Predicting Unplanned Hospital Readmissions Using Outcome-Oriented Predictive Process Mining

Abdulaziz Aljebreen[1](✉), Allan Pang[1,2], Marc de Kamps[1], and Owen Johnson[1]

[1] University of Leeds, Leeds, UK
{ml17asa,ugm5a2p,m.dekamps,o.a.johnson}@leeds.ac.uk
[2] Royal Centre for Defence Medicine, Birmingham, UK

Abstract. Many hospitals in the world are under pressure to improve their efficiency and effectiveness so that they can achieve better health outcomes with limited resources. One common measure of performance is the rate of unplanned hospital readmissions (UHRs) within 30-days. Emergency readmissions for the same disease can be assumed to indicate inappropriate discharge or poor planning, are costly, increase patients' mortality risks and put additional pressure on bed capacity. Data Mining (DM) techniques have been used to predict UHRs based on clinical and demographic features, but these ignore the process perspective. Predictive Process Monitoring (PPM) is a process mining technique using completed traces to make predictions for in progress cases with machine learning (ML) algorithms. The Outcome-Oriented PPM (OOPPM) is a sub-technique of PPM focusing on predicting categorical outcomes of process. Adaptation of OOPPM in healthcare settings has been limited to date. Here, we illustrate how to implement OOPPM in a healthcare context through an application of an OOPPM pipeline to hospital admissions using the open access MIMIC-IV dataset. Clinical, demographical and process features were used to build an extended event log, which was then employed for UHRs prediction. Results show prediction using OOPPM techniques outperformed traditional DM techniques. OOPPM tests using tree-based ML algorithms achieved better results compared to OOPPM tests using other ML algorithms. Our results suggest OOPPM can make a significant contribution to better understanding of hospital performance.

Keywords: Predictive Process Monitoring · Unplanned Hospital Readmission · 30-days Hospital Readmissions · Discharge Decisions · Process Mining · Healthcare · Electronic Health Record · EHR · MIMIC-IV

1 Introduction

Hospitals can be seen as highly complex health systems tasked with the delivery of high-quality healthcare services to their users using standardised processes, procedures, technologies, and medicines. Many hospitals in the world are under pressure to improve

A. Delgado and T. Slaats (Eds.): ICPM 2024 Workshops, LNBIP 533, pp. 421–433, 2025.
https://doi.org/10.1007/978-3-031-82225-4_31

their efficiency and effectiveness so that they can achieve better health outcomes with limited resources. One common measure of the performance of hospitals is their rate of unplanned hospital readmissions (UHRs) within a defined period, with 30 days being commonly adopted [1]. UHRs happen when a patient returns to hospital through emergency services for the same medical condition causing disruption to normal operations and distress to patients. UHRs indicate inappropriate discharge or poor planning. UHRs leads to more premature discharge decisions due to pressure on hospital, creating more UHRs [2]. A vicious cycle that merits further research.

UHRs within 30 days represent 20% of total UHRs was accounting for $17.4 billion of additional hospital payments in USA in 2021 and add pressures onto hospital systems by increasing demand on hospital services with poor mortality outcomes [1, 2]. While the causes of UHRs are multifactorial, we hypothesise that decisions during the hospital admissions may contribute to UHRs. UHRs have been shown to be potentially avoidable if proper healthcare service would have been provided [1]. If such UHRs could be predicted and highlighted to the medical team, these could influence decision making to prevent future UHRs. Previous attempts to predict UHRs have included the use of statistical analysis [3], machine learning (ML) [4], deep learning (DL) [5] and Natural Language Processing (NLP) [6]. Our literature search found that none of the existing work included process data (e.g., event sequences) for the prediction.

Process Mining (PM) main techniques such process discovery and conformance checking has been adopted in healthcare domain to analyse compliance with guidelines. Healthcare process represented through a group of events, include its activities, time, and objects. Following a process view of a patient's clinical pathway through hospital we can define a typical pathway from admission to discharge as a *case*. For example, Fig. 1 below illustrates a simplified clinical pathway with possible UHR as an outcome, indicating multiple opportunities for better prediction before the discharge event.

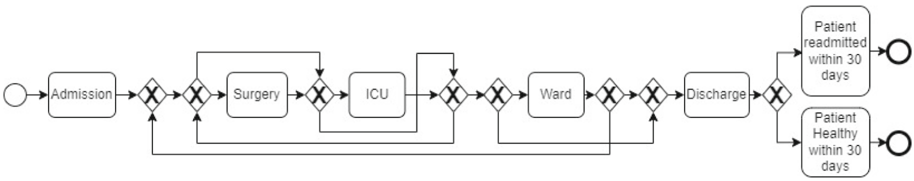

Fig. 1. Hospital admission process shows UHRs probability as an outcome.

Predictive (business) process monitoring (PPM) technique offers prediction ability to ongoing cases at different points throughout the process [7]. Predictions could be the outcome of a case; next activity/activities; execution time or expected load on a resource. Prediction in PM for healthcare (PM4H) has been considered a challenge by the PM4H community under *Challenge 2: discover beyond discovery* [8]. One PPM technique that focuses on predicting categorical case outcomes is called Outcome-Oriented PPM (OOPPM). Here, we are interested in the prediction of UHRs as an outcome but recognise that the approach should be generalisable to other outcomes and activities. OOPPM has been used in healthcare to support clinical decisions such as predicting of discharge location for patients [9]. OOPPM has also been used to predict unplanned Intensive

Care Unit (ICU) readmissions where it surpasses the baseline of Data Mining (DM) techniques [10]. From our review of the literature, we believe our work is the first work that utilises OOPPM technique to predict UHRs.

The rest of this paper is structured as follows: Sect. 2 gives background on the concepts and related work on OOPPM, Sect. 3 illustrates the OOPPM framework, Sect. 4 describes our implementation of using OOPPM for UHRs prediction and present the results, and Sect. 5 discuss and conclude our findings and future work.

2 Background

2.1 MIMIC-IV

Our data is drawn from Medical Information Mart for Intensive Care IV (MIMIC-IV), a widely used, open-access Electronic Healthcare Record (EHR) database [11]. We have selected MIMIC-IV to apply OOPPM in response to a challenge identified by PM4H community *Challenge 4: Deal with Reality* specifying the importance of using real life data [8]. MIMIC-IV includes anonymised, detailed data for patients who were admitted to an ICU or Emergency Department (ED) at Beth Israel Deaconess Medical Centre in Boston, USA. It contains information on more than 380,000 patients receiving care between 2008 to 2019. The MIMIC-IV database is rich with event data, making it an appropriate choice for process mining [8].

2.2 UHRs Prediction

UHR prediction is a complex task as it requires multiple data features to make the prediction. Prediction of UHRs can be implemented for a specific cohort (e.g., heart failure patients) or more broadly for all patients. In [3], statistical analysis using logistic regression was applied to identify variables associated with UHRs on older adult patients in Sweden. Multiple ML algorithms were used by [4] to predict UHRs in MIMIC-III (an earlier, smaller version of the MIMIC-IV dataset) using demographics, aggregated vital signs and diagnoses. DL algorithms such as Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs) were used to predict UHRs for pneumonia patients in Taiwan [5]. Clinical notes from MIMIC-III were able to predict UHRs successfully using NLP techniques [6]. None of the previous work considered process control-flow and the events order as a feature to be used for prediction.

2.3 Prediction in PM for Healthcare

Several methodologies are available for process mining projects. The Process Mining Project Management PM^2 [12] is a generic method contains six stages, where the use of ML is included in step 4 named Mining & Analysis. An extension of PM2 is the *ClearPath* method which was developed for clinical pathway discovery and incorporates process simulation approach [13]. The L^* life-cycle method included the prediction as part of last stage Operational Support [14]. Current PM methodologies do not provide explicit support for the detailed implementation of predictions within process mining context. However, it should be possible to incorporate PPM as a dedicated method to make predictions within these existing process mining methodologies.

2.4 Outcome-Oriented Predictive Process Monitoring

OOPPM, like other PM approaches, uses *event logs* where the *case_id*, *activity* and *timestamp* must be present. However, this information is not enough for OOPPM to make predictions and more data is needed to create *extended event logs*. We can augment event log with *event attributes* (dynamic) or *case attributes* (static). These attributes can be categorical (e.g. a patient's ethnicity), numerical (e.g. patient age) or textual (e.g., clinical note). In healthcare, data types such as age, are a case attribute since age will not vary as a result of the execution of activities during a hospital admission while data types such as Blood Pressure (BP) is a value which will change during patient hospitalisation due to multiple BP readings.

The collection of sequenced events produced by a case is called *trace*, where a trace can contain all or part of a case events. Each possibility of sequenced events represents a complete or part of a trace is called *prefix*. OOPPM uses prefixes for prediction, since the ongoing cases (e.g., incomplete traces) are employed for prediction. To make predictions, trace prefixes should be encoded into a feature vector so that it can be labelled for use in classifiers (e.g., a decision tree) [7].

The use of OOPPM in healthcare to predict process outcomes has been described in the literature. The closest to our work is the prediction of the unplanned ICU readmissions likelihood before discharging patient from ICU using control-flow with clinical metrics like laboratory tests [10]. Jonas et al. employed OOPPM using demographic, lab test values and stay information to predict where a heart failure patients should be discharged to, since the discharge location is strongly associated with UHRs [9]. OOPPM and Time-Oriented Predictive Process Monitoring (TOPPM) was used to predict next activity and its timestamp prediction for ED patients after surgery in Norway [15]. It is to be noted not all the above works followed the framework of OOPPM as described Sect. 3.

3 Methodology

In our work, we used OOPPM to predict UHRs within 30-days. We have selected the OOPPM framework suggested by [7] after they conducted a systematic literature review on OOPPM, where two phases are implemented: offline phase used for learning (see Fig. 2) and online phase for testing and application (see Fig. 3).

3.1 Extracting and Filtering Prefixes

In OOPPM, where prefix logs are extracted from event logs, classifiers use prefix logs to make predictions. This is since OOPPM when applied in the online phase, it will consume incomplete traces in order to make predictions for them as early as possible. So, the use of prefix log will provide us with training data that will match the data we want to test. However, considering all possible prefixes will raise challenges by increasing classifiers learning time and bias toward cases with longer traces as they will produce more prefixes.

There are several methods to filter prefixes, one of them is by limiting the length of the prefix to a certain number of events only [7]. Instead of fixing one prefix length, gaps can be identified to have prefixes with different lengths using a base number (e.g., 1) then add gaps accordingly (e.g., for gap = 3, we will have 1, 4, 7, 10 prefix length). Another method is to define execution time of prefix so only events in prefixes within the execution time will be considered [10].

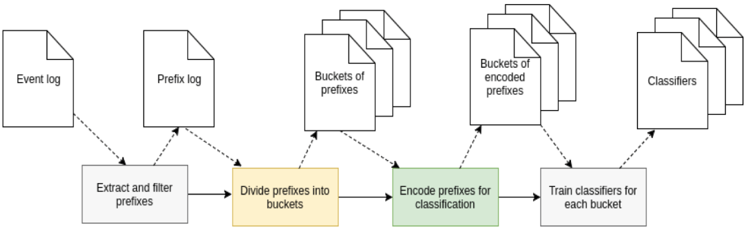

Fig. 2. OOPPM offline phase. [7]

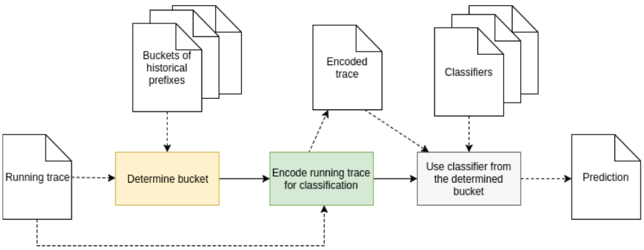

Fig. 3. OOPPM online phase. [7]

3.2 Divide Prefixes into Buckets

At this stage, the prefix log needs to be divided into *buckets*, where each bucket will have a dedicated classifier. During the online phase, cases will be assigned to a similar bucket based on its prefix to make the prediction. There are several bucketing approaches used in OOPPM. The first approach is "single bucket" or "no bucket" where all prefixes are kept in one bucket, leading to having one classifier only [10]. The second approach is to use "process states" or "decision points" available in process model based on event log and train a classifier for each state [16]. The prediction for ongoing cases will be based on the state of process they are in regardless of the followed path. The third approach is to cluster similar encoded prefixes using ML clustering algorithms (e.g., DBScan), which might ignore process structure, and then build classifier for each cluster [16]. The fourth method is to bucket the prefixes based on length, and classifiers are trained accordingly [17]. The last approach is to utilize domain knowledge by manually establishing rules for bucketing prefixes (e.g., execution stages) through consultation with domain experts [7].

3.3 Encode Prefixes for Classification

The input for classifiers should be in fixed size vectors representation, so we need to encode all bucketed prefixes' traces. This raises complications since moving forward in a case execution will add more information, but the number of features in the prefix should not increase. The solution is by applying sequence encoding which is a combination of trace abstraction and feature extraction methods [7]. At this stage, a compromise between generalisation of method (i.e., applicability for all prefixes) and information lost is to be considered. We will discuss sequence encoding methods application on numerical and categorical attributes of case and events, as unstructured textual attributes are outside the scope of our work.

For case attributes, we use the sequence encoding method named *static,* by directly adding them to feature vector without any modifications [7]. If the case attribute is categorical, then we use a baseline approach known as "one hot encoding" to convert the categorical feature into multiple binary vectors based on number of distinct categorical values. This method is to be used on conjunction with other sequence encoding methods, since these methods are concerned with event attributes.

First event attributes sequence encoding method named *Last State*, where both case and event attributes of the recent state is included in one feature vector [16]. Last event numerical attributes will be added as it is, hot encoding will be applied on this event categorial attributes and control-flow, while remining attributes for other events will be zeros. The last state method enables us to use different lengths of prefix traces buckets.

The second event sequence encoding method is *aggregation.* In this method, we aggregate all events from the start of the case in one feature vector regardless of events order, so we avoid losing of information from previous states while maintaining fixed size feature vector. Aggregation of control-flow can be achieved by counting executed activities, or to indicate whether a certain activity has been executed or not [16]. Numerical event attributes could be aggregated using statistics functions (e.g., average, or max), while categorical event attributes are hot encoded then their frequency will be accumulated. The aggregation method presents a way to preserve all the trace data but with the price of losing control-flow relationships and patterns. The aggregation method can also be applied to traces with different lengths.

Index is the third event sequence encoding method proposed by [17] to overcome the partial information loss of events order in aggregation method. This is achieved by creating a single feature in the feature vector for each event attribute executed in the trace. Encoding of control-flow and categorical event attributes is done through hot encoding and numerical event attributes will be included as it is. The issue with this method is the feature vector length depends on number of executed events, putting restriction on its use with heterogenous buckets with different traces length. With comparison to the previous two methods, the index method will create large dimensional feature vector when there are long traces leading to classifiers training complications.

3.4 Train Classifiers for Each Bucket

The problem in classification in OOPPM can be related to the problem of early sequence classification in ML literature [7]. In addition to the *accuracy*, *precision*, *recall*, and *f-1* metrics used to evaluate ML classification models, *earliness* and *computation time* are an importation evaluation metrics for OOPPM classification model, as it is designed to work in an online mode with ongoing cases. ML algorithms including logistic regression (LR), Support Vector Machines (SVM), decision tree (DT), random forest (RF), and Gradient Boosted Machines (GBM) can used in OOPPM in addition to DL algorithms like neural networks (NN) [7, 10].

4 Predicting UHRs Using OOPPM

In this section, we will discuss the implementation for our work for each stage of OOPPM. Our experiments were run on Google Colaboratory using Python 3.4.

4.1 Creating Extended Event Log

In our work, we utilised data available in MIMIC-IV dataset to create an extended event log for hospital admissions with labels. OOPPM does not consider the creating of event log as a main step. However, creating event log and extending it with useful features to be used for prediction is a crucial stage which requires more attention.

Data Filtering. We started with reading the *admissions* table since it contains basic information about hospital admissions and their unique id (*hadm_id*). We grouped admission types from nine different types into *elective* and *emergency*. There are four possible outcomes of interest – death in hospital, no further hospital admissions, emergency readmission within 30 days and subsequent readmissions that were planned or unplanned but after 30 days. For this case study we focussed on outcome concerned with readmissions. To identify patients who have been readmitted within 30-days, we sorted the table based on patient id *subject_id* and hadm_id, added a new column to record next visit admission time, subtracted next admission day from current visit discharge day, checked if the next visit was within 30 days and of emergency type, then label the patients *readmission* status accordingly. We got 350,579 admissions followed by UHR and 80,652 admissions who were not followed by UHR. We filtered out patients who had only single admission (= 101,198) and admissions were a patient died in hospital (= 8,772). This has left us with total admissions of 325,119, where 244,607 admissions (75%) were not followed by UHR and 80512 followed by UHR (25%).

Extracting Data Features. After identifying the study cohort, we started working on preparing features to be used for prediction. These features were divided into demographic, clinical and process related features. For demographic, we selected age, gender, insurance, ethnicity, and marital status. Patients age values are distributed from 18 to 91, where patients older than 91 are anonymised as 91 by MIMIC-IV [11]. To enhance the prediction results, we grouped patients into eight groups represent a ten-year age band starting from 18, 28 etc. with last group being patients who are 88 and over. MIMIC-IV has two genders available (male and female), and three types of insurance known as: Medicaid (public), Medicare (public) and Other (private, military, cash payment, …). MIMIC-IV holds 31 different ethnicity types for patients, which we grouped into eight main groups based on ethnicity name. For marital status, there are five values: single, married, widowed, divorced and unknown.

MIMIC-IV is rich with clinical data, so the selection of clinical features was built on previous studies and clinical knowledge where such features were important for UHR prediction. The patient history was considered by calculating total number of previous UHRs and add this information for each hospital admission. Patients who were admitted through emergency department were flagged. The available patient body mass index (BMI) values were extracted and categorise into four categories: underweight, normal weight, overweight, and obese. Lab tests are important indication for patient health, so we have calculated the number of abnormal lab tests of patient per admission. It is important to identify whether a patient is having chronic diseases, so we have used chronic diseases codes developed by AHRQ to calculate how many chronic diseases the patient have. Number of Medications given to patients during their hospitalisation was included as it is an important representation of treatment provided.

Patients' diagnosis in MIMIC-IV are coded using two versions of the International Classification of Diseases (ICD) which are (ICD-9) and (ICD-10). MIMIC-IV covers patient data from 2008 to 2019 and initially used ICD-9 but switched to ICD-10 when it was implemented in the hospital systems. To overcome this challenge, we used the 18 categories of diseases defined in ICD-9 (e.g., respiratory) and for each patient calculated the number of diseases per category diagnosed within each admission. The same task was done with patients having their diagnoses registered in ICD-10, considering the changes in diagnoses codes and re-arrangement of categories. This approach allowed us to work with 18 diagnosis categories without adding the complexity of encoding the ICD codes for patient into thousands of dimensions and helped to improve the prediction accuracy and reduce the computation significantly.

Data features to be used for OOPPM should include control-flow information as this is the driving concept of OOPPM. However, other process related features can be considered as well to maximise classifiers learning and ensure better use of process-oriented data. We have looked at several process aspects of hospital admissions in terms of time, process context and number of events. For time, we calculated the length of stay (LOS) (i.e., process completion time), grouped the LOS into 4 quartiles to reduce the outliers' effect, and calculated time spent in ICUs during the admission. We also investigated process context from geographical view, where admission and discharge locations were considered. There are many events registered for patients in MIMIC-IV and we selected data on how many times a patient was admitted to ICU and number of surgeries per admission. We end up with 18 data features to be used in classifiers.

Extracting Events. Before building the extended event log to be used by OOPPM, we must also identify suitable events for our purpose. MIMIC-IV has multiple different events, making it challenging to choose appropriate events without complicating the following stages. The events selection for OOPPM should be different from other process mining tasks like process discovery, since including less-informative events will add more dimensions which will complicate the classification task. On the other hand, we should include as much information as possible about different events to ensure that processes are considered in prediction. For this case study we considered events available in the *transfers* table which shows when and where patients were admitted, transferred to within hospital and discharge. The transfer events enable us to look at the process through the eyes of a patient's experience, reflecting one of the key challenges for PM4H [8]. Other events were considered in the prediction as high-level data features (e.g., surgeries total) but not in the same detail as complete events.

The transfer events in MIMIC-IV are categorised into 77 different events (38 different events for admitted to a clinical unit, 38 different events for transferred to a clinical unit and 1 discharge event). We removed events with activities named "admitted into Unknown" or "transferred to Unknown". We grouped all admission events into one event "Admission" except for admission to ICU or surgery, all admission and transferred to ICU under one event "ICU", all admission and transferred to a surgery unit to be "Surgery", all the remaining "transferred to" became "Ward" and we renamed the discharge event. We end up with an extended event log contains 942,368 events categorised into 2423 trace variants. For result comparison purpose, we kept another copy of the event log with original events without grouping to be tested.

4.2 Implementation of OOPPM

Extracting and Filtering Prefixes. Healthcare processes are known for their heterogeneity [8], and with the high number of cases we used for training in our case study, it was essential to make the prefix logs smaller. We have considered only transfer events and aggregated them to help reduce prefix log size, removed cases with data quality issues (e.g., started with discharge event) and fixed decision point (i.e., prediction place) to be before the discharge time. However, with the 2423 trace variants we had, it was a challenge to apply any method to filter prefixes, so we chose to use all traces without filtering, taking into consideration the effect on the classifiers. Complete prefixes were used during the offline phase to train the classifiers, while in online phase we removed the discharged event from prefixes log.

Divide Prefixes into Buckets. In our work, we experimented using single cluster approach. We have also implemented clustering algorithm (K-Mean) with 3 clusters to create prefixes buckets, after testing best performing clustering algorithms and clusters number. Since our prediction place cannot be fixed in a specific state during process (e.g., patient get discharged from ICU without transfer to ward), the process state clustering method is not applicable. With the high number of heterogeneous prefixes we had, application of prefix length was excluded. Although with domain expert involvement in our work, we did not see the need to bucket prefixes based on domain knowledge in this case study.

Encode Prefixes for Classification. We used static encoding for all case attributes, while for event attributes we chose the aggregation approach (i.e., counting of grouped and ungrouped dataset of activities) as previous work suggests it can give better results when compared to other prefix encoding approaches [7].

Train Classifiers for Each Bucket. We have chosen to build a baseline classifier using LR algorithm. The tree-based classifiers are mostly used in OOPPM literature [7], so we tested DT, RF, and GBM. To complement our work and compare classifiers results, NN was included in our experiments. We have not applied hyperparameter tuning in our work as our goal is to test the usefulness of OOPPM with healthcare data. In addition to the common evaluation metrics used for classifiers, we included the area under the receiving operating characteristic curve (AUROC) as it will not be affected by our imbalanced data where 75% of the labels are 0 and 25% are 1. We did not include earliness in our evaluation since we have a fixed decision point where the prediction is required. Computing time was calculated for training and testing of the classifiers.

Execution of Offline and Online Phases. To enable us to simulate the implementation of offline and online phases, we split data into two chunks: first chunk contains 67% of data used for training classifiers (offline phase), and second chunk contains 33% of data used for testing and evaluating the classifiers (online phase). Since MIMIC-IV is anonymised on a temporal level, the splitting could not be executed on chronological order which would be ideal in this step. We have used testing data for the online phase where it was bucketed, encoded then classified.

4.3 Results

We have summarized our results in Table. 1 by mentioning the classifier name and applied approached followed: use of OOPPM, use of clustering and when the aggregated events were used. It was followed by classifiers evaluation metrics then training and predicting time in seconds. It can be noticed applying OOPPM enhanced accuracy in all cases comparing to DM approach. Clustering step did not show effect on the results, while the use of aggregated events has slightly affected results positively or negatively comparing to original events without aggregation. RF models has achieved the highest accuracy and precision, NN models scored best AUROC and F1 values, and DT gained best Recall. DT models were the fastest to learn and predict followed by GBM and RF, then NN and LR. The work done by [4] to predict 30-days UHRs using several ML models using DM approach in MIMIC-III where the RF model was the best achieving AUROC 0.66 and accuracy of 0.65 which is lower than our results in term of accuracy for all classifiers and in AUROC except for our baseline LR classifier.

5 Discussion and Conclusion

Our implementation of OOPPM was challenging due to the nature of healthcare processes and our aim to predict UHRs on hospital level for all patients instead of choosing a cohort of patients with specific disease. We had many prefixes in testing data with only one event, which has limited the learning from control-flow chances, suggesting more work on prefix filtering is needed. A limitation in our work was to ensure the patient is readmitted for the same disease, since MIMIC-IV is not providing this information clearly [11]. Even if the clinical decision was to discharge the patients with high probability of UHR instead of keeping them in hospital, predicting UHRs could help clinicians in reviewing delivered treatment, highlight potential corrections, and indicate more care is need for a patient after their discharge. It was noticed the patient history (i.e., number of previous admissions) and diseases diagnosed within the admission was the most informative features for the classifiers.

Table 1. Summary of classifiers results.

Classifier	OOPPM	Clustering	Events Aggregated	Accuracy	AUROC	Precision	Recall	F1	Learning Time	Prediction Time
Logistic Regression	No	–	–	0.76	0.55	0.69	0.54	0.53	103.68	0.13
	Yes	No	No	0.77	0.55	0.69	0.55	0.54	128.22	0.08
	Yes	Yes	No	0.77	0.55	0.69	0.55	0.54	101.24	0.05
	Yes	No	Yes	0.77	0.56	0.70	0.56	0.55	228.4	0.14
	Yes	Yes	Yes	0.77	0.56	0.70	0.56	0.55	296.02	0.08
Decision Trees	No	–	–	0.75	0.67	0.67	0.67	0.67	3.59	0.04
	Yes	No	No	0.77	0.69	0.70	0.70	0.70	5.91	0.05
	Yes	Yes	No	0.77	0.69	0.70	0.70	0.70	13.76	0.23
	Yes	No	Yes	0.77	0.68	0.68	0.68	0.68	5.5	0.04
	Yes	Yes	Yes	0.77	0.68	0.68	0.68	0.68	8.03	0.11
Random Forest	No	–	–	0.79	0.66	0.73	0.66	0.67	52.42	4.65
	Yes	No	No	0.82	0.70	0.75	0.68	0.69	55.91	5.06
	Yes	Yes	No	0.82	0.70	0.75	0.68	0.69	96.21	4.24
	Yes	No	Yes	0.80	0.68	0.74	0.65	0.67	74.88	6.2
	Yes	Yes	Yes	0.80	0.68	0.74	0.65	0.67	71.93	5.94
Gradient Boosting Machines	No	–	–	0.79	0.66	0.72	0.65	0.67	44.13	1.77
	Yes	No	No	0.80	0.67	0.72	0.66	0.68	44.35	2.55
	Yes	Yes	No	0.80	0.67	0.72	0.66	0.68	42.97	2.67
	Yes	No	Yes	0.79	0.66	0.72	0.63	0.65	41.5	3.03
	Yes	Yes	Yes	0.79	0.66	0.72	0.63	0.65	39.18	2.6
Neural Networks	No	–	–	0.79	0.87	0.72	0.70	0.71	84.07	4.26
	Yes	No	No	0.80	0.87	0.73	0.68	0.69	89.9	4.51
	Yes	Yes	No	0.80	0.87	0.73	0.68	0.69	85.53	5.39
	Yes	No	Yes	0.80	0.87	0.73	0.69	0.70	80.04	3.88
	Yes	Yes	Yes	0.80	0.87	0.73	0.67	0.69	84.48	4.58

Features engineering for prediction in PM4H settings requires more research and recommendations from PM4H community, as they are distinguished from other domains. In our work, we used intra-case attributes for prediction, but looking into inter-case attributes where shared information about concurrent ongoing cases can be useful [7]. The integration of OOPPM in healthcare information systems (HISs) satisfy a need identified by PM4H community *Challenge 9: completement HISs with the process perspective* [11]. OOPPM has performed better than traditional DM in healthcare settings, and thus its integration into PM^2 and other PM methodologies is recommended. We urge PM4H to implement OOPPM in their research work as there is large area for enhancements.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Jencks, S., Williams, M., Coleman, E.: Rehospitalizations among patients in the Medicare fee-for-service program. N. Engl. J. Med. **360**(14), 1418–1428 (2009)
2. Kalra, A., Fisher, R., Axelrod, P.: Decreased length of stay and cumulative hospitalized days despite increased patient admissions and readmissions in an area of urban poverty. J. Gen. Intern. Med. **25**, 930–935 (2010)
3. Glans, M., Kragh, A., Jakobsson, U.: Risk factors for hospital readmission in older adults within 30 days of discharge–a comparative retrospective study. BMC Geriatrics **20** (2020)
4. Assaf, R., Jayousi, R.: 30-day hospital readmission prediction using MIMIC data. In: 14th International Conference on Application of Information and Communication Technologies (AICT), pp. 1–6. IEEE (2020)
5. Tey, F., Liu, C., Chien, T.: Predicting the 14-day hospital readmission of patients with pneumonia using artificial neural networks (ANN). Int. J. Environ. Res. Public Health **18**(10) (2021)
6. Thapa, N., Seifollahi, S., Taheri, S.: Hospital readmission prediction using clinical admission notes. In: 2022 Australasian Computer Science Week Proceedings, pp. 193–199 (2022)
7. Teinemaa, I., Dumas, M., Rosa, M.: Outcome-oriented predictive process monitoring: review and benchmark. ACM Trans. Knowl. Discov. Data **13**(2), 1–57 (2019)
8. Munoz-Gama, J., Martin, N., Fernandez-Llatas, C.: Process mining for healthcare: characteristics and challenges. J. Biomed. Inform. **1**(127), (2022)
9. Cremerius, J., König, M., Warmuth, C.: Patient discharge classification based on the hospital treatment process. In: International Conference on Process Mining, pp. 314–326. Springer International Publishing, Cham (2021)
10. Chen, Q., Lu, Y., Tam, C.S.: Outcome-oriented predictive process monitoring to predict unplanned ICU readmission in MIMIC-IV database. In: ECIS (2022)
11. Johnson, A., Bulgarelli, L., Shen, L.: MIMIC-IV, a freely accessible electronic health record dataset. Sci. Data **10**(1) (2023)
12. Van Eck, M., Lu, X., Leemans, S.: PM: a process mining project methodology. In: International conference on advanced information systems engineering, pp. 297–313. Springer International Publishing, Cham (2015)
13. Johnson, O., Ba Dhafari, T., Kurniati, A.: The ClearPath method for care pathway process mining and simulation. In: Business Process Management Workshops: BPM 2018 International Workshops, pp. 239–250. Springer International Publishing (2019)
14. Van Der Aalst, W.: Process mining: discovering and improving Spaghetti and Lasagna processes. In: Symposium on Computational Intelligence and Data Mining, pp. 1–7. IEEE (2011)
15. Aravazhi, A., Helgheim, B., Aadahl, P.: Decision-making based on predictive process monitoring of patient treatment processes: a case study of emergency patients. Adv. Oper. Res. (1) (2023)
16. Di Francescomarino, C., Dumas, M., Maggi, F.: Clustering-based predictive process monitoring. IEEE Trans. Serv. Comput. **12**(6), 896–909 (2016)
17. Leontjeva, A., Conforti, R., Di Francescomarino, C.: Complex symbolic sequence encodings for predictive monitoring of business processes. In: 13th International Conference on Business Process Management, pp. 297–313. Springer International Publishing. (2015)