COP-E-CAT: Cleaning and Organization Pipeline for EHR Computational and Analytic Tasks

Aishwarya Mandyam[*]
Department of Computer Science
Princeton University
Princeton, NJ, USA
aishwarya@princeton.edu

Elizabeth C. Yoo[*]
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ, USA
elizabeth.yoo@princeton.edu

Jeff Soules[*]
Center for Computational Mathematics
Flatiron Institute
New York, NY, USA
jsoules@flatironinstitute.org

Krzysztof Laudanski
Department of Anesthesiology and Critical Care
Hospital of the University of Pennsylvania
Leonard Davis Institute for Health Economics
Philadelphia, PA, USA
krzysztof.laudanski@uphs.upenn.edu

Barbara E. Engelhardt
Department of Computer Science
Center for Statistics and Machine Learning
Princeton University
Princeton, NJ, USA
bee@princeton.edu

ABSTRACT

In order to ensure that analyses of complex electronic healthcare record (EHR) data are reproducible and generalizable, it is crucial for researchers to use comparable preprocessing, filtering, and imputation strategies. We introduce **COP-E-CAT**: **C**leaning and **O**rganization **P**ipeline for **E**HR **C**omputational and **A**nalytic **T**asks, an open-source processing and analysis software for MIMIC-IV, a ubiquitous benchmark EHR dataset. COP-E-CAT allows users to select filtering characteristics and preprocess covariates to generate data structures for use in downstream analysis tasks. This user-friendly approach shows promise in facilitating reproducibility and comparability among studies that leverage the MIMIC-IV data, and enhances EHR accessibility to a wider spectrum of researchers than current data processing methods. We demonstrate the versatility of our workflow by describing three use cases: ensemble prediction, reinforcement learning, and dimension reduction. The software is available at: `https://github.com/eyeshoe/cop-e-cat`.

CCS CONCEPTS

• **Applied computing** → **Health informatics**; • **Information systems** → **Data cleaning**; **Extraction, transformation and loading**.

KEYWORDS

health informatics, reinforcement learning, electronic health records

ACM Reference Format:
Aishwarya Mandyam, Elizabeth C. Yoo, Jeff Soules, Krzysztof Laudanski, and Barbara E. Engelhardt. 2021. COP-E-CAT: Cleaning and Organization Pipeline for EHR Computational and Analytic Tasks. In *12th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '21), August 1–4, 2021, Gainesville, FL, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3459930.3469536

1 INTRODUCTION

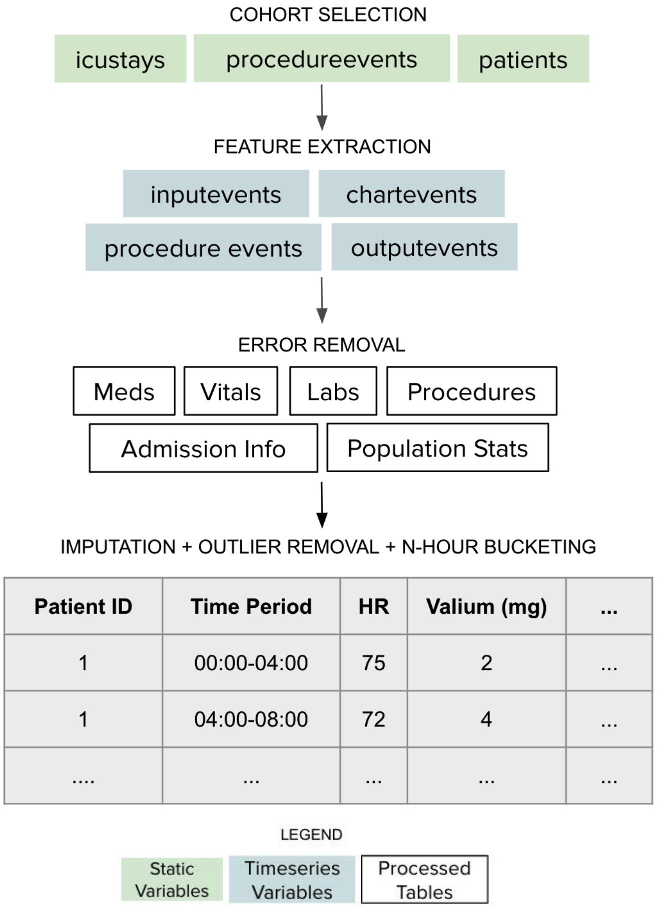
Electronic health record (EHR) datasets enable progress in understanding and improving healthcare in many ways; they can be used to develop decision making algorithms and quantify clinically meaningful patterns in patient trajectories. However, due to issues of patient privacy and interoperability among digital systems, EHR data are traditionally difficult to acquire and process for the research community at large. The Medical Information Mart for Intensive Care (MIMIC) dataset series [11, 14, 8, 9] is the first open-access, freely available, large single-source EHR database that alleviates this difficulty. The latest version of this dataset, MIMIC-IV, contains de-identified EHR data from over 524,000 distinct hospital admissions and over 257,000 patients collected between 2008 and 2019 from the Beth Israel Deaconess Medical Center in Boston, Massachusetts [9]. These EHR data contain information about vital signs, medicine and interventions, lab results, procedures, and metadata about patient demographics and hospital stay information.

The ubiquity of the MIMIC EHR database series is evident – MIMIC-III, the predecessor to MIMIC-IV, has over 2400 total citations since its release in 2016 [4]. Despite the popularity of the series, preprocessing and de-noising MIMIC-IV can be an arduous task. Prior work notes that the lack of preprocessing standards to clean the inherent noise in these data leads to poor reproducibility among downstream analyses [6, 10, 18]. This indicates a need for a standard data processing workflow that cleans, denoises, and removes

[*] Author contributed equally to this research.

BCB '21, August 1–4, 2021, Gainesville, FL, USA

ACM ISBN 978-1-4503-8450-6/21/08.
https://doi.org/10.1145/3459930.3469536

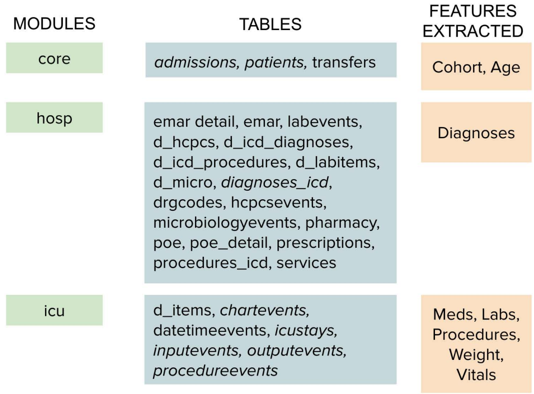


Patient ID	Time Period	HR	Valium (mg)	...
1	00:00-04:00	75	2	...
1	04:00-08:00	72	4	...
....	...	...	...	...

Figure 1: COP-E-CAT organization workflow to build state spaces. COP-E-CAT first selects a cohort, then removes noise present in raw data, imputes missing values, and generates a state space representation for this cohort. Each state captures information about a user-specified set of covariates during one of the user-specified uniform time intervals that partition a patient's trajectory spanning their hospital stay. COP-E-CAT is highly customizable; the imputation, error removal and outlier removal methods can be specified in a json file that is read in when constructing the state spaces.

errors in the data. Additionally, varied imputation and normalization choices at the preprocessing stage may lead to substantially different results across similar data analysis tasks because of preprocessing artifacts [1, 2]. A consistent and easy-to-use workflow ensures that downstream conclusions are robust and reproducible.

Workflows such as MIMIC-EXTRACT [18] and FIDDLE [15] have been developed to facilitate data cleaning and preprocessing for MIMIC-III. However, these workflows are not compatible with MIMIC-IV due to fundamental structural differences in MIMIC-III and MIMIC-IV [9]. The MIMIC-III database stores all hospital events, regardless of medical category, in a single shared schema that includes all tables. MIMIC-IV, however, stores data in three separate modules (realized as separate database schemas) according to the data source and scope: `core` stores patient-tracking data, `hosp` stores data from hospital-wide EHRs, and `icu` stores all data originating specifically from ICU stays (Figure 2). Moreover, MIMIC-IV contains new tables sourced from the electronic Medicine Administration Record (eMAR) system. These eMAR tables record each instance of medication administration and capture an unprecedented degree of granularity. Thus, MIMIC-IV's new structure and its accompanying level of detail require new querying strategies.



MODULES	TABLES	FEATURES EXTRACTED
core	*admissions*, *patients*, transfers	Cohort, Age
hosp	emar detail, emar, labevents, d_hcpcs, d_icd_diagnoses, d_icd_procedures, d_labitems, d_micro, *diagnoses_icd*, drgcodes, hcpcsevents, microbiologyevents, pharmacy, poe, poe_detail, prescriptions, procedures_icd, services	Diagnoses
icu	d_items, *chartevents*, datetimeevents, *icustays*, *inputevents*, *outputevents*, *procedureevents*	Meds, Labs, Procedures, Weight, Vitals

Figure 2: Overview of the MIMIC-IV database. Italicized tables are those that COP-E-CAT uses.

An effective MIMIC-IV organization and analysis pipeline should have the following characteristics. First, the pipeline should clean, smooth, impute, and prepare time-series data for use in downstream analysis tasks. Second, it should enable users to tune the details of these preprocessing steps, such as the imputation method or covariate selection options, and record these choices along with final results to enable reproducibility. Third, the pipeline should be able to operate on raw MIMIC-IV data without any additional time-consuming, pipeline-specific repackaging steps. The user should be able to directly apply this pipeline to their instance of MIMIC-IV.

We introduce COP-E-CAT, a new workflow for MIMIC-IV, with these characteristics in mind. COP-E-CAT is an open-source, highly customizable processing and analysis pipeline for MIMIC-IV data that: (1) removes noise present in raw data, (2) imputes missing values, and (3) generates a state-space representation for a designated patient cohort using specified covariates. The state space representation transforms the raw MIMIC-IV data into regular data structures that can be directly used in analysis tasks. Each row of the state space, known as a state, captures information about a user-specified set of covariates during one of the user-specified uniform time intervals that partition a patient's trajectory spanning the duration of their hospital stay. For example, a state space with a four-hour time interval contains information about every four hour period in every patient's trajectory (Figure 1).

COP-E-CAT standardizes EHR preprocessing for the healthcare research community and serves as a benchmark for reproducibility and comparisons. This pipeline additionally lowers the barrier-to-entry to EHR data analysis by providing cleaned, organized data frames that can be directly used in downstream analysis tasks. We describe the use of this pipeline through three vignettes for common downstream data analytic applications: ensemble prediction to estimate triage levels of ICU patients, predicting electrolyte repletion with reinforcement learning [13], and performing dimension reduction across patient records [16].

We first describe related efforts to democratize EHR data. Then we detail the implementation of COP-E-CAT and discuss how a user can construct a state space representation. We next present three vignettes describing the use of COP-E-CAT. We conclude with a discussion of possible extensions of this work.

2 RELATED WORK

There have been several prior efforts to systematize EHR data analysis for earlier versions of the MIMIC dataset series. Currently, MIMIC-EXTRACT [18] and FIDDLE [15] are two open-source preprocessing pipelines compatible with the MIMIC-III database. MIMIC-EXTRACT processes raw vital sign and laboratory measurements into data structures that can be used in time-series modeling and prediction tasks. Its ability to encode interventions including ventilation, vasopressors, and fluid bolus therapies greatly expands the tractability of downstream analyses. However, MIMIC-EXTRACT buckets patient data into hourly aggregates, a fixed design decision that may result in information loss. COP-E-CAT instead enables the user to specify the time interval for the state spaces, which allows the user to control the granularity of the state spaces. Additionally, MIMIC-EXTRACT uses ICD-9 diagnosis codes to derive cohorts, which is appropriate for MIMIC-III. MIMIC-IV includes both ICD-9 and ICD-10 diagnosis codes, and COP-E-CAT allows a user to filter for a cohort based on both code types.

FIDDLE is an EHR data preprocessing pipeline [15] that readily generalizes across different datasets such as MIMIC-III and eICU Collaborative Research Database [12]. FIDDLE is largely data-driven because it relies on the underlying distributions of the data without consideration of clinical knowledge; however, EHR databases often contain artifacts such as mislabeled drugs and erroneous dosages that need to be addressed case-by-case, often with input from a subject-matter expert. As such, results through FIDDLE data preprocessing may contain artifacts that are not clinically meaningful.

Although MIMIC-EXTRACT and FIDDLE are open-source, both require users to repackage raw data prior to pipeline usage. COP-E-CAT instead enables researchers to use the MIMIC-IV dataset off-the-shelf with limited configuration steps, making MIMIC-IV accessible to a wide range of researchers, including those with less experience in data configuration (Listing 1).

3 METHODS

COP-E-CAT uses embedded SQL queries to generate a set of output tables that aggregate ICU-level information separated by category of measurement. The output tables are then used to generate a state space (Figure 1).

3.1 Cohort Selection

The first step in COP-E-CAT is patient cohort selection. The goal of this step is to select patients that meet a user-specified set of criteria. COP-E-CAT is currently intended to work with ICU-level cohorts, but can be easily reconfigured to work with non-ICU cohorts. MIMIC-IV includes over 69,000 unique hospital admissions in the ICU. By default, COP-E-CAT selects all ICU patients who are at least 18 years old, and were in the ICU between 1 and 8 days. A user can specify additional characteristics including ICD diagnoses, mortality, race, and gender.

3.2 Covariate Selection

Next, COP-E-CAT gathers a user-specified set of covariates for the patients in the cohort. Supported covariates span labs, medicines, procedures, vitals, and static information such as weight. Curating this selection of features and potentially cross-referencing multiple tables for measurements is a labor-intensive procedure. COP-E-CAT provides a mapping of covariates to their corresponding MIMIC-IV `itemids`, as well as pointers to the tables where these measurements are located. A user can additionally filter the cohort based on covariates (e.g., to reduce the cohort to patients that have received electrolyte repletion).

During covariate selection, COP-E-CAT also removes extreme outliers. These outliers include measurements that are labeled `999999`, likely filler values, as well as lab measurements and vitals with biologically infeasible (e.g., negative) values (Figure 3). The result of this covariate selection step is a set of data frames that categorize ICU level information about the cohort (Figure 1 1).

At this point, the data frames contain all of the specified covariates for the selected cohort and omit noisy values that are likely to skew the output of downstream algorithms. COP-E-CAT now generates population-level statistics such as mean and standard deviation for each covariate. These statistics can be used to manually verify that the pipeline works as intended. Next, COP-E-CAT discretizes this information into *state spaces*. Recall that in a state space representation, each state reports all specified covariates during uniform time intervals across the complete patient trajectory for all patients.

3.3 State Spaces

State spaces are a useful data structure for downstream analytic tasks for three reasons. First, they discretize the data into uniform time intervals, each of which can be treated independently or combined with earlier and later intervals from the same patient. Second, they impute values as necessary to smooth the sampling intervals. MIMIC-IV raw data are sampled at irregular intervals across patients, but most downstream analysis tasks require the input to be uniform time slices capturing patient state. COP-E-CAT selectively imputes missing measurements of certain labs and vitals that are likely to stay consistent in adjacent time segments. COP-E-CAT does not impute any information about medications. Third, these state space representations normalize units of measurement to the appropriate standard unit of measurement (e.g., mL, mg, bpm) for a given covariate to ensure comparability across patients and time intervals.

We generate state spaces using Algorithm 1. In the resulting state space, each column corresponds to a distinct covariate, and each row records all specified covariates per patient, per user-specified constant time interval (Figure 4).

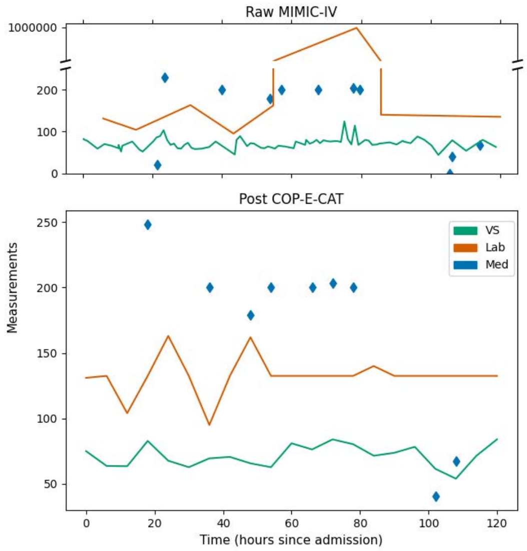
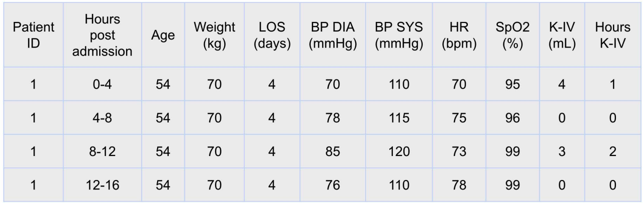


Figure 3: After processing by COP-E-CAT, source data from MIMIC-IV has been structured as regularly spaced state frames. This figure shows a vital sign (diastolic blood pressure, in mmHg), a lab value (serum glucose, in mg/dL), and a medication administration (Dextrose 5% IV) to a composite patient record. Original MIMIC-IV data has been cleaned to remove error values (999999) and outliers. High-frequency samples have been smoothed and missing samples are imputed to ensure one data point per time period.



Patient ID	Hours post admission	Age	Weight (kg)	LOS (days)	BP DIA (mmHg)	BP SYS (mmHg)	HR (bpm)	SpO2 (%)	K-IV (mL)	Hours K-IV
1	0-4	54	70	4	70	110	70	95	4	1
1	4-8	54	70	4	78	115	75	96	0	0
1	8-12	54	70	4	85	120	73	99	3	2
1	12-16	54	70	4	76	110	78	99	0	0

Figure 4: A hypothetical state space in which static values are repeated across rows for a given patient. The units for covariates are either in mg, mL or a standardized unit.

3.4 Usage

The Github repository README contains extensive instructions on how to use COP-E-CAT.

Listing 1: Usage of COP-E-CAT with default settings

```
copecat = CopECat()
copecat.generate_state_spaces()
```

A user can construct state spaces and customize the covariate and cohort selection parameters by changing the default arguments passed to the `CopECat()` class (Listing 1). The user can also customize imputation and outlier removal decisions using the `CopECat()` class.

Algorithm 1: Algorithm to generate a state representation for a given patient p. The resulting state space representation is an $m \times n$ matrix, where m is the number of constant-width intervals required to cover the patient's trajectory and n is the number of covariates.

```
Result: A state space representation STATE_SPACE_P for
        patient P

STATE_FEATURES, TABLES selected features & their sources
STATE_SPACE_P = [] collection of states for the patient
interval user-specified constant time interval
time = time of patient admission

while time < discharge_time_P do
    STATE = []
    for feature in STATE_FEATURES do
        feature_P ← TABLES[feature, P, time]
        if feature_P(is null and should be imputed) then
            feature_P ← imputed value
        end
        feature_P ← NORMALIZE_UNITS(feature_P)
        STATE += feature_P
    end
    time += interval
    STATE_SPACE_P += STATE
end
return STATE_SPACE_P
```

4 VIGNETTES

We now describe three downstream applications that use the COP-E-CAT workflow for MIMIC-IV data exploration and quality control. These applications span a range of analysis tasks: ensemble prediction, reinforcement learning, and dimension reduction. We present these vignettes to demonstrate the versatility of COP-E-CAT. A subset of the code associated with each of these vignettes is available in the COP-E-CAT Github repository.

4.1 Triaging ICU patients

Given a set of ICU patients diagnosed with the same disease, predicting which patients are likely to deteriorate rapidly within the next 24 hours is useful for optimally allocating limited resources. For example, if we have a cohort of patients diagnosed with acute respiratory distress syndrome (ARDS), we want to predict which patients require immediate, severe interventions such as mechanical ventilation. We can use gradient tree boosting to classify patients into levels of urgency. Gradient tree boosting (GSB) is an ensemble method that builds weak learners sequentially by fitting each learner to the residual errors of its predecessor. Boosting has demonstrated strong performance in predicting behavior of temporal features in medical data and is a natural model choice in clinical prediction tasks [7, 5].

We first use COP-E-CAT to filter for patients whose ICD-9 and ICD-10 codes correspond to ARDS. Among the list of parameters

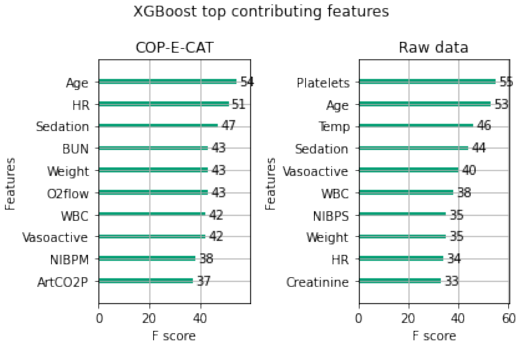


Figure 5: Comparison of top ten contributing features to gradient tree boosting performance, ranked by their F1 score. The plot on the left corresponds to boosting trained on post-COP-E-CAT data, and the plot on the right corresponds to boosting trained on raw MIMIC-IV data.

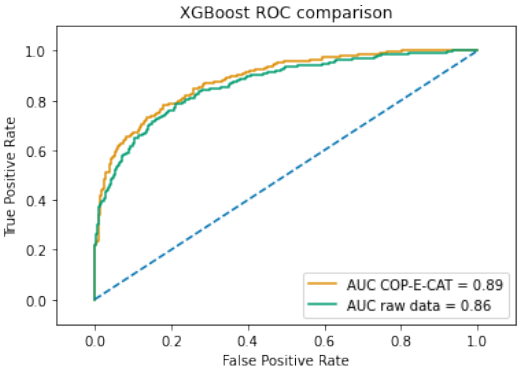


Figure 6: Comparison of gradient tree boosting performance between MIMIC-IV data that underwent preprocessing via COP-E-CAT and raw MIMIC-IV data. Boosting on post-COP-E-CAT MIMIC-IV data results in an ROC AUC of 0.89**, an** 0.03 **absolute improvement from boosting on raw MIMIC-IV data that results in a ROC-AUC of** 0.86**.**

that COP-E-CAT offers, we select 65 features including static demographic information, vitals, labs, drugs, and procedures (Table 1). Temporal data such as vital signs and lab tests for each patient are bucketed into 4-hour intervals. We additionally specify that COP-E-CAT impute missing values with the mean of a patient's measurements observed up to that point. For all measurements within a 4-hour interval, we select the mean as the sample statistic to represent that interval.

The resulting state space representation can be used as input to a boosting model. This model predicts mortality in the 24-hour interval following the time of a patient's last observation and identifies the top contributing features (Figure 5). From fitting the model, we are able to identify 23% of patients that are likely to expire in the next 24 hours with 0.89 receiver operating characteristic curve area under curve (ROC AUC) (Figure 6). We find that COP-E-CAT reduces the time needed to generate a state space representation specific to predicting respiratory decompensation and allows us to focus on model architectures and analysis.

4.2 Predicting Electrolyte Repletion

The overuse of intravenous fluid administration and oral medicine prescription intended to balance electrolyte levels in patients has been linked to complications post-surgery and patient mortality [17]. An algorithm that accurately captures patterns in electrolyte measurements among patients and predicts the optimal dosage and route of electrolyte repletion can prevent side effects of electrolyte imbalance [13]. This algorithm can supplement clinical decision making and reduce costs to hospitals and patients. We model this problem as a Markov decision process (MDP). An MDP organizes this problem into a set of states and potential actions; the transition from a state using a particular action results in a long-term reward. We can use this MDP as an input to a reinforcement learning algorithm like Fitted Q-Iteration [3], which learns a function that estimates expected reward. This allows us to find the sequence of actions that maximizes cumulative reward.

We can use COP-E-CAT to construct the MDP for this problem. In particular, we want a patient cohort that receives electrolyte repletion either through IV or oral administration. We are interested in three electrolytes: potassium, phosphates, and magnesium. In order to find this cohort, we first use the default filtering specifications for ICU patients (greater than 18 years old, and in the hospital for between 1 and 8 days). We then filter the cohort to only include patients who received electrolyte repletion in the covariate selection step. We are interested in separating medications based on their route of usage; this means separating orally administered drugs from IV administered fluids. We can specify these constraints when filtering the medications from `mimic-icu/inputevents.csv`. The resulting state space includes features relevant to electrolyte repletion divided into 6-hour time intervals. These state spaces can be directly used to construct the time-interval state representation and to compute rewards for the MDP. The MDP is used in FQI (Figure 7) to predict the time and route of repletion for the three electrolytes we are interested in. We find that this electrolyte repletion algorithm's predictions match a doctor's actions between 85% and 90% of the time depending on the route of repletion (IV or orally) and the electrolyte being repleted. COP-E-CAT streamlines the data preprocessing step and allows us to focus on designing the MDP and experimenting with different model structures.

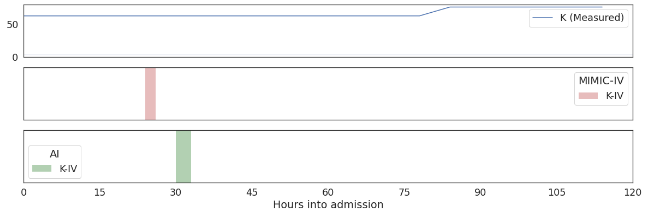


Figure 7: Comparison of reinforcement learning algorithm recommendations for potassium repletion through an IV to actual doctor's actions in the MIMIC-IV dataset for a sample patient. The algorithm recommendations (AI), seen here in green, are very similar to doctor's actions for this sample patient.

4.3 Dimension reduction

Dimension reduction is an essential first step in analysis of high-dimensional data that enhances tractability and reduces the effect of noise on downstream prediction tasks. Student's t-distributed Gaussian process latent variable model (tGPLVM) is a flexible non-parametric model for nonlinear manifold estimation that finds low-dimensional structure in unprocessed count data from high-throughput single-cell RNA-sequencing [16]. By using a robust Student's t-distribution noise model, a weighted sum of non-smooth covariance kernel functions, and a sparse kernel structure, tGPLVM is able to accommodate noisy, unprocessed high-dimensional data. We observe that EHR data contains noisy and sparse features similar to those in single-cell RNA-sequencing data, so we use tGPLVM to perform dimension reduction on MIMIC-IV data. In particular, we are interested in exploring low-dimensional representations of the labs and vital sign trajectories of patients diagnosed with pneumonia.

We use COP-E-CAT to generate a state space representation to use as input for tGPLVM. We select the cohort by filtering for the ICD-9 and ICD-10 codes for *pneumonia*. Because we are particularly interested in vitals and labs that serve as clinical markers of pneumonia, we use the following covariates:

- Vitals: Inspired O2 Fraction, Arterial CO2 Pressure, pH (Venous), Arterial O2 pressure
- Labs: Chloride, Creatinine, Creatinine, Hematocrit, Hemoglobin, Internalized Normalized Ratio, Lactate, Platelets, Arterial O2 pressure, Potassium, Sodium, Urine Output, White Blood Cell count

We impute missing vital signs with the the most recently observed measurements, and missing labs with the per-patient mode of all available measurements. We then bucket these trajectories by selecting a 1-hour `time-delta` interval. When trained on the resulting state-space representation, tGPLVM identifies clusters among the patient trajectories that principal component analysis fails to find due to linearity assumptions.

Dimension reduction algorithms generally require a tabular dataset in which each row is a sample and each column represents a feature. COP-E-CAT enables the use of dimensionality reduction algorithms because the resulting state space representations are clean, normalized, and tabular unlike the raw MIMIC-IV data.

5 CONCLUSION

In summary, COP-E-CAT is the first processing and analysis framework designed specifically for MIMIC-IV, a widely used EHR database. We demonstrate COP-E-CAT's ability to derive state spaces representations, useful data structures that can be subsequently used in downstream tasks such as gradient tree boosting for decompensation prediction, reinforcement learning for clinical decision making, and a Gaussian process latent variable model for dimension reduction. A natural extension to this work is incorporating more covariates and encoding classes of interventions such as medication and procedures in clinically meaningful ways. Additional work includes expanding COP-E-CAT to select cohorts and covariates from other MIMIC-IV modules such as `hosp` and `derived`. The concepts used to preprocess MIMIC-IV here can also be extended to other EHR databases; we anticipate that the majority of work when expanding to other databases lies in identifying how covariates are encoded. Since we introduce COP-E-CAT as open source software, we anticipate that users will continue to expand the current capabilities and downstream applications that this pipeline enables.

6 ACKNOWLEDGEMENTS

We would like to thank Niranjani Prasad for initial versions of the code and for useful discussions. The authors gratefully acknowledge the Covid-19 Fast Grants funding of this work.

REFERENCES

[1] Andrew L. Beam, Arjun K. Manrai, and Marzyeh Ghassemi. 2020. Challenges to the Reproducibility of Machine Learning Models in Health Care. *JAMA*, 323, 4, (January 2020), 305–306. ISSN: 0098-7484. DOI: 10.1001/jama.2019.20866. eprint: https://jamanetwork.com/journals/jama/articlepdf/2758612/jama_beam_2020_vp_190172.pdf. https://doi.org/10.1001/jama.2019.20866.

[2] Spiros Denaxas, Kenan Direk, Arturo Gonzalez-Izquierdo, Maria Pikoula, Aylin Cakiroglu, Jason Moore, Harry Hemingway, and Liam Smeeth. 2017. Methods for enhancing the reproducibility of biomedical research findings using electronic health records. *BioData Mining*, 10, 1, (December 2017), 31. ISSN: 1756-0381. DOI: 10.1186/s13040-017-0151-7.

[3] Damien Ernst, Pierre Geurts, and Louis Wehenkel. 2005. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.*, 6, (December 2005), 503–556. ISSN: 1532-4435.

[4] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101, 23, e215–e220.

[5] Stephanie L Hyland, Martin Faltys, Matthias Hüser, Xinrui Lyu, Thomas Gumbsch, Cristóbal Esteban, Christian Bock, Max Horn, Michael Moor, Bastian Rieck, et al. 2019. Machine learning for early prediction of circulatory failure in the intensive care unit. *arXiv preprint arXiv:1904.07990*.

[6] Alistair E. W. Johnson, Tom J. Pollard, and Roger G. Mark. 2017. Reproducibility in critical care: a mortality prediction case study. en. In *Machine Learning for Healthcare Conference*.

PMLR, (November 2017), 361–376. Retrieved 03/29/2021 from http://proceedings.mlr.press/v68/johnson17a.html.

[7] Alistair E.W. Johnson and Roger G. Mark. 2018. Real-time mortality prediction in the Intensive Care Unit. *AMIA Annual Symposium Proceedings*, 2017, (April 2018), 994–1003. ISSN: 1942-597X. Retrieved 01/18/2021 from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5977709/.

[8] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3, 1, 160035. DOI: 10.1038/sdata.2016.35.

[9] Johnson, Alistair, Bulgarelli, Lucas, Pollard, Tom, Horng, Steven, Celi, Leo Anthony, and Mark, Roger. [n. d.] MIMIC-IV. type: dataset. (). DOI: 10.13026/S6N6-XD98. Retrieved 03/19/2021 from https://physionet.org/content/mimiciv/1.0/.

[10] Matthew B. A. McDermott, Shirly Wang, Nikki Marinsek, Rajesh Ranganath, Luca Foschini, and Marzyeh Ghassemi. 2021. Reproducibility in machine learning for health research: Still a ways to go. en. *Science Translational Medicine*, 13, 586, (March 2021), eabb1655. ISSN: 1946-6234, 1946-6242. DOI: 10.1126/scitranslmed.abb1655. Retrieved 03/26/2021 from https://stm.sciencemag.org/lookup/doi/10.1126/scitranslmed.abb1655.

[11] G.B. Moody and R.G. Mark. 1996. A database to support development and evaluation of intelligent intensive care monitoring. *Computers in Cardiology 1996*, 657–660. ISSN: 0276-6547. DOI: 10.1109/cic.1996.542622.

[12] Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. 2018. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. en. *Scientific Data*, 5, 1, (September 2018), 180178. ISSN: 2052-4463. DOI: 10.1038/sdata.2018.178. Retrieved 03/31/2021 from https://www.nature.com/articles/sdata2018178.

[13] Niranjani Prasad. 2020. Methods for reinforcement learning in clinical decision support. https://dataspace.princeton.edu/handle/88435/dsp018s45qc694.

[14] M Saeed, C Lieu, G Raber, RG Mark, and Mohammed Saeed. 2002. MIMIC II: A Massive Temporal ICU Patient Database to Support Research in Intelligent Patient Monitoring. *Computers in Cardiology*, 641–644. DOI: 10.1109/cic.2002.1166854.

[15] Shengpu Tang, Parmida Davarmanesh, Song Yanmeng, Danai Koutra, Michael W Sjoding, and Jenna Wiens. 2020. Democratizing ehr analyses with fiddle: a flexible data-driven preprocessing pipeline for structured clinical data. *Journal of the American Medical Informatics Association*, 27, 12, 1921–1934. ISSN: 1527-974X. DOI: 10.1093/jamia/ocaa139.

[16] Archit Verma and Barbara E. Engelhardt. 2018. A robust nonlinear low-dimensional manifold for single cell rna-seq data. *bioRxiv*. DOI: 10.1101/443044. eprint: https://www.biorxiv.org/content/early/2018/10/14/443044.full.pdf. https://www.biorxiv.org/content/early/2018/10/14/443044.

[17] Anders Winther Voldby and Birgitte Brandstrup. 2016. Fluid therapy in the perioperative setting—a clinical review. *Journal of Intensive Care*, 4, 1, (April 2016), 27. ISSN: 2052-0492. DOI: 10.1186/s40560-016-0154-3. Retrieved 03/28/2021 from https://doi.org/10.1186/s40560-016-0154-3.

[18] Shirly Wang, Matthew B. A. McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C. Hughes, and Tristan Naumann. 2020. Mimic-extract: a data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM Conference on Health, Inference, and Learning*. ACM, (April 2020), 222–235. ISBN: 9781450370462. DOI: 10.1145/3368555.3384469. https://dl.acm.org/doi/10.1145/3368555.3384469.

A TRIAGING ICU PATIENTS

Table 1: The clinical covariates modeled by gradient tree boosting.

Static	Age, Gender, Weight, Ethnicity, Admit unit, Admit type
Vitals	Heart rate, Respiratory rate, Temperature, Arterial pH, Non-invasive blood pressure (systolic, diastolic, mean), FiO2, O_2 saturation pulseoxymetry (SpO_2), Tidal volume, Minute volume, Peak Insp. Pressure, Arterial pressure (O2, CO2), Inspiratory ratio, Expiratory ratio, Insp time, Apnea interval, O2 flow, PEEP set, RR set, Cuff pressure, Mean airway pressure, Plateau pressure, PSV level
Labs	Creatinine, Glucose, BUN, WBC, Hemoglobin, Sodium, Potassium, DD, Fibrinogen, Troponin, CRP, Albumin, PT, PTT
Drugs	Anticoagulants, Beta-blockers, Calcium channel blockers, Dextrose, Fluids, Insulin, Sedation Loop Diuretics, Nutrition (PN, PO), Packed RBC, Paralytics, Vasoactive drugs
Procedures	Invasive MV, Non-invasive MV, CRRT, Peritoneal dialysis, Blood culture, EKG, X-ray

B ELECTROLYTE REPLETION

Table 2: Features used to predict electrolyte repletion and their associated COP-E-CAT category and MIMIC-IV table they are found in.

Category	MIMIC-IV Table	Feature
Metadata	mimic-core/patients	anchor_age gender expired
	mimic-icu/procedureevents	patient_weight
	mimic-icu/icustays	los (length of stay)
Vitals	mimic-icu/chartevents	heart rate respiratory rate oxygen saturation temperature systolic bp diastolic bp
Meds	mimic-icu/inputevents	Potassium-IV Potassium Calcium-IV Calcium Phosphate-IV Phosphate Vasopressors BetaBlockers CaBlockers Loop Diuretics Insulin Dextrose Parenteral-Nutrition Other Nutrition
Labs	mimic-icu/chartevents	Potassium Calcium Magnesium Phosphate Sodium Hemoglobin Chloride Anion Gap Creatinine CPK LDH ALT WBC Glucose
Comorb	mimic-hosp/diagnoses_icd	CAD AFib CHF ESRD CKD Paralysis Parathyroid Rhabdomyolysis Sarcoidosis Sepsis
Procedures + Other	mimic-icu/procedureevents	Dialysis RBC Transfusion
	mimic-icu/outputevents	Urine Output