Predicting Sequences of Clinical Events by using a Personalized Temporal Latent Embedding Model

Cristóbal Esteban
Siemens AG and
Ludwig Maximilian
University of Munich
Munich, Germany
cristobal.esteban@siemens.com

Danilo Schmidt
Charité
University Hospital of Berlin
Berlin, Germany
danilo.schmidt@charite.de

Denis Krompaß
Siemens AG
Munich, Germany
denis.krompass@siemens.com

Volker Tresp
Siemens AG and
Ludwig Maximilian
University of Munich
Munich, Germany
volker.tresp@siemens.com

***Abstract*—As a result of the recent trend towards digitization —which increasingly affects evidence-based medicine, accountable care, personalized medicine, and medical "Big Data" analysis— growing amounts of clinical data are becoming available for analysis. In this paper, we follow the idea that one can model clinical processes based on clinical data, which can then be the basis for many useful applications. We model the whole clinical evolution of each individual patient, which is composed of thousands of events such as ordered tests, lab results and diagnoses. Specifically, we base our work on a dataset provided by the Charité University Hospital of Berlin which is composed of patients that suffered from kidney failure and either obtained an organ transplant or are still waiting for one. These patients face a lifelong treatment and periodic visits to the clinic. Our goal is to develop a system to predict the sequence of events recorded in the electronic medical record of each patient, and thus to develop the basis for a future clinical decision support system. For modelling, we use machine learning approaches which are based on a combination of the embedding of entities and events in a multidimensional latent space, in combination with Neural Network predictive models. Similar approaches have been highly successful in statistical models for recommendation systems, language models, and knowledge graphs. We extend existing embedding models to the clinical domain, in particular with respect to temporal sequences, long-term memories and personalization. We compare the performance of our proposed models with standard approaches such as K-nearest neighbors method, Naive Bayes classifier and Logistic Regression, and obtained favorable results with our proposed model.**

I. Introduction

It is well known that data observed in clinical practice can lead to important insights and can complement information gathered from controlled clinical studies [1]. One argument is that data from clinical practice reflects the natural mix of patients whereas patients participating in clinical studies typically have another composition: they are carefully selected, they should not have other problems as the one under study, and they should not receive any other treatment. Also, a future personalized medicine needs to be based on many attributes from a large number of patients, information that can be collected from data recorded during the clinical practice [2], [3].

In this paper we focus on the prediction of clinical events, such as decisions, procedures, measurements and other observations. We model the whole evolution of each individual patient, which is composed of thousands of single events. A good predictive system could have many applications, for example, as part of a decision support system that predicts common practice in a clinical setting and which could alert in case of unusual orders. Eventually, a predictive system could also be used to optimize decisions, although here, confounding variables can be a problem. If many dimensions are measured, the available information might include direct or indirect information on important confounders, alleviating the problem [4], [3].

We are addressing the issue from a "Big Data" perspective and use a large data set collected from patients that suffered from kidney failure. The data was collected in the Charité hospital in Berlin and it is the largest data collection of its kind in Europe. Once the kidney has failed, patients face a lifelong treatment and periodic visits to the clinic for the rest of their lives. Until the hospital finds a new kidney for the patient, the patient must attend to the clinic multiple times per week in order to receive dialysis, which is a treatment that replaces many of the functions of the kidney. After the transplant has been performed, the patient receives immunosuppressive therapy to avoid the rejection of the transplanted kidney. The patient must be periodically controlled to check the status of the kidney, adjust the treatment and take care of associated diseases, such as those that arise due to the immunosuppressive therapy. The usual procedure at the Charité University Hospital of Berlin for these periodic evaluations is that the visiting patient undergoes some laboratory testing in the morning, followed by the prescription of pertinent medications in the afternoon based on the results of the test.

The dataset contains every event that happened to each patient concerning the kidney failure and all its associated events: medications prescribed, hospitalizations, diagnoses, laboratory tests, etc. [5], [6]. The dataset started being recorded more than 30 years ago and it is composed of more than 4000 patients that underwent a renal transplantation or are waiting for it. For example, the database contains more than 1200 medications that have been prescribed more than 250000 times, and the results of more than 450000 laboratory analysis. The database

has been the basis for many studies in the past [7], [8], [9], [10]. In this work we study if future events for a patient can be predicted given the past events of the same patient. This is particularly important for the estimation of drug-drug interactions (DDI) and adverse drug reactions (ADR) in patients after renal transplantation.

Note that the data is extremely high-dimensional (there are thousands of diagnosis, procedures, lab results to consider) and sparse, since most combinations are unobserved. In recent years a number of approaches for this type of data situation have been developed in other application fields. These approaches are based on the concept of a low-dimensional latent embedding of the entities and events of interest in combination with Neural Network models and showed superior predictive performance in their respective domains. Examples are leading language models in natural language processing [12], the winning entries in the Netflix competition for the development of movie recommendation systems [11] and approaches for learning in knowledge graphs [13]. A new aspect here is that the temporal sequence of events plays an important role. In this paper we extend these models to be applicable towards temporal sequential models for the prediction of events in a clinical setting and we develop a new model that extends the Markov property of language models towards a personalized model and a long-term memory. We compare the prediction accuracy of these approaches with other leading modelling approaches such as a nearest neighbor methods, Naive Bayes classifier and Logistic Regression models.

The paper is organized as follows. In the next section we introduce the proposed models for this work. In Section IV we describe details of the nephrology use case and describe the data structure in detail. In Section V we explain the experimental set ups and present its results. Section VII contains our conclusions and an outlook.

II. Related Work

There have been efforts within the medical domain to simultaneously predict a reduced number of events [14] [15] and also to detect patterns within a larger amount of events [16]. Our dataset consists of sequences of high-dimensional sparse data and in this situation latent embedding approaches as used in language models [12], collaborative filtering [11] and knowledge graph models [13] have been very successful. In these models, the latent embeddings represent general entities such as users, items, or simply words, and the idea is that the embeddings represent the essence of the entities in form of low-dimensional real-valued representations. Latent embeddings were introduced as a suitable strategy for clinical data in [17] by predicting hospital readmissions. In this work we will show how to predict the sequence of a large amount of clinical events by developing a temporal latent embedding model.

III. Temporal Latent Embeddings for Predicting Clinical Events

In this section we extend latent embedding models to be applicable to clinical data which consist of temporal sequences of high-dimensional sparse events. In particular, in our approach the latent embeddings describe the state of the patient at a given time. Another extension is that we complement the short-term memory of language models with a long-term memory by including a representation of the complete clinical history of the patient.

A. The Basic Data Structures

A recorded event in our data is based on the schema *event(Time, Patient, EventType, Value)*. *Time* stands for the time of the event and is represented as the day of the event. Note that several events can happen at the same time. *Patient* stands for the patient ID and *EventType* for the type of the event, such as a specific diagnosis, a specific prescribed medication, a specific laboratory result and so on. For events like prescribed medications the value is equal to 1 if this particular event happens for the patient at time *Time=t* and is equal to 0 otherwise. For laboratory results such as Calcium or Blood count, we used a binary encoding and represented each measurement as three event types, i.e., *LabValueHigh*, *LabValueNormal* and *LabValueLow*.

These events can be stored in a three-way tensor $\underline{X}$ with dimensions *Time*, *Patient*, and *EventType*. The tensor entry $x_{t,i,j}$ with $t = 0,\dots,T$, $i = 0,\dots,I$, $j = 0,\dots,J$ is the value of the tuple *event(Time=t, Patient=i, EventType=j, Value)*. The tensor is extremely sparse and is stored in form of a sparse data structure. The task of the learning approach is to predict tensor entries for patients in the test set. In particular we predict entries in a second tensor $\underline{\Theta}$, with the same dimensions as $\underline{X}$, that contains the patients in the test set. The relationship between both is defined by the sigmoid function $P(x_{t,i,j} = 1) = \text{sig}(\theta_{t,i,j})$.

There are a number of interesting challenges in the dataset. Time plays an essential role and we are dealing with sequences of events but *absolute* time is of little value and a patient-specific normalization of time is non-trivial. Also the tensor $\underline{X}$ initially contains only data about the patients in the training dataset; our real goal of course is to obtain valid predictions for test patients which are not part of the training data without an expensive retraining of the model.

In the next subsections we will describe the Temporal Latent Embedding models we have used in the experiments. In the next subsection we describe the model which is based on the complete patient history up to time t. Subsection III-C then describes a Markov model that is based only on a recent history and Subsection III-D describes a combination of both.

B. Patient History Embedding

We define an aggregation tensor $\underline{\tilde{X}}$ with entries $\tilde{x}_{t,i,j}$. Here, $\tilde{x}_{t,i,j}$ is an aggregation of $\{x_{t',i,j}\}_{t'=1,\dots,t}$, i.e., of all events that happened to patient i up to time t. In the experiments we used different aggregation functions (see Section V). $\tilde{x}_{t,i,j}$ is supplemented with dimensions encoding background patient information such as age, gender and so on.

We then model

$$\theta_{t,i,j} = f_j(h^{\text{hist}}_{t-1,i}).$$

Here, $h^{\text{hist}}_{t,i}$ is an r-dimensional real vector that represents the embedding of patient i at time t, based on all information

observed for that patient until time t. We call r the rank of the embedding.

Since we want to apply the learned model easily to new patients, we assume that the embeddings can be calculated as a linear function of the events that are associated with patient i up to time t, with

$$h^{\text{hist}}_{t,i} = A\tilde{x}_{t,i,:}$$

where $\tilde{x}_{t,i,:}$ is a J-dimensional vector and $A \in R^{r \times J}$ is a matrix of learned weights. Thus $h^{\text{hist}}_{t,i}$ is a latent representation of the history of the patient i until time t. In a related but slightly different interpretation, we can also think of the j-th column of A as the latent embedding representation of event type j. As in other embedding approaches, the model has the ability to form similar latent representations for event types which have a similar semantics, i.e. for medications with comparable effects.

Note, that if $f_j(\cdot)$ is a linear map, we obtain a factorization approach, as used in collaborative filtering. In our experiments, the functions $f_j(\cdot)$ are nonlinear maps and are modelled by a multi-layer Perceptron (MLP) with J outputs, as also used in [12] and [13].

C. Markov Embeddings

In a K-th order Markov model, the events in the last K time steps are used to predict the event in the next time step. Markov models are used in language models where an event would correspond to an observed word [12]. Some of the leading approaches in computational linguistics [18], [19] are then using learned word embeddings to realize a number of applications and we will also pursue this approach in this paper.

More precisely, our model is

$$\theta_{t,i,j} = f_j(h^{\text{Mar}}_{t-1,i}, h^{\text{Mar}}_{t-2,i}, \ldots, h^{\text{Mar}}_{t-K,i}).$$

Note that in this model $h^{\text{Mar}}_{i,t}$ is an an r-dimensional embedding of all the observed events for patient i at time t. Note also that, in contrast to the situation in language models, several events can happen at the same time.

As before, we assume that there is a linear map of the form

$$h^{\text{Mar}}_{t,i} = Bx_{t,i,:}$$

where $x_{t,i,:}$ is a J-dimensional vector that contains all observed events for patient i at time t.

We can think of $h^{\text{Mar}}_{t,i}$ as the latent representation of patient i at time t based on all events that happened to the patient at time t. In contrast, $h^{\text{hist}}_{t,i}$ was the presentation of all events that happened to patient i until time t.

Again the $j-th$ column of B is representing latent embedding of event type j. The overall architecture in shown in Figure 1.

D. Personalized Markov Embeddings

The Markov model so far is independent of the individual patient history but it makes sense to assume that this history would be relevant for predicting events. Thus, we include $h^{\text{hist}}_{t,i}$ in the Markov model in the form

$$\theta_{t,i,j} = f_j(h^{\text{hist}}_{t,i}, h^{\text{Mar}}_{i,t-1}, h^{\text{Mar}}_{i,t-2}, \ldots, h^{\text{Mar}}_{i,t-K}).$$

The overall architecture is shown in Figure 2.

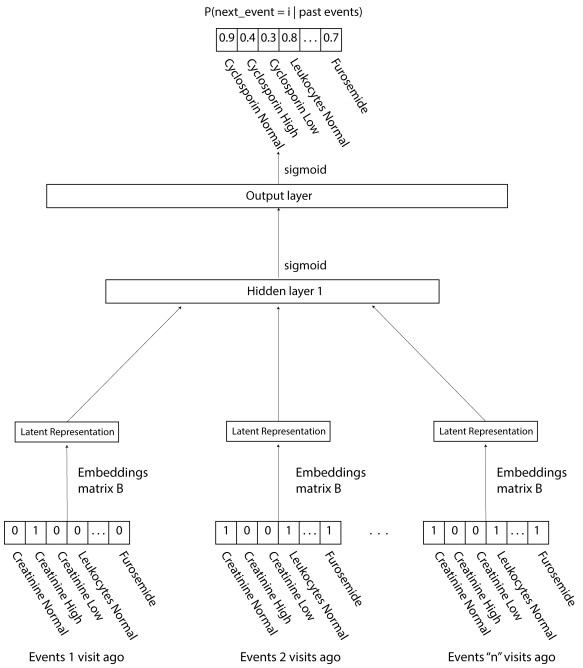

Fig. 1. Markov embedding model for predicting sequences of clinical events by taking the previous time steps as inputs.

E. Modelling the Function

In the language models of [12], $f_j(\cdot)$ was modelled as a standard multi-layer Perceptron neural network (MLP) with one hidden layer. A similar representation was used in modelling knowledge graphs as described in [13]. We use the same MLP structure here, where we also experimented with different numbers of hidden layers. In the following, the set of all MLP parameters is denoted by $W = \{w\}$.

F. Cost Function

We derive a cost function based on the Bernoulli likelihood function, also known as Binary Cross Entropy, which has the form:

$$\text{cost}(A, B, W) =$$

$$\sum_{t,i,j \in \text{Tr}} -x_{t,i,j} \log(\text{sig}(\theta_{t,i,j})) - (1 - x_{t,i,j}) \log(1 - \text{sig}(\theta_{t,i,j}))$$

$$+\lambda_w \sum_{w \in W} w^2 + \lambda_a \sum_{l=1}^{r} \sum_{j=1}^{J} a_{i,j}^2 + \lambda_b \sum_{l=1}^{r} \sum_{j=1}^{J} b_{i,j}^2$$

Note that we added regularization terms to penalize large MLP parameters w and large embedding parameters $a_{i,j}$ and $b_{i,j}$. Here, λ_w, λ_a, and λ_b are regularization parameters. Tr stands for the training data set and sig is the sigmoid function.

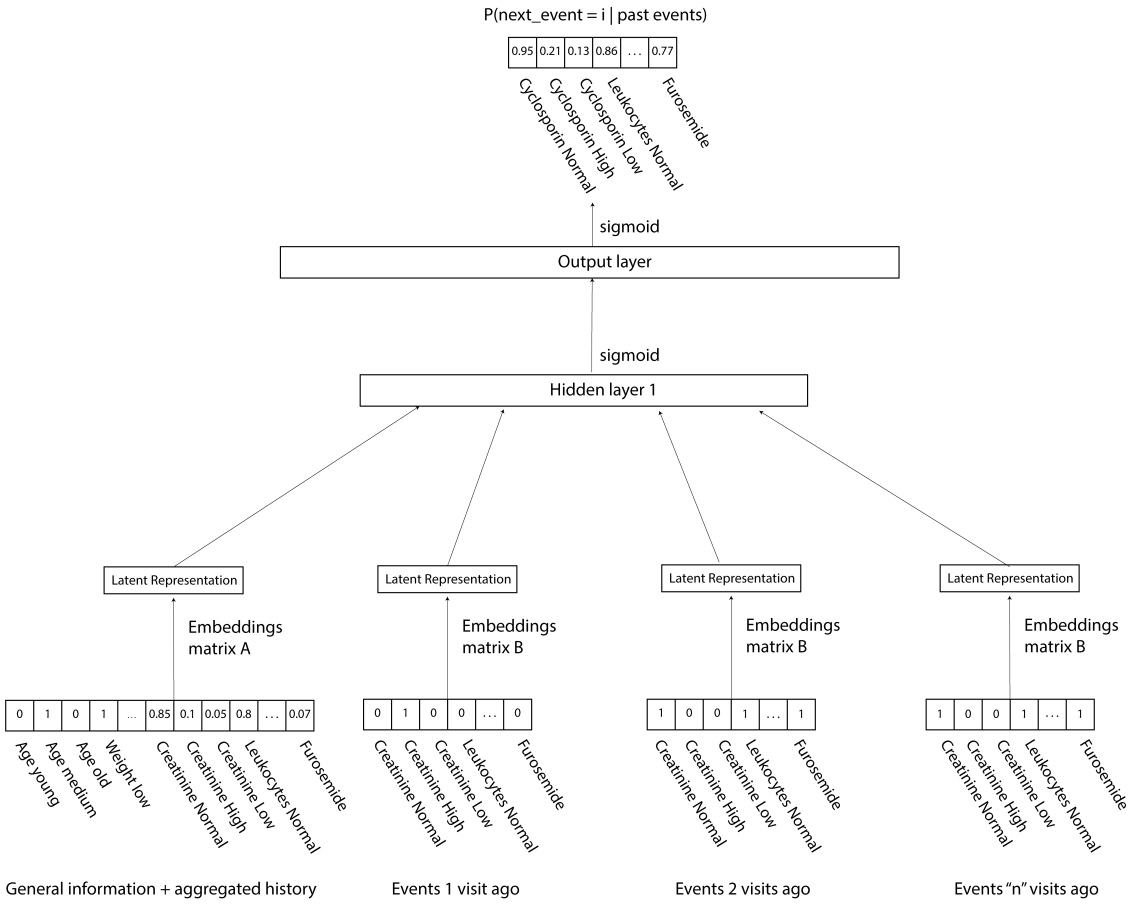

Fig. 2. Personalized Markov embedding model. It predicts the observed events within the next time step given the patient history and the previous time steps as inputs.

IV. The Use Case

A. Kidney Diseases and their Treatments

Kidney diseases are causing a significant financial burden on all health systems worldwide. Here we describe the situation in Germany. It is estimated that alone the treatment of end-stage renal disease (ESRD) with chronic renal replacement therapies accounts for more than 2.5 billion Euros annually, and the incidence of dialysis-dependent renal insufficiency is rising by 5-8% each year [20]. Despite progress in diagnosis, prophylaxis and therapy of chronic kidney diseases, renal transplantation remains the therapy of choice for all patients with ESRD. Kidney transplantation leads to a significant improvement of quality of life, to substantial cost savings and most importantly to a significant survival benefit in relation to all other renal replacement therapies. Only approximately 2300 kidney transplantations were performed in Germany in 2013 but more than 8000 patients are registered on the waiting list for a kidney transplant [21]. With excellent short term success rates, nowadays the reduction of complications and the increase of long-term graft survival are the main goals after transplantation, especially on the background of the dramatic organ shortage. It is not only important to reduce - or better avoid - severe and/or life-threatening complications such as acute rejection, malignancy and severe opportunistic infections, but it is also of utmost importance to ameliorate the many other serious side effects, which increase cardiovascular risk, decrease renal function, necessitate costly co-medication or hospitalisations and also have an impact on the quality of life after successful transplantation.

Despite the fact that renal transplantation is much cheaper than regular dialysis treatment it is a complex and costly procedure. Due to the outlined complexities, patients should remain in life-long specialized posttransplant care. Patients have not only to take immunosuppressants, but also have to take numerous drugs for prophylaxis and treatment of pre-existing and/or concomitant diseases, which are at least in part aggravated by the immunosuppressants. As a consequence most patients have to take 5-10 different medications every day during their entire life. The many drugs and the multiple side effects of the routinely administered medication are causing a substantial cost burden. There is not only a medical need but also a financial necessity to reduce side effects, diagnostic procedures, therapeutic interventions, hospitalisations and ultimately improve patient safety. This will directly lead to a better quality of life, cost savings and better allocation of medical resources.

B. Relevance of Event Modelling

The long-term goal of the research described in this paper is to improve patient treatment by, e.g., prescribing the most

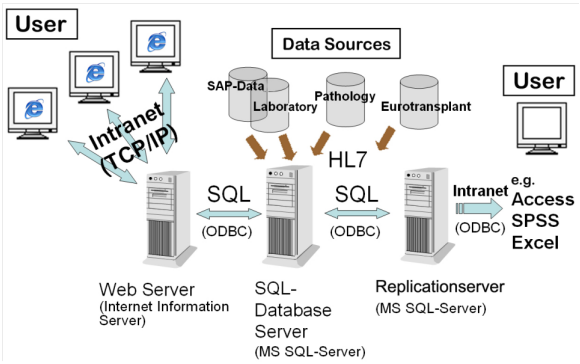

Fig. 3. TBase®architecture

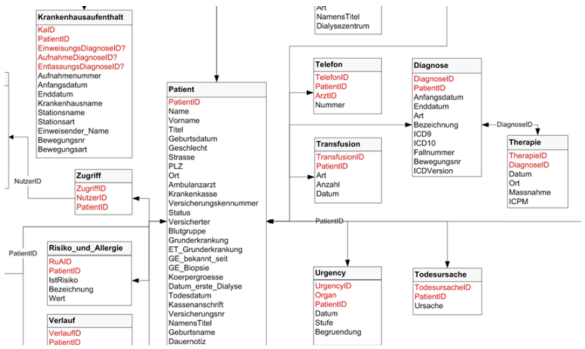

Fig. 4. View on the TBase®Schema

effective drugs to the patient my minimizing side effects. Particularly in focus are drug-drug interaction (DDI) and adverse drug reactions (ADR) in patients after renal transplantation. Of high interest are the effects of decisions on key outcome parameters such as patient and graft survival, renal function as well as hospitalisations. Lastly, the goal is to implement a clinical decision support system directly into the electronic patient file, in order to prevent dangerous DDI, reduce dosing errors and provide the physician and patient with timely and adequate information on new prescriptions.

C. TBase®

In close collaboration with the department of Artificial Intelligence of the Humboldt University, the Charité - Universitätsmedizin Berlin developed an electronic patient record (TBase®) for renal allograft recipients in 1999. The main idea was to combine a database for the daily patient care on the one hand with a validated database for medical research of the other hand. The combination of daily medical routine with a research database was the key concept, in order to collect data of high quality, which are constantly validated by the user. Due to clinical needs only accurate and reliable data can be used in daily routine practice. By this means, we have created a continuous internal validation process and almost completely avoid missing data. Since 2000 TBase®is used in the clinical routine of the Charité and all relevant patient data is automatically transferred. Due to the increase of the options of medical diagnostics, the extent of the information of the clinical data has also increased dramatically. The elaborate and flexible structure (see Figure 3) of the patient record and the database made it possible to integrate a large number of electronic data of several subsystems with different data structures over the years. Currently TBase®automatically integrates essential data from the laboratory, clinical pharmacology, nuclear medicine, findings from radiology and administrative data from the SAP-system of the Charité. TBase®is now under patronage of Deutsche Transplantationsgesellschaft (DTG) and Eurotransplant, Leiden, The Netherlands, and was implemented in 8 German transplant centres. Figure 4 provides an impression of the schema of TBase®.

V. Experiments

A. Setup of the Experiments

The data contains every event that happened to each patient concerning the kidney failure and all its associated effects, including prescribed medications, hospitalizations, diagnosis, laboratory tests and so on. In this paper we will consider events from year 2005 and onwards due to the improvement of the data quality from that year. Also, in order to have a better control of the experiments, we will work with a subset of the variables available in the dataset. Specifically, we will try to model three aspects of the patient evolution:

1) Medication prescriptions: which medications are prescribed in each situation.
2) Ordered laboratory tests: which laboratory tests are ordered in each situation.
3) Laboratory test results: which will be the outcome of the ordered laboratory tests.

Each entry in the database is labelled with the date in which the event happened. Our task will consist in predicting all the events that will happen to a patient on his or her next visit to the clinic given his past visits, as illustrated in Figure 5.

A very common situation is that the patient gets some laboratory tests done during the morning, and then based on the results of those tests, the doctor prescribes some medications to the patient in the afternoon. Therefore, we can define a second type of experiment by only considering days that have both laboratory tests performed and medications prescribed, and assuming that the laboratory tests always happen before the medications. Specifically, we will try to predict which will be the medications prescribed in the afternoon given the results of the laboratory tests performed in the morning and the events that happened in the previous visits. This way we can see how the model behaves in intra-day predictions. Figure 6 shows a representation of the experiment.

After selecting the subset of the dataset that we will use and performing the binary encoding, our pre-processed dataset consists of a table where each row represents one visit to the clinic. Each of these rows belongs to a patient, has an associated date and contains all the events that occurred during

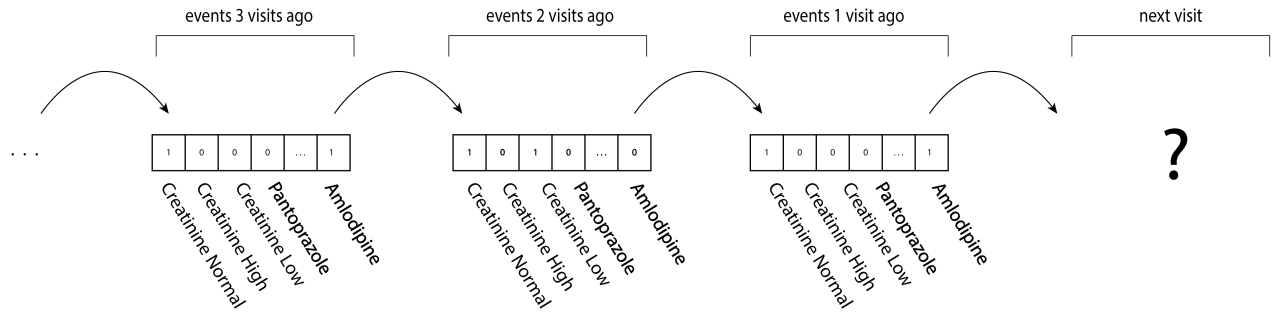

Fig. 5. Full visit predictions. We predict all the events that will happen within the next visit given the previous visits.

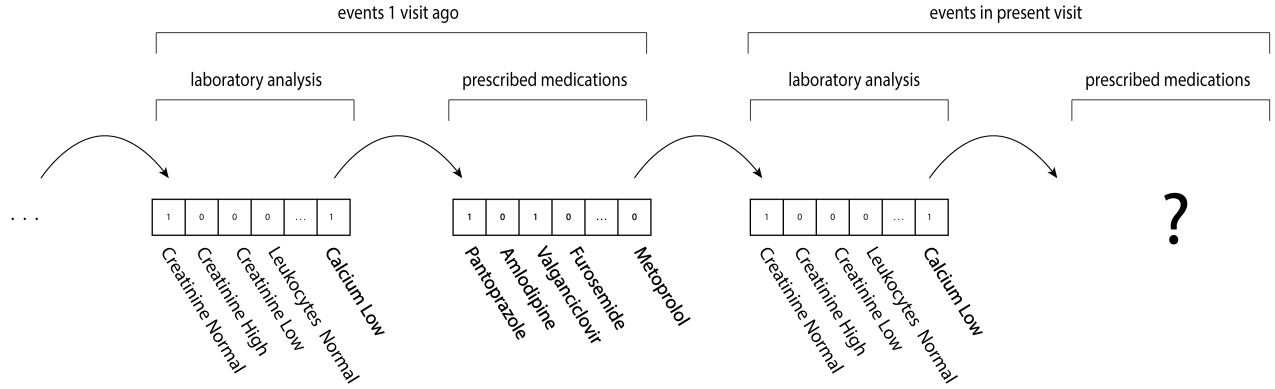

Fig. 6. Intra-day predictions. We predict the medications that will be prescribed in the afternoon given the laboratory analysis that were performed in the morning and the previous visits.

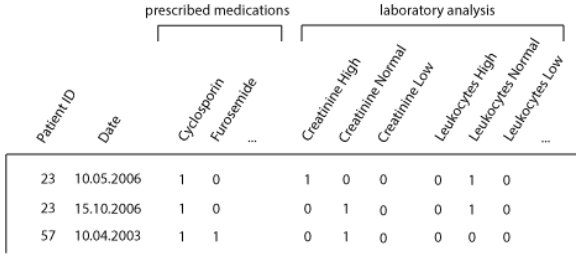

Fig. 7. Example of pre-processed data.

that visit in binary format. An example of how our pre-processed data look like can be found in Figure 7.

B. Hyperparameter Fitting

The model contains several hyperparameters that need to be optimized, being the most relevant ones the rank r of the embeddings, the order of the Markov model K, the number of hidden units in the Neural Network, the learning rate and the regularization parameters. In order to fit these hyperparameters, we randomly split the data into three subsets: 60% of the patients were assigned to the training set with totally about 100 thousand visits, 20% of the patients were assigned to the validation set and another 20% to the test set, with approximately 33 thousand visits each. Note that, under this configuration, we evaluate the performance of the model by predicting the future events of patients that the model has never seen before, and therefore increasing the difficulty of the task.

In Figure 8 we can see how the area under the Precision-Recall curve on the validation set improves as we increase the order of the Markov model K. We observe that the performance stabilizes with an input window of size six. A 6-th order Markov model (without the personalization) has around 28 thousand inputs (4666 input events multiplied by 6 time steps). The number of outputs of the Neural Network is 2383, i.e. 2383 events are predicted.

C. Baseline Models

We will compare the performance of our model with various classic Machine Learning algorithms. Specifically, our baseline models will be: Naive Bayes classifier, K-nearest neighbor classifier and Logistic Regression. Additionally, we will also use what we named "constant predictor", which consists in predicting always for each event the occurrence rate of such event (thus the most common event is given the highest probability of happening, followed by the second most common event, and so on). Random Forests were also considered to be included in this work, but after some trials they were discarded due to the excessive amount of time they required to be trained with this dataset, due to the large number of events to be predicted (nevertheless in the few experiments we performed with them, they never got to outperform our

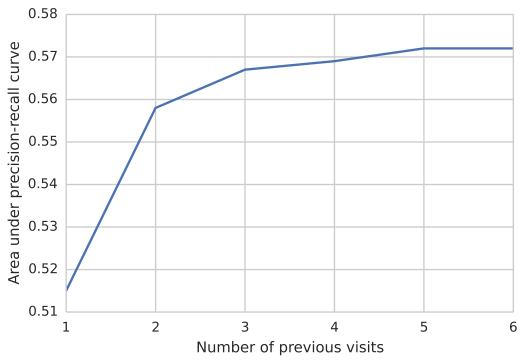

Fig. 8. Area Under the Precision Recall Curve improves as we increase the number of past visits (order of the Markov model K) used to predict the events that will be observed in the next visit.

proposed models). When comparing the performance between these models, we report for each model the mean area under the Precision-Recall curve (AUPRC) and mean area under Receiver Operating Characteristics curve (AUROC) together with their associated standard errors after repeating each experiment ten times with different random splits of the data. We made sure that these ten random splits were identical for each model. Most of these baseline models were taken from Scikit-learn [22], which is the main open source machine learning library for the Python programming language.

D. Model Training and Evaluation

We trained the proposed models by using mini-batch Adaptive Gradient Descent (AdaGrad) [23] combined with an early stopping strategy and using a mini-batch size of 128 samples. Our main goal will be to maximize the area under the Precision-Recall curve (AUPRC) of our predictions. We chose this score due to the high sparsity of the data (the density of ones is around 1%) and because we are mainly interested in predicting the very small amount of events that will happen, as opposed to the task of predicting which events will not be observed. Nevertheless, we will also report the area under Receiver Operating Characteristics curve (AUROC) because it is often reported in related scenarios.

The proposed models were implemented in Theano [25], [24], which is a graph-based computation library, especially well suited for training Neural Networks. The experiments were conducted using a Intel(R) Xeon(R) CPU E7-4850 v2 processor with 1TB of RAM and 48 cores at 1.2 Ghz with 2 threads per core. The reported computation times were all achieved using one thread.

E. Full Visit Predictions

As explained earlier in this section, our first experimental setting consists in predicting all the events that will happen to the patients during their next visit to the clinic given the events that were observed in their previous visits, as it is illustrated in Figure 5.

Therefore, we predict the events that will happen to a patient in her or his next visit to the clinic given the events that were observed in her or his six previous visits to the clinic, i.e. $K = 6$. Table I shows the results obtained after repeating the experiments with ten different random splits of the data. We can see that the Markov embedding model, which corresponds to the architecture shown in Figure 1, outperforms all our baseline models. Our proposed Markov embedding model obtained an AUPRC score of 0.574, being Logistic Regression the second best model with an AUPRC score of 0.554. We can also see how the random predictor achieved a very low AUPRC score due to the high sparsity of the data, which means that optimizing the AUPRC for this dataset is a hard task.

In the last column of Table I we also report the time that it took to train for each model with the best set of hyperparameters in the first random split. Note that one of the advantages of the proposed model is that the rank of the embeddings matrix B can always be reduced in order to decrease the computational cost required to train the model. Besides, given constant hyperparameters, the parameters of the model will increase linearly with the amount of different event types present in our dataset (e.g. number of medications, number of diseases...), whereas the parameters of other models such us the Logistic Regression will grow quadratically in this situation since for every additional event that we include we are adding both one input and one output.

TABLE I. SCORES FOR FULL VISIT PREDICTIONS. AUPRC STANDS FOR AREA UNDER PRECISION-RECALL CURVE. AUROC STANDS FOR AREA UNDER ROC CURVE.

	AUPRC	AUROC	Time (hours)
Markov Embeddings	0.574 $\pm$ 0.0014	0.977 $\pm$ 0.0001	6.11
Logistic Regression	0.554 $\pm$ 0.0020	0.970 $\pm$ 0.0005	4.31
KNN	0.482 $\pm$ 0.0012	0.951 $\pm$ 0.0002	17.74
Naive Bayes	0.432 $\pm$ 0.0019	0.843 $\pm$ 0.0015	39.1
Constant predictions	0.350 $\pm$ 0.0011	0.964 $\pm$ 0.0001	0.001
Random	0.011 $\pm$ 0.0001	0.5	-

We repeated the same experiment with the Personalized Markov Embedding model as represented in Figure 2. The additional information that we input to the model is composed of the aggregated history and general information of each patient. In order to create the aggregated history, for each sample that we input to the model we create a vector composed of the sum of all the events that are recorded for that particular patient until the date of the visit we want to predict. Our experiments showed that instead of directly using this count of the data as long term memory, we have two options that work better. The first option consists in computing the frequency of appearance of each event by dividing each row of the memory by the number of visits used to make the count. The second option consists in normalizing the count between 0 and 1. We will use both the appearance frequency of each event and the normalized count as our long term memory. Regarding the background information, it is composed of static or slow changing variables that we also converted to a binary format. Specifically, the background information is composed of the following variables: age, gender, blood type, time from first dialysis, time from the first time the patient was seen, weight and primary disease. We can see in Table II how the personalization of the Markov embedding model improved its performance. During our experiments, we observed that among all the variables that compose the additional information used

in this experiment, the inclusion of the frequency of appearance of each event is the factor that contributed most to the improvement of the performance of the model. Last row in II shows the performance of the model when making the predictions using just the aggregated patient history as input, as described in Section III-B.

TABLE II. SCORES FOR FULL VISIT PREDICTIONS WITH AND WITHOUT LONG TERM MEMORY AND BACKGROUND INFORMATION. AUPRC STANDS FOR AREA UNDER PRECISION-RECALL CURVE. AUROC STANDS FOR AREA UNDER ROC CURVE.

	AUPRC	AUROC
Personalized Markov embeddings	0.584 ± 0.0011	0.978 ± 0.0001
Markov embeddings	0.574 ± 0.0014	0.977 ± 0.0001
Patient history embedding	0.487 ± 0.0016	0.974 ± 0.0002

Regarding the architecture of the personalized Markov embedding model, we also tested the option of having just one embeddings matrix shared between the long term memory and the visits within the time window, i.e. $A = B$, but we found that the best strategy for our use case is to use separate embeddings matrix for the long term memory and the background information as it is shown in Figure 2.

We also tried to initialize the embedding matrices by using an autoencoder. This brought a speed up of around 30% to the optimization process of the model. However, this advantage vanished when we considered both the model optimization time and the training time of the autoencoder.

F. Intra-day Predictions

Our second experiment type consists in predicting which medications will be prescribed in the afternoon given the results of the laboratory tests performed in the morning and the events that happened in the six previous visits. Figure 6 shows a representation of the experiment. The architecture of the model will be similar to the one for the Markov embedding model (Figure 1), but including one more time step in the input window that will contain the information regarding all the observed events in the present day. Therefore, for this experiment the order of the Markov model K will be equal to seven, instead of six as it was in the case of full visit predictions. We can see the result of the experiment in Table III, which shows that also in this setting our proposed model outperforms the baseline models. The Markov embedding model for intra-day predictions achieved an AUPRC score of 0.277, which is lower than the score achieved when doing full visit predictions because the dataset is even more sparse when we only take into account the medications. Logistic Regression is again the second best result, and we can also observe how in this case the performance of the constant predictor is almost as bad as the random predictor, which means that this is even a harder task than the full visit predictions.

Another interesting experiment is to compare this result with the one obtained when doing full visit predictions. That is, we will measure the performance of predicting medication prescriptions both considering the laboratory tests performed in the same day and not considering them. Table IV shows that incorporating intra-day information actually improves the performance of the predictions.

TABLE III. SCORES FOR INTRA-DAY PREDICTIONS. AUPRC STANDS FOR AREA UNDER PRECISION-RECALL CURVE. AUROC STANDS FOR AREA UNDER ROC CURVE.

	AUPRC	AUROC
Markov embeddings intra-day	0.277 ± 0.0026	0.935 ± 0.0007
Logistic Regression intra-day	0.238 ± 0.0041	0.916 ± 0.0014
KNN	0.184 ± 0.0027	0.873 ± 0.0002
Naive Bayes intra-day	0.231 ± 0.0013	0.686 ± 0.0020
Constant predictions intra-day	0.008 ± 0.0013	0.564 ± 0.0064
Random intra-day	0.006 ± 0.0064	0.5

TABLE IV. SCORES FOR INTRA-DAY PREDICTIONS WITH AND WITHOUT CONSIDERING THE PRESENT DAY. AUPRC STANDS FOR AREA UNDER PRECISION-RECALL CURVE. AUROC STANDS FOR AREA UNDER ROC CURVE.

	AUPRC	AUROC
Markov embeddings intra-day	0.277 ± 0.0026	0.935 ± 0.0007
Markov embeddings	0.250 ± 0.0022	0.931 ± 0.0006

Besides, as we did with full visit predictions, we will make intra-day predictions incorporating a long term memory and background information of the patients. Table V shows how we improved the performance of the predictions with the personalized Markov embedding model.

TABLE V. SCORES FOR INTRA-DAY PREDICTIONS WITH AND WITHOUT MEMORY AND BACKGROUND INFORMATION. AUPRC STANDS FOR AREA UNDER PRECISION-RECALL CURVE. AUROC STANDS FOR AREA UNDER ROC CURVE.

	AUPRC	AUROC
Personalized Markov embeddings intra-day	0.289 ± 0.0027	0.938 ± 0.0005
Markov embeddings intra-day	0.277 ± 0.0026	0.935 ± 0.0007

G. Sensitivity Analysis

We performed a sensitivity analysis in order to evaluate how the model reacts to changes in the inputs. We performed this analysis using the medication named Tacrolimus, because it is one of the main immunosuppressants used in our database but it is not as frequent as other immunosuppressants such as Cyclosporin.

When doing the intra-day predictions as illustrated in Figure 6, and if we look exclusively at the score obtained in the prediction of Tacrolimus prescription (i.e. predicting whether or not Tacrolimus prescription will be observed next), we obtain an AUPRC score of 0.629, whereas the random prediction score 0.160. The sensitivity analysis will consist in suppressing one by one the events in the input and check how the absence of such input affects to the AUPRC score.

After performing this analysis we rank our input variables according to how much the AUPRC score of predicting Tacrolimus prescription was degraded when suppressing each of them. Even though this is a simplified analysis since we do not analyze how each variable influences the output when combined with other variables, we can infer that the higher a variable is ranked, the higher is the importance that has been assigned to it by our model for this task.

Most of the prescriptions of Tacrolimus present in our database correspond to an increase or decrease of the amount of medication that a patient is taking. The dosage of Tacrolimus that a patient takes has to be adjusted when certain criteria are met. The factors that the physicians take into account to decide whether or not the dosage of Tacrolimus has to be changed are the amount of Tacrolimus in blood and the excess of Creatinine in blood. Out of almost 5000 events, the laboratory results for "Low Tacrolimus", "High Tacrolimus" and "Normal Tacrolimus" occupy the positions second, third and fourth respectively in our sensitivity ranking. The laboratory result of "High Creatinine" occupies the position number 27. Therefore we can see how the model has learnt to predict the prescription of Tacrolimus giving a very high importance to the same observations that the physicians use. Moreover, other factors that are also correlated with the prescription of Tacrolimus are also present in the top 10 entries of the sensitivity ranking. For example in the position number 8 we find "High C-reactive protein", which is an infection marker that, when observed, indicates that the Tacrolimus dosage has to be reduced. Also in position 10 we find "High Glucose" which is a side effect of Tacrolimus that often leads to the reduction of the Tacrolimus dosage.

VI. Future Work

We will try to improve the model by introducing other elements that proved to be successful in deep Neural Networks such as drop out regularization and temporal convolutional layers. We will also explore the possibility of including additional information in the model such as the size of the time gap between the visits.

Besides, [19] showed that Recurrent Neural Networks provide the best performance in the task of language modelling. Therefore we will explore such models for our use case.

Regarding the data, we will extend our model to predict more event types within this dataset, and we will also apply our model to other datasets and use cases.

Our project also serves to encourage the TBase system to collect more information that would be valuable for decision support, such as patient symptoms and a precise time stamp for each event. Future work will also include the incorporation of textual information as present in pathology reports and information from molecular tests, e.g., genetics. Finally, we plan to make more extensive use of background ontologies which for example can be used to map different medications with identical active components to a common representation.

VII. Conclusion

We presented a model capable of predicting clinical events that is scalable and provides an acceptable performance for our current use case, which consist of modelling a subset of the variables that compose the evolution of the patients in our dataset.

Our work already lead to new requirements for improving the medical documentation. For example a detailed documentation of the patients symptoms would be a very valuable information for improving the model.

We showed how the proposed model performed better than our baseline models both making full visit predictions and intra-day predictions. We also showed how to integrate both the background information of each patient and a long term memory in order to improve the performance of the model.

Our model currently predicts common practice in a clinic which can already be useful in many ways, for example in alerting staff in case of unusual decisions. Of course the ultimate goal of a clinical decision support system should be not just replicating the decisions that are most often taken by the physicians in each situation, but to provide recommendations that lead to the best outcome possible. The basis for achieving this goal is a predictive model as presented in this paper.

Acknowledgments

This work was supported by the European Union 7th Framework Programme through the Marie Curie Initial Training Network Machine Learning for Personalized Medicine MLPM2012, Grant No. 316861. We also acknowledge support by the German Federal Ministry of Economics and Technology under the program "Smart Data", grant number 01MT14001.

References

[1] Charles Safran, Meryl Bloomrosen, Edward Hammond, Steven Labkoff, Suzanne Markel-Fox, Paul C. Tang, Don E. Detmer. Toward a National Framework for the Secondary Use of Health Data: An American Medical Informatics Association White Paper. Journal of the American Medical Informatics Association, 2007.

[2] Travis B. Murdoch, Allan S. Detsky. The Inevitable Application of Big Data to Health Care JAMA, 2013.

[3] Volker Tresp, Sonja Zillner, Maria J. Costa, Yi Huang, Alexander Cavallaro, Peter A. Fasching, Andre Reis, Martin Sedlmayr, Thomas Ganslandt, Klemens Budde, Carl Hinrichs, Danilo Schmidt, Philipp Daumke, Daniel Sonntag, Thomas Wittenberg, Patricia G. Oppelt, and Denis Krompass. Towards a New Science of a Clinical Data Intelligence. NIPS 2013 Workshop on Machine Learning for Clinical Data Analysis and Healthcare, 2013.

[4] Sebastian Schneeweiss. Learning from Big Health Care Data. N Engl J Med, 2014.

[5] The Resource Description Framework (RDF) as a modern Structure for Medical Data Gabriela Lindemann, Danilo Schmidt, Thomas Schrader, Dietmar Keune World Academy of Science, Engineering and Technology International Journal of Medical, Health, Biomedical and Pharmaceutical Engineering Vol:1, No:7, 2007.

[6] K. Schrter, G. Lindemann, L. Fritsche: TBase2 A web-based Electronic Patient Record. Fundamenta Informaticae 43, 343-353, IOS Press, Amsterdam, 2000.

[7] Huber L., Naik M., Budde K., Desensitization of HLA-incompatible kidney recipients. N Engl J Med. 27;365(17):1643, Oct 2011.

[8] Budde K. et al., Everolimus-based, calcineurin-inhibitor-free regimen in recipients of de-novo kidney transplants: an open-label, randomised, controlled trial. The Lancet. 5;377(9768):837-47, Mar 2011.

[9] Budde K. et al., Conversion from cyclosporine to everolimus at 4.5 months posttransplant: 3-year results from the randomized ZEUS study. Am J Transplant. 12(6):1528-40, Jun 2012.

[10] Bissler JJ, Kingswood JC, Radzikowska E, Zonnenberg BA, Frost M, Belousova E, Sauter M, Nonomura N, Brakemeier S, de Vries PJ, Whittemore VH, Chen D, Sahmoud T, Shah G, Lincy J, Lebwohl D, Budde K. Everolimus for angiomyolipoma associated with tuberous sclerosis complex or sporadic lymphangioleiomyomatosis: a multicentre, randomised, double-blind, placebo-controlled trial. The Lancet, 2012

[11] Yehuda Koren, Robert M. Bell and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems, IEEE Computer, 2009.

[12] Yoshua Bengio, Réjean Ducharme, Pascal Vincent and Christian Jauvin, A Neural Probabilistic Language Model, Journal of Machine Learning Research, 3(1137–1155), 2003.

[13] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs: From Multi-Relational Link Prediction to Automated Knowledge Graph Construction. To appear in the Proceedings of the IEEE, (invited paper), 2015.

[14] MW. Lorenz, HS. Markus, ML. Bots, M. Rosvall, M. Sitzer, Prediction of Clinical Cardiovascular Events With Carotid Intima-Media Thickness. Circulation in American Heart Association, 2006.

[15] DH. O'Leary, JF. Polak, Intima-media thickness: a tool for atherosclerosis imaging and event prediction. The American Journal of cardiology, 2002.

[16] F. Wang, N. Lee, J. Hu, J. Sun, S. Ebadollahi, Towards heterogeneous temporal clinical event pattern discovery: a convolutional approach. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012.

[17] Denis Krompaß, Cristóbal Esteban, Volker Tresp, Martin Sedlmayr and Thomas Ganslandt. Exploiting Latent Embeddings of Nominal Clinical Data for Predicting Hospital Readmission. KI - Künstliche Intelligenz, December 2014.

[18] Ronan Collobert and Jason Weston and Léon Bottou and Michael Karlen and Koray Kavukcuoglu and Pavel P. Kuksa. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research, 2012.

[19] Mikolov Tomas. Statistical Language Models based on Neural Networks. PhD thesis, Brno University of Technology, 2012.

[20] Lysaght MJ. Maintenance dialysis population dynamics: current trends and long-term implications. J Am Soc Nephrol, 2002.

[21] Nierentransplantation, http://www.dso.de/organspende-und-transplantation/transplantation/ nierentransplantation.html.

[22] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. Scikit-learn: Machine Learning in Python, in Journal of Machine Learning Research 12 (2825–2830), 2011.

[23] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research, pages 2121-2159, 2011.

[24] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio. Theano: new features and speed improvements. NIPS deep learning workshop, 2012.

[25] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio. Theano: A CPU and GPU Math Expression Compiler. Proceedings of the Python for Scientific Computing Conference (SciPy), 2010.