Neural Networks 124 (2020) 109–116

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2019 Special Issue

Medi-Care AI: Predicting medications from billing codes via robust recurrent neural networks

Deyin Liu [a], Yuanbo Lin Wu [b,c,*], Xue Li [d], Lin Qi [a]

[a] School of Information Engineering, Zhengzhou University, China
[b] Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, China
[c] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230000, China
[d] Dalian Neusoft University of Information, China

ARTICLE INFO

Article history:
Available online 23 January 2020

Keywords:
Billing codes
Robust recurrent neural networks
Health care data
Medication prediction

ABSTRACT

In this paper, we present an effective deep prediction framework based on robust recurrent neural networks (RNNs) to predict the likely therapeutic classes of medications a patient is taking, given a sequence of diagnostic billing codes in their record. Accurately capturing the list of medications currently taken by a given patient is extremely challenging due to undefined errors and omissions. We present a general robust framework that explicitly models the possible contamination through overtime decay mechanism on the input billing codes and noise injection into the recurrent hidden states, respectively. By doing this, billing codes are reformulated into its temporal patterns with decay rates on each medical variable, and the hidden states of RNNs are regularized by random noises which serve as dropout to improved RNNs robustness towards data variability in terms of missing values and multiple errors. The proposed method is extensively evaluated on real health care data to demonstrate its effectiveness in suggesting medication orders from contaminated values.

1. Introduction

There has been growing interest in exploiting the large amounts of data existed in electronic medical records for both clinical events and secondary research. While leveraging large historical data in electronic health records (EHR) holds great promise, its potential is weakened by multiple errors and omissions in those records. Some studies show that over 50% of electronic medication lists contain omissions (SelinCaglar, Henneman, Blank, Smithline, & Henneman, 2011), and even 25% of all medications taken by patents are not recorded. To ensure the corrections of medication lists, great efforts have been dedicated to improve the communications between patients and providers (Keogh et al., 2016), however, manually maintaining these lists would be extremely human–labor intensive. Thus, it demands a generic yet robust predictive model that is able to suggest medication consultation to the patients next visit in the context of medication documentation contaminations.

Recently, Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), and Gated Recurrent Unit (GRU) (Cho et al., 2014) have been explored for modeling diseases and patient diagnosis in health care modality (Choi, Bahadori, Schuetz, Stewart, & Sun, 2016; Choi, Xiao, Stewart, & Sun, 2018; Lipton, Kale, Elkan and Wetzel, 2016; Wang, Zhang, He and Zha, 2018). For instance, a temporal model based on RNNs, namely Doctor AI, is developed to predict future physician diagnosis and medication orders. This intelligent system demonstrates that historical EHR data can be leveraged to forecast the patient status at the next visit and present medication to a physician would like to refer at the moment. However, little efforts are put into systematically modeling the EHR with missing values (Che, Purushotham, Cho, Sontag, & Liu, 2016) since it is difficult to capture the missing patterns in medical billing codes. Simple solutions such as omitting the missing data and to perform analysis only on the observed data, or filling in the missing values through smoothing/interpolation (Kreindler & Lumsden, 2012), spectral analysis (Mondal & Perciva, 2010; Wang, Lin, Wu, & Zhang, 2017; Wang, Wu, Lin and Gao, 2018; Wang, Wu and Zhang, 2018; Wang et al., 2016), and multiple imputations (White, Royston, & Wood, 2011) offer plausible ways to the missing values in data series. However, these solutions often result in suboptimal analysis and poor predictions because the imputations are disparate from the prediction models and missing patterns are not properly described (Wells, Chagin, Nowacki, & Kattan, 2013).

* Corresponding author at: School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230000, China.
E-mail addresses: iedyzzu@outlook.com (D. Liu), xiaoxian.wu9188@gmail.com (Y.L. Wu), lixue@neusoft.edu.cn (X. Li), ielqi@zzu.edu.cn (L. Qi).

https://doi.org/10.1016/j.neunet.2020.01.001
0893-6080/

A recent finding demonstrates that missing values in time series data are usually *informative missing*, that is, the missing values and patterns are related to the target labels in supervised learning tasks. For example, Che et al. (2016) show that the missing rates in time series health care data are usually highly correlated with the labels of interests such as mortality and ICD-9 diagnoses. Hence, it demands an appropriate strategy to describe the decaying on diagnostic measurements over time. Moreover, the diagnostic billing codes are characterized of more than missing values in patient records, whereas in most cases they are combined with multiple errors and omissions. Thus, we use the terminology *noise* to generally refer to all potential incorrectness of medication lists.

1.1. Our approach

Inspired by the noise-based regularizer of RNNs, a.k.a dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014; Wager, Wang, & Liang, 2013), we impose a multiplicative noise into the hidden states to ensure the robustness of recurrence and also preserve the underlying RNN in the context of noise injection. Hence, in this paper we develop a robust RNN model, an effective new way to deal with incomplete billing codes in medical domain whilst being capable of predicting the future medication orders given the missing codes in sequence. The key idea is to not only model the input codes by explicitly encoding the missing patterns over time, but also inject random noise into the transition function of recurrence. Intuitively, the explicit noise injection into the hidden states of RNNs can serve as regularizer to drop the observation difference that will be potentially added into the hidden states. Thus, the RNNs are trained to fit its parameters to maximize the corresponding marginal likelihood of observations in the context of high variability. The proposed model is experimentally evaluated on real EHR datasets to demonstrate its effectiveness in identifying missing actual information in relation to therapeutic classes.

1.2. Contributions

The contributions of this paper can be summarized as follows.

- We present a robust RNN based medication prediction framework to effectively cope with sequential billing codes that are contaminated by missing values and multiple errors.
- The proposed approach is designed to predict the complete set of medications a patient is actively taking at a given moment from a sequence of diagnostic billing coeds in the context of non-trivial billing record noise. This is, to our best knowledge, the first effort to *explicitly* model both the medication care data and delving the RNNs into the medical domain.
- Insightful analysis to our approach is provided in this paper. Extensive experiments on health care datasets are conducted to demonstrate the superiority of our method over state-of-the-art by achieving the performance gain on AUC by 13% and 7% on the Physio-net challenge dataset (Silva, Moody, Scott, Celi, & Mark, 2012) and MIMIC-III (Johnson et al., 2016), respectively.

The rest of this paper is organized as follows. Section 2 reviews some related works. We detail the proposed predictive model in Section 4 with some background described in Section 3 in advance. Section 5 reports extensive experiments over the real-valued medical datasets, and the paper is concluded in Section 6.

2. Related work

2.1. Modeling medical event sequences

Common approaches to modeling medical event sequences include continuous-time Markov chain based models (Johnson & Willsky, 2013) and their extension using Bayesian networks (Weiss, Natarajan, & Page, 2012) as well as intensity function methodologies such as Hawkes processes (Choi, Du, Chen, Song, & Sun, 2015). It is known that continuous-time Markov chain methods are computationally expensive because modeling multi-labeled point processes would expand rapidly their state-space. On the other hand, Hawkes processes with intensity functions depend linearly with respect to the past observations, while they are limited in capturing temporal dynamics. Moreover, there is no study on these models to deal with missing values or incorrect data. In this paper, we address these challenges by designing a robust recurrent neural network which has shown to be effective in learning complex yet potentially missing data in sequential patterns regarding health-care systems.

2.2. Deep learning models for EHR

It has witnessed some attempts to apply neural network models a.k.a deep learning methods to study EHR since deep learning models are capable of learning complex data patterns. The earlier work is the use of an LSTM model that produced reasonable accuracy (micro-AUC 0.86) in a 128-dim multi-label prediction of diagnoses from regularly sampled, continuously real-valued physiologic variables in an Intensive Care Unit (ICU) setting (Lipton, Kale, Elkan et al., 2016). One successful framework is Doctor AI (Choi et al., 2016) which is a predictive temporal model using RNNs to predict the diagnosis and medication codes for a subsequent visit of patients. They used a GRU model in a multi-label context to predict the medications, billing codes, and time of the next patient visit from a sequence of that same information for previous visits. It can achieve an improvement over a single-hidden-layer MLP (reach a recall@30 of 70.5 by a 20 margin). This is a successful showcase of using the strength of recurrence, *i.e.,* to predict the next element in a sequence. However, aforementioned deep learning paradigms are not able to effectively cope with EHR with errors and omissions.

Prior efforts have been dedicated into modeling missing data in sequences with RNNs in clinical time series (Che et al., 2016; Lipton, Kale and Wetzel, 2016; Wells et al., 2013). A very recent work yet contemporary with our work, namely GRU-Decay (Che et al., 2016), used a GRU model with imputation on missing data by a decay term to predict the mortality/ICD-9 diagnosis categories from medication orders and billing codes. Our method contrasts with GRU-Decay (Che et al., 2016) in the way of managing the RNN to tackle the missing values. Instead of using the same decay mechanism on both input sequence and the hidden state as the GRU-Decay performed, we propose to dealing with the raw inputs and hidden states in different strategies wherein the input billing codes are multiplied by decay rate on each variable (the same as GRU-Decay Che et al., 2016), and the hidden states are injected into noises in the multiplicative form.

3. Background

3.1. Medical billing codes

In our experiments, codes are from the International Classification of Disease, Ninth Revision (ICD-9). The ICD-9 hierarchy consists of 21 chapters roughly corresponding to a single organ system or pathologic class. Leaf-level codes in the tree represent

Table 1
The top level classes for ICD-9 chapters.

Code range	Description
001–139	Infectious and parasitic diseases
140–239	Neoplasms
240–279	Endocrine, nutritional and metabolic diseases, immunity disorders
280–289	Blood diseases and blood-forming organs
290–319	Mental disorders
320–359	Nervous system diseases
360–389	Sense system diseases
390–459	Circulatory system diseases
460–519	Respiratory system diseases
520–579	Digestive system diseases
580–629	Genitourinary system diseases
630–679	Complications of pregnancy, childbirth, and the puerperium
680–709	Skin and subcutaneous tissue
710–739	Musculoskeletal system and connective tissue
740–759	Congenital anomalies
760–779	Conditions originating in the perinatal period
780–799	Symptoms, signs and ill-defined conditions
800–999	Injury and poisoning

single diseases or disease subtypes. For each time a patient has billable contact with the health-care system through which the time stamped billing codes are associated with the patient record, indicating the medical conditions that are related to the reasoning for the visit. However, these billing codes are more often unreliable or incomplete, and thus making the electronic medical records unable to track the set of medications that the patient is actively taking. The code range and descriptions are shown in Table 1.

3.2. Recurrent neural networks

An recurrent neural network (RNN) considers a sequence of observations, $\mathbf{X}_{1:T} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, and to handle the sequential time-series the RNN introduces the hidden state $\mathbf{h}_t$ at time step t, as a parametric function $f_W(\mathbf{h}_{t-1}, \mathbf{x}_{t-1})$ of the previous state $\mathbf{h}_{t-1}$ and the previous observation $\mathbf{x}_{t-1}$. The parameter W is shared across all steps which would greatly reduce the total number of parameters we need to learn. The function f_W is the transition function of the RNN, which defines a recurrence relation for the hidden states and renders $\mathbf{h}_t$ a function of all the past observations $\mathbf{x}_{1:t-1}$.

The particular form of f_W determines the variants of RNN including Long–Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Units (GRU) (Cho et al., 2014). In this paper, we will study GRU which has shown very similar performance to LSTM but employs a simper architecture. First, we would reiterate the mathematical formulation of GRU as follows

$$\begin{aligned} &\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1}),\ \mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1}),\\ &\hat{\mathbf{h}}_t = tanh(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1})),\\ &\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \hat{\mathbf{h}}_t, \end{aligned} \tag{1}$$

where $\odot$ is an element-wise multiplication. $\mathbf{z}_t$ is an update gate that determines the degree to which the unit updates its activation. $\mathbf{r}_t$ is a reset gate, and σ is the sigmoid function. The candidate activation $\hat{\mathbf{h}}_t$ is computed similarly to that of traditional recurrent unit. When $\mathbf{r}_t$ is close to zero, the reset gate make the unit act as reading the first symbol of an input sequence and forgets the previously computed state.

4. Robust recurrent neural networks for medication predictions

In this section, we develop a new framework for clinical medication predictions in the context of missing information and multiple errors. We first formulate the prediction problem setting, and then detail the architecture with explicit noise injection into the recurrent hidden states. Finally, we present the training procedure on the proposed model.

4.1. Problem setting

For each patient, the temporal observations are represented by multivariate time series with D variables of length T as $\mathbf{X}_{1:T} \in \mathbb{R}^{T\times D}$, where $\mathbf{x}_t \in \mathbb{R}^D$ denotes the tth observations, namely measurements of all variables and x_t^d denotes the dth variable of $\mathbf{x}_t$. In the medication records, the variables correspond to multiple medication codes, such as the codes 493 (asthma) and 428 (heat failure) from ICD-9. For each time stamp, we may extract high-level codes for prediction purpose and denote it by $\mathbf{y}_t$. Generic Product Identifier (GPI) medication codes are extracted from the medication orders. This is because the input ICD-9 codes are represented sequentially while the medications are represented as a list that changes over time. Also, many of the GPI medication codes are very granular, for example, the pulmonary tuberculosis (ICD-9 code 011) can be divided into 70 subcategories (011.01, 011.01, …, 011.95, 011.96).

In this paper, we are interested in learning an effective vector representation for each patient from his billing codes over time with multiple missing values at each time stamp $t = 1, \ldots, T$, and predicting diagnosis and medication categories in the next visit $\mathbf{y}_{T+1}$. We investigate the use of RNN to learn such billing code representations, treating the hidden layers as the representation for the patient status and use them for the prediction tasks. To account for the situation of missing/incorrect values in EHR, we propose robust RNN architecture, which effectively models the missing patterns from time series onwards through the temporal decay mechanism (Che et al., 2016; Vodovotz, An, & Androulakis, 2013; Zhou & Hripcsak, 2007) and injects noises into the hidden states of RNN at each time step.

4.2. Robust RNNs with noise injection

To effectively learn representations from missing or incorrect values in billing codes, we propose to incorporate different strategies in regard to the input billing codes and the hidden states, respectively. For the missing values in billing codes of EHR, we employ the decay mechanism which has been designed for modeling the influence of missing values in health care domain (Vodovotz et al., 2013). This is based on the property that the values of missing variables tend to be close to some default value if its last measurement is observed a long time ago. This property should be considered as critical for disease diagnosis and treatment. Also, the influence of the input dimensions will fade away over time if some dimension is found missing for a while. On the other hand, the hidden states of RNNs should be injected with random noises which is more advantageous by preventing the dimensions of hidden states from co-adapting and it can force the individual units to capture useful features (Dieng, Ranganath, Altosaar, & Blei, 2018).

Specifically, we inject a decay rate into each variable of the billing code series. In this way, the decay rate differs from variable to variable and indicative to unknown possible missing patterns. To this end, the vector of a decay rate is formulated as

$$\gamma_t = \exp\{-\max(\mathbf{0}, \mathbf{W}_\gamma\delta_t + \mathbf{b}_\gamma)\}, \tag{2}$$

where $\mathbf{W}_\gamma$ and $\mathbf{b}_\gamma$ are trainable parameters jointly with the LSTM. $\exp\{\cdot\}$ is the exponential negative rectifier to keep each decay rate monotonically decreasing ranged between 0 and 1. δ_t^d is the time

interval for each variable d since its last observation, which can be defined as

$$\delta_t^d = \begin{cases} s_t - s_{t-1} + \delta_{t-1}^d, t > 1 \\ 0, t = 1 \end{cases} \tag{3}$$

In Eq. (3), s_t denotes the time stamp when the tth observation is obtained and we assume that the first observation is made at time $t = 0$ ($s_1 = 0$). Hence, for a missing variable code, we adopt the decay vector γ_t to decay it overtime but towards an empirical mean instead of using its last observation. And the decaying measurement billing code vector can be formulated by applying the decay scheme into:

$$x_t^d \leftarrow \gamma_{x_t}^d x_{t'}^d + (1 - \gamma_{x_t}^d)\hat{x}^d, \tag{4}$$

where $x_{t'}^d$ is the last observation of the dth variable ($t' < t$) and $\hat{x}^d$ is the empirical mean of the dth variable. We remark that when the input billing code is decaying, the parameter $\mathbf{W}_{\gamma_x}$ should be constrained to be diagonal so as to ensure the decay rates of variables are not affecting each other.

To augment the RNN's capability of coping with multiple errors in sequential EHR billing codes, we explicitly redefine the hidden states by injecting noises. This strategy is able to effectively fit the parameters of RNN by maximizing the likelihood of data observations because the next predicted output from RNN is determined as $p(\mathbf{x}_t|\mathbf{x}_{1:t-1}) = p(\mathbf{x}_t|\mathbf{h}_t)$.[1] Thus, we define the GRU with noise as follows

$$\begin{aligned} &\epsilon_{1:T} \sim \{0, (1-\delta)\}^d; \\ &\mathbf{h}_t = f_W(\mathbf{x}_{t-1}, \mathbf{h}_{t-1}, \epsilon_t) = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} \odot \epsilon_t + \mathbf{z}_t \odot \hat{\mathbf{h}}_t \odot \epsilon_t. \end{aligned} \tag{5}$$

In Eq. (5), the noise component $\epsilon_{1:T}$ is an independent drawn from a scaled Bernoulli $(1 - \delta)$ random variable. In this paper, it is used to create the dropout noise via the element-wise product $(\cdot)$ of each time hidden state $\mathbf{h}_{t_1}$. In other words, dropout noise corresponds to setting $\mathbf{h}$ to 0 with probability δ, and to $\mathbf{h}/(1 - \delta)$ else. Intuitively, this multiplicative form of noise injection can induce the robustness of RNN to how future data may be different from the observations. Also, this can be regarded as a regularization on RNN to normalize the hidden states, which is similar to noise-based regularizer for neural networks, namely dropout (Srivastava et al., 2014; Wager et al., 2013). This explicit regularization is equivalent to fitting the RNN loss to maximize the likelihood of the data observations, while being with a penalty function of its parameters. This type of regularization that involves noise variables can help the RNNs learn long-term dependencies in sequential data even in the context of high variability because dropout-based regularization can only drop differences that are added to network's hidden state at each time-step. And thus this dropout scheme allows up to use per-step sampling while still being able to capture the long-term dependencies (Semeniuta, Severyn, & Barth, 2016).

4.3. The architecture of the prediction model

As shown in Fig. 1, the proposed robust neural network architecture receives input at each time stamp t corresponding to patient visits in sequences. The billing codes $\mathbf{x}_t$ are in the form of multi-label categories. The input sequential billing codes are modeled with the decay of missing values, and then fed into the stacked multiple layers of GRU to project the inputs into lower dimensional space, and also learn the status of the patients at each time stamp as real-valued vectors. For predicting the diagnosis codes and the medication codes at each time stamp t, a softmax layer is stacked on top of the GRU, using the hidden state $\mathbf{h}_t$ as the input, that is, $\mathbf{y}_{t+1} = softmax(\mathbf{W}_{code}^T \mathbf{h}_t + \mathbf{b}_{code})$. Thus, the objective of our model is to learn the weights $\mathbf{W}_{[z,r,h,code,\gamma]}$, $\mathbf{U}_{[z,r,h]}$, $\mathbf{b}_{[z,r,h,code,\gamma]}$. In particular, the values of all $\mathbf{W}$ and $\mathbf{U}$ are initialized to orthogonal matrices using singular value decomposition of matrices from the normal distribution (Saxe, McClelland, & Ganguli, 2013). All values of $\mathbf{b}$ are initialized to be zeros. Therefore, for each patient we employ the cross entropy as the loss function for the code prediction, which is defined as

$$\mathcal{L}(\mathbf{W}, \mathbf{U}, \mathbf{b}) = \sum_{t=1}^{n-1} (\ddot{\mathbf{y}}_{t+1} \log(\mathbf{y}_{t+1}) + (1 - \ddot{\mathbf{y}}_{t+1}) \log(1 - \mathbf{y}_{t+1})), \tag{6}$$

where $\ddot{\mathbf{y}}$ is the ground truth medication category.

Algorithm 1: The proposed robust RNN framework for medication prediction from billing codes.

Data: Input billing codes in sequence $\mathbf{x}_{1:T}$, initial hidden state $\mathbf{h}_0$, noise distribution $\varphi(\cdot : 1, \sigma)$.
Result: Set of learned parameters of GRU: $\mathbf{W}_{[z,r,h,code,\gamma]}$, $\mathbf{U}_{[z,r,h]}$, $\mathbf{b}_{[z,r,h,code,\gamma]}$.

```
Initialize the set of parameters ;
while stopping criterion not met do
    for t from 1 to T do
        Sample noise from ε_t ~ φ(ε_t : 1, σ) ;
        Compute the decayed inputs
        x_t^d ← γ_{x_t}^d x_{t'}^d + (1 − γ_{x_t}^d) x̂^d ;
        Compute state
        h_t = (1 − z_t) ⊙ h_{t−1} ⊙ ε_t + z_t ⊙ ĥ_t ⊙ ε_t ;
    Compute loss as in Eq. (6) ;
    Update the network parameters ;
```

5. Experiments

In this section, we demonstrate the performance of our model on two real-world health-care datasets, and compare it to several strong machine learning and deep learning competitors in the classification tasks.

5.1. Data preparation and experimental setting

We conduct experiments on two health-care datasets: Physio-net challenge dataset (Silva et al., 2012) and MIMIC-III (Johnson et al., 2016).

- Physio-net challenge 2012 dataset (Physio-Net): This PhysioNet Challenge dataset (Silva et al., 2012) is a publicly available collection of multivariate clinical time series from 8000 intensive care unit (ICU) records. Each record is a multivariate time series of roughly 48 h and contains 33 variables such as albumin, heart-rate, glucose etc. We use the training subset A in our experiments since ground truth outcomes are only publicly available on this subset. We conduct the prediction of 4 tasks on this dataset: in-hospital mortality, length-of-stay less than 3 days, had a cardia condition or not, and whether the patient was recovering from surgery. This can be treated as a multi-task classification problem.
- MIMIC-III: This is a publicly available dataset collected at Beth Israel Deaconess Medical Center from 2001 to 2012 (Johnson et al., 2016). It contains over 58,000 hospital admission records, and we extract 99 time series features from

[1] The likelihood $p(\mathbf{x}_t|\mathbf{h}_t)$ can be in the form of the exponential family.

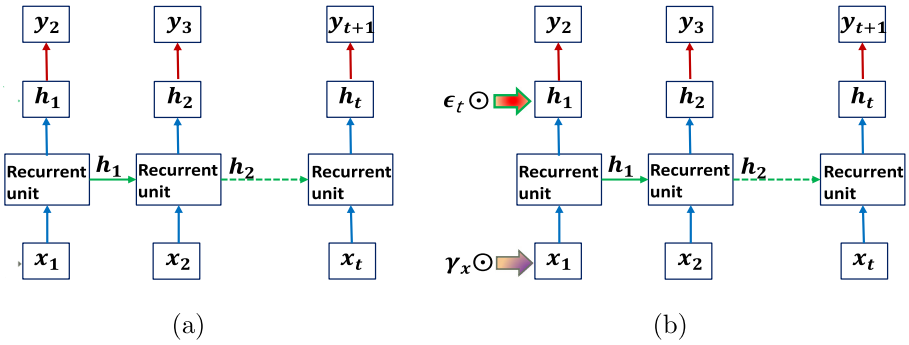

Fig. 1. The overview of our framework with robust RNN to solve the problem of forecasting the medication codes assigned to a patient for his next visit. (a) A conventional RNN model. (b) The proposed model. The input sequential data in regard to a patient ($\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$) are embedded with decay mechanism (γ_x) to model the potential missing pattern, and the stacked recurrent layers with multiplicative noise regularization (ϵ_t) learn the status of the patient at each time stamp. Given the learnt status ($\mathbf{h}_t$), the framework is to generate the codes observed in the next time stamp.

19,714 admission records for 4 modalities which are very useful for monitoring ICU patients (Che et al., 2016). These modalities include input-events (fluids to patients, *e.g.,* insulin), output-events (fluids out of the patient, *e.g.,* urine), lab-events (lab test results, *e.g.,* pH values), and prescription-events (active drugs prescribed by doctors, *e.g.,* aspirin). We use the first 48 h data after admission from each time series, and conduct the predictive ICD-9 code task: predict ICD-9 diagnostic categories (*e.g.,* respiratory system diagnosis) for each admission, which can be treated as a multi-label problem.

For the training on all the models, we use 85% of the patients as the training set, and 15% as the testing set. All the RNN models are trained with 50 epoches *i.e.,* 50 iterations over the entire training data, and then evaluate the performance against the testing set. To avoid over-fitting, we apply the dropout between the GRU layer and the final prediction layer, and also between the multiple stacked GRU layers. The dropout rate is 0.3 and the norm-2 regularization is applied into the weight matrix of $\mathbf{W}_{code}$. The dimensionality of the hidden states $\mathbf{h}$ of the GRU is set to be 2048 to ensure the expressive power. We train the models using truncated back-propagation through time with average stochastic gradient descent (Polyak & Juditsky, 1992). To avoid the problem of exploding gradients, we clip the gradients to a maximum norm of 0.25.

5.2. Evaluation metrics

For the evaluation on the task in a multi-label context, the performance of all methods is evaluated against two metrics: the micro-averaged area under the ROC curve (AUC) and the top-k recall. The measure of AUC treats each instance with equal weight, regardless of the nature of the positive labels for that instance (Bajor & Lasko, 2017), which would not give a score advantage to instances with very prevalent or very rare labels. The micro-averaged AUC considers each of the multiple label predictions as either true or false, and then computes the binary AUC if they all belong to the same 2-class problem, Thus, the micro-average AUC $\mathcal{A}_\mu$ can be defined as

$$\mathcal{A}_\mu = \frac{|(\mathbf{x}, \mathbf{x}', l, l') : f(\mathbf{x}, l) \geqslant f(\mathbf{x}', l'), (\mathbf{x}, l) \in \mathcal{S}, (\mathbf{x}', l') \in \bar{\mathcal{S}}|}{|\mathcal{S}||\mathcal{S}'|} \tag{7}$$

where $\mathcal{S} = \{\mathbf{x}, l\} : l \in Y$ is the set of [instance, label] pairs with a positive label, and $Y = \{y_d : y_d = 1, \ldots, D\}$ is the set of positive labels for the input $\mathbf{x}$.

The top-k recall mimics the behavior of doctors examining differential diagnosis which suggest the doctor is listing most probable diagnoses and treat the patients accordingly to identity the patients status. The top-k recall is defined as

$$\text{top} - k \,\text{recall} = \frac{\#\text{TP in the top k predictions}}{\#\text{TP}}, \tag{8}$$

where #TP denotes the number of true positives. Thus, a machine with high top-k recall translates to a doctor with effective diagnostic skills. In this end, it turns out to make top-k recall a suitable measure for the performance of prediction models on medications.

5.3. Baselines

We consider baselines in two categories: (1) RNN based methods: Doctor-AI (Choi et al., 2016), GRU-Decay (Che et al., 2016), LSTM-ICU (Lipton, Kale, Elkan et al., 2016); MiME (Choi et al., 2018); SRL-RNN (Wang, Zhang et al., 2018); (2) Non-RNN based methods: Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF).

- Doctor-AI (Choi et al., 2016): Doctor AI is a temporal model using RNN to assess the history of patients to make multi-label predictions on physician diagnosis and the next medication order list.
- GRU-Decay (Che et al., 2016): To tackle the missing values in EHR data, GRU-Decay is based on Gated Recurrent Units and exploits the missing patterns for effective imputation and improves the prediction performance.
- LSTM-ICU (Lipton, Kale, Elkan et al., 2016): It is a study to empirically evaluate the ability of LSTMs to recognize patterns in multivariate time series of clinical measurements. They consider multi-label classification of diagnoses by training a model to classify 128 diagnoses given frequently but irregularly sampled clinical measurements.
- MiME (Choi et al., 2018) A Multilevel Medical Embedding (MiME) approach to learn the multilevel embedding of EHR data that only relies on this inherent EHR structure without the need for external labels.
- SRL-RNN (Wang, Zhang et al., 2018) A Supervised Reinforcement Learning with Recurrent Neural Network (SRL-RNN), which fuses them into a synergistic learning framework.
- Logistic Regression (LR): Logistic regression is a common method to predict the codes in the next visit $\mathbf{x}_t$ using the past $\mathbf{x}_{t-1}$. Following Choi et al. (2016), we use the data from L time lags before and aggregate the data $\mathbf{x}_{t-1} + \mathbf{x}_{t-2} +, + \mathbf{x}_{t-L}$ for some duration L to create the feature for prediction on $\mathbf{x}_t$.

Table 2
Comparison results of AUC on the real-valued datasets for multi-task predictions.

Method		Physio-Net	MIMIC-III
RNN	Ours	**0.90**	**0.78**
	Doctor-AI (Choi et al., 2016)	0.77	0.71
	GRU-Decay (Che et al., 2016)	0.84	0.76
	LSTM-ICU (Lipton, Kale, Elkan et al., 2016)	0.76	0.70
	SRL-RNN (Wang, Zhang et al., 2018)	0.86	0.74
Non-RNN	Logistic Regression	0.64	0.66
	SVM	0.71	0.69
	Random Forest	0.71	0.73
	Logistic Regression-mean	0.66	0.67
	SVM-mean	0.72	0.71
	Random Forest-mean	0.72	0.73

- Support Vector Machine (SVM): A multi-label SVM is trained to obtain multiple classifiers to each diagnostic code and each medication category.
- Random Forest (RF): The random forest is not easily constructed to work on sequences, and we represented the input data as bag-of-code vector $b \in \mathbb{R}^D$. As RF cannot be operated on large-size dataset, we break down it into an ensemble of ten independent forests while each one trained on one tenth of the training data, and their averaged score is used for test prediction.

5.4. Results and discussions

Prediction performance. In the first experiment, we evaluate all methods on Physio-Net and MIMIC-III datasets. Table 2 shows the prediction performance of al the models on the multi-task predictions on real datasets: all 4 tasks on Physio-Net and 20 ICD-9 code tasks on the MIMIC-III. The proposed method achieves the best AUC score across all tasks on both the datasets. We notice that all RNN models perform better than non-RNN methods because the deep recurrent layers help these models capture the temporal relationship that is useful in solving prediction tasks. Moreover, explicitly modeling the missing values in both the input signals and the hidden states, such as GRU-Decay and our method, can further improve the prediction results due to the capability of fitting the parameters robust to noisy time-series data.

Table 3 compares the results of the proposed method with different algorithms in three settings: predicting only the diagnosis codes (Dx), predicting only the medication codes (Rx), and jointly predicting both Dx and Rx codes. The experimental results show that the proposed method is able to outperform the baseline algorithms by a noticeable margin. The results also confirm that RNN based approaches achieve superior performance to non-RNN methods. This is mainly because RNNs are able to learn succinct feature representations of patients by accumulating the relevant information from their history visits and the current set of codes, which outperform the hand-crafted features of Non-RNN baselines. Moreover, in the case of missing values and incorrectness in billing codes, our method achieves the best results on all measures in the merit of explicit modeling on billing code variables and robust improvement on recurrence.

To further examine the capability of our method in a real-world medical care setting where patients may have varying lengths of their medical records, we conduct an experiment to study the affect of billing code history duration on the prediction performance. To this end, we select 5800 patients from MIMIC-III who had more than 100 visits. We consider the RNN based deep models to predict the diagnosis codes at visit at different times and calculate the mean values of recall@10 across the selected patient samples. Fig. 2 shows the experimental results of different RNN based models. It can be observed that all methods are increasing their performance on prediction as they see longer patient visit records, and certainly our approach achieved the best prediction performance amongst all RNN-based models. This is mainly because the recurrence is well-suited to time-series and the prediction is more faithful given longer sequence inputs. Also, it is inferred that those patients with high visit count are more likely caught in severely ill, and therefore their future is easier to predict.

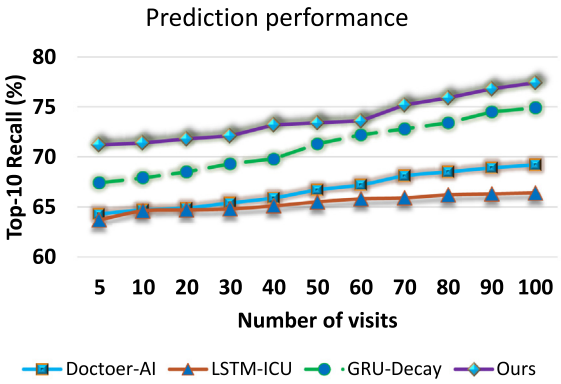

Fig. 2. The prediction performance with respect to the duration of patient medical history.

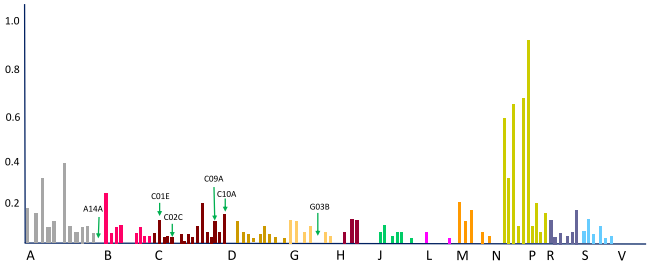

Fig. 3. The medication predictions for a patient with only one ICD-9 code. Each vertical bar represents the prediction for a single medication class and the height indicates the confidence of the prediction. See the texts for details.

More discussions. As the spread σ controls the noise level and determines the amount of regularization into RNN, we discuss on the property of different noise distributions, *i.e.*, Gaussian and Bernoulli, and the impact on the training of RNN. The experimental results are reported in Table 4. It can be found that what really matters with different distributions is the variance σ which determines the degree of regularization into the RNN. And the RNN regularization is not very sensitive to different types of distribution, for example, on both the health-care datasets the AUC values with Gaussian distribution are very similar to Bernoulli while for each specific distribution the spread σ affects the performance.

To further examine the capability of our model in predicting medications in missing billings, we study a case on a patient with Parkinson's disease in which his/her record has at least five years of data consisting of only codes for Parkinson's disease whereas the data contains medications for high cholesterol, hypertension without explicit labels referring to Parkinson's disease. In fact, the medication entities listed as true labels are not suggested for paralysis agitans (Parkinson's disease), while the patient was surely taking them even though not documented into the ICD-9 sequence. As shown in Fig. 3, in the case of missing medication items, the model is still able to predict reasonable medications for

Table 3
Comparison results of accuracy in forecasting future medical activities on the MIMIC-III dataset.

Method		Dx Recall @k			Rx Recall @k			[Dx, Rx] Recall @k		
		$k = 10$	$k = 20$	$k = 30$	$k = 10$	$k = 20$	$k = 30$	$k = 10$	$k = 20$	$k = 30$
RNN	Ours	**71.2**	**77.8**	**85.1**	**77.2**	**86.2**	**92.0**	**59.8**	**73.5**	**80.2**
	Doctor-AI (Choi et al., 2016)	64.3	74.3	79.6	68.2	79.7	85.5	55.0	66.3	72.5
	GRU-Decay (Che et al., 2016)	67.4	75.9	82.6	73.5	83.7	89.0	57.4	69.0	76.1
	LSTM-ICU (Lipton, Kale, Elkan et al., 2016)	63.7	74.3	79.5	68.0	79.1	84.8	54.8	62.9	72.4
	MiME (Choi et al., 2018)	–	–	–	–	–	–	–	–	–
	SRL-RNN (Wang, Zhang et al., 2018)	–	–	–	–	–	–	–	–	–
Non-RNN	Logistic Regression	43.2	54.0	60.8	45.8	60.0	69.0	36.0	46.3	52.5
	SVM	46.2	57.9	65.1	47.8	63.4	69.9	38.0	48.5	56.1
	Random Forest	47.8	58.9	67.2	48.6	63.5	69.8	37.8	48.0	55.0
	Logistic Regression-mean	44.7	55.1	62.4	46.2	60.8	69.7	37.0	46.5	52.8
	SVM-mean	46.8	59.6	66.0	49.2	65.4	71.0	39.8	49.6	57.8
	Random Forest-mean	48.2	59.7	67.3	49.0	65.0	71.1	39.0	48.2	55.7

Table 4
The study on different noise distributions. The micro-averaged AUC values are reported on two datasets.

Distribution	σ	Physio-net	MIMIC-III
Gaussian	0.53	0.82	0.70
	0.92	0.86	0.73
	1.10	0.90	0.78
	1.50	0.87	0.75
Bernoulli	0.33	0.79	0.71
	0.41	0.84	0.72
	0.50	0.89	0.75
	0.80	0.87	0.72

Table 5
A case study: Top prediction and true labels for a patient with Parkinson's disease.

Top predictions		Prob.
N04B	Dopaminergic agents	98.2%
N03A	Antiepileptics	38.4%
N02B	Other analgesics and antipyretics	35.7%
N06A	Antidepressants	31.2%
N02A	Opioids	24.7%
True labels		**Prob.**
C10A	Lipid modifying agents, plain	17.4%
C09A	Ace inhibitors, plain	13.2%
C01E	Other cardiac preparations	7.8%
C02C	Antiadrenergic agents, peripherally acting	3.7%
G03B	Androgens	3.1%
A14A	Anabolic steroids	2.4%

a patient with Parkinson's disease, such as Dopaminergic agents and Antiepileptics, which are primary treatment for the disease. The top prediction probabilities and missing true labels on each treatment regarding a patient are reported in Table 5. Thus, our model is useful to identifying missing medications in the clinical scenario, such as reconciling information in a large scale from a range of electronic and human sources to establish the ground truth of medications that are taken on a particular day.

6. Conclusions and future work

In this paper, we present an effective approach to medicare system, which is a RNN-based deep learning model that can learn robust patient representation from a large amount of longitudinal patient billing code records and predict future medication lists. We demonstrate the effectiveness of our method which achieved improved recall accuracy values in the real medical practice with observed missing values or incorrect records. In the future work, we would strive to improve the performance of the recurrent networks by including additional input data, such as laboratory test results, demographics, and perhaps vital signs related to rare diseases. One interesting direction is to figure out a pathway to convert the medication data into reliably-ordered sequences, so as to fully exploit the strength of recurrent networks for medication prediction.

Acknowledgment

This research was partially supported by NSFC U19A2073.

References

Bajor, J. M., & Lasko, T. A. (2017). Predicting medications from diagnostic codes with recurrent neural networks. In *ICLR*.
Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2016). Recurrent neural networks for multivariate time series with missing values. arXiv:1606.01865.
Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv:1406.1078.
Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., & Sun, J. (2016). Doctor ai: Predicting clinical events via recurrent neural networks. In *Proceedings of the 1st machine learning for healthcare conference*.
Choi, E., Du, N., Chen, R., Song, L., & Sun, J. (2015). Constructing disease network and temporal progression model via context-sensitive hawkes processes. In *ICDM*.
Choi, E., Xiao, C., Stewart, W., & Sun, J. (2018). Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In *NIPS*.
Dieng, A. B., Ranganath, R., Altosaar, J., & Blei, D. M. (2018). Noisein: Unbiased regularization for recurrent neural networks. arXiv:1805.01500.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. In *Neural computation*.
Johnson, A. E., Pollard, T. J., Shen, L., wei H. Lehman, L., Feng, M., Ghassemi, M., et al. (2016). Mimic-iii, a freely accessible critical care database. In *Scientific data*.
Johnson, M. J., & Willsky, A. S. (2013). Bayesian nonparametric hidden semi-markov models. *Journal of Machine Learning Research (JMLR)*, *14*(1), 673–701.
Keogh, C., Kachalia, A., Fiumara, K., Goulart, D., Coblyn, J., & Desai, S. P. (2016). Ambulatory medication reconciliation: Using a collaborative approach to process improvement at an academic medical center. *Joint Commission Journal on Quality and Patient Safety*, *42*(4), 186–194.
Kreindler, D. M., & Lumsden, C. J. (2012). The effects of the irregular sample and missing data in time series analysis. In *Nonlinear dynamical systems analysis for the behavioral sciences using real data*.
Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzel, R. (2016). Learning to diagnose with lstm recurrent neural networks. In *ICLR*.
Lipton, Z. C., Kale, D., & Wetzel, R. (2016). Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Proceedings of the 1st machine learning for healthcare conference*.
Mondal, D., & Percival, D. B. (2010). Wavelet variance analysis for gappy time series. *Annals of the Institute of Statistical Mathematics*, *62*(5), 943–966.
Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, *30*(4), 838–855.
Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120.
SelinCaglar, Henneman, P. L., Blank, F. S., Smithline, H. A., & Henneman, E. A. (2011). Emergency department medication lists are not accurate. *The Journal of Emergency Medicine*, *40*, 613–616.

Semeniuta, S., Severyn, A., & Barth, E. (2016). Recurrent dropout without memory loss. arXiv:1603.05118.

Silva, I., Moody, G., Scott, D. J., Celi, L. A., & Mark, R. G. (2012). Predicting in-hospital mortality of icu patients: The physionetcomputing in cardiology challenge. In *Computing in cardiology*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, *15*(1), 1926–1958.

Vodovotz, Y., An, G., & Androulakis, I. P. (2013). A systems engineering perspective on homeostasis and disease. *Frontiers in Bioengineering and Biotechnology*.

Wager, S., Wang, S., & Liang, P. (2013). Dropout training as adaptive regularization. In *NIPS*.

Wang, Y., Lin, X., Wu, L., & Zhang, W. (2017). Effective multi-query expansions: Collaborative deep networks for robust landmark retrieval. *IEEE Transactions on Image Processing*, *26*(3), 1393–1404.

Wang, Y., Wu, L., Lin, X., & Gao, J. (2018). Multi-view spectral clustering via structured low-rank matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(10), 4833–4843.

Wang, Y., Wu, L., & Zhang, W. (2018). Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering. *Neural Networks*, *103*, 1–8.

Wang, L., Zhang, W., He, X., & Zha, H. (2018). Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *ACM SIGKDD*.

Wang, Y., Zhang, W., Wu, L., Lin, X., Fang, M., & Pan, S. (2016). Terative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clusteringe. In *IJCAI*.

Weiss, J., Natarajan, S., & Page, D. (2012). Multiplicative forests for continuous-time processes. In *NIPS*.

Wells, B. J., Chagin, K. M., Nowacki, A. S., & Kattan, M. W. (2013). Strategies for handling missing data in electronic health record derived data. *EGEMS*, *1*(3).

White, I. R., Royston, P., & Wood, A. M. (2011). Multiple imputation using chained equations: issues and guidance for practice. *Statistics in Medicine*, *30*(4), 377–399.

Zhou, L., & Hripcsak, G. (2007). Temporal reasoning with medical data review with emphasis on medical natural language processing. *Journal of Biomedical Informatics*.