Doctor AI: Predicting Clinical Events via Recurrent Neural Networks

Edward Choi[1], Mohammad Taha Bahadori[1], Andy Schuetz[2],
Walter F. Stewart[2], Jimeng Sun[1]

[1] Georgia Institute of Technology, [2] Sutter Health

Abstract

Large amount of Electronic Health Record (EHR) data have been collected over millions of patients over multiple years. The rich longitudinal EHR data documented the collective experiences of physicians including diagnosis, medication prescription and procedures. We argue it is possible now to leverage the EHR data to model how physicians behave, and we call our model *Doctor AI*. Towards this direction of modeling clinical behavior of physicians, we develop a successful application of Recurrent Neural Networks (RNN) to jointly forecast the future disease diagnosis and medication prescription along with their timing. Unlike traditional classification models where a single target is of interest, our model can assess the entire history of patients and make continuous and multilabel predictions based on patients' historical data. We evaluate the performance of the proposed method on a large real-world EHR data over 260K patients over 8 years. We observed Doctor AI can perform differential diagnosis with similar accuracy to physicians. In particular, Doctor AI achieves up to 79% recall@30, significantly higher than several baselines. Moreover, we demonstrate great generalizability of Doctor AI by applying the resulting models on data from a completely different medication institution achieving comparable performance.

1 Introduction

The broad adoption of Electronic Health Records (EHR) has continuously generated large amount of patient data that documents rich clinical interactions over time. This high-dimensional longitudinal data has created an opportunity to perform sophisticated temporal analysis that was not possible before. Forecasting clinical events for patients is an especially challenging, yet important task. Our goal is to develop a temporal prediction model that mimics physician practice based on the collective memory of many physicians, i.e., large amount of EHR data over a long period of time. Successfully forecasting clinical events can not only facilitate patient-specific care and timely intervention, but also potentially reduce healthcare cost.

Although related problems such as disease progression modeling have been studied by many researchers over several decades, e.g. [16, 6, 27], most works do not achieve required accuracy and scalability, or need excessive expert domain knowledge, partly due to the lack of rich longitudinal EHR data and scalable computational architecture. Thanks to the recent advances in recurrent neural network, we propose *Doctor AI* system that can diagnose multiple disease conditions and prescribe relevant medications based on historical EHR data. Furthermore, the Doctor AI tries to

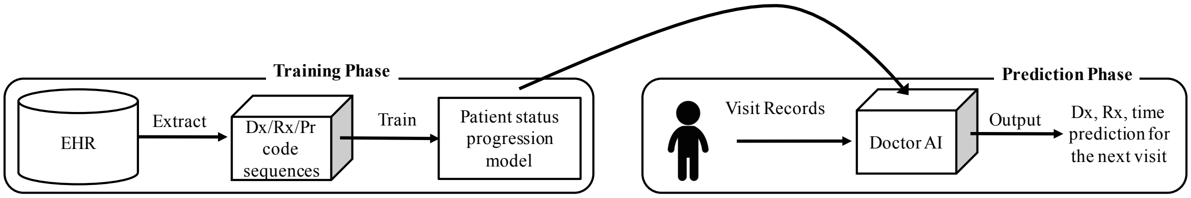


Figure 1: Doctor AI extracts clinical events as multilable point process data from EHR and learns a model for patient status dynamics. Given a new patient's record, it can forecast the patient's diagnoses (Dx), prescribed medications (Rx), and the time until his/her next visit.

predict when the patient will make the next visit. Our ultimate goal is to have Doctor AI help both health providers and patients.

The problem in general can be described as a multilabel marked point process modeling task. The task is different from common sequential learning tasks such as those in natural language processing as it requires prediction of multiple categories over the continuous time axis. The key challenge in this task is to find a flexible model that is capable of predicting multiple event types for patients. The two main classes of techniques, continuous-time Markov chain based models [36, 27, 21], and intensity based point process modeling techniques such as Hawkes processes [30, 54, 8] have been proposed but they are expensive to generalize to nonlinear and multilabel settings. Furthermore, they often make strong assumptions about the data generation process which might not be valid in large-scale EHR datasets.

The key idea of this paper is to learn an effective representation of the patient status over time and to leverage such representation to predict future clinical events of the patients such as diagnoses and medication prescriptions and their timings. To learn such patient representations we propose to use recurrent neural networks (RNN), considering the fact that patients have different length of medical records and that recurrent neural networks have been shown to be particularly successful for representation learning in sequential data, *e.g.* [13, 14, 45, 24, 51]. In particular, we make the following main contributions in this paper:

- We demonstrate a successful application of RNNs in representing the status of patients, predicting the future clinical events and the timing of the events. The trained RNN is able to achieve above 64% (79%) recall in its top 10 (30) predicted diagnosis codes, demonstrating great potential as a computerized differential diagnosis guide.

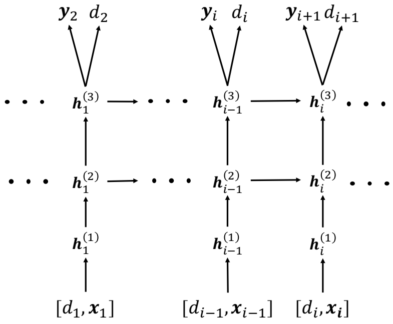
- We propose an efficient initialization scheme for RNNs using Skip-gram embedding [34] and show that it improves the performance of the RNN in both accuracy and speed.

- We empirically confirm that RNNs can be used to transfer knowledge across multiple medical institutions. This suggests that hospitals with insufficient patient records can adopt the models learned from larger data of other health institutions to improve the quality of their clinical service.

2 Methodology

This section describes the main proposed neural network model. After mathematically describing the problem as a multilabel point process modeling, we outline our approach for addressing the

Figure 2: This diagram shows how we have applied RNNs to solve the problem of joint forecasting of next visits' time and the codes assigned during each visit. The first layer simply embeds the high-dimensional input vectors in a lower dimensional space. The next layers are the recurrent units (here two layers), which learn the status of the patient at each timestamp as a real-valued vector. Given the status vector, we use two dense layers to generate the codes observed in the next timestamp and the duration until next visit.


$\boldsymbol{y}_2$ d_2
$\boldsymbol{y}_i$ d_i
$\boldsymbol{y}_{i+1}$ d_{i+1}
$\boldsymbol{h}_1^{(3)}$
$\boldsymbol{h}_{i-1}^{(3)}$
$\boldsymbol{h}_i^{(3)}$
$\boldsymbol{h}_1^{(2)}$
$\boldsymbol{h}_{i-1}^{(2)}$
$\boldsymbol{h}_i^{(2)}$
$\boldsymbol{h}_1^{(1)}$
$\boldsymbol{h}_{i-1}^{(1)}$
$\boldsymbol{h}_i^{(1)}$
$[d_1, \boldsymbol{x}_1]$
$[d_{i-1}, \boldsymbol{x}_{i-1}]$
$[d_i, \boldsymbol{x}_i]$

challenges. The proceeding sections are devoted to details of the model and the technical details of the learning procedure.

Problem setting. For each patient, we are given a sample of size n from a univariate multilabel marked point process in the form of $(t_i, \boldsymbol{x}_i)$ for $i = 1, \ldots, n$. Each pair represents an event, such as a hospital visit, during which multiple medical codes such as ICD-9 diagnosis codes, procedure codes, or medication codes are assigned to a patient. The multi-hot label vector $\boldsymbol{x}_i \in \{0,1\}^p$ represents the medical codes assigned at time t_i, where p denotes the number of unique medical codes. At each timestamp, we may extract higher-level codes for prediction purposes and denote it by $\boldsymbol{y}_i$, see the details in Section 4.1. The number of events for each patient may differ.

Description of neural network architecture. Our goal is to learn an effective vector representation for the status of patients at each timestamp t_i. Using the representation for the status of patients, we would be able to jointly predict different future quantities about this patient such as future diagnoses and medications $\boldsymbol{y}_{i+1}$ and the time duration until next event $d_{i+1} = t_{i+1} - t_i$. Finally, we would like to perform all these steps jointly in a single supervised learning scheme. As we discussed in the introduction, we use recurrent neural networks to learn such patient representations. We treat the state vector of RNNs as the latent representation for the patient status and use it for predicting multiple forms of outputs.

The proposed architecture is shown if Figure 2. The input at each timestamp t_i is the concatenation of the multi-hot coding $\boldsymbol{x}_i$ of the multilabel categories and the duration d_i since the last event. In our datasets, this input have as large as $40,000$ dimensions. Thus, the next layer maps into a lower dimensional space. Then, we pass the lower dimensional vector through units of an RNN. The RNN units can be simple RNN units [28] or more complex recurrent units such as Long Short-Term Memory (LSTM) [17, 15] or Gated Recurrent Units (GRU) [9]. We also can stack multiple units of RNN on top of each other to increase the representative power of the network. Finally, we use a softmax layer to predict the future codes and a rectified linear unit to predict the time duration until next event.

Details of the RNN. Specifically, we implemented our RNN with GRU. Although LSTM has drawn much attention from many researchers, GRU has recently shown to have similar performance as LSTM, while employing a simpler architecture [9]. In order to precisely describe the network used in this work, we reiterate the mathematical formulation of GRU as follows:

$$
\begin{aligned}
\boldsymbol{z}_i &= \sigma(\boldsymbol{W}_z\boldsymbol{x}_i + \boldsymbol{U}_z\boldsymbol{h}_{i-1} + \boldsymbol{b}_z)\\
\boldsymbol{r}_i &= \sigma(\boldsymbol{W}_r\boldsymbol{x}_i + \boldsymbol{U}_r\boldsymbol{h}_{i-1} + \boldsymbol{b}_r)\\
\tilde{\boldsymbol{h}}_i &= \tanh(\boldsymbol{W}_h\boldsymbol{x}_i + \boldsymbol{r}_i \circ \boldsymbol{U}_h\boldsymbol{h}_{i-1} + \boldsymbol{b}_h)\\
\boldsymbol{h}_i &= \boldsymbol{z}_i \circ \boldsymbol{h}_{i-1} + (1-\boldsymbol{z}_i) \circ \tilde{\boldsymbol{h}}_i
\end{aligned}
$$

where $\boldsymbol{z}_i$ and $\boldsymbol{r}_i$ respectively represent the update gate and the reset gate, $\tilde{\boldsymbol{h}}_i$ the intermediate memory unit, $\boldsymbol{h}_i$ the hidden layer, all at timestep t_i. For predicting the diagnosis codes and the medication codes at each timestep t_i, a Softmax layer is stacked on top of the GRU, using the hidden layer $\boldsymbol{h}_i$ as the input: $\widehat{\boldsymbol{y}}_{i+1} = \mathrm{softmax}({\boldsymbol{W}_{code}}^\top \boldsymbol{h}_i + \boldsymbol{b}_{code})$. For predicting the time duration until the next visit, a rectified linear unit (ReLU) is placed on top of the GRU, again using the hidden layer $\boldsymbol{h}_i$ as the input: $\widehat{d}_{i+1} = \max({\boldsymbol{w}_{time}}^\top \boldsymbol{h}_i + b_{time}, 0)$. The objective of training our model is to learn the weights $\boldsymbol{W}_{\{z,r,h,code\}}$, $\boldsymbol{U}_{\{z,r,h\}}$, $\boldsymbol{b}_{\{z,r,h,code\}}$, $\boldsymbol{w}_{time}$ and b_{time}. The values of all $\boldsymbol{W}$'s and $\boldsymbol{U}$'s were initialized to orthonormal matrices using singular value decomposition of matrices generated from the normal distribution [40]. The initial value of $\boldsymbol{w}_{time}$ was chosen from the uniform distribution between -0.1 and 0.1. All $\boldsymbol{b}$'s and b_{time} were initialized to zeros. The joint loss function consists of the cross entropy for the code prediction and the squared loss for the time duration prediction, as described below for a single patient:

$$
\mathcal{L}(\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{b}, \boldsymbol{w}_{time}, b_{time}) = \sum_{i=1}^{n-1} \left\{ \Big(\boldsymbol{y}_{i+1} \log(\widehat{\boldsymbol{y}}_{i+1}) + (1-\boldsymbol{y}_{i+1}) \log(1-\widehat{\boldsymbol{y}}_{i+1}) \Big) + \frac{1}{2} \| d_{i+1} - \widehat{d}_{i+1} \|_2^2 \right\}
$$

As mentioned above, the multi-hot vectors $\boldsymbol{x}_i$ of almost 40,000 dimensions are first projected to a lower dimensional space, then put into the GRU. We employed two different approaches for this: (1) We put an extra layer of a certain size between the multi-hot input $\boldsymbol{x}_i$ and the GRU, and call it the embedding layer. We denote the weight matrix between the multi-hot input vector and the embedding layer as $\boldsymbol{W}_{emb}$. Then we learn the weight $\boldsymbol{W}_{emb}$ as we train the entire model. (2) We initialize the weight $\boldsymbol{W}_{emb}$ with a matrix generated by Skip-gram algorithm [34], then refine the weight $\boldsymbol{W}_{emb}$ as we train the entire model. This can be seen as using the pre-trained Skip-gram vectors as the input to the RNN and fine-tuning them with the joint prediction task. The brief description of learning the Skip-gram vectors from the EHR is provided in Appendix A. The first and second approach can be formulated as follows:

$$
\boldsymbol{h}_i^{(1)} = [\tanh({\boldsymbol{x}_i}^\top \boldsymbol{W}_{emb} + \boldsymbol{b}_{emb}),\ d_i] \tag{1}
$$

$$
\boldsymbol{h}_i^{(1)} = [{\boldsymbol{x}_i}^\top \boldsymbol{W}_{emb},\ d_i] \tag{2}
$$

where $[\cdot,\cdot]$ is the concatenation operation used for appending the time duration to the multi-hot vector $\boldsymbol{h}_i^{(1)}$ to make it an input vector to the GRU.

3 Related Work

In this section, we briefly overview the common approaches to modeling multilabel event data with special focus on the models that have been applied to medical data.

Discretization vs Continuous-time modeling. There are two main approaches to modeling point process data: with or without discretization (binning) of time. When the time axis is discretized, the point process data can be converted to binary time series (or time series of count data if binning is coarse) and analyzed via time series analysis techniques [47, 1, 38]. However, this approach is inefficient as it produces long time series whose elements are mostly zero. Furthermore, discretization of time introduces noise in the time stamps of visits. Finally, these models are often not able to model the duration until next event. Thus, it is advantageous not to discretize the data both in terms of modeling and computation.

Continuous-time models. Among the continuous-time models, there are two main techniques: continuous-time Markov chain based models [36, 11, 21, 26, 31] and intensity function modeling techniques such as Cox and Hawkes processes [30, 53, 29, 8]. The latter has been shown to have computational advantages over the former. Moreover, modeling multilabel marked point processes with continuous-time Markov chains expands their state-space and make them even more expensive.

However, Hawkes processes only depend linearly on the past observation times; while there are limited classes of non-linear Hawkes process [54], the temporal dynamics can be more complex. Moreover, there is no scalable multi-label extension for Hawkes processes. Finally, Hawkes processes are known to have a flat loss function near optimal value of the parameters which renders the gradient-based learning algorithms inefficient [48]. In this paper we address these challenges by designing a recurrent neural network which has been shown to be successful in learning complex sequential patterns.

Health care. There have been active research in modeling the temporal progression of diseases [35]. Generally, most works can be divided into two groups: works that focus on a specific disease and works that focus on a broader range of diseases.

Specific-purpose progression modeling. There have been many studies that focus on modeling the temporal progression of a specific disease based on either intensive use of domain-specific knowledge [10, 19, 46] or taking advantage of advanced statistical methods [31, 20, 44, 52]. Specifically, studies have been conducted on Alzheimer's disease [19, 52, 44], glaucoma [31], chronic kidney disease [46], diabetes mellitus [10], and abdominal aortic aneurysm [20]

General-purpose progression modeling. Recently, [49, 8, 38] proposed more general approaches to modeling the progression of wider range of diseases. As discussed earlier, [8] used Hawkes process, and [38] discretized time in order to model multiple patients and multiple diseases. [49] proposed a graphical model based on Markov Jump Process to predict the stage progression of chronic obstructive pulmonary disease (COPD) and its co-morbid diseases.

One of the main challenges in using these algorithms is scalability. The datasets used in previous works typically contain a few thousands of patients and a few hundreds of codes. Even the largest dataset used by [38] contains 13,180 patients and 8,722 codes, which is significantly smaller than Sutter dataset described in Table 1. Need for domain-specific knowledge is also a big challenge. For example, [49] not only used a smaller dataset (3,705 patients and 264 codes) but also used co-morbidity information to improve the performance of their algorithm. Such expert knowledge is difficult to obtain from typical EHR data.

Table 1: Basic statistics of the Sutter Health clinical records dataset.

# of patients	263,706	Total # of codes	38,594
Avg. # of visits	54.61	Total # of 3-digit Dx codes	1,183
Avg. # of codes per visit	3.22	# of top level Rx codes	595
Max # of codes per visit	62	Avg. duration between visits	76.12 days

4 Experiments

In this section, we describe the details of our experiments, the datasets that we have used and the baselines. Throughout this section, we demonstrate the success of the proposed approach in forecasting the future events of the patients. We make the source code of Doctor AI publicly available at https://github.com/mp2893/doctorai.

4.1 Dataset description

We use a health records dataset provided by Sutter Health; its basic statistics are summarized in Table 1.

Population and source of data The source population for this study was primary care patients from Sutter Palo Alto Medical Foundation (PAMF) Clinics, a multispecialty group practice with large primary care practices that has used EHR for more than 8 years. The dataset was extracted from a case-control study for heart failure nested within Sutter-PAMF. The dataset consists of encounter orders, medication orders, problem list records and procedure orders. The data are fully de-identified and do not include any personal health information (PHI).

Data processing For input, we used diagnosis codes, medication codes, and procedure codes. Diagnosis codes, which are presented in the ICD-9 format, could be found in the encounter orders, medication orders, problem list records and procedure orders. Medication and procedure codes could be found in medication orders and procedure orders respectively. We extracted all diagnosis, medication and procedure codes from the dataset for each patient, and laid them out in a temporal order. If a patient received multiple codes in a single visit, those codes were assigned the same timestamp. By excluding patients that made less than two visits, we were left with 263,706 patients who made on average 54.61 visits per person.

Medical code grouping The number of ICD-9 diagnosis codes are approximately 11,000. The number is approximately 18,000 for medication codes. Many codes in this set are very granular; for example, pulmonary tuberculosis (ICD-9 code 011) is divided into 70 subcategories (ICD-9 code 011.01, 011.02, ..., 011.95, 011.96). In a practical perspective, however, simply knowing that a patient is likely to have pulmonary tuberculosis is enough to increase the doctor's awareness of the severity of the clinical situation. Therefore, in order to predict future diagnosis codes and medication codes,we group the codes into a higher-order codes to decrease the granularity among the codes that we predict. For the diagnosis codes, we use the 3-digit ICD-9 code system, where the the number of unique codes are 1,183 in our dataset. For the medication codes, we use the Sutter in-house medication grouper, which groups the medication codes into 595 unique codes. Therefore, the future code prediction problem reduces from approximately a 29,000 class classification to a 1,778 class classification. The $\boldsymbol{y}_i$ in Figure 2 is the 1,778-dimensional vector representing the grouped diagnosis codes and medication codes.

Training specifics. For training all models including the baselines, we used 85% of the patients as the training set and 15% as the test set. Since we trained the RNN models for 20 epochs without evaluating against the test set, it is very unlikely that the models were overfit. We used dropout between the GRU layer and the prediction layer (*i.e.* code prediction and time duration prediction). Dropout was also used between GRU layers if we were using a multi-layer GRU. We also applied norm-2 regularization on both $\boldsymbol{W}_{code}$ and $\boldsymbol{w}_{time}$. Both regularization coefficients were set to 0.001. The size of the hidden layer $\boldsymbol{h}_i$ of the GRU was set to 2000 to guarantee a sufficient expressive power. After running sets of preliminary experiments where we tried the size from 100 to 2000, we noticed that the code prediction performance started to saturate around 1600∼1800. All models were implemented with Theano [2] and trained on a machine equipped with two Nvidia Tesla K80 GPUs.

4.2 Evaluation metrics

We use the following metrics for evaluating the performance of the algorithms in predicting the codes and the time duration until next visit.

Top-*k* recall. In order to evaluate the performance of algorithms for forecasting next events, we use the top-k recall measure defined as follows:

$$\text{top-}k\text{ recall} = \frac{\#\text{ of true positives in the top } k \text{ predictions}}{\#\text{ of true positives}}$$

This metric is consistent with the differential diagnosis framework where the machine suggests k possible codes and we measure the fraction of true codes that are correctly retrieved. We choose $k = 10, 20, 30$ because as shown in Table 1 on average every visit includes more than three codes. Thus, selecting small k may result in inaccurate evaluation.

Top-k recall mimics the behavior of doctors conducting differential diagnosis, where doctors list most probable diagnoses and treat patients accordingly to identify the patient status. Therefore a machine with a high Top-k recall translates to a doctor with an effective diagnostic skill. This makes Top-k an attractive tool for evaluating the performance of the models for our problem.

We select the maximum k to be 30 in order to evaluate the performance of the models not only for simple cases but also for complex cases. Near 50.7% of the Sutter patients have been assigned with more than 10 diagnosis and medication codes at least once. Since it is those complex cases that are of interest to predict and analyze, we had to choose k to be large enough.

Coefficient of determination or R^2 is a metric for evaluation of predictive performance of regression and forecasting algorithms. It compares the accuracy of prediction with respect to simple prediction by mean of the target variable.

$$R^2 = 1 - \frac{\sum_i (y_i - \widehat{y_i})^2}{\sum_i (y_i - \overline{y_i})^2}$$

Given the fact that the time duration varies significantly over time and the interest in accurately predicting the durations decreases as the patients visit after a long period of time, we measure the R^2 performance of the algorithms in predicting $\log(d_i)$ to lower the impact of anomalous long durations in the performance metric. In the same spirit, we train all models to predict the logarithm of the time duration between visits.

4.3 Baselines

We compare our model against several baselines as described below. Some of the existing techniques based on CTMC and latent space models were not scalable enough to be trained in the entire dataset in a reasonable amount of time; thus comparison could not be fair.

Intuitive baselines. We compare our algorithms against simple baselines that are based on experts' intuition about the dynamics of events in clinical settings. The first baseline is to use a patient's medical codes in his last visit as the prediction for his current visit. This baseline is competitive when the status of a patient with a chronic condition stabilizes over time. We can make this baseline stronger, by using the top-k most frequent labels observed in visits prior to the current visits. In the experiments we observe that the second baseline is quite competitive.

Logistic and Neural Network time series models. A common way to perform prediction task is to use $(\mathbf{x}_{i-1}, d_{i-1})$ to predict the codes in the next event $\mathbf{x}_i$. We can use logistic regression or multilayer preceptron (MLP). To make this baseline stronger, we can use the data from L time lags before and concatenate the $(\mathbf{x}_{i-\ell}, d_{i-\ell})$ for $\ell = 1, \ldots, L$ to create the features for prediction of $\mathbf{x}_i$. Similarly, we can have a model that predicts the time until next event using rectified linear units as the output activation. While increasing the number of lags allows the model to capture longer history, it results in two disadvantages compared to RNNs: it increases the number of parameters of the model and also prevents the model to be used for prediction of the first L visits of the patients. Because of this limitation, we do not choose L to be bigger than 5 because the model loses its practicality for many patients with short visit history. Due to lack of space, we describe the details of MLP design in Appendix B.1.

Manual feature extraction. It is common in information retrieval to extract features from text data to improve the performance of learning algorithms. We define a set of *tf-idf* like features [32] based on the history of patients to not only provide potentially better features, but also give learning algorithms access to longer past information. The *idf* vector is calculated based on the entire dataset. To calculate the *tf* for each patient at each visit, we find the term frequencies in all visits until the current visit. We also generate five values based on the duration information: last, min, max, mean, and std of durations until the current visit. We concatenate the *tf-idf* vector and the duration features with the multihot vector of the current visit and use it in the learning algorithms, i.e., multilayer preceptron in our experiments. Further details of the feature extraction process are provided in Appendix B.2.

4.4 Results

Table 2 compares the results of different algorithms with RNN based Doctor AI. We report the results in three settings: when we are interested in (1) only predicting disease codes (Dx), (2) only medication codes (Rx), and (3) jointly predicting Dx, Rx, and time to next visit. The results confirm that the proposed approach is able to outperform the baseline algorithms by a large margin. Note that the recall values for the joint task are lower than those for single Dx or Rx prediction because the hypothesis space is larger for the joint prediction task. Comparing RNN-based and most frequent past pattern algorithm with the lagged multilayer perceptron algorithm, we postulate that the status of the patients in this dataset depends on more than 5 lags. This can be because this dataset is collected for study of heart failure which shows long-term dynamics.

The superior performance of RNN based approaches can be attributed to the efficient representation that they learn for patients at each visit [4, 41]. RNNs are able to learn succinct vector representations of patients by accumulating the relevant information from their history and the current set of codes. Given our network structure and the loss function, we can observe from the experimental results that the hidden layer of the RNN significantly outperforms at learning efficient representations for the patient status than ad-hoc feature engineering (*e.g. tf-idf* features used in the baseline models) such that a simple linear classifier can effectively predict the codes of the next

Table 2: Accuracy of algorithms in forecasting future medical activities.

	Dx Only Recall @k			Rx Only Recall @k			Dx,Rx,Time Recall @k			
Algorithms	$k = 10$	$k = 20$	$k = 30$	$k = 10$	$k = 20$	$k = 30$	$k = 10$	$k = 20$	$k = 30$	R^2
Last visit	29.17			13.81			26.25			—
Most freq.	56.63	67.39	71.68	62.99	69.02	70.07	48.11	60.23	66.00	—
Logistic (L=1)	22.97	32.20	36.58	28.01	39.75	43.79	17.66	26.12	31.23	0.0013
MLP (L=1)	26.09	39.19	48.04	32.27	51.12	61.50	19.49	30.80	38.13	0.0017
Logistic (L=5)	26.04	39.17	48.19	32.39	51.06	61.03	18.79	29.13	35.63	0.0013
MLP (L=5)	26.14	39.41	48.28	32.39	51.18	61.66	19.32	30.77	38.08	0.0002
Feature Ext.	26.12	39.33	48.20	32.27	51.12	61.65	19.60	30.80	38.13	0.0022
RNN-1	63.12	73.11	78.49	67.99	79.55	**85.53**	53.86	65.10	71.24	0.2519
RNN-2	63.32	73.32	78.71	67.87	79.47	85.43	53.61	64.93	71.14	0.2528
RNN-1-IR	63.24	73.33	78.73	**68.31**	**79.77**	85.52	54.37	65.68	71.85	0.2492
RNN-2-IR	**64.30**	**74.31**	**79.58**	68.16	79.74	85.48	**54.96**	**66.31**	**72.48**	**0.2534**

visit.

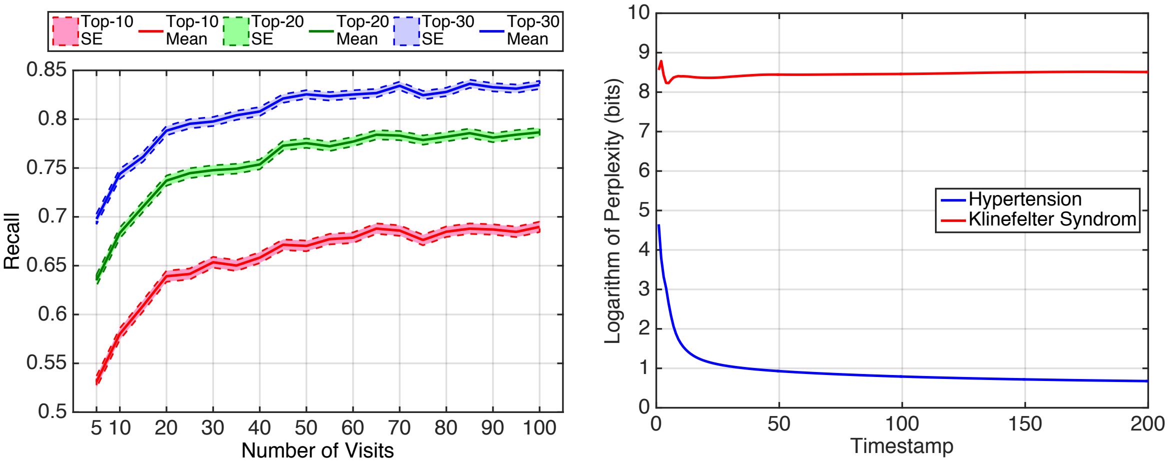
The results also suggest that increasing the number of lags in the logistic and MLP approaches may not significantly improve the performance. Given the size of the dataset, this can be due to overparameterization of the model that may push the models to the boundary of high-dimensionality. While we use L_1 regularization in both cases, the result indicates that it cannot fully prevent noise accumulation due to noisy input dimensions.

Table 2 confirms that learning patient representation with RNN is easier with the input vectors that are already efficient representations of the medical codes. The RNN trained with the Skip-gram vectors (denoted by RNN-IR) consistently outperforms the RNN that learns the weight matrix $\boldsymbol{W}_{emb}$ directly from the data, with only one exception, the medication prediction Recall@30, although the differences are insignificant. The results also confirm that having multiple layers when using RNN improves its ability to learn more efficient representations. The results also indicate that a single layer RNN might have enough representative power to capture the dynamics of medications, and adding more layers may not improve the performance.

The results also indicate that our approach significantly improves the accuracy of predicting the time duration until the next visit compared to the baselines. However, the absolute value of R^2 metric shows that accurate prediction of time intervals remains as a challenge. We believe achieving significantly better time prediction without extra features should be difficult because the timing of a clinical visit can be affected by many personal factors such as financial status, location of residence, means of transportation, and life style, to name a few. Thus, without such sensitive personal information, which is rarely included in the EHR, accurate prediction of time intervals should be unlikely.

4.5 Understanding the behavior of the network

In order to study the applicability of our model in a real-world setting where patients have varying length of medical records, we conducted an additional experiment to study the relationship between the length of the patient medical history and the prediction performance. To this end, we selected



(a)

(b)

Figure 3: Characterizing behavior of the trained network: (a) Prediction performance of Doctor AI as it sees a longer history of the patients. (b) Change in the perplexity of response to a frequent code (hypertension) and an infrequent code (Klinefelter's syndrome).

5,800 patients from the test set who had more than 100 visits. We used the best trained model to predict the diagnosis codes at visits at different times and found the mean and standard error of recall across the selected patients. Figure 3a shows the result of the experiment. We believe that the increase in performance can be due to two reasons: (1) RNN is able to learn a better estimate of the patient status as it sees longer patient records and (2) the patient's status stabilizes over time and the prediction task becomes easier.

Another experiment was conducted to understand the behavior of the network by giving synthetic inputs. We chose hypertension (ICD-9 code 401.9) as an example of a frequently observed diagnosis, and Klinefelter's syndrome (ICD-9 code 758.7) as an example of an infrequent diagnosis. We created two synthetic patients who respectively have 200 visits of 401.9 and 758.7. Then we used the best performing network to predict the diagnosis codes for the next visits. The results in Figure 3b shows contrasting patterns: when the input is one of the frequent codes such as hypertension, the network quickly learns a more specific set of output codes as next disease. When we select an infrequent code like Klinefelter's syndrom as the input, the network's output is more diverse and mostly the frequently observed codes. The top 30 codes after convergence shown in Table 4 in Appendix C confirm the disparity of the diversity of the predicted codes for the two cases.

In order to take a closer look at the performance of Doctor AI, in Table 3 (in Appendix C) we list the predicted, true, and historical diagnosis codes for five visits of different patients. The blue items represent the correct predictions. The results are promising and show that, given the history of the patient, the Doctor AI can predict the true diagnostic codes. The results highly mimic the way a human doctor will interpret the disease predictions from the history. For all five of the cases shown in Table 3, the set of predicted diseases contain most, if not all of the true diseases. For example, in the first case, the top 3 predicted diseases match the true diseases. A human doctor would likely predict similar diseases to the ones predicted with Doctor AI, since old myocardial infarction and chronic ischemic heart disease can be associated with infections and diabetes [43].

In the fourth case, visual disturbances can be associated with migraines and essential hyper-

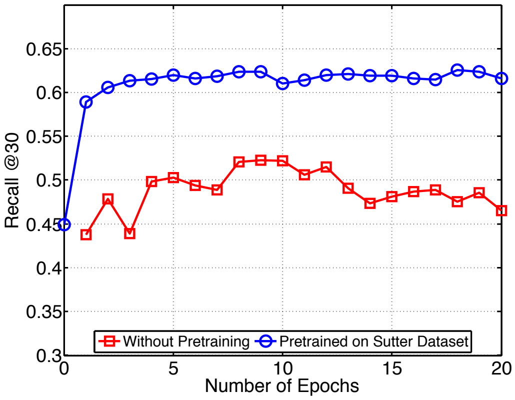


Figure 4: The impact of pre-training on improving the performance on smaller datasets. In the first experiment, we first train the model on a small dataset (red curve). In the second experiment, we pretrain the model on our large dataset and use it for initializing the training of the smaller dataset. This procedure results in more than 10% improvement in the performance.

tension [23]. Further, essential hypertension may be linked to cognitive function [25], which plays a role in anxiety disorders and dissociative and somatoform disorders. Regarding codes that are guessed incorrectly with the fourth case, they can still be plausible given the history. For example, cataracts, and disorders of refraction and accomodation could have been guessed based on a history of visual disturbances, as well as strabismus and disorders of binocular eyemovements. Allergic rhinitis could have been guessed, because there was a history of allergic rhinitis. In summary, Doctor AI is able to very accurately predict the true diagnoses in the sample patients. The results are promising and should motivate future studies involving the application of Doctor AI on different datasets exhibiting other populations of patients.

4.6 Knowledge transfer across hospitals

As we observed from the previous experiments, the dynamics of clinical events are complex, which requires models with a high representative power. However, many institutions have not yet collected large scale datasets, and training such models could easily lead to overfitting. To address this challenge, we resort to the recent advances in domain adaptation techniques for deep neural networks [33, 3, 50, 18, 37].

MIMIC II [39], a publicly available clinical dataset collected from ICU patients over 7 years of observation was chosen to conduct the experiment. This dataset differs from the Sutter dataset in that it consists of demographically and diagnostically different patients. The number of patients who made at least two visits is 2,695, and the number of unique diagnosis code (3-digit ICD-9 code) is 767, which is a subset of Sutter dataset. From the dataset, we extracted sequences of 3-digit ICD-9 codes. We chose 2,290 patients for training, 405 for testing. We chose the 2-layer RNN with 1000 dimensional hidden layer, and performed two experiments: 1) We trained the model only on the MIMIC II dataset. 2) We initialized the coefficients of the model with the values learned from the 3-digit ICD-9 sequences of Sutter data, then we refined the coefficients with the MIMIC II dataset. Figure 4 shows the vast improvement of the prediction performance induced by the knowledge transfer from the Sutter data.

4.7 Discussion

In healthcare, it is desirable to have models that are interpretable; i.e., models' parameters can be translated to clinically meaningful quantities that physicians or patients understand. However, there is a well-known trade-off between interpretability (intelligibleness) and accuracy of machine learning algorithms [5]. Neural networks, and in particular Recurrent Neural Networks, are more on the accurate side rather than the intelligible side of the trade-off spectrum. Although, several recent attempts have been made to interpret the parameters of RNNs, such as [12] and this problem remains an active line of research [22, 7]. Interpretability can be considered a limitation of RNN in healthcare applications, although we believe the significant gain in accuracy in this approach outweighs the limitation.

While interpretability of the model is desirable in healthcare applications, the accurate black-box models can still be useful. In fact, the common clinical practice does not require practitioners to fully understand the underlying mechanism of how a particular drug or lab test works. For example, pharmaceutical companies are not required to show exactly why their proposed medicine works. They only need to show efficacy through randomized control trials. In the same spirit, with this early evidence from this work, Doctor AI can be prospectively validated through randomized clinical trials in order to apply in practice.

Finally, it is worth mentioning that interpretability of machine learning algorithms usually requires a discussion about causality and whether the algorithm can uncover causal relationships. Despite the fact that most of the common machine learning algorithms are not guaranteed to find causal relationships between features and labels, we still find them useful, because they provide *accurate prediction* of the target label [42]. Similarly, Doctor AI's main objective is to provide accurate diagnostic and medication prediction which can be of great value in practice as a clinical decision support tool.

5 Conclusion

In this work, we proposed Doctor AI system, which is a RNN-based model that can learn efficient patient representation from a large amount of longitidinal patient records and predict future events of patients. We demonstrated Doctor AI on two large real-world EHR datasets, which achieved 64.30% recall@10 and significantly outperformed many baselines. The empirical analysis by a medical expert confirmed that Doctor AI not only mimics the predictive power of human doctors, but also provides diagnostic results that are clinically meaningful.

Our proposed Doctor AI can potentially open up new avenues for further improvement of computational health problems. Applications and extensions of Doctor AI can be studied in diverse clinical settings including expanding toward rich and unstructured data sources such as medical images and clinical notes.

Acknowledgments

This work was supported by the National Science Foundation, award IIS- #1418511 and CCF-#1533768, Children's Healthcare of Atlanta, CDC I-SMILE project, Google Faculty Award, AWS Research Award, Microsoft Azure Research Award and UCB.

References

[1] Mohammad Taha Bahadori, Yan Liu, and Eric P Xing. Fast structure learning in generalized stochastic processes with latent factors. In *KDD*, pages 284–292, 2013.

[2] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[3] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. *Unsupervised and Transfer Learning Challenges in Machine Learning*, 7:19, 2012.

[4] Yoshua Bengio, Aaron Courville, and Pierre Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

[5] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noémie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD*, 2015.

[6] Wendy W Chapman, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310, 2001.

[7] Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*, 2015.

[8] Edward Choi, Nan Du, Robert Chen, Le Song, and Jimeng Sun. Constructing disease network and temporal progression model via context-sensitive hawkes process. In *ICDM*, 2015.

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[10] Willem De Winter, Joost DeJongh, Teun Post, Bart Ploeger, Richard Urquhart, Ian Moules, David Eckland, and Meindert Danhof. A mechanism-based disease progression model for comparison of long-term effects of pioglitazone, metformin and gliclazide on disease processes underlying type 2 diabetes mellitus. *Journal of pharmacokinetics and pharmacodynamics*, 33(3):313–343, 2006.

[11] Yohann Foucher, Magali Giral, Jean-Paul Soulillou, and Jean-Pierre Daures. A semi-markov model for multistate and interval-censored data with multiple terminal events. application in renal transplantation. *Statistics in medicine*, 26(30):5381–5393, 2007.

[12] Joydeep Ghosh and Vijay Karamcheti. Sequence learning with recurrent networks: analysis of internal representations. In *Aerospace Sensing*, pages 449–460. International Society for Optics and Photonics, 1992.

[13] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[14] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, pages 1764–1772, 2014.

[15] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *PAMI*, 2009.

[16] David Heckerman. A tractable inference algorithm for diagnosing multiple diseases. In *UAI*, 1990.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[18] Judy Hoffman, Sergio Guadarrama, Eric S Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems*, pages 3536–3544, 2014.

[19] Kaori Ito, Sima Ahadieh, Brian Corrigan, Jonathan French, Terence Fullerton, Thomas Tensfeldt, Alzheimer's Disease Working Group, et al. Disease progression meta-analysis model in alzheimer's disease. *Alzheimer's & Dementia*, 6(1):39–53, 2010.

[20] Christopher H Jackson, Linda D Sharples, Simon G Thompson, Stephen W Duffy, and Elisabeth Couto. Multistate markov models for disease progression with classification error. *JRSS-D*, 2003.

[21] Matthew J Johnson and Alan S Willsky. Bayesian nonparametric hidden semi-markov models. *The Journal of Machine Learning Research*, 14(1):673–701, 2013.

[22] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

[23] Norman M Keith, Henry P Wagener, and Nelson W Barker. Some different types of essential hypertension: their course and prognosis. *The American Journal of the Medical Sciences*, 197(3):332–343, 1939.

[24] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[25] Johanna Kuusisto, Keijo Koivisto, L Mykkänen, Eeva-Liisa Helkala, Matti Vanhanen, T Hänninen, K Pyörälä, Paavo Riekkinen, and Markku Laakso. Essential hypertension and cognitive function. the role of hyperinsulinemia. *Hypertension*, 22(5):771–779, 1993.

[26] Jane Lange. *Latent Continuous Time Markov Chains for Partially-Observed Multistate Disease Processes*. PhD thesis, 2014.

[27] Jane M Lange, Rebecca A Hubbard, Lurdes YT Inoue, and Vladimir N Minin. A joint model for multistate disease processes and random informative observation times, with applications to electronic medical records data. *Biometrics*, 71(1):90–101, 2015.

[28] Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

[29] Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *ICML*, pages 1413–1421, 2014.

[30] Thomas Josef Liniger. *Multivariate hawkes processes*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 18403, 2009, 2009.

[31] Yu-Ying Liu, Hiroshi Ishikawa, Mei Chen, Gadi Wollstein, Joel S Schuman, and James M Rehg. Longitudinal modeling of glaucoma progression using 2-dimensional continuous-time hidden markov model. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*, pages 444–451. 2013.

[32] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. Scoring, term weighting, and the vector space model. In *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[33] Grégoire Mesnil, Yann Dauphin, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian J Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, David Warde-Farley, et al. Unsupervised and transfer learning challenge: a deep learning approach. *ICML Unsupervised and Transfer Learning*, 27:97–110, 2012.

[34] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[35] DR Mould. Models for disease progression: new approaches and uses. *Clinical Pharmacology & Therapeutics*, 92(1):125–131, 2012.

[36] Uri Nodelman, Christian R Shelton, and Daphne Koller. Continuous time bayesian networks. In *UAI*, pages 378–387. Morgan Kaufmann Publishers Inc., 2002.

[37] Tom Le Paine, Pooya Khorrami, Wei Han, and Thomas S Huang. An analysis of unsupervised pre-training in light of recent advances. *arXiv preprint arXiv:1412.6597*, 2014.

[38] Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David M Blei. The survival filter: Joint survival analysis with a latent time series. In *UAI*, 2015.

[39] Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

[40] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[41] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[42] Peter Spirtes. Introduction to causal inference. *JMLR*, 11:1643–1662, 2010.

[43] Victor J Stevens, Carol A Rouzer, Vincent M Monnier, and Anthony Cerami. Diabetic cataract formation: potential role of glycosylation of lens crystallins. *PNAS*, 75(6):2918–2922, 1978.

[44] Rafid Sukkar, Edward Katz, Yanwei Zhang, David Raunig, and Bradley T Wyman. Disease progression modeling using hidden markov models. In *EMBC*, 2012.

[45] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.

[46] Navdeep Tangri, Lesley A Stevens, John Griffith, Hocine Tighiouart, Ognjenka Djurdjev, David Naimark, Adeera Levin, and Andrew S Levey. A predictive model for progression of chronic kidney disease to kidney failure. *Jama*, 305(15):1553–1559, 2011.

[47] Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.

[48] Alejandro Veen and Frederic P Schoenberg. Estimation of space–time branching process models in seismology using an em–type algorithm. *JASA*, 103(482):614–624, 2008.

[49] Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *KDD*, 2014.

[50] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

[51] Wojciech Zaremba and Ilya Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.

[52] Jiayu Zhou, Jun Liu, Vaibhav A Narayan, and Jieping Ye. Modeling disease progression via fused sparse group lasso. In *KDD*, pages 1095–1103, 2012.

[53] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS*, pages 641–649, 2013.

[54] Lingjiong Zhu. *Nonlinear Hawkes Processes.* PhD thesis, New York University, 2013.

A Learning the Skip-gram vectors from the EHR

Learning efficient representations of medical codes (*e.g.* diagnosis codes, medication codes, and procedure codes) may lead to improved performance of many clinical applications. We specifically used Skip-gram [34] to learn real-valued multidimensional vectors to capture the latent representation of medical codes from the EHR.

We processed the Sutter PAMF dataset so that diagnosis codes, medication codes, procedure codes are laid out in a temporal order. If there are multiple codes at a single visit, they were laid out in a random order. Then using the context window size of 5 to the left and 5 to the right, and applying Skip-gram, we were able to project diagnosis codes, medication codes and procedure codes into the same lower dimensional space, where similar or related codes are embedded close to one another. For example, hypertension, obesity, hyperlipidemia all share similar values compared to pneumonia or bronchitis. The trained Skip-gram vectors are then plugged into RNN so that a multi-hot vector can be converted to vector representations of medical codes.

B Details of the baselines

B.1 Details of training procedure for baselines

We use a multilayer perceptron with a hidden layer of width 2,000. We use dropout with $p = 0.3$ and apply L_1 regularization to all of the weight matrices. The activation functions in the first and output layers are selected to be sigmoid and softmax functions respectively. For prediction of time intervals, we used rectified linear units. The regularization parameter has been increased with the number of lags to keep the performance of the model in high-dimensional regimes.

B.2 Manual feature extraction details

The process of extracting *tf-idf* like features for each visit is as follows: First, we compute the p-dimensional count vector by counting the number of visits in which each code has been observed. The *idf* vector is computing by taking the logarithm of total number of visits divided by the count vector.

Then, the *tf* vector is computed for each visit by counting the number of codes observed in *all visits* until the current visit. This process is used to ensure that the features include information from past visits. Finally, we use the element-wise product of the *tf* and *idf* vectors as the *tf-idf* vector.

C Detailed inspection of results

The detailed results are shown in Table 3.

Table 3: Comparison of the diagnoses by Doctor AI and the true future diagnoses.

Predicted		True		History	
ICD9	Description	ICD9	Description	ICD9	Description
412	**Old myocardial infarction**				
V58	**Encounter for other and unspecified procedures**				
414	**Other forms of chronic ischemic heart disease**			465	Acute upper respiratory infec. of multiple or unspec. sites
272	Disorders of lipoid metabolism	**414**	**Other forms of chronic ischemic heart disease**	250	Diabetes mellitus
250	Diabetes mellitus	**412**	**Old myocardial infarction**	366	Cataract
585	Chronic kidney disease (CKD)	**V58**	**Encounter for other and unspecified procedures**	V58	Encounter for other and unspecified procedures
428	Heart failure			362	Other retinal disorders
285	Other and unspecified anemias				
V04	Need for prophylactic vaccin. and inocul. against certain diseases				
V76	Special screening for malignant neoplasms				
V07	**Need for isolation and other prophylactic measures**				
477	Allergic rhinitis				
780	General symptoms			782	Symptoms involving skin and other integumentary tissue
401	**Essential hypertension**	**V07**	**Need for isolation and other prophylactic measures**	477	Allergic rhinitis
786	**Symptoms involving respiratory system**	**401**	**Essential hypertension**	V07	Need for isolation and other prophylactic measures
493	Asthma	**786**	**Symptoms involving respiratory system**	564	Functional digestive disorders, not elsewhere classified
300	Anxiety, dissociative and somatoform disorders	782	Symptoms involving skin and other integumentary tissue	401	Essential hypertension
461	Acute sinusitis				
530	Diseases of esophagus				
719	Other and unspecified disorders of joint				
453	Other venous embolism and thrombosis				
V58	**Encounter for other and unspecified procedures**				
719	**Other and unspecified disorders of joint**				
V12	**Personal history of certain other diseases**	**715**	**Osteoarthrosis and allied disorders**	453	Other venous embolism and thrombosis
V43	Organ or tissue replaced by other means	**V12**	**Personal history of certain other diseases**	956	Injury to peripheral nerve(s) of pelvic girdle and lower limb
729	Other disorders of soft tissues	**719**	**Other and unspecified disorders of joint**	V43	Organ or tissue replaced by other means
715	**Osteoarthrosis and allied disorders**	**V58**	**Encounter for other and unspecified procedures**		
733	Other disorders of bone and cartilage				
726	Peripheral enthesopathies and allied syndromes				
451	Phlebitis and thrombophlebitis				
477	Allergic rhinitis				
780	**General symptoms**				
300	**Anxiety, dissociative and somatoform disorders**			782	Symptoms involving skin and other integumentary tissue
401	**Essential hypertension**	**401**	**Essential hypertension**	477	Allergic rhinitis
346	**Migraine**	**780**	**General symptoms**	692	Contact dermatitis and other eczema
366	Cataract	**346**	**Migraine**	368	Visual disturbances
V43	Organ or tissue replaced by other means	**300**	**Anxiety, dissociative and somatoform disorders**	378	Strabismus and other disorders of binocular eye movements
367	Disorders of refraction and accommodation				
368	Visual disturbances				
272	Disorders of lipoid metabolism				
428	**Heart failure**				
427	**Cardiac dysrhythmias**				
272	**Disorders of lipoid metabolism**	**250**	**Diabetes mellitus**	466	Acute bronchitis and bronchiolitis
401	Essential hypertension	402	Hypertensive heart disease	428	Heart failure
786	Symptoms involving respiratory system	**428**	**Heart failure**	786	Symptoms involving respiratory system
185	Malignant neoplasm of prostate	**272**	**Disorders of lipoid metabolism**	785	Symptoms involving cardiovascular system
250	**Diabetes mellitus**	**427**	**Cardiac dysrhythmias**	250	Diabetes mellitus
414	Other forms of chronic ischemic heart disease				
788	Symptoms involving urinary system				
424	Other diseases of endocardium				

Table 4: Comparison of the diagnoses by Doctor AI for a frequent and an infrequent disease code after 200 time step.

Hypertension		**Klinefelter's syndrome**	
ICD9	Description	ICD9	Description
401	Essential hypertension	272	Disorders of lipoid metabolism
272	Disorders of lipoid metabolism	V70	General medical examination
786	Symptoms involving respiratory system and other chest symptoms	V04	Need for prophylactic vaccination and inoculation against certain diseases
V06	Need for prophylactic vaccination and inoculation against combinations of diseases	730	Osteomyelitis, periostitis, and other infections involving bone
790	Nonspecific findings on examination of blood	780	General symptoms
V76	Special screening for malignant neoplasms	783	Symptoms concerning nutrition, metabolism, and development
V04	Need for prophylactic vaccination and inoculation against certain diseases	295	Schizophrenic disorders
V70	General medical examination	V76	Special screening for malignant neoplasms
780	General symptoms	141	Malignant neoplasm of tongue
276	Disorders of fluid, electrolyte, and acid-base balance	V06	Need for prophylactic vaccination and inoculation against combinations of diseases
782	Symptoms involving skin and other integumentary tissue	250	Diabetes mellitus
268	Vitamin D deficiency	782	Symptoms involving skin and other integumentary tissue
719	Other and unspecified disorders of joint	786	Symptoms involving respiratory system and other chest symptoms
427	Cardiac dysrhythmias	208	Leukemia of unspecified cell type
380	Disorders of external ear	401	Essential hypertension
250	Diabetes mellitus	790	Nonspecific findings on examination of blood
599	Other disorders of urethra and urinary tract	280	Iron deficiency anemias
V72	Special investigations and examinations	607	Disorders of penis
789	Other symptoms involving abdomen and pelvis	281	Other deficiency anemias
729	Other disorders of soft tissues	V03	Need for prophylactic vaccination and inoculation against bacterial diseases
682	Other cellulitis and abscess	332	Parkinson's disease
V03	Need for prophylactic vaccination and inoculation against bacterial diseases	255	Disorders of adrenal glands
724	Other and unspecified disorders of back	799	Other ill-defined and unknown causes of morbidity and mortality
V58	Encounter for other and unspecified procedures and aftercare	244	Acquired hypothyroidism
278	Overweight, obesity and other hyperalimentation	V58	Encounter for other and unspecified procedures and aftercare
V82	Special screening for other conditions	151	Malignant neoplasm of stomach
V65	Other persons seeking consultation	294	Persistent mental disorders due to conditions classified elsewhere
585	Chronic kidney disease (CKD)	V72	Special investigations and examinations
274	Gout	344	Other paralytic syndromes
V49	Other conditions influencing health status	146	Malignant neoplasm of oropharynx