)
23. Van Der Aalst, W.: *Data science in action: Process mining*. Springer (2016)