Towards Explainable Predictive Models for Electronic Health Records

Amela Fejza · Pierre Genevès · Nabil Layaïda · Jean-Luc Bosson

Received: date / Accepted: date

Abstract Early identification of patients at risk of developing complications during their hospital stay is currently a challenging issue in healthcare. Complications include hospital-acquired infections, admissions to intensive care units, and in-hospital mortality. Being able to accurately predict the patients' outcomes is a crucial prerequisite for tailoring the care that certain patients receive, if it is believed that they will do poorly without additional intervention. We consider the problem of complication risk prediction, such as inpatient mortality, from the electronic health records of the patients. We study the question of making predictions on the first day at the hospital, and of making updated mortality predictions day after day during the patient's stay. We develop distributed models that are scalable and interpretable. Key insights include analysing diagnoses known at admission and drugs served, which evolve during the hospital stay. We leverage a distributed architecture to learn interpretable models from training datasets of gigantic size. We test our analyses with more than one million of patients from hundreds of hospitals, and report on the lessons learned from these experiments.

Keywords Big medical data · EHR · Artificial intelligence · Machine learning · Predictive models · Predictive Analytics · Complication risk · Interpretability · Scalability

This research was partially supported by the ANR project CLEAR (ANR-16-CE25-0010) and by MIAI@Grenoble Alpes, (ANR-19-P3IA-0003).

Amela Fejza, Pierre Genevès and Nabil Layaïda
Tyrex team, Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France E-mail: {amela.fejza,pierre.geneves,nabil.layaida}@inria.fr

Jean-Luc Bosson
Univ. Grenoble Alpes, CNRS, Public Health department CHU Grenoble Alpes, Grenoble INP, TIMC-IMAG, 38000 Grenoble, France E-mail: JLBosson@chu-grenoble.fr

1 Introduction

One major expectation of data science in healthcare is the ability to leverage on digitized health information and computer systems to better apprehend and improve care. Over the past few years the adoption of electronic health records (EHRs) in hospitals has surged to an unprecedented level. In the USA for example, more than 84% of hospitals have adopted a basic EHR system, up from only 15% in 2010 [1, 13]. The availability of EHR data opens the way to the development of quantitative models for patients that can be used to predict health status, as well as to help prevent disease, adverse effects, and ultimately death.

We consider the problem of predicting important clinical outcomes such as inpatient mortality, based on EHR data. This raises many challenges including dealing with the very high number of potential predictor variables in EHRs. Traditional approaches have overcome this complexity by extracting only a very limited number of considered variables [6,15]. These approaches basically trade predictive accuracy for simplicity and feasibility of model implementation. Other approaches have dealt with this complexity by developing black box machine learning models that retain predictor variables from a large set of possible inputs, especially with deep learning [3, 20, 22, 24]. These approaches often trade some model interpretability for more predictive accuracy.

Predictive accuracy is crucial as wrong predictions might have critical consequences. False positives might overwhelm the hospital staff, and false negatives can miss to trigger important alarms, exposing patients to poor clinical outcomes. However, model interpretability is essential as it allows physicians to get better insights on the factors that influence the predictions, understand, edit and fix predictive models when needed [5]. The search for tradeoffs between predictive accuracy and model interpretability is challenging.

We consider complication risk prediction and focus on two aspects of this problem: (i) how to make accurate predictions with interpretable models; and (ii) how to take into account evolving clinical information during hospital stay. Our main contributions are the following:

- we show that with relatively simple and interpretable models it is possible to make quite accurate risk predictions, based on data concerning admitting diagnoses and drugs served on the first day. Specifically, we devise models based on stacked logistic regressions that are interpretable[1], and we implement them in a distributed and highly scalable way so that they can learn from very large volumes of EHR data – a key ingredient to reach high predictive accuracy in this setting.
- we further develop mortality risk prediction models to make updated predictions when new clinical information becomes available during hospitalization. We analyze the evolution of drugs served and find in particular that analysing only the latest drugs served in conjunction with the admitting diagnoses yields quite accurate risk estimations of the patient perspectives of evolution.
- we report on lessons learned through practical experiments with real EHR data from more than one million of patients admitted to US hospitals, which is, to the best of our knowledge, one of the largest such experimental study conducted so far.
- we propose measures for general and instance-level interpretations of the predictions. As a step towards more model interpretability, these measures summarize the most influential features, so as to help in better understanding the reasons of a given predicted outcome.

Outline. The rest of the paper is organized as follows: we first present the data and methods used in § 2. In § 3 we present results obtained when making predictions of clinical outcomes on the first day at the hospital. In § 4 we investigate to which extent the predictive models can benefit from the availability of supplemental information becoming available during the hospital stay to make updated predictions. In § 5 we report on further investigations on the quality of risk estimations and discuss model explainability. Finally, we review related works in § 6 before concluding in § 7.

2 Methods

2.1 Data source

We used EHR data from the Premier healthcare database which is one of the largest clinical databases in the United States, gathering information from millions of patients over a period of 12 months from 417 hospitals in the USA [21]. These hospitals are believed to be broadly representative of the United States hospital experience. The database contains hospital discharge files that are dated records of all billable items (including therapeutic and diagnostic procedures, medication, and laboratory usage) which are all linked to a given admission [16]. We focus on hospital admissions of adults hospitalized for at least 3 days, excluding elective admissions. The snapshot of the database used in our work comprises the EHR data of 1,271,733 hospital admissions.

2.2 Outcomes

For a given patient, we consider the problem of predicting the occurrence of several important clinical outcomes:

- death: in-hospital mortality, defined as a discharge disposition of "expired" [10, 22];
- hospital-acquired infections (HAI) developed during the stay [23];
- admissions to intensive care unit (ICU) on or after the second day, excluding direct admissions on the first day;
- pressure ulcers (PU) developed during the stay (not present at admission).

Patients who experienced a given outcome are considered positive cases for this outcome; those who did not are considered negative cases. Table 1 presents the distribution of patients with respect to the considered outcomes.

Problem studied	Positive cases	Negative cases	Ratio
Mortality	28,236	857,005	3.29%
HAI	22,402	862,839	2.59%
ICU Admission	32,310	852,931	3.78%
Pressure Ulcers	23,742	861,499	2.75%

Table 1: Number of instances for each case study.

2.3 Preparing the data for supervised learning

Our methodology assumes no a priori clinical knowledge. For a given patient, we first extract a list E of elementary features including the age, gender, and admission type. Our models also use the list of admitting diagnoses known for a given patient as available in the EHR data at admission[2], which we denote by A. Procedures can be performed during the hospital stay. We denote the list of procedures performed

[1] We consider interpretability of the predictive models to be at least as important as accuracy so that the models can be further developed, edited or fixed in collaboration with medical doctors.

[2] We use a list of unique identifiers encoded using the International Classification of Diseases, Ninth Revision, Clinical Modification known as ICD-9-CM.

on the i^{th} day of the stay (with $i > 0$) by P_i. We also consider the lists of drugs served, on a daily basis: D_i denotes the list of drug names (and their quantities) served on the i^{th} day.

We filter out unused procedures and drugs, and use a perfect hash function to encode the features. The feature matrix is very sparse so in the implementation we use a sparse representation of feature vectors. Most patients are admitted at the hospital with at least one admitting diagnosis (among 5,094 possible diagnoses). A small proportion of patients receive procedures during their stay ($\sim 20\%$ of patients receive procedures on the first day). The total number of possible procedures is 11,338. Furthermore, during the stay, a total of 10,739 possible drugs can be served. On the first day of stay, a patient is served 8.6 drugs on average. Figure 1 shows the distribution of the considered population in terms of the number of drugs received on the first day. Figure 2 shows an excerpt of the data for a sample patient.

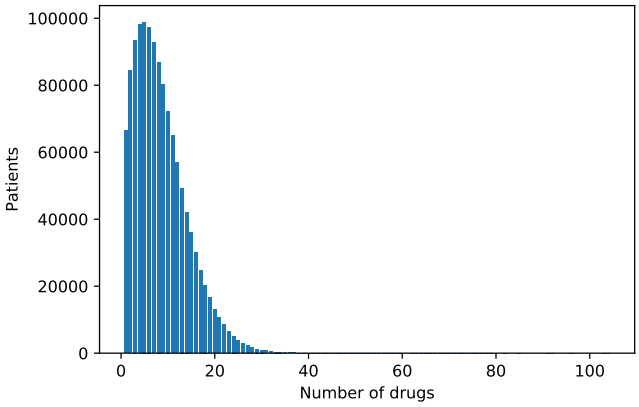


Fig. 1: Population distribution in terms of the number of drugs received on the first day ($|D_1|$).

Reduction of dimensionality and its impacts on the analysis. We observe that some features concern only very few patients (e.g. some drugs are almost never served). This can cause issues in our approach because the test (or train) dataset may not even contain a patient with such a feature. We thus filter out the features which are used by less than a certain amount of patients (100 in our case). We verify that this has negligible impact on the quality of predictions in terms of AuROC and accuracy. One important advantage is that we obtain simpler models which are trained much more efficiently. We do this when predicting at t0+24h. The updated number of features contains 2,702 different drugs and 622 different diagnosis codes at admission time. We do not apply similar filters when studying the prediction in evolving days. This is because the set of drugs used by more than 100 patients may change from one day to another. Considering the intersection of the sets of common drugs used between different days might filter out some drugs whose importance is instrumental for the predictive quality on a particular day.

```
[(434456800,
  (82, 'M', 1,
   ['A(0)': [('578.1', 'A', '999', 999)]],
   ['D(1)': [('250258001120000', '2'),
      ('460460947620000', '1'),
      ('380381000310000', '2'),
      ('300305857300000', '1'),
      ('300305850250000', '1'),
     ...
      ('250250043500000', '1')]
    ],
    ['D(2)': [('380381000310000', '1'),
     ('320320721000000', '1'),
     ('300301825750000', '1'),
     ('250257025740000', '1.85'),
     ...
     ('250250052970000', '1')]
   ]))]
```

Fig. 2: Data excerpt for a 82 years old male patient who went out alive on the third day.

2.4 Development of models

2.4.1 Interpretability

Following [5], we pay specific attention to the interpretability of the predictive models we develop. Model interpretability (or "intelligibility" as found in [5]) refers to the ability to understand, validate, edit, and trust a learned model, which is particularly important in critical applications in healthcare such as the one we consider here. Accurate models such as deep neural nets and random forests are usually not interpretable, but more interpretable models such as logistic regression are usually less accurate. This often imposes a tradeoff between accuracy and interpretability. We choose to preserve interpretability and develop classifiers based on logistic regression. Advantages of logistic regression include yielding insights on the factors that influence the predictions, such as an interpretable vector of weights associated to features, and predictions that can be interpreted as probabilities. We further report on model interpretability in Section 5.

2.4.2 Mathematical formulation

Logistic regression can be formulated as the optimization problem $\min_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w})$ in which the objective function is of the form:

$$f(\boldsymbol{w}) = \lambda R(\boldsymbol{w}) + \frac{1}{n}\sum_{i=1}^{n} \theta_i L(\boldsymbol{w}; \boldsymbol{x_i}, y_i)$$

where n is the number of instances in the training set, and for $1 \leq i \leq n$:

- $\boldsymbol{w}$ is the vector of weights we are looking for.
- the vectors $\boldsymbol{x_i} \in \mathbb{R}^d$ are the instances of the training data set: each vector $\boldsymbol{x_i}$ is composed of the d values corresponding to features retained for a given admission.
- $y_i \in \{0, 1\}$ are their corresponding labels, which we want to predict (e.g. for the mortality case study, 0 means the patient survived and 1 means the patient died at the hospital)
- $R(\boldsymbol{w})$ is the regularizer that controls the complexity of the model. For the purpose of favoring simple models and avoiding overfitting, in the reported experiments we used $R(\boldsymbol{w}) = \frac{1}{2}||\boldsymbol{w}||_2^2$.
- λ is the regularization parameter that defines the trade-off between the two goals of minimizing the loss (i.e., training error) and minimizing model complexity (i.e., to avoid overfitting). In the reported experiments we used $\lambda = \frac{1}{2}$.
- θ_i is the weight factor that we use to compensate for class imbalance. The classes we consider are heavily imbalanced (as shown in Table 1): in-hospital death for instance can be considered as a rare event. Notice that we do not use downsampling (that would drastically reduce the set of negative instances for the purpose of rebalancing classes); instead we apply the weighting technique [14] that allows our models to learn from all instances of imbalanced training sets. θ_i is thus in charge of adjusting the impact of the error associated to each instance proportionally to class imbalance: $\theta_i = \tau \cdot y_i + (1 - \tau) \cdot (1 - y_i)$ where τ is the fraction of negative instances in the training set.
- the loss function L measures the error of the model on the training data set, we use the logistic loss:

$$L(\boldsymbol{w}; \boldsymbol{x_i}, y_i) = \ln(1 + e^{(1-2y_i)\boldsymbol{w}^T\boldsymbol{x_i}})$$

Given a new instance $\boldsymbol{x}$ of the test data set, the model makes a prediction by applying the logistic function:

$$f(z) = \frac{1}{1 + e^{-z}}$$

where $z = \boldsymbol{w}^T\boldsymbol{x}$. The raw output $f(z)$ has a probabilistic interpretation: the probability that x is positive. In the sequel we rely on this probability to build further models (in particular using the stacking technique: see the combined model built from such probabilities in § 4). We also use the common threshold $t = 0.5$ such that if $f(z) > t$, the outcome is predicted as positive (and negative otherwise)[3].

[3] We make t vary in $[0, 1]$ for computing ROC curves.

2.4.3 Scalability with distributed computations

a particularity of our study is that we want our models to be able to learn from very large amounts of training data (coming from many hospitals). We typically consider models for which both n and d are large: for instance $n > 8 * 10^5$ and $d > 16 \cdot 10^3$ when we train models using features found in EAD_1. This is achieved by implementing a distributed version of logistic regression, including a distributed version of the L-BFGS optimization algorithm which we use to solve the aforementioned optimization problem. L-BFGS is known for often achieving faster convergence compared with other first-order optimization techniques [2]. We use a cluster composed of one driver machine and a set of worker machines[4]. Each worker machine receives a fraction of the training data set. The driver machine then triggers several rounds of distributed computations performed independently by worker machines, until convergence is reached. The software was implemented using the Python programming language and the Apache Spark machine learning library [19].

2.5 Model evaluation and statistical analysis

Patients were randomly split into disjoint train and test subsets. We perform k-fold cross validation with $k = 5$ unless indicated otherwise ($k = 10$ when indicated). Model accuracy is reported in terms of several metrics on the (naturally imbalanced) test set, which is used exclusively for evaluation purposes. We report on the receiver operator characteristic (ROC) curves and especially on the area under the ROC curve (AuROC). For the sake of completeness, we also include the commonly used Accuracy metric [9]. Since we deal with highly skewed datasets (as shown in Table 1), we also report on the precision-recall (PR) curves and on the area under the PR curve (AuPR), as additional metrics [7].

2.6 Prediction timing

We consider making predictions at different times. First, we consider making predictions on the first day at the hospital. We report on corresponding results, for all considered clinical outcomes, in § 3. We then report on how to make new mortality predictions, day after day, whenever new EHR information becomes available, and present corresponding results in § 4.

[4] Reported experiments were conducted with 5 machines (1 driver and 4 workers), each equipped with two Intel Xeon CPU (1.90GHz-2.6Ghz), with 24 to 40 cores, 60-160 GB of RAM, and a 1GB ethernet network.

3 Results on Predictions on the First Day

On the first day, we consider predictive models built with different sets of features (that we later combine). We name the models we consider after the sets of features they rely on. For example we consider the model EA for making predictions at hospital admission time t_0 (i.e. at the moment when the patient arrives at the hospital). This model uses the elementary features E and the diagnoses A known at admission. We also consider making predictions whenever the set of drugs served on the first day is known (typically at $t_0 + 24h$). For this purpose, we consider the model ED_1 of [10] that uses elementary features and drugs served on the first day. All the considered models systematically use the elementary features E, so we often omit E in model names in the sequel.

3.1 Mortality risk prediction

For predicting in-hospital mortality, AuROC was 77.8% and AuPR was 12.7% with the D_1 model, indicating significant predictive power of the drugs served on the first day (as already known from [10]). Over the total considered population of 1,271,733 patients, 885,241 ($\sim$ 70%) of them have non-empty admitting diagnosis information at admission time ($A \neq \emptyset$). AuROC was 76.4% and AuPR was 10.9% with the A model, which is aimed to leverage this information for making predictions directly at admission time. This indicates predictive power of the admitting diagnoses as well. It thus makes sense to study how these models could be combined to obtain more accurate predictions for the concerned population of 885,241 patients. We study combinations of the predictions made at admission with predictions made at $t_0 + 24h$ with the knowledge of the set of drugs served on the first day.

More generally, we consider different model combinations:

- we consider models obtained by the flattening and concatenation of features found in several basic models. In the sequel, we denote by $\mathcal{C}(B_1, B_2, ..., B_n)$ (or equivalently by $B_1B_2...B_n$) the single model obtained from the (ordered) concatenation of the features used in the basic models $B_1, B_2, ..., B_n$. For instance we consider the model $\mathcal{C}(A,D_1)$ in which all the features found in A and D_1 are respectively concatenated, in order.
- we also use ensemble techniques and in particular the stacking technique [8] to create combined models. Using logistic regressions as basic models to be combined offers a significant advantage. We can not only reuse the binary output of the basic classifiers but also their raw probabilities (obtained before thresholding) as features of a stacked model. In the sequel, we denote by $\mathcal{S}(B_1, B_2, ..., B_n)$ the combined model obtained by stacking the basic models $B_1, B_2, ..., B_n$. Such a stacked model learns to which extent each basic model should be trusted, respectively in relation with the others. It does so by considering their raw output probabilities as features, which are analysed in comparison with the ground truth. For example, we consider the model $\mathcal{S}(A,D_1)$ built from the stacking of the two models A and D_1.

Table 2 gives an overview of the AuROC, Accuracy, AuPR and obtained with the basic and combined models considered, on the same population, having admitting diagnosis information. Table 2 indicates the average, minimum and maximum values of each metric obtained with a 5-fold cross-validation process.

We observe that the combined models yield significantly more accurate predictions than the basic ones, improving over comparable earlier works. For predicting inpatient mortality, with the AD_1 model AuROC was 80.4% and AuPR was 14.2%, compared to respectively 77.4% and 12.3% obtained with the D_1 model of [10].

Figure 3 presents the ROC curve obtained for a run of the $\mathcal{C}(A,D_1)$ model on a given train and test set. The PR curve is shown on Figure 4. Table 3 presents sizes of train and test sets, and Table 4 presents the confusion matrix and associated metrics.

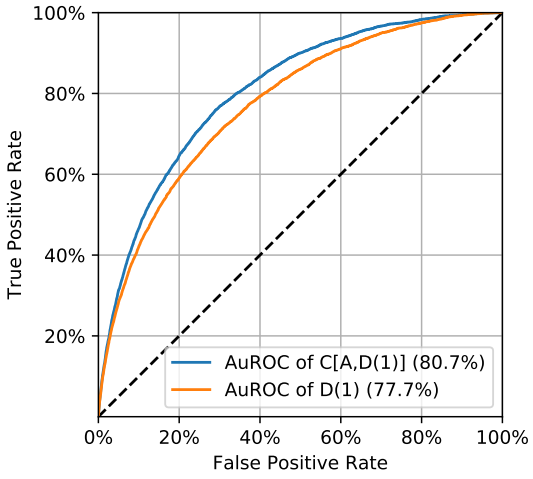


Fig. 3: Mortality prediction at $t_0 + 24h$: comparison of ROC curves obtained with our model $\mathcal{C}(A,D_1)$ and the model D_1 found in [10].

Model	AuROC %	Accuracy %	AuPR %
A	76.4 (76.0-76.8)	65.4 (65.2-65.6)	10.9 (10.6-11.2)
D_1 [10]	77.4 (77.2-77.7)	74.5 (74.5-74.8)	12.3 (12.0-12.5)
$\mathcal{S}(A,D_1)$	80.1 (79.9-80.2)	69.2 (68.9-69.4)	14.0 (13.5-14.3)
$\mathcal{C}(A,D_1)$	80.4 (80.2-80.7)	75.3 (75.2-75.5)	14.2 (13.8-14.6)

Table 2: Mortality risk predictions on the first day.

Mortality case study	Train set	Test set
Total size	708,373	176,868
Positive instances	22,660	5,576
Negative instances	685,713	171,292

Table 3: Number of instances for train and test sets.

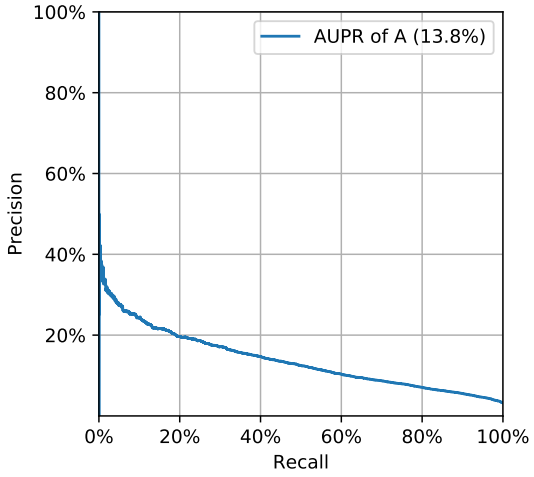


Fig. 4: PR curve for mortality prediction at $t_0 + 24h$.

Table 4: Sample confusion matrix.

		Prediction outcome P′	Prediction outcome N′	Total
Actual Value	P	3,928 TP	1,648 FN	5,576
	N	41,629 FP	129,663 TN	171,292
	Total	45,557	131,311	

TP is the number of true positives, FP the number of false positives, TN the number of true negatives and FN the number of false negatives. N is the total number of actual negatives, and P is the total number of actual positives.

True Positive Rate	70.4%	TP/P	recall
False Negative Rate	29.6%	FN/P	miss rate
True Negative Rate	75.7%	TN/N	specificity
False Positive Rate	24.3%	FP/N	fall-out
Negative Predictive Value	98.7%	TN/(TN+FN)	
Positive Predictive Value	8.6%	TP/(TP+FP)	precision
False Discovery Rate	91.4%	FP/(TP+FP)	
Accuracy	75.5%	(TP+TN)/(P+N)	
Error	24.5%	(FP+FN)/(P+N)	
AuROC	80.7%		
AuPR	13.8%		

3.2 Risk prediction for HAI, ICU admission, and pressure ulcers

Table 5 presents results obtained when predicting all the other considered clinical outcomes using the $\mathcal{C}(A, D_1)$ model. To the best of our knowledge, our models outperform state-of-the-art interpretable models found in the literature for predicting hospital-acquired infections (AuROC was 84.8% vs 80.3% in [10]), and for predicting pressure ulcers (AuROC was 82.2% vs 80.9% in [10]). This suggests that classifiers trained from large amounts of diagnoses and drugs served found in EHR data can produce valid predictions across a variety of clinical outcomes (not only mortality) on the first day at the hospital.

4 Results with Evolving Data

In this Section we study the problem of making mortality[5] predictions no longer at hospital admission time but at a later stage during the hospital stay, while taking into account new clinical information becoming available since admission.

We consider making inpatient mortality predictions on a daily basis. We investigate interpretable models that predict

[5] Notice that the labeling of our data (for supervised learning) conveys the information of whether an ICU admission, HAI or PU occured during the hospital stay, but not exactly *when* it occurred. This is why in this Section we exclusively focus on predicting mortality when new data arrives (the date of death corresponding to the last date of the stay).

Table 5: Predictive accuracy on the first day.

Case study	AuROC %	Accuracy %	AuPR %
Mortality	80.4 (80.2-80.7)	75.3 (75.2-75.5)	14.2 (13.8-14.6)
HAI	84.8 (84.5-85.1)	84.2 (84.1-84.3)	20.6 (20.3-21.0)
ICU	64.0 (63.7-64.3)	58.2 (58.1-58.5)	5.9 (5.7-6.0)
PU	82.2 (81.6-82.6)	78.3(78.2-78.5)	16.3 (16.0-16.5)

on day k using data available up to that day. Therefore, in addition to elementary features (E), diagnoses (A) known at admission and drugs served on the first day (D_1), we now consider the procedures P_i done on day i as well as the drugs D_i served on day i, for $i = 1$ to k as bases for the predictions.

4.1 Preliminary observations

Figure 5 gives insights on the number of patients remaining hospitalized at a certain day (no matter how long they stay). For each day i, it illustrates the subset of patients who have at least one drug served on that day (i.e. for which $D_i \neq \emptyset$), and the subset of patients who have a least one procedure on that day (i.e. for which $P_i \neq \emptyset$) , respectively. The vast majority of patients (more than 99.8%) are served drugs during their stay whereas only a small proportion of the population receive new procedures.

We first studied the evolution of procedures during hospitalization. In particular, we created separate models using E and P_i as features for each day i; but their combinations with ensemble techniques did not yield any significant improvement in prediction accuracy over the global population[6]. One possible explanation for this is that the number of patients with $P_i \neq \emptyset$ for $i \geq 1$ remains too limited (as shown in Figure 5). For this reason, we concentrate on the evolution of drugs served (D_i for $i \geq 1$) in the sequel.

4.2 Daily mortality risk predictions

For making predictions on a certain day k, we consider a variety of models built from different sets of features, that we combine with ensemble techniques (in a similar manner than for the first day – except that the set of basic models is now much richer as we can consider various models and several

[6] We did not obtain significant improvements when restricting to the patients having new procedures on the last day neither. Specifically, we filtered the population so as to retain only those patients who have at least one procedure at a certain day i. Since $|P_i|$=11,338, we conducted these analyses only until day 2 (on which only 140,747 patients received procedures). AuROC obtained with P_2 was in the 69-70% range.

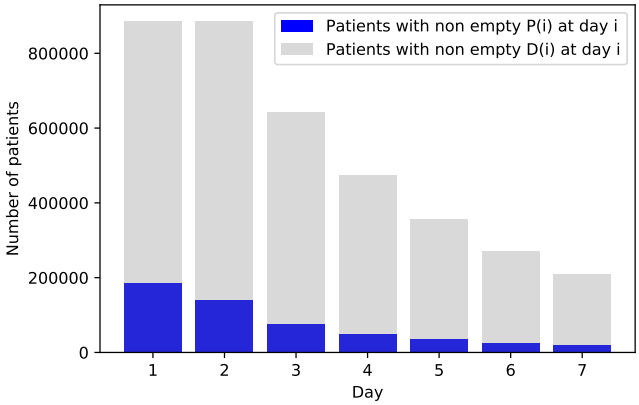


Fig. 5: Histogram illustrating the number of patients having at least one procedure or drug served on a given day.

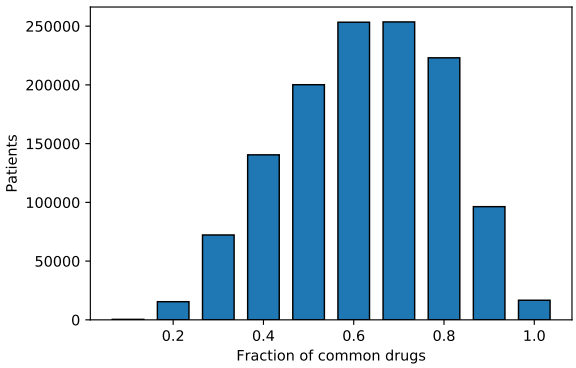
days). First, we consider all the models made with the features of one particular day [E,D_i] for $i = 1...k$. We thus obtain k such models, each one composed of $|E|$+$|D_i|$ features. More generally, we also consider models in which we incorporate a sliding window of historical data available since admission by considering the features [ED_{k-w},D_{k-w+1}, ..., D_k] for w decreasing from $k+1$ to 1. These models use up to $n * k$+$|E|$ features, where n is the number of possible drugs served ($n = 10,739$). This can represent a very large number of features. To avoid running into the curse of dimensionality, we define a threshold for the maximum acceptable ratio between the number of features and the number of training instances (that decreases with higher values of k as shown in Figure 5). We arbitrarily set this ratio to 10, which allows us to conduct analyses with sliding windows until the 6$^{\text{th}}$ day.

For instance, Table 6 presents the results obtained with all the basic models when predicting mortality on the 4th day of stay. First, we observe that AuROC, Accuracy and AuPR all raise when moving from a model D_k to a model $D_{k'}$ (for $1 \leq k < k' \leq 4$). This suggests that recent drugs served carry more accurate information on the current patient's situation perspective of evolution. In particular, information on the last drugs served (D_k) improve the accuracy of predictions at day k. Models that do not leverage the latest drug information become less accurate with time, as shown in Table 6. Second, taking into account historical

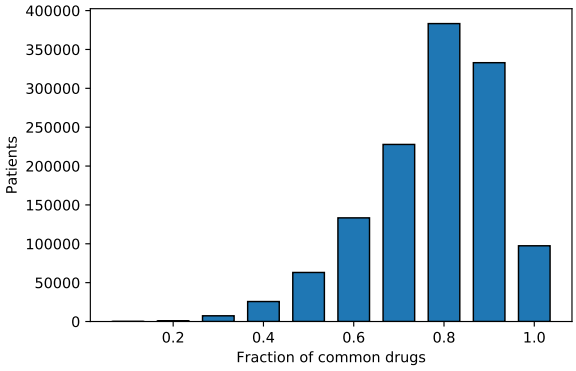
drugs served in the past few days (since admission) slightly improves AuROC and Accuracy.

This raises the question of how much historical data (since admission) it is worth to consider for making predictions (or in other terms, identifying tradeoffs between predictive accuracy and model complexity). Table 7 presents the performance metrics obtained when predicting with models with different sizes of moving windows of historical data. Notice that the AuROC obtained with a model on day i is not directly comparable to the AuROC obtained for predictions at admission time in § 3 because they do not correspond to the same population (some patients left the hospital or died before day i). Each row of Table 7 reports results for a different population filtered based on lengths of stay: predictions made at day i concern the population of patients that stayed for at least $i+1$ days.

Results suggest that $[D_k]$ models provide an interesting tradeoff (between accuracy and complexity) for predicting on day k compared to all the other models. One possible explanation for the limited accuracy improvements obtained with historical data since admission is that D_j carries most of the information from D_{j-1} for any j. Figure 6 gives an overview of the (high) similarities between drugs served on consecutive days. We computed the number of common drugs served in two consecutive days for a given patient, which we normalize with respect to the total number of drugs served. The distribution of the population in terms of this ratio is shown in Figure 6a for the first two days, and in Figure 6b for the next two days, respectively. We observe that for the majority of patients, the set of drugs served tends to only slightly change from a day to another, a majority of drugs being continuously served day after day.



(a) Between day 1 and day 2.



(b) Between day 2 and day 3.

Fig. 6: Similarities between drugs served on different days.

4.3 Discussion

All the predictions we made in this Section consider drug served data from the first day and onwards, but do not take into account the admitting diagnoses A (as opposed to § 3). These models can thus be particularly useful for making predictions for patients with no available admitting diagnosis ($\sim 30\%$ of the overall population).

For patients having an admitting diagnosis ($\sim 70\%$ of the overall population), we study combinations of the predictions made at admission (from § 3) with predictions made later during the stay. Figure 7 illustrates the different kinds of combined models that we consider. For instance, AuROC of the combined model $\mathcal{S}(A,D_1,D_2,D_3)$ was 82.2%. This suggests that data known at admission still helps in improving the accuracy of mortality predictions made at a later stage during the hospital stay. Table 8 presents the weights of the combined model $\mathcal{S}(A,D_1,D_2,D_3)$. We observe that the most important weight is associated with the A basic model. Then, the most important weights are associated, in decreasing order of importance, from the most recent to the oldest models. The higher influence on the final prediction thus comes from the admitting diagnoses, and then, from the drugs served in the most recent days.

So far, results suggest that when admitting diagnoses are available ($A \neq \emptyset$), then they should be used as they increase the predictive accuracy. Otherwise (when $A = \emptyset$), relying on the last drugs served (D_k) provides a reasonable tradeoff for making mortality prediction at a certain day $k > 0$ of the stay.

We now concentrate on studying how models with admitting diagnoses (A) can be combined with the models offering reasonable tradeoffs in terms of accuracy and complexity obtained so far (i.e. D_k). Each row of Table 9 reports results for a population filtered based on lengths of stay (predictions made at day i concern the population of patients that stayed for at least $i+1$ days). We observe that the AuROC of the predictions made with A decreases over time with the remaining population (as also illustrated by the blue line in Figure 8). We also observe a (much slighter) decrease in AuROC when predicting with the latest drugs served. It is rather intuitive that predictions made only with data known at admission (A) become less accurate with time. This sug-

Table 6: Results for mortality prediction on day 4 (considered population: patients that stay for at least 5 days).

Prediction	Model features	AuROC	Accuracy	AuPR
Day 4	D_1	(72.5-73.1)	(70.6-71.2)	(11.5-12.3)
Day 4	D_2	(76.4-76.8)	(74.6-74.9)	(15.2-15.4)
Day 4	D_3	(78.3-79.2)	(76.5-76.8)	(18.0-18.5)
Day 4	D_4	(80.8-81.1)	(78.8-79.1)	(20.6-22.0)
Day 4	D_1D_2	(76.8-77.0)	(75.4-75.8)	(15.1-15.6)
Day 4	$D_1D_2D_3$	(79.0-79.3)	(77.7-78.1)	(18.0-18.6)
Day 4	$D_1D_2D_3D_4$	(81.1-81.5)	(79.8-80.2)	(21.0-21.5)
Day 4	$D_2D_3D_4$	(80.9-81.4)	(79.3-79.7)	(21.0-22.0)
Day 4	D_3D_4	(80.8-81.4)	(79.1-79.4)	(21.0-22.0)

Table 7: Predictive accuracy with different historical windows (min and max values obtained with 5-fold cross validation).

Day	Model	AuROC	Accuracy	AuPR
2	D_2	(81.9-82.3)	(79.1-79.5)	(17.1-18.2)
	D_1D_2	(82.5-82.7)	(79.9-80.1)	(17.3-18.1)
3	D_3	(81.0-81.7)	(79.1-79.2)	(18.9-21.0)
	D_2D_3	(81.4-82.0)	(79.5-79.7)	(19.4-20.7)
	$D_1D_2D_3$	(81.6-82.3)	(80.0-80.2)	(19.3-20.7)
4	D_4	(80.8-81.1)	(78.8-79.1)	(20.6-22.0)
	D_3D_4	(80.8-81.4)	(79.1-79.4)	(21.0-22.2)
	$D_2D_3D_4$	(80.9-81.4)	(79.3-79.7)	(21.0-22.0)
	$D_1D_2D_3D_4$	(81.1-81.5)	(79.8-80.2)	(21.0-21.5)
5	D_5	(80.8-81.0)	(78.6-78.8)	(23.0-23.4)
	D_4D_5	(80.7-81.2)	(78.9-79.3)	(22.7-23.7)
	$D_3D_4D_5$	(80.5-81.2)	(79.1-79.4)	(22.5-23.7)
	$D_2D_3D_4D_5$	(80.3-81.1)	(79.4-79.7)	(22.0-23.6)
	$D_1D_2D_3D_4D_5$	(80.4-81.2)	(79.9-80.1)	(22.5-23.8)

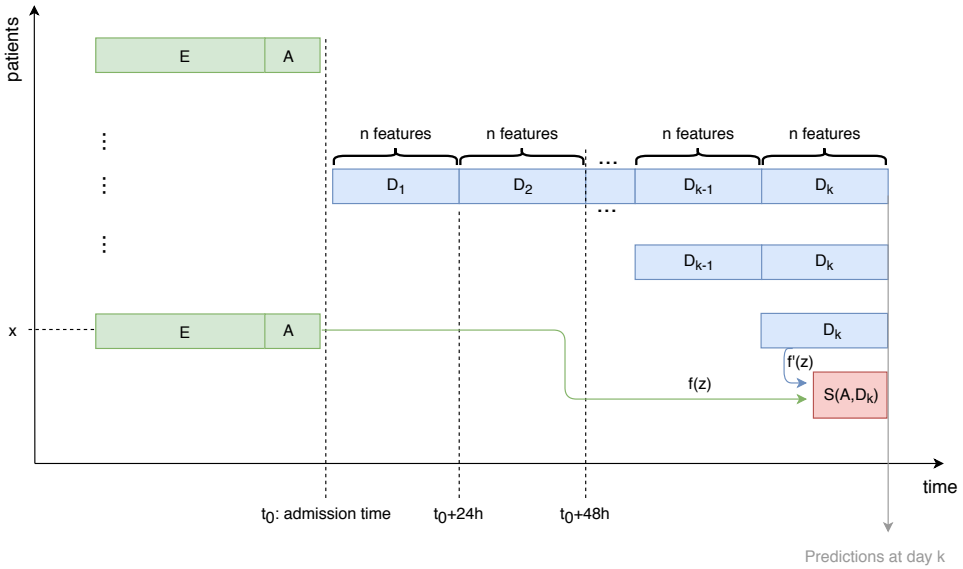


Fig. 7: The different kinds of models considered.

Weights	Model features
2.237623	probability using A
1.160006	probability using D_1
1.292319	probability using D_2
1.436751	probability using D_3

Table 8: Weights of the combined model $\mathcal{S}(A, D_1,D_2,D_3)$.

gests that, due to changing conditions, we are progressively left with more complex cases (that stay longer at the hospital), whereas the population with imminent outcome progressively leaves day after day (either dead or alive).

Results show that the combined (stacked) model always outperforms the other basic models in terms of AuROC. After a certain period though, say on day k, models considering admitting diagnoses (A) start to be outdated enough so that predictions made only with this model progressively carry less useful additional predictive power to be leveraged by the stacked model. Therefore the net increase in predictive power brought by the $\mathcal{S}(A,D_k)$ model erodes with time. This starts to occur significantly for $k \geq 6$, as illustrated on Table 9 and Figure 8. In conclusion, the joint analysis of the evolution of drugs served with admitting diagnoses helps in improving the AuROC of predictions made at any moment during the hospital stay, especially for short stays.

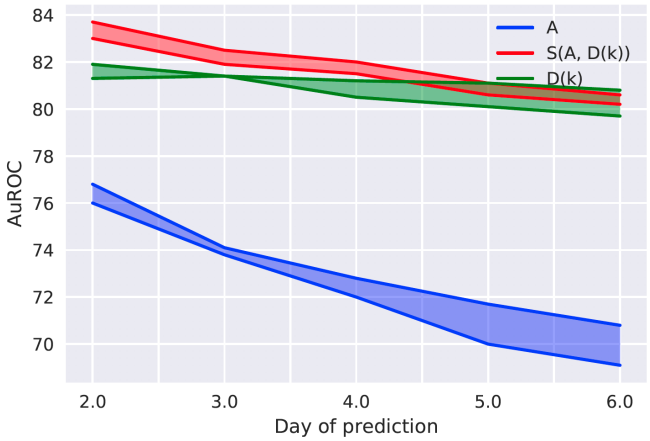


Fig. 8: Ranges of AuROCs obtained with A, D_k and $\mathcal{S}(A,D_k)$.

5 Insights on Predicted Risks and Model Interpretability

5.1 Analysis of risk predictions

From the logistic regression models that we build, we obtain a raw output $f(z)$ which has a probabilistic interpretation (as defined in Section 2.4.2). We then build a binary classifier by thresholding this probability, predicting positively (i.e. that the outcome will occur) when $f(z) > 0.5$ and negatively otherwise. So far, we mainly focused on evaluating the "discriminative power" of the binary classifier (after thresholding), in particular using AuROC as commonly found in the literature. However, it is also natural to wonder how correct the estimated probabilities are. For instance, from the perspective of the binary classifier, two patients with estimated probabilities of 0.51 and 0.82 will be equally classified as positive. However, it might be important to know which patient has the highest risk – and whether the one with estimated probability of 0.82 indeed exhibits actual higher risk than the one with 0.51. This can be helpful when affecting budget for prevention, for instance. Rather than focusing on the discriminative power of the predictive algorithm, we thus focus on assessing the quality of estimated probabilities, before thresholding. Since we do not have any ground truth for the actual risks, we study how estimated probabilities relate to prevalence. For this purpose, we check how the estimated probabilities stand when compared to the fraction of actual positive instances in the population. Figure 9 presents the percentage of positive instances for each interval of estimated probabilities (for mortality risk prediction). Specifically, we order the estimated probabilities and separate them into bins (on the x-axis). Each bin corresponds to a subset of the population: the set of patients for which the estimated probability fits within the bin range bounds. We indicate on the y-axis the ratio between the number of dead people over the total amount of people in that bin (prevalence). For instance, in the whole (unbalanced) population,

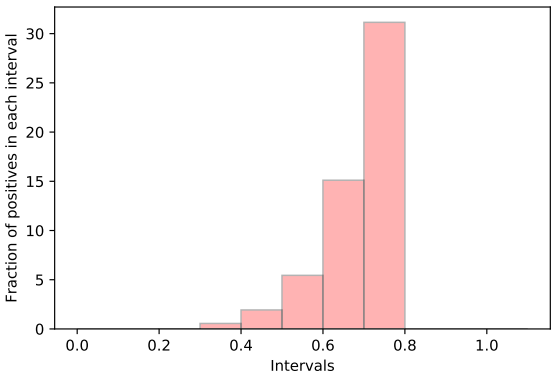


Fig. 9: Fraction of positive instances in the overall (unbalanced) population in terms of estimated probabilities (mortality risk prediction).

there are 9399 patients that are associated with an estimated probability in the range [0.6–0.7], from which 1420 actually died and 7979 survived. For this subpopulation, the actual

Prediction on	Model	AuROC	Accuracy	AuPR
Day 2	A	(76.0-76.8)	(65.2-65.6)	(10.9-11.2)
	D_2	(81.3-81.9)	(79.2-79.3)	(16.3-18.0)
	$\mathcal{S}(A,D_2)$	(83.0-83.7)	(73.5-74.0)	(18.5-19.1)
Day 3	A	(73.8-74.1)	(63.1-63.6)	(10.4-11.1)
	D_3	(80.6-81.4)	(79.1-79.2)	(18.6-19.9)
	$\mathcal{S}(A,D_3)$	(81.9-82.5)	(73.2-74.0)	(19.8-20.7)
Day 4	A	(72.0-72.8)	(61.0-61.8)	(10.7-11.3)
	D_4	(80.5-81.2)	(78.8-79.2)	(20.6-21.8)
	$\mathcal{S}(A,D_4)$	(81.5-82.0)	(72.9-73.2)	(21.2-22.9)
Day 5	A	(70.0-71.7)	(59.7-60.4)	(11.3-11.7)
	D_5	(80.1-81.1)	(78.4-78.9)	(21.5-24.2)
	$\mathcal{S}(A,D_5)$	(80.6-81.1)	(72.1-72.8)	(22.4-23.3)
Day 6	A	(69.1-70.8)	(58.3-58.8)	(11.4-12.4)
	D_6	(79.7-80.8)	(77.9-78.4)	(23.3-26.0)
	$\mathcal{S}(A,D_6)$	(80.2-80.6)	(71.6-72.2)	(23.9-24.7)

Table 9: Quality of predictions with stacked models.

prevalence is 15.1%. We observe that the ratio of positive instances in the whole population increases for each population subclass associated with higher estimated probabilities.

We now further compare the mortality risk estimated by our algorithm to the (a posteriori known) actual proportion of dead people for population subclasses. Since we deal with an unbalanced dataset, we first apply undersampling in order to generate a balanced test set (with the same amount of positive and negative instances). Figure 10 illustrates the percentage of positive instances in terms of predicted risks, where we order the different population classes by increasing order of predicted risk, in the manner of calibration theory as performed in [4, 25]. To check how well estimated probabilities stand with respect to prevalence, we compare the estimations (the blue curve) obtained in Figure 10 with the line (x=y) that represents the ideal case. We can thus assess how far from the (ex-post) reality our (ex-ante) predicted probabilities stand, based on how close the points of this curve are to this ideal line. This representation allows to draw important observations:

1. For high predicted probabilities (e.g. greater than 0.6) the curve departs from the (y=x) ideal line, in a way that indicates that our algorithm is too optimistic. For instance, when the predicted mortality risk is 0.6 on average, we know that the actual risk in this population is 0.8 on average. In other terms, our algorithm correctly identifies those cases (they are predicted as positive since >0.5) but in an optimistic way (the actual risk being higher on average). We make the converse observation for predicted probabilities lower than 0.45 which indicates pessimism in predictions, as shown in Figure 10.
2. Most importantly, the curve is strictly monotonic with respect to increasing predicted risks. This indicates that the higher the bin predicted probability is, the higher the actual risk.

For these reasons, we consider that the estimated probabilities actually help in providing a more precise insight on the patient's situation and perspectives of evolution. Hence our predictive system also outputs raw estimated probabilities, in addition to the binary predictions.

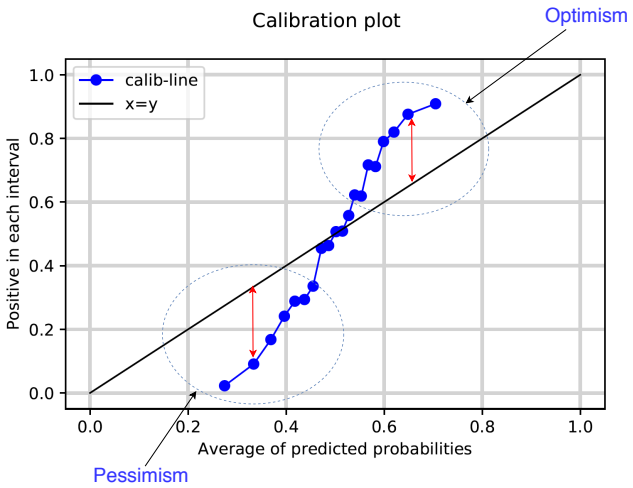


Fig. 10: Calibration curve for $\mathcal{S}(A,D_1)$.

5.2 General explanations

Apart from achieving accurate risk estimations, it is often equally or even more important to be able to explain them in a comprehensible way to domain experts (i.e. clinicians in this case). A straightforward approach to understand the variables that influence the most the predictions, is to observe the most significant weights of the model. An identification of the top most ones can be of great value for model interpretability. Also, we are interested to know to which extent variables associated with the greatest weights are simi-

lar between model instances, looking for stability of top influencers. Specifically, we study to which extent the logistic regression weights vary, when different training sets are randomly picked. In particular, we look for the most significant weights obtained from a model instance to another. We rank the weights obtained from two different model instances in terms of their absolute values, and compare them. For example, we retained the 100 most important weights corresponding to the features of $\mathcal{C}(A,D_1)$ (with the most significant absolute values) obtained for several random training sets. We observe that the vast majority of the topmost weights remain the same accross each different model instance. A systematic pairwise comparison of the lists of topmost weights for each model instance shows that the least proportion of common weights between model instances is 70% to 80%. This shows that top influencing variables tend to be the same accross different model instances.

Formally, we compute k model instances from k randomly picked train sets. For each model instance, we extract the set S_k of the n variables with greatest weights (in absolute value). For a given pair (k, n), we define the Top Global Variables (TGV) as the intersection of the sets $S_{1,..,k}$.

For example, Table 10 presents an excerpt of TGV (for $k = 2$ and $n = 100$) along with the ranking of the features and their impact (that can be either positive or negative) in predicting the outcome. Features are sorted by decreasing order of weights' absolute values. Features of type D correspond to drugs whereas features of type A correspond to admitting codes. Here a "positive" role is to be understood as a mathematically positive contribution in favor of the outcome (higher mortality risk). Table 10 illustrates diversity of main predictors: both drugs and admitting diagnoses can have positive or negative influence in favor of the predicted outcome. The clinical interpretation of those top most influencing variables is beyond the scope of this paper. However, the point is that our approach allows such a common vector (shared by most model instances) to be given to medical experts for further clinical research.

5.3 Instance-Level Explanations

While previous general explanations help pinpoint features with the highest impact, based on the model weights (see e.g. Table 10), some of the variables in TGV might not be present for a particular patient. In that case, it is still unclear which combination of features led to the predicted class (0 or 1). This is why we also want to have an explanation on a per-instance basis. The main question that arises here is: why is this patient classified in that class? For instance, why is he classified as having a significant mortality risk? Most known attempts in providing instance-level explanations fail to generalize to high-dimensional spaces (with e.g. thousands of features), as noticed in [17]. Inspired by the developments found in [17] we consider three concepts that provide elements of answer in seeking to explain predictions at instance-level: Top Positive Values, Minimal Piece of Evidence and Excess Confidence Level, that we present below.

Top Positive Values. We seek to explain the classification by observing the features which are present for a particular patient, from which we examine those with the greatest weights. We build on this idea and compute the smallest set containing these features:

We define Top Positive Values (TPV) as the smallest set defined as follows (inspired from [17]):

- TPV $\subseteq$ features
- model.predict(features) = 1:
- model.predict(TPV) = 1
- $\nexists$ TPV' $\subset$ TPV : model.predict(TPV') = 1 (TPV is minimal)

For computing TPV, we propose the Algorithm 1. For a given patient classified as positive, we consider the list of present features (given as a sparse vector representation). Each feature is associated with a weight given by the model. We sort them in decreasing order of their corresponding weights. Since the patient is classified as positive, we know that the initial vector of present features is not empty. We repeatedly use the same model to predict with an updated set of features, until we reach the case where the prediction changes (it becomes 1).

Data: features; // sparse vector of features for a given patient, sorted in decreasing order of weights importance, and such that model.predict(features)==1
Result: TPV
TPV ← {features.getFirstElement()}; // initialized with the singleton features' ← features.removeFirstElement();
while *model.predict(TPV) == 0* **do**
  TPV ← TPV ∪ {features'.getFirstElement()};
  features' ← features'.removeFirstElement();
end
return TPV;

Algorithm 1: Pseudo code for computing TPV.

For a given patient, TPV provides us with the minimal set of features which is sufficient to predict 1. However, it does not indicate that the absence of this set would lead the model towards a different prediction for that patient. In other terms, even without TPV, the model might still predict 1 due to the counterbalancing effect of some other variables. At that stage, we are left with one question: can we identify a "minimal piece of evidence" such that, if not present, would have led the predictive model to take a different decision?

Minimal Piece of Evidence (MPE). We study how the predicted class changes when features associated with the most

Feature code	Type	Role (+/-)	Rank	Short description
197.6	A	+	52	SEC MAL NEO RETROPERITON&PERIT
155.0	A	+	53	MALIGNANT NEOPLASM OF LIVER PR
162.8	A	+	54	MAL NEOPLSM OTH PART BRONCHUS
197.0	A	+	55	SEC MALIGNANT NEOPLASM OF LUNG
250250005000000	D	+	70	ATROPINE OP SOL 1% 5ML
519.8	A	-	1	OTH DISEASES RESPIRATORY SYSTEM
995.1	A	-	2	ANGIONEUROTIC EDEMA NEC
250636041650000	D	-	57	METHOTREXATE PWDR VL 1GM
250250037940000	D	-	58	LIDOCAINE, XYLOCAINE AMP 4% 5ML
250250005720000	D	-	59	BENAZEPRIL/AMLODIPINE, LOTREL TAB 10/2.5MG

Table 10: Description of some of the variables with high influence (a subset of TGV).

important weights are removed. Our method, inspired from [17], consists in being able to provide a set of features (present for the patient), such that removing all of them would cause a change of class. We are interested in a minimal set, such that when all the features of that set are put to zero, the class predicted by the model changes.

As we did for TPV, we analyse the features that are present for a particular patient and we examine the ones with the greatest weights. We define the Minimal Piece of Evidence (MPE) as the smallest set that causes the instance to be classified differently, as follows:

- MPE $\subseteq$ features
- model.predict(features) = 1
- model.predict(features$\backslash$MPE) = 0
- $\nexists$ MPE‘ $\subset$ MPE : model.predict(features$\backslash$MPE‘) = 0 (MPE is minimal)

where (features$\backslash$MPE) denotes the result of removing the features found on MPE from the list of features present in the patient.

Data: features; // sparse vector of features for a given patient, sorted in decreasing order of weights importance such that model.predict(features)==1
Result: MPE; // sparse vector of features
MPE;
while *model.predict(features) == 1* **do**
  MPE← MPE.append(features.getFirstElement());
  features ← features.removeFirstElement();
end
return MPE;

Algorithm 2: Pseudo code for computing MPE.

We propose the Algorithm 2 to compute MPE. Features are ordered in decreasing order of weight importance. Since the patient is classified as positive, we know that the set of features associated with positive weights is not empty. We start by setting to zero the value of the first present feature associated with the top most weight and we predict again with the updated instance. If the prediction is still the same, we keep removing the features in order, until the prediction changes, meaning that we repeatedly use the same predictive model on updated instances in which we successively affect the feature values (setting them to zero). We keep track of the removed features in a set (MPE), which eventually yields the minimal piece of evidence we are looking for.

Example. Figure 11 shows the sample features for a particular patient (a female patient, 74 years old with admission type 1 and a hospital length of stay of twelve days). For this patient, the ground truth is 1, meaning that the patient eventually died during her hospital stay. The class predicted by our system is 1 with an estimated 0.7926 probability. We can see that among the 3,328 possible features, only 22 of them are present for this particular patient.

```
Row(patientKey=532881591,
    lengthStay=12,
    label=1.0,
    features=SparseVector(3328,
    {0: 74.0, 2: 1.0, 3: 1.0, 203: 1.0,
     286: 1.0, 661: 1.0, 672: 1.0,
     715: 1.0, 719: 1.0, 744: 1.0,
     1134: 1.0, 1363: 1.0, 1446: 1.0,
     1564: 1.0, 1566: 1.0, 2204: 1.0,
     2318: 1.0, 2320: 1.0, 2362: 1.0,
     2404: 1.0, 2428: 1.0, 3054: 1.0}),
     rawPrediction=DenseVector([-1.3409, 1.3409]),
     probability=DenseVector([0.2074, 0.7926]),
     prediction=1.0)
```

Fig. 11: Sample sparse representation of features for a given patient.

We observe that for this patient, TPV's cardinality is 5:

TPV= {719: 1.0, 715: 1.0, 286: 1.0,
1363: 1.0, 1566: 1.0 }

We know that this set of features (further described in Figure 11) is sufficient for the system to raise an alarm about

mortality risk. The probability obtained when predicting only with TPV is 0.5187 (slightly above the threshold).

For the same patient of Figure 11, the cardinality of MPE is 11:

MPE= {719, 715, 286, 1363, 1566, 203, 3054, 1564, 661, 2362, 2404 }

The probability obtained when predicting without MPE is 0.4963 (slightly below the threshold). Notice that when predicting while removing MPE features one after the other, the estimated probability progressively decreases, and we can therefore examine the individual contribution of each MPE feature on the overall estimated probability. For example, when removing the feature 719, the estimated probability decreases from 0.7926 to 0.7532. This also allows to compare the relative contributions of variables.

Excess Confidence Level. We define the Excess Confidence Level (ECL) as the set MPE $\setminus$ TPV. For the particular case of the patient of Figure 11, we observe that TPV $\subset$ MPE, and Table 12 shows the details of the variables in ECL. In this example, we observe that even though TPV provides us with useful information, the absence of TPV does not lead the model to take a different decision. This is because the variables shown in Table 12 compensate the absence of TPV in predicting positively. ECL thus provides an additional insight on why the patient's perspectives of evolution are considered severe by the algorithm.

Notice that TGV, TPV, MPE and ECL are defined in a general manner which makes them usable when seeking explanations with any LR-based binary classifier. TPV and MPE were already used in a different application for explaining document classification techniques [17]. The present work shows that they are useful as well for healthcare analytics.

6 Related Works

The interest in developing predictive systems for EHR data has soared recently. The automated identification of at-risk profiles is a topic that has been actively investigated under various forms, including: prediction of hospital length-of-stay, readmissions, discharge diagnostics, occurrence of hospital-acquired infections, admissions to intensive care units, and in-hospital mortality. Several lines of work can be identified from the perspective of the methods used.

The first line of works gathers "score-based approaches". These works build on decades of research by clinicians and statisticians for attempting to measure the complexity of a patient's situation according to a yardstick index. The basic idea boils down to computing an aggregate score or index from EHR data. For a given patient, the value of the index is meant to represent the severity of the patient's condition and perspectives of evolution. Typical examples include the seminal Charlson comorbidity index [18], and the Medication Regimen Complexity Index (MRCI) [11]. The MRCI is a global score meant to indicate the complexity of a prescribed medication regimen. It aggregates 65 aspects related to the drug dosage form, dosing frequency and instructions. The greater the MRCI, the more complex the patient's situation is. The minimum MRCI score is 2 (e.g. one tablet taken once a day) and there is no maximum. The main advantages of score-based approaches, besides their simplicity, are that scores are well-defined and commonly accepted among clinicians, easily implementable, computationally cheap, and understandable even by non-experts. However, if scores can give a rapid estimation of a general patient's condition, their usage for predicting particular outcomes is still open. The work [15] shows positive correlations between the MRCI value at admission and the occurrence of complications later during the hospital stay. It remains unclear though to which extent such correlations might actually be leveraged in a predictive system. More generally, score-based approaches suffer from significant drawbacks when it comes to building accurate predictive systems. Reducing a priori the whole patient's situation and outcome to a single scalar value is questionable for two reasons. First, this aggregation is performed independently from any specific outcome to be predicted (like a particular complication). Second, many subtleties in EHR data (like drug interactions) are potentially discarded during the score computation. This may result in rough approximations. In the case of MRCI for instance, the same MRCI value may denote distinct situations with radically different perspectives of evolution.

A very recent line of work consists in analysing EHR data in a more comprehensive way, trying not to resort to a priori simplifications such as scores but trying instead to preserve as much as possible of the original EHR information to analyse it in a fine-grained way. For instance, the works found in [3, 5, 10, 22] apply supervised machine learning techniques to a wide range of features selected from EHR data. These works can be subdivided into two further subcategories: (i) those that trade model interpretability for numeric accuracy and (ii) those which preserve model interpretability, usually at the price of some loss in accuracy.

Among the first category, we find the works [3, 22] that develop classifiers based on deep neural networks (DNNs). The models proposed in [22] typically achieve areas under the ROC curve within the 0.79-0.89 range for mortality prediction at admission on their dataset (they do not report on AuPR nor on accuracy though). It would be interesting to compare the complete picture of the predictive performances of the models proposed in [22] with the ones proposed in the present paper on the same dataset. Unfortunately [22]

Feature ID	Charge Code	Type	Role	Short description
719	250250030560000	D	+	HEPARIN NA FLUSH VL 10U/ML 10ML
715	250250030520000	D	+	HEPARIN NA FLUSH VL 100U/ML 10ML
286	250250011800000	D	+	D50% 50ML SYRINGE
1363	250250052630000	D	+	PIPERACILLIN/TAZO, ZOSYN VL 3/0.375GM
1566	250250005000000	D	+	SOD BICARB VL 8.4% 50MEQ 50ML

Table 11: TPV for the sample patient of Figure 11.

Feature ID	Code	Type	Role	Short description
203	250250008600000	D	+	CA CHL INJ 10% 10ML (1GM)
3054	584.9	A	+	UNSPECIFIED ACUTE RENAL FAILURE
1564	250250058730000	D	+	SOD BICARB INJ 8.4% 10MEQ 10ML
661	250250028610000	D	+	GENTAMICIN, GARAMYCIN VL 40MG/ML 1ML
2362	250258001300000	D	+	D5% 1000ML
2404	250258002920000	D	+	WATER STERILE 1000ML

Table 12: Excess confidence level for the sample patient of Figure 11.

does not contain enough details on the features used in their models to allow for a reimplementation of their technique. Further comparisons with [22] are thus inappropriate because the datasets are different: they consider a lower number (216,221) of patients but with more data per patient (including historical data before admission, which we do not have in our dataset). In comparison, we analyse many more (1,271,733) patients with less data per patient (no history in particular). Notice that this makes our models more broadly applicable since they apply to patients for which we have no history at all.

It is also worth noticing that the results reported in [22] are achieved with an important sacrifice: a major drawback of DNNs is their lack of interpretability, as notoriously known. Preliminary works on explaining "black-box" models are rather in early stage [12]. Interpretability happens to be crucial for healthcare models so that they can be given to domain experts (e.g. clinicians) to be checked and fixed when necessary [5] using domain-expert medical knowledge. This is one reason why simple linear models (such as logistic regression) might be preferred over DNNs even when their accuracy is significantly lower, as detailed in [5].

Finally, the results described in [22] come at another price too: the computational cost. The computing power necessary for learning DNN classifiers with large amounts of data and large feature spaces is significant (an earlier version of [22] mentions more than 201,000 GPU hours of computation using Google Vizier for building the DNNs and setting up the hyper-parameters that crucially affect their performance). Compared to [22], our predictions are obtained with interpretable models, which yield interpretable weights (See 5) in particular. Our models are also lighter computationally and more scalable (one model is trained on 1.2 million of instances in around 4 minutes on a commodity cluster of 5 machines).

In the second category of works, we find [5] that advocates the use of the so-called "intelligible" models and apply them to two use cases in healthcare. The first use case is concerned with the prediction of pneumonia risk. The goal is to predict probability of death so that patients at high risk can be admitted to the hospital, while patients at low risk are treated as outpatients. Their intelligible model provides an AUC in the 0.84-0.86 range, and uncovers patterns in the data that previously had prevented complex learned models from being fielded in this domain. The second use case is the prediction of hospital readmission, for which the developed model provides an AUC in the range 0.75-0.78. The datasets considered in [5] are uncomparable to the dataset we use in the present paper. For the prediction of pneumonia risk, [5] considers a dataset of 14,199 pneumonia patients with 46 features. For the prediction of hospital readmission, [5] considers 296,724 patients from a large hospital, with 3,956 features for each patient. Features include lab test results, summaries of doctor notes, and details of previous hospitalizations. Our problem formulation is different as we concentrate on predicting the risks of in-hospital mortality and other complications for any patient admitted to the hospital. We also consider a larger dataset (>1.2M patients) with more features, thanks to a distributed implementation.

The work found in [10] also applies linear models such as logistic regression for building binary classifiers. The goal is similar: making predictions of complications at hospital admission time. In § 3 we applied our method on the same dataset so the results can be directly compared. Our method provides significant improvements in accuracy: for mortality prediction our method achieves an AuROC in the range 80.2-80.7% (Accuracy: 75.3%) compared to 77.9% (Accu-

racy: 75%) reported in [10]. Similar improvements in accuracy can also be observed when predicting other complications (HAI, PU) considered in [10]. By leveraging admitting diagnoses (A), our method thus makes it possible to obtain even more accurate predictions when compared to [10]. The models proposed in [10] remain useful for patients with no admitting diagnosis. Finally and more importantly, compared to [10], we investigate the problem of making updated mortality predictions whenever more clinical data become available during the hospital stay (Section 4), which is not considered in [10].

7 Conclusion

We develop a distributed supervised machine learning system for predicting clinical outcomes based on EHR data. We propose interpretable and highly scalable models, capable of leveraging the knowledge of admitting diagnoses and drugs served during the hospital stay. These models can be used to make predictions concerning the risk of hospital-acquired infections, pressure ulcers, and inpatient mortality. We study how mortality risk models can be extended with the analysis of the evolution of drugs served during the stay. The distributed implementation trains models with millions of patient profiles. We also assess the system predictions and propose further explanations for the risks predicted, using concepts inspired from both calibration theory and document classification. Finally, we report on lessons learned with a large-scale experimental study with real EHR data from US hospitals.

References

1. J. Adler-Milstein, A. J. Holmgren, P. Kralovec, C. Worzala, T. Searcy, and V. Patel. Electronic health record adoption in u.s. hospitals: The emergence of a digital 'advanced use' divide. *Journal of the American Medical Informatics Association*, 24(6):1142–1148, Nov 2017. url.
2. G. Andrew and J. Gao. Scalable training of l^1-regularized log-linear models. In *Machine Learning, Proceedings of the International Conference (ICML)*, pages 33–40, 2007. url.
3. A. Avati, K. Jung, S. Harman, L. Downing, A. Y. Ng, and N. H. Shah. Improving palliative care with deep learning. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine*, pages 311–316, 11 2017. url.
4. A. Bella. Chapter 6 calibration of machine learning models. 2016.
5. R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730, August 2015. url.
6. Y. Choi, C. Y.-I. Chiu, and D. Sontag. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41–50, 2016. url.
7. J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *Machine Learning, Proceedings of the International Conference (ICML)*, pages 233–240, June 2006. url.
8. T. G. Dietterich. Ensemble methods in machine learning. In *Intl workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
9. T. Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27(8):861–874, jun 2006. url.
10. P. Genevès, T. Calmant, N. Layaïda, M. Lepelley, S. Artemova, and J.-L. Bosson. Scalable machine learning for predicting at-risk profiles upon hospital admission. *Big Data Research*, (12):23–34, 2018. url.
11. J. George, Y. Phun, M. J. Bailey, D. C. Kong, and K. Stewart. Development and validation of the medication regimen complexity index. *Annals of Pharmacotherapy*, 38(9):1369–1376, 2004. url.
12. R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti. A survey of methods for explaining black box models. *CoRR*, abs/1802.01933, 2018. url.
13. J. Henry, Y. Pylypchuk, T. Searcy, and V. Patel. Adoption of electronic health record systems among u.s. non-federal acute care hospitals: 2008-2015. url, may 2016.
14. G. King and L. Zeng. Logistic regression in rare events data. *Political Analysis*, 9(2):137–163, 2001. url.
15. M. Lepelley, C. Genty, A. Lecoanet, B. Allenet, P. Bedouch, M.-R. Mallaret, P. Gillois, and J.-L. Bosson. Electronic medication regimen complexity index at admission and complications during hospitalization in medical wards: a tool to improve quality of care? *International Journal for Quality in Health Care*, 2017. url.
16. R. Makadia and P. B. Ryan. Transforming the premier perspective®hospital database into the observational medical outcomes partnership (omop) common data model. *eGEMs*, 2(1):1110, 2014. url.
17. D. Martens and F. Provost. Explaining data-driven document classifications. *MIS Q.*, 38(1):73–100, Mar. 2014.
18. C. ME, P. P, A. KL, and M. CR. A new method of classifying prognostic comorbidity in longitudinal studies: development and validation. *J Chronic Dis.*, 40(5):373–83, 1987. url.
19. X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar. Mllib: Machine learning in Apache Spark. *CoRR*, abs/1505.06807, 2015. url.
20. R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific Reports*, 6, May 2016. url.
21. Premier. Healthcare database, Feb 2018. url.
22. A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, P. Sundberg, H. Yee, K. Zhang, Y. Zhang, G. Flores, G. E. Duggan, J. Irvine, Q. Le, K. Litsch, A. Mossin, J. Tansuwan, D. Wang, J. Wexler, J. Wilson, D. Ludwig, S. L. Volchenboum, K. Chou, M. Pearson, S. Madabushi, N. H. Shah, A. J. Butte, M. D. Howell, C. Cui, G. S. Corrado, and J. Dean. Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(1):18, 2018. url, An earlier version appeared in eprint arXiv:1801.07860.
23. D. Roosan, M. Samore, M. Jones, Y. Livnat, and J. Clutter. Big-data based decision-support systems to improve clinicians' cognition. In *Proceedings of the IEEE International Conference on Healthcare Informatics*, pages 285–288. IEEE, December 2016. url.
24. B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi. Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE Journal of Biomedical and Health Informatics*, 2017. url.

25. B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 609–616, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.