Domain Knowledge Guided Deep Learning with Electronic Health Records

Changchang Yin*, Rongjian Zhao*, Buyue Qian†, Xin Lv*, Ping Zhang‡
*School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, China
Email: {lentery,zrj1120}@stu.xjtu.edu.cn, lvxin1@xjtu.edu.cn
†National Engineering Lab for Big Data Analytics, Xi'an Jiaotong University, Xi'an, Shaanxi, China
Email: qianbuyue@xjtu.edu.cn
‡The Ohio State University, Columbus, Ohio, USA 43210
Email: zhang.10631@osu.edu

***Abstract*—Due to their promising performance in clinical risk prediction with Electronic Health Records (EHRs), deep learning methods have attracted significant interest from healthcare researchers. However, there are 4 challenges: (i) Data insufficiency. Many methods require large amounts of training data to achieve satisfactory results. (ii) Interpretability. Results from many methods are hard to explain to clinicians (e.g., why the models make particular predictions and which events cause clinical outcomes). (iii) Domain knowledge integration. No existing method dynamically exploits complicated medical knowledge (e.g., relations such as *cause* and *is-caused-by* between clinical events). (iv) Time interval information. Most existing methods only consider the relative order of visits from EHRs, but ignore the irregular time intervals between neighboring visits. In the study, we propose a new model, Domain Knowledge Guided Recurrent Neural Networks (DG-RNN), by directly introducing domain knowledge from the medical knowledge graph into an RNN architecture, as well as taking the irregular time intervals into account. Experimental results on heart failure risk prediction tasks show that our model not only outperforms state-of-the-art deep-learning based risk prediction models, but also associates individual medical events with heart failure onset, thus paving the way for interpretable accurate clinical risk predictions.**

***Index Terms*—deep learning, RNN, EHR, knowledge graph, risk prediction**

I. Introduction

There has been a rapid growth in volume and diversity of Electronic Health Records (EHRs) during the last decades, which makes it possible to apply clinical prediction models to improve the quality of clinical care. EHRs are temporal sequence data and consist of diagnosis codes, medications, lab results, and various clinical notes. Patient health information contained in the massive EHRs is extremely useful in different tasks within the medical domain, such as risk prediction [1], computational phenotyping [2], and patient similarity analysis [3]. In this paper, we focus on clinical risk prediction tasks.

Most state-of-the-art clinical risk prediction models are based on deep learning, and trained in an end-to-end way. Recurrent Neural Network (RNN), a popular deep learning model for modeling sequences, has achieved good performance in clinical risk prediction tasks recently [4]–[6]. However, there are still some challenges in the field. (i) Although some existing approaches achieve decent performance in prediction tasks [1], [3], [4], a large number of parameters need more data to train. Most models have limited performances when EHRs data are insufficient, especially for some rare diseases. (ii) Although some existing approaches [5], [6] try to introduce medical domain knowledge, they mainly use the International Classification of Diseases (ICD) code hierarchy to initialize better medical concept embeddings, but do not dynamically exploit complicated medical knowledge (e.g., relations such as *cause* and *is-caused-by* between diseases) for each patient in the whole prediction process. (iii) Most of these approaches lack medical interpretability. It is hard to associate previous individual medical event inputs with later clinical outcomes. (iv) Many existing models [4]–[6] just input EHRs events according to their time order but ignore the time intervals between neighbouring events. These limitations make it difficult to convince doctors for clinical usages. It is crucial to develop robust and interpretable models to combat the limitations.

In this study, we leverage a long short-term memory [7] (LSTM, an RNN architecture) to model the sequence of EHRs entities of each patient, which considers both EHRs medical event series and their time of occurrence. Then, we adopt a graph-based attention mechanism to integrate EHRs information with a public medical knowledge graph KnowLife[1] [8]. With the help of complementary information from this knowledge graph, our model could perform well even if the size of available EHRs data is small. Finally, we use a global max-pooling layer and a fully connected layer to predict a patient's risk for future clinical outcomes. By analyzing the fully connected layer's outputs, the max-pooling layer's outputs and returned indices, our model is able to compute the contribution rate of each timestamp's input medical event, thus paving the way for interpretable clinical risk predictions.

We compare our proposed Domain Knowledge Guided Recurrent Neural Network (DG-RNN) model with both traditional machine-learning methods (e.g., logistic regression) and recent deep-learning methods (e.g., RETAIN, KAME) on heart failure risk prediction tasks. We conduct experiments on both a publicly available MIMIC-III dataset [9] and our proprietary EHRs data. DG-RNN outperforms all the baselines in both datasets and various settings, which demonstrates the

[1]http://knowlife.mpi-inf.mpg.de/

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00084

TABLE I
NOTATIONS USED IN OUR MODEL.

Variable	Description
v_t	The t^{th} medical event
e_t	The embedding of event v_t
p_t	The time encoding for the t^{th} input
$v_{t,m}$	The m^{th} adjacent node of v_t in knowledge graph
$e_{t,m}$	The embedding of $v_{t,m}$
$\alpha_{t,m}$	The attention weight of $e_{t,m}$
g_t	The graph attention result for the t^{th} input
h_{2t-1}, h_{2t}	The LSTM's output vectors for the t^{th} input
C_{2t-1}, C_{2t}	The LSTM's hidden states for the t^{th} input
h_c	The concatenation output of the output vectors
o_g	The global max pooling output
r_i	The i^{th} patient's disease risk score
y_i	The i^{th} patient's disease risk probability
Q_t	The contribution risk of the output vector h_t
$\mathbf{CR}_t$	The contribution rate of medical event v_t

effectiveness of the proposed model. Moreover, after DG-RNN is well trained, it is used to find the EHRs events with high contribution rates to heart failure. Our results show that the selected events are well aligned with domain knowledge, which demonstrates the interpretability of DG-RNN.

In sum, our contributions are as follows:

- We develop a DG-RNN deep learning framework, which introduces knowledge graph into the risk prediction model via a dynamic attention mechanism. The new framework is able to accurately predict clinical risks, even if the patients' EHRs data are insufficient.
- We leverage a global pooling operation to make our model interpretable. The model can output the contribution of each input medical event to the clinical outcome.
- We introduce time encoding to consider the irregular time intervals between medical events, which are important for many medical applications.
- We evaluate DG-RNN on 2 real-world EHRs datasets for the heart failure risk prediction problem. Experimental results show our model not only outperforms state-of-the-art predictive models but also identifies medical events relevant to heart failure onset.

The rest of the paper is organized as follows. In Section II, we describe our model in detail. In Section III, we conduct experiments on two real-world EHRs datasets. We review the related studies in Section IV. Section V concludes our work.

II. METHOD

In this work, we present a new domain knowledge guided RNN model (DG-RNN) to predict clinical risk, as is shown in Figure 1. The inputs of DG-RNN consist of the medical events and their time of occurrence. For example, at t^{th} step of DG-RNN, the embedding e_t of a medical event and its time encoding p_t are sent to LSTM. The LSTM generates an output vector h_{2t-1} and a hidden state C_{2t-1}. Then, the knowledge graph attention module takes the sub graph adjacent to the t^{th} event and C_{2t-1} as inputs, and generates an attention vector g_t, which is input to the LSTM again and another output vector h_{2t} is produced. Note that the unit of our model produces two output vectors for one input event, which can help to compute the contribution rates of the initial medical event and the potential information supplemented by medical knowledge graph respectively. Next, all the output vectors are concatenated and a global pooling operation is followed. At last, we use a fully connected layer to predict the clinical risk.

A. Basic Notations

In the work, each patient's EHRs data consist of a sequence of visits which include several different medical events. Following previous studies (e.g., [3]), we sort all the medical events according to their time of occurrence. Rather than embedding the medical concepts from EHRs and knowledge graph in two different ways [5], [6], [10], we map all the concepts into the same feature space. For each patient i, we denote $V_i = \{v_1, v_2, ..., v_t, ..., v_{|V_i|}\}$ as his/her sequence of medical events and y_i^* as the ground truth that whether the patient will be diagnosed with a given specific disease after the hold-off window. The embeddings of the patient's medical events are denoted as $E_i = \{e_1, e_2, ..., e_t, ..., e_{|E_i|}\}$, where $e_t \in R^d$. We display some important notations in Table I.

B. Time Encoding.

In order for the model to make use of the time intervals of EHRs events, we infuse the time information into the model. When the event embeddings are sent to LSTM, we simultaneously input "time encoding" to the model. Our time encoding is similar to the position encoding in Transformer [11]. Firstly, we computes each event's relative time to the criterion operation date and the time interval between neighboring events. Then, we use sine and cosine functions of the time intervals to present the time encoding for the t^{th} event:

$$p_{t,4j} = \sin\left((date_o - date_t)/10000^{j/d}\right)$$

$$p_{t,4j+1} = \cos\left((date_o - date_t)/10000^{j/d}\right)$$

$$p_{t,4j+2} = \cos\left((date_t - date_{t-1})/10000^{j/d}\right)$$

$$p_{t,4j+3} = \cos\left((date_t - date_{t-1})/10000^{j/d}\right)$$

$$0 \leq j \ < \ d$$

where $date_o$ denotes the criterion operation date, $date_t$ denotes the t^{th} event's date, and $p_t \in R^{4d}$ denotes the time encoding vector, and j is the dimension of EHRs event embeddings. The lengths of generated time encoding vectors are four times of the medical event embedding vectors. The wavelengths form a geometric progression from 2π to $10000 * 2\pi$. This function selection is followed to the position encoding of Transformer [11]. The main difference from position encoding is that our time encoding considers two kinds of time intervals while Transformer encodes the absolute positions into vectors. Having the medical event embedding and the time encoding, we input both vectors into LSTM.

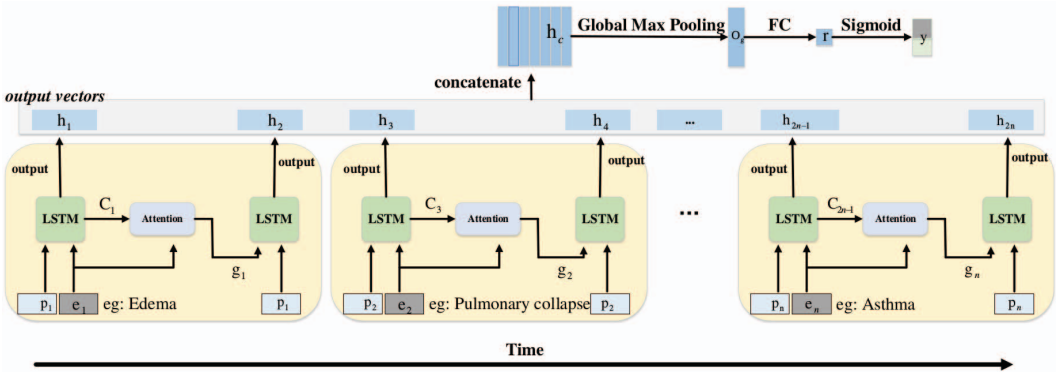


Fig. 1. Framework of DG-RNN. The model's inputs include the medical event embeddings (e_1, e_2, ..., e_n) and the corresponding time encoding vectors (p_1, p_2, ..., p_n). For each event input (e.g., h_n), DG-RNN generates two output vectors (e.g., h_{2n-1} and h_{2n}). All the output vectors are concatenated and then a global max pooling operation is followed. Finally, a fully connected layer (FC) and a sigmoid function are used to predict the clinical risk of disease.

C. LSTM Architecture

Because of the various lengths of clinical event sequences from different patients, RNN are suitable for modeling EHRs data. Moreover, the middle hidden states of RNN are helpful for attending the knowledge graph information. Given medical event embedding and time encoding vectors, we build our model based on LSTM [7] for its ability to recall long term information. The LSTM model can be described as follows:

$$\begin{aligned} i_t &= \sigma(W_i\hat{e}_t + W_{it}\hat{p}_t + U_i h_{t-1} + b_i) \\ f_t &= \sigma(W_f\hat{e}_t + W_{ft}\hat{p}_t + U_f h_{t-1} + b_f) \\ o_t &= \sigma(W_o\hat{e}_t + W_{ot}\hat{p}_t + U_o h_{t-1} + b_o) \\ C_t &= \sigma(W_{ce}\hat{e}_t + W_{ct}\hat{p}_t + U_c h_{t-1} + b_c) * i_t + C_{t-1} * f_t \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{1}$$

where σ is the sigmoid function, t denotes the t^{th} step of LSTM, and C_t is corresponding hidden state, and h_t is the output vector. $W_i, W_f, W_o, W_{ce} \in R^{k\times d}$, $W_{it}, W_{ft}, W_{ot}, W_{ct} \in R^{k\times 4d}$, $U_i, U_f, U_o, U_c \in R^{k\times d}$, $b_i, b_f, b_o, b_c \in R^k$ are learnable parameters of the LSTM. $\hat{p}_t$ is the time encoding vector. $\hat{e}_t$ is the input event embedding. For each event, for example the n^{th} event, we input the embedding e_n and the graph attention result g_n respectively, and obtain two output vectors (h_{2n-1} and h_{2n}). Thus, if t is odd number, $\hat{e}_t = e_{(t+1)/2}$. Otherwise, $\hat{e}_t = g_{t/2}$. We assume that g_n and e_n have the same time encoding. Therefore, every time encoding vector is used twice, $\hat{p}_t = p_{(t+1)/2}$.

D. Knowledge Graph Attention Mechanism

A knowledge graph is used to dynamically introduce medical domain knowledge. We embed different relations (e.g., *causes* and *is-caused-by*) and entities (e.g., *diagnosis*) of the knowledge graph into d dimension feature space. Given the t^{th} input event v_t, we denote the relations of v_t in knowledge graph as $R_t = \{(r_{t,1}, v_{t,1}), (r_{t,2}, v_{t,2}), ..., (r_{t,|R_t|}, v_{t,|R_t|})\}$. The attention mechanism is designed to automatically focus on useful related tail entities and to find some potential information. Formally, it takes as input the hidden state C_{2t-1} of the LSTM and the related relations R_t, and then generate corresponding weights as follows:

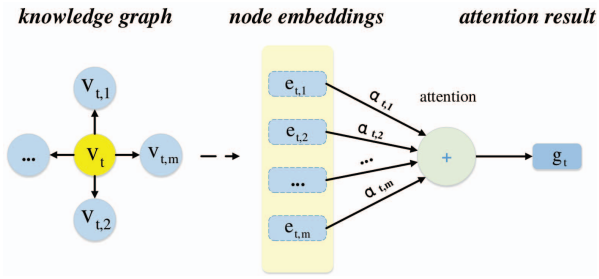


Fig. 2. Attention mechanism. The left part of the figure is a sub graph of medical knowledge graph. The node v_t means the current input medical event. Other nodes (e.g., $v_{t,1}$ and $v_{t,2}$) are the adjacent nodes of v_t. All the embeddings of the adjacent nodes (e.g., $e_{t,1}$ and $e_{t,2}$) are used to compute the graph attention vector g_t.

$$\alpha_{t,m} = \frac{exp(\beta_{t,m})}{\sum_{j=1}^{|R_t|} exp(\beta_{tj})} \tag{2}$$

$$\beta_{t,m} = (W_r r_{t,m})^{\mathrm{T}} tanh(W_h e_t + W_a C_{2t-1} + W_t e_{t,m})$$

where $W_r, W_h, W_a, W_t \in R^{d\times d}$ are learnable parameters, and $r_{t,m}$, $e_{t,m} \in R^d$ are the relation and tail entity embeddings. Given the weights, a soft attention is used to produce the vector g_t, as shown in Figure 2. Then g_t is input to the LSTM, as shown in Figure 1.

$$g_t = \sum_{m=1}^{|R_t|} \alpha_{t,m} e_{t,m} \tag{3}$$

E. Global Max Pooling Operation

RNN-based models are sometimes inefficient due to their long-term dependency. When the input sequence is too long, it is easy for the models to forget the earlier data. Therefore, we adopt a global pooling operation to shorten the distance between the earlier inputs and the final outputs. As is shown in Figure 1, all the outputs of the LSTM are concatenated and then a global pooling operation is followed. The output o_g is

fed through the fully connected layer to produce the clinical risk of patient i, which is defined as:

$$r_i = W_s o_g + b_s \tag{4}$$
$$y_i = sigmoid(r_i)$$

where $W_s \in R^k$ and $b_s \in R$ are the learnable parameters, r_i and y_i denote the clinical risk score and probability respectively. Because of the shortened distance between the inputs and the outputs, the pooling operation makes it more efficient to propagate the gradients. Besides, the global pooling operation is useful to compute the contribution rates of the outputs and their corresponding input medical events.

F. Objective Function

Based on Eq. (4), we use the cross-entropy between the ground truth y_i^* and the predicted result y_i to calculate the loss for each patient as follows:

$$L(y_i, y_i^*) = -(y_i^* \log(y_i) + (1 - y_i^*) \log(1 - y_i)) \tag{5}$$

Note that in our implementation, we use the average loss of batch patients each time to optimize the model. Algorithm 1 describes the overall training process of our proposed DG-RNN.

G. Interpretability

Interpretability is very important for machine learning models of clinical applications. The global pooling operation leveraged in our architecture is able to associate the contribution of each input medical event to the final clinical outcome, paving the way for interpretable clinical risk predictions.

In Figure 1, given the output vectors, the global max pooling layer is followed and produces the patient feature vector h_c, which is used to predict clinical risk. The max pooling operation can return the indices of each dimension number in h_c. It means we can track the vectors which provide specific elements of h_c. After the fully connected layer, we can calculate every dimension's contribution risk, as in show in the upper right part of Figure 3. We assume that every dimension's contribution risk of h_c attributes to the dimension's corresponding output vector. Every output vector's contribution can roughly represent the corresponding input EHRs event's contribution. Because DG-RNN produces two output vectors for one input event, we should sum the corresponding two output vectors' contribution rate for the input event. For a case patient, the contribution rate of the t^{th} medical event is the sum of the two output vectors' (h_{2t-1} and h_{2t}) contribution rates, which is calculated as follow:

$$\mathbf{CR}_t = \frac{Q_{2t-1} + Q_{2t}}{\sum_{j=1}^{2n} max(Q_j, 0)} \tag{6}$$

where Q_j denotes the j^{th} output vector's contribution to the risk score r_i in the Eq. (4).

Figure 3 gives a toy example for illustrating the interpretability of DG-RNN. As is shown in Figure 1, the LSTM generates a lot of output vectors h_*. Our model generates

Algorithm 1 DG-RNN Model

Input: Patient' EHRs data and medical knowledge graph
Output: Clinical risk y

1: Randomly initialize basic embedding matrix of medical events E, LSTM parameters W_i, W_{it}, W_f, W_{ft}, W_o, W_{ot}, W_{ce}, W_{ct}, U_i, U_f, U_o, U_c, b_i, b_f, b_o and b_c, attention parameters W_r, W_h, W_a and W_t, fully connected layer parameters W_s and b_s.
2: **repeat**
3: $V_i \longleftarrow i^{th}$ patient's EHRs data;
4: **for** event v_t in V_i do **do**
5: Obtain the embedding of v_t, represented as e_t;
6: Obtain the time encoding vector, represented as p_t;
7: Input e_t and p_t to LSTM and compute the hidden state C_{2t-1} and output vector h_{2t-1} according to the Eq. (1);
8: Obtain the relations of v_t in knowledge graph G, represented as R_t;
9: Calculate the attention weights $\alpha_{t,m}$ of the relations R_t according to the Eq. (2);
10: Compute the attentional vector g_t according to the Eq. (3);
11: Input g_t and p_t to LSTM and compute the hidden state C_{2t} and the output vector h_{2t} according to the Eq. (1);
12: **end for**
13: Obtain vector h_c by concatenating the output vectors;
14: Obtained vector o_g by applying max pooling over h_c;
15: Make prediction y using the Eq. (4);
16: Calculate the prediction loss L using the Eq. (5);
17: Update parameters according to the gradient of L;
18: **until** convergence

$2n$ (varying from 100 to 200 in our proprietary EHRs experiments) output vectors for each patient and the vectors are 512-dimensional. In Figure 3, in order to illustrate the interpretability clearly, we just display 2 input events and 4 corresponding 6-dimensional output vectors (h_1, h_2, h_3, h_4). Given h_c and fully connected parameters, the output risk is computed (2.30). The first dimension's contribution risk is 0.21 and the contribution rate is 9.1%, which comes from the fourth output vector h_4. Similarly, the fifth dimension's contribution rate also comes from h_4. Thus, the contribution rate of the fifth vector h_4 is computed by summing the two contribution rates. Then, we compute the contribution rate of the input e_2 by summing the contribution rates of h_3 and h_4, $\mathbf{CR}_2$ = 36.9%.

Besides, because the odd and the even numbered inputs of LSTM is EHRs event embedding (e.g., e_n) and knowledge graph attention result (g_n), we assume that the contribution rates of all the odd numbered vectors, like h_{2n-1}, comes from EHRs data, and the contribution rates of all the even numbered output vectors, like h_{2n}, comes from knowledge graph data. Therefore, we can compute the contribution rates of EHRs and medical knowledge graph, by summing the corresponding

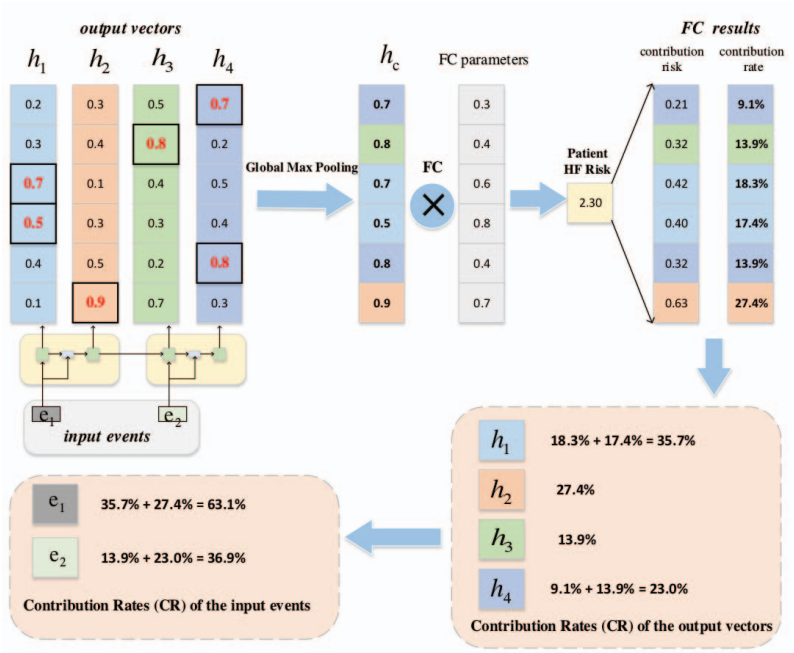


Fig. 3. A toy example for illustrating the interpretability of DG-RNN. We display two input events' embeddings (e_1 and e_2) and four corresponding output vectors (h_1, h_2, h_3, h_4) of LSTM in Figure 1. After global max pooling layer and fully connected layer (FC), DG-RNN outputs the patient's heart failure (HF) risk (2.30). Then the contribution risks of the six dimensions of h_c are computed. The six dimensions' contribution risks come from the four output vectors (h_1, h_2, h_3, h_4). Therefore, each output vector's contribution risk is calculated by summing the corresponding dimensions' contribution risks. Finally, the input events' contribution rates are calculated according to Eq. (6).

TABLE II
STATISTICS OF DATASETS.

	MIMIC-III	EHR-120	EHR-90	EHR-60	EHR-30	EHR-14	EHR-7
number of case patients	425	442	462	494	517	536	554
number of control patients	1275	1326	1386	1482	1551	1608	1662
number of events	20528	134666	140984	152389	160584	169636	176460
number of unique events	2410	967	974	978	983	989	995
average EHRs length	12.07	76.17	76.29	77.11	77.65	79.12	79.62
average event number per visit	9.47	2.17	2.36	2.29	2.41	2.35	2.39

output vectors' contribution rates.

III. EXPERIMENTS

In order to evaluate the effectiveness of our model, we compare DG-RNN with some state-of-art methods on heart failure risk prediction tasks. The experiments are conducted on two different real-world EHRs datasets.

A. Datasets

The first dataset is extracted from a real-world proprietary EHRs database. Firstly, we select the patients diagnosed with heart failure as case patients. Then, for each case, we select 3 control patients according to their *year of birth* and *gender*. For every selected case, we set an operation criterion date, which is the heart failure confirmation date for the case patient. Every control's criterion date is the same as that of his/her corresponding case. Finally, we trace back from the operation criterion date, hold off the EHRs records in a prediction window. There are various hold-off windows: 7, 14, 30, 60, 90 and 120 days. Following previous studies (e.g., [3]), we prepare the data by concatenating every patient's the medical events according to the time of occurrence (we ignore the orders of medical events with the same time-stamps). Thus, every patient's data are represented as a sequence of medical events along with their time of occurrence. We find that the EHRs lengths (the lengths of patients' event sequences) of controls are much longer than that of the cases. All the RNN based models can easily classify them just by the length of EHRs data rather than the clinical meaning of the medical events in EHR. Therefore, we select all the controls and cases with the similar EHRs lengths. In our experiments, the EHRs lengths of the chosen patients are in [50, 100]. Although we select the patients according to EHRs lengths, DG-RNN can handle any length of EHRs data via LSTM.

The second dataset is a public dataset MIMIC-III [9], which includes thousands of ICU patient data. We select the patients data in the same way as the first dataset. The main difference

is that we select all the controls and cases with EHRs length in [10, 20]. Because most case patients' EHRs lengths are less than 20, we set the upper bound as 20. Besides, due to the short lengths of EHRs data in MIMIC-III, we don't set hold-off window and use all events before the operation criterion date to predict heart failure risk. Thus, the task becomes whether the patients will be diagnosed with heart failure in the next visit.

For both datasets, we remove the medical events which appear less than 10 times in the datasets. Because most events in the datasets are diagnosis, we also remove other types of events (e.g. medications), but our model can handle different kinds of medical events. The statistics of the selected datasets are listed in Table II.

B. KnowLife: A Public Knowledge Graph

We leverage KnowLife [8], a knowledge graph that consists of millions of entities and dozens of relationships, in our experiments. Each entity has different properties and a unique code CUI, which can be converted into ICD9 code for disease entities. In our model, we use all the *disease* entities, and their relations to construct a sub-graph. Because the aforementioned datasets only include diagnosis codes, we select two relations between diagnosis entities in KnowLife, which are *causes* and *is-caused-by*. The embeddings of the entities and relations are initialized with TransE [12], which is implemented by [13] [2]. When training our model, the embeddings of entities are fine-tuned. Although only two relations are used in the experiments, DG-RNN can handle different kinds of relations due to the relation embedding and the attention mechanism.

C. Methods for Comparison

To validate the performance of the proposed framework for risk prediction task, we implement the following models, including three traditional machine learning methods, five deep learning methods and three versions of our model.

Random Forest (RF): We compute the counts of each medical events for each patient and normalize the vectors to zero mean and unit variance. The resulting vector is used to train the Random Forest model.

Logistic Regression (LR): We train the logistic regression model with the same vectors as random forest.

Support Vector Machine (SVM): We train the support vector machine model with the same vectors as random forest. The support vector machine is trained with four different kernels, including *poly*, *rbf*, *linear* and *sigmoid*. The kernel with the best performance in validation set is used to predict the risk in test set.

GRU and **LSTM**: GRU [14] and LSTM [7] are classical RNN based models, which both introduce various gates to improve RNN's performance.

RETAIN: The REverse Time AttentIoN model (RETAIN) [4] is the first work that tries to interpretate model's disease risk prediction results with two attention modules. The attention modules generate weights for every medical events. The weights are helpful to analyze different events' contributions to the output risk.

GRAM: GRAM [5] uses a medical knowledge graph to learn the medical event representations. Better representations can help predict the future visits information.

KAME: KAME [6] proposes an attention mechanism to exploits general knowledge to improve the prediction accuracy.

DG-RNN: DG-RNN is our proposed model which introduces medical domain knowledge into RNN by using a knowledge graph attention module. A global max pooling operation is introduced to shorten the distance between the earlier EHRs records and the outputs, and help to interpretate the model's output.

DG-RNN-nk: DG-RNN-nk, which does not use medical knowledge graph and the attention module, is a variant version of DG-RNN. It can help to validate the effectiveness of our attention mechanism.

DG-RNN-np: In order to evaluate the global max pooling operation's effectiveness, by removing the global pooling operation, we implement the last version DG-RNN-np, which predicts the risk based on the last output vector of RNN.

The three traditional methods are implemented with scikit-learn[3]. A grid search is adopted to find the best parameter settings. Besides, note that GRAM and KAME originally aim at predicting all diagnosis codes in the next visit. Therefore, we have to modify the two baselines to adapt to the risk prediction task. For a fair comparison to our proposed method, we use KnowLife as the medical domain knowledge for both of them.

D. Implementation Details

We implement all the baselines and our proposed DG-RNN models with PyTorch 0.4.1[4]. For training models, we use Adam optimizer with a mini-batch of 64 patients. We train on 1 GPU (TITAN XP) for 50 epochs, with a learning rate of 0.0001. We randomly divide the datasets into 10 sets. All the experiment results are averaged from 10-fold cross validation, in which 7 sets are used for training every time, 1 set for validation and 2 sets for test. The validation sets are used to determine the best values of parameters in the training iterations. We use the area under the receiver operating characteristic curve (AUROC) in the test sets as a measure for comparing the performance of all the methods in two datasets. We use 512-dimensional embeddings to represent entities. We use BCELoss as loss function.

E. Results of Risk Prediction

As is shown in Table III, we can observe that DG-RNN achieves the best performance compared with all the baselines, which demonstrates the effectiveness of the proposed model.

The overall performance of traditional machine-learning approaches is worse than the deep learning approaches. We speculate there are two possible reasons. The first is the difference in the representation of medical events. The traditional approaches use high-dimensional one-hot representation

[2]https://github.com/thunlp/OpenKE

[3]https://scikit-learn.org/stable/

[4]https://pytorch.org/

TABLE III
AUROC OF THE HEART FAILURE PREDICTION TASK.

Model	MIMIC-III	EHR-120	EHR-90	EHR-60	EHR-30	EHR-14	EHR-7
LR	0.6993	0.6883	0.6956	0.6932	0.7139	0.7347	0.7386
RF	0.6946	0.6726	0.6913	0.6965	0.7212	0.7217	0.7336
SVM	0.6501	0.6173	0.6339	0.6213	0.6258	0.6323	0.6372
GRU	0.7231	0.6504	0.6670	0.6939	0.7178	0.7438	0.7638
LSTM	0.7133	0.6628	0.6792	0.6982	0.7282	0.7459	0.7631
RETAIN	0.7049	0.6962	0.7115	0.7318	0.7437	0.7561	0.7683
GRAM	0.7232	0.7081	0.7292	0.7378	0.7525	0.7648	0.7656
KAME	0.7269	0.7168	0.7319	0.7392	0.7573	0.7662	0.7717
DG-RNN-nk	0.7238	0.7158	0.7310	0.7368	0.7486	0.7583	0.7663
DG-RNN-np	0.7051	0.6995	0.7075	0.7182	0.7425	0.7596	0.7723
DG-RNN	**0.7375**	**0.7288**	**0.7437**	**0.7510**	**0.7663**	**0.7789**	**0.7863**

TABLE IV
THE AVERAGE EPOCHS FOR THE MODELS TO CONVERGE.

Model	EHR-120	EHR-60	EHR-30
DG-RNN-nk	5.6	5.3	4.8
DG-RNN-np	22.1	15.4	11.9
DG-RNN	10.5	7.6	6.6

while the deep learning approaches adopt medical concept embedding by mapping each concept into a relatively low-dimensional vector, which can represent the clinical meaning of the medical concept. The second possible reason is those deep learning methods are better to model the high-dimensional and sparse data for the task of risk prediction.

Among the five deep learning baselines, KAME and GRAM perform better than other models, which can demonstrate that medical knowledge is useful in clinical applications.

Among the three versions of the proposed models, DG-RNN is our main model and achieves the best performance. Without the help of medical knowledge graph, DG-RNN-nk' performance becomes worse than that of DG-RNN, which demonstrates that the introduced medical knowledge is very helpful. The performance of DG-RNN-np is also worse than that of DG-RNN, which demonstrates that the global pooling operation is useful to improve the model's performance.

DG-RNN outperforms GRAM and KAME, which also introduce medical knowledge and leverage RNN for the risk prediction. Compared to our proposed DG-RNN: GRAM just uses knowledge graph to initialize a static embedding, but not to dynamically attend to sub-graph; KAME only dynamically attends to the sub-graph in the last admission, but not fully utilizes all the previous attention information. Moreover, our DG-RNN applies global pooling operation over the output vectors of RNN, which further improves the performance.

Additionally, we find that global pooling operation can accelerate the convergence rate, which can be demonstrated in Table IV. We compare three versions of our models, DG-RNN, DG-RNN-nk and DG-RNN-np in their convergence rates. The experiments are conducted in three settings of our proprietary EHRs datasets. We can observe that DG-RNN-nk converges fastest. Because DG-RNN has an extra knowledge graph attention module and more parameters, it needs more epochs to convergence. However, compared with DG-RNN-np, DG-RNN converges much faster with global pooling operation. The reason may be that the global pooling operation shortens the distance between the earlier EHRs records and the outputs, so the parameter propagation becomes more efficient.

F. Contribution Rate Analysis

Apart from the superior performance, another advantage of DG-RNN is its interpretability. DG-RNN can be used to analyze different medical events' contributions to the risk of clinical outcomes. In this subsection, we firstly show a case study to evaluate the interpretability, and then analyze the contributions of various events and knowledge graph.

Case Study. We apply DG-RNN-nk and DG-RNN to predict the heart failure (HF) risk of a patient from test set, who has been diagnosed with heart failure later. Figure 4 (a) and Figure 4 (b) respectively show the DG-RNN-nk's and DG-RNN's prediction results and contributions of prior medical events to HF. In the figure, we just display those events with relatively high absolute value of contribution. Given a patient i, the HR risk score r_i is computed with the Eq. (4). The score is also the sum of all the input events' contribution, which is in $(-\infty, \infty)$. With a sigmoid function, we can obtain the patient's future HF probability $y_i = sigmoid(r_i)$. In our experiments, we set the threshold of HF risk score as 0. Only the positive risk score means that the patient will be diagnosed with HF. In Figure 4, only DG-RNN is able to correctly predict the patient's heart failure with a positive score of +0.27, compared with DG-RNN-nk's -0.71. Because we input the EHRs event embedding and graph attention result to DG-RNN respectively, DG-RNN can calculate the event's and the attention module's contribution rates. The red color head of each bar in Figure 4 (b) means the contribution of knowledge graph. The sum of knowledge graph's contributions is 0.09 (i.e., the sum of the red bar), which demonstrates that knowledge graph does provide some useful information for HF risk prediction. Figure 4 (c) show that how the models' output HF risks change when the medical events are input to models along the time. In the last 170 days prior to HF confirmation date, DG-

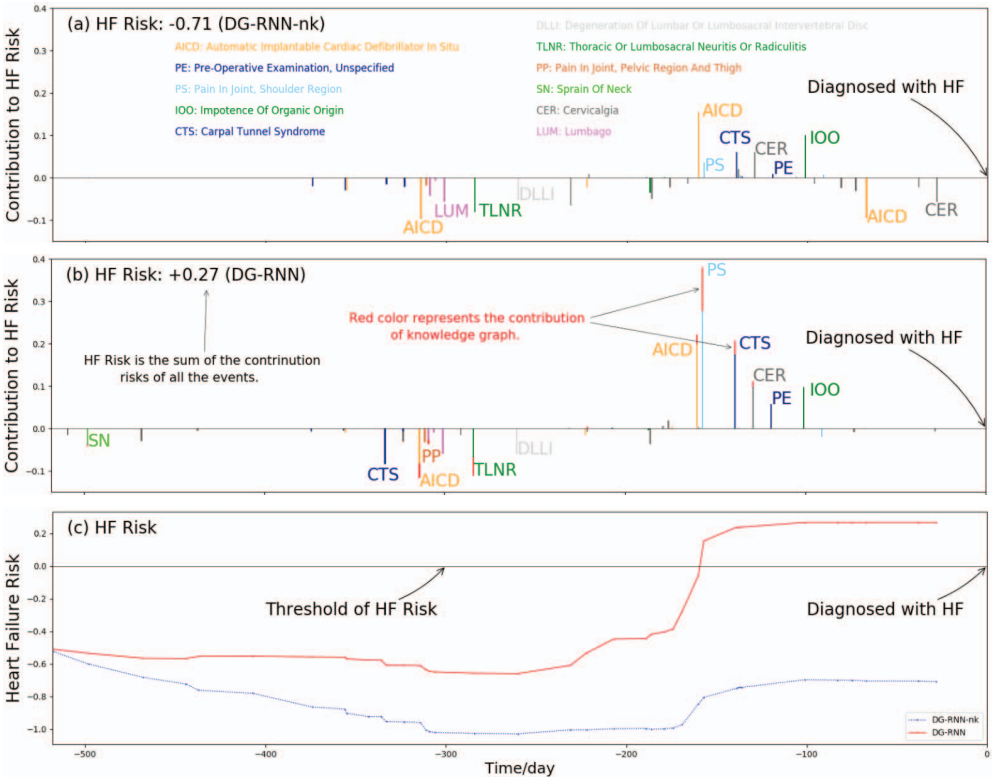


Fig. 4. (a) DG-RNN-nk's contribution of medical events to patient's clinical outcome risk. The x-axis means the time before the heart failure (HF) confirmation date, while the y-axis indicate the magnitude of each event's contributions to HF risk prediction. (b) DG-RNN's contribution of medical events to patient's clinical outcome risk. We introduce knowledge graph information to our DG-RNN. The red head of each bar means the knowledge graph attention module's contribution risk. (c) Two models' output clinical risks along the time.

TABLE V
TOP 10 MEDICAL EVENTS WITH THE HIGHEST AVERAGE CONTRIBUTION RATES (**AVG-CR**) TO HF, THEIR DISTANCE TO HF IN KNOWLIFE. THE DISTANCE "-" MEANS THERE IS NO CONNECTED PATH BETWEEN THE CORRESPONDING MEDICAL EVENT AND HEART FAILURE.

Name	Distance	AVG-CR
Obstructive sleep apnea	3	0.1393
Proliferative diabetic retinopathy	-	0.1360
Macular degeneration NOS	4	0.1331
Long-term use anticoagul	-	0.1283
Atrial flutter	4	0.1280
Malignant neoplasm bronchus	-	0.1108
Acute respiratory failure	-	0.1062
Cardiovascular abnormal function	-	0.0991
End stage renal disease	2	0.0963
Thrombocytopenia NOS	3	0.0937

RNN begins to correctly predict the positive risk score of HF, while DG-RNN does not. It demonstrates that after introducing medical knowledge graph, DG-RNN becomes more competent in clinical risk predictions.

Average Contribution Rate (AVG-CR) of Clinical Events. After obtaining the contributions of every patient's events, we compute every unique event's average contribution rate (**AVG-CR**) on population basis. For each medical event, the average contribution rate is obtained by averaging its contribution rates to different patients, whose EHRs data contain the event. Table V displays the EHR-7 setting's top 10 medical events with the highest contributions to heart failure (HF), their AVG-CR and distances between HF and the events in KnowLife. The distance "-" means there is no connected path between the corresponding medical event and HF. We can observe that in Table V there are 5 medical events near HF in our knowledge graph, while the other events are not directly connected with HF. It shows that our model considers the initial EHR data and the medical domain knowledge simultaneously. Besides, the top 10 events contain risk factors of HF (e.g., sleep apnea), and some common complications (e.g., renal disease), which aligns well with clinicians' knowledge[5].

Contribution of Knowledge Graph. DG-RNN is also able to analyze the overall contribution of knowledge graph to the prediction. For each patient, the initial EHR's contribution rate is obtained by summing the contribution risks of the odd numbered output vectors $\{Q_{2t-1}\}$, where $t \in \{1, 2, ..., n\}$. Similarly, we can get the knowledge graph's contribution rate by summing $\{Q_{2t}\}$. Figure 5 shows the average contribution rates across all the patients in test set and their changing

[5]https://www.mayoclinic.org/diseases-conditions/heart-failure/symptoms-causes/syc-20373142

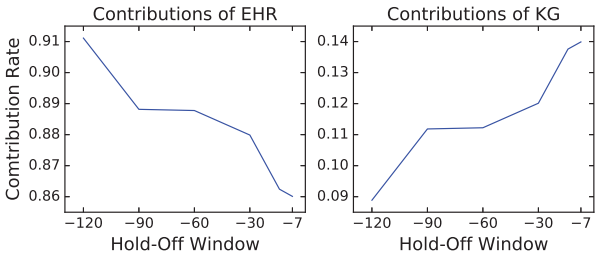


Fig. 5. Average contribution rates of different hold off windows across all the patients in test set. The two figures show the average contribution rates of Electronic Health Records (EHR) and Knowledge Graph (KG) respectively.

trend over the hold-off window size. It is obvious that the smaller the hold-off window is, the higher the knowledge graph's contribution rate is. We can also observe the same trend of DG-RNN's performance in Table III. When the hold-off window becomes small, more medical events related with heart failure appear in case patients' EHR. Knowledge graph can provide more information based on the related events. Thus, the performance gap between DG-RNN and DG-RNN-nk becomes larger, as is shown in Table III.

G. Models' Performances with Varying Training Data Size

To evaluate the various models' performances when training data are insufficient, we randomly remove some patient data in the training set. Figure 6 shows the AUC of different models' heart failure prediction for increasing data size on the two datasets EHR-120 and EHR-7. We can observe that in general, deep learning approaches' overall performances are better than that of traditional machine learning approaches. Besides, compared with the data-driven methods (RETAIN, LSTM, GRU), the knowledge-combined methods (DG-RNN, KAME, GRAM) have less performance degradation when conducting experiments on smaller datasets, which confirms our assumption that introducing medical domain knowledge can alleviate the data insufficiency problem. Moreover, when the training data size is less, the performance gaps between DG-RNN and baselines (including KAME and GRAM) are larger, which demonstrates that the proposed dynamical attention mechanism is better to utilize the complementary information of knowledge graph than the attention mechanisms of GRAM and KAME.

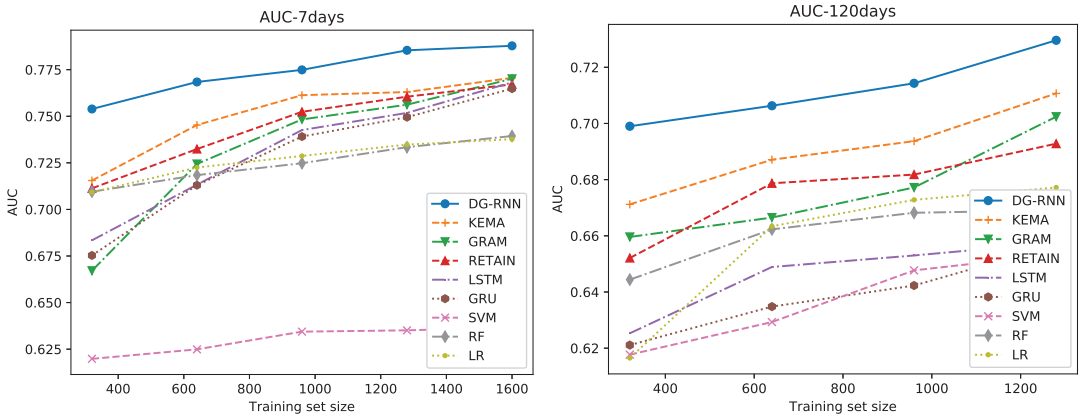
IV. Related Work

In recent years, with the rapid development of deep learning, many deep learning models, such as convolutional neural networks (CNN) [10], [15], [16] and recurrent neural networks (RNN) [17], [18] have shown their superior ability for diverse prediction tasks. In this section, we mainly focus on RNN based models. For example, RNN is successfully applied in modeling sequential EHRs data to predict diagnosis [19], [20]. Besides, RNN can be used for patient subtyping [2], modeling disease progression [21], and mining time series healthcare data with missing values [17], [22]. For some RNN based approaches, the relationships between subsequent visits are usually ignored. To address the issue, Dipole [23] adopts bidirectional recurrent neural networks (BRNNs) with different attention mechanisms and significantly improves the prediction accuracy. When preprocessing the EHRs data, most existing models ignore the time intervals between neighbouring medical events. However, the time intervals are common and important in many healthcare applications. Therefore, a time-aware patient subtyping model [2] is proposed to handle irregular time intervals in longitudinal patient records. It is demonstrated that taking time intervals into account can significantly improve the model's performance. In this study, DG-RNN also considers the time intervals with a time encoding operation.

In order to pursue better performance, many models attempt to introduce medical domain knowledge. GRAM [5] and KAME [6] both use attention mechanism to supplement medical knowledge into their models. GRAM takes knowledge DAG as a knowledge prior to cope with data insufficiency and learn medically interpretable representations to make accurate predictions. It can be regarded as a static attention mechanism cause different patient shares the same embedding in the graph. KAME learns meaningful and interpretable medical code representations on the given knowledge graph for making accurate predictions. KAME's attention results are only used to predict the risk, but not to represent patients' health state. Thus each time, KAME can only utilize the medical knowledge related to the last visit, but not all the previous visits. Attention mechanism is also adopted to interpret the approaches. RETAIN [4], [24] propose a two-level neural attention model that detects influential past visits and significant clinical variables within those visits, which is clinically interpretable but not able to achieve relatively high accuracy. There is a trade-off in clinical applications where both accuracy and interpretability are important. Therefore, we propose domain knowledge guided recurrent neural networks, which can dynamically attend medical knowledge graph, to achieve a better accuracy while remaining clinically interpretable.

V. Conclusion

In this work, we presented DG-RNN, a domain knowledge guided deep learning framework that introduces medical knowledge graph information into RNN-based models for clinical risk prediction. We leveraged a knowledge graph attention mechanism to dynamically attend the adjacent nodes in the knowledge graph of given patients' EHRs. We adopted a global pooling operation to improve performance and accelerate the convergence rate. Experimental results on real-world EHRs demonstrated that the proposed DG-RNN model outperforms existing risk prediction models, especially when training data are insufficient. DG-RNN model also outputs contributions of individual medical events to final clinical outcomes, thus paving the way for interpretable clinical risk predictions.



(a) Hold-off window: 7 days

(b) Hold-off window: 120 days

Fig. 6. Test AUC of heart failure prediction for increasing data size.

VI. Acknowledgement

This work is sponsored by "China Northwest Cohort Study" under the National Key Research and Development Program of China with grant No. 2018YFC130078; "Multi-model Based Patient Similarity Learning for Medical Data Modelling and Learning" under National Natural Science Foundation of China General Program with grant No. 61672420; Project of China Knowledge Center for Engineering Science and Technology; National Natural Science Foundation of China Innovation Research Team No. 61721002; Ministry of Education Innovation Research Team No. IRT_17R86; Key Project of Natural Science Foundation of China under grant No. 61532015.

References

[1] Y. Cheng, F. Wang, P. Zhang, and J. Hu, "Risk prediction with electronic health records: A deep learning approach," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, 2016.

[2] I. M. Baytas, C. Xiao, X. Zhang *et al.*, "Patient subtyping via time-aware LSTM networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[3] Z. Zhu, C. Yin, B. Qian *et al.*, "Measuring patient similarities via a deep architecture with medical concept embedding," in *IEEE 16th International Conference on Data Mining, ICDM*, 2016.

[4] E. Choi, M. T. Bahadori, J. Sun *et al.*, "RETAIN: an interpretable predictive model for healthcare using reverse time attention mechanism," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, 2016.

[5] E. Choi, M. T. Bahadori, L. Song *et al.*, "GRAM: graph-based attention model for healthcare representation learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[6] F. Ma, Q. You, H. Xiao *et al.*, "KAME: knowledge-based attention model for diagnosis prediction in healthcare," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM*, 2018.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, no. 8, 1997.

[8] P. Ernst, A. Siu, and G. Weikum, "Knowlife: a versatile approach for constructing a large knowledge graph for biomedical sciences," *BMC Bioinformatics*, 2015.

[9] A. E. Johnson, T. J. Pollard, L. Shen *et al.*, "Mimic-iii, a freely accessible critical care database," 2016.

[10] F. Ma, J. Gao, Q. Suo *et al.*, "Risk prediction on electronic health records with prior medical knowledge," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*, 2018.

[11] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017, pp. 6000–6010.

[12] A. Bordes, N. Usunier, A. García-Durán *et al.*, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems: 27th Annual Conference on Neural Information Processing Systems*, 2013.

[13] Y. Lin, Z. Liu *et al.*, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[14] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.

[15] N. Razavian, J. Marcus, and D. Sontag, "Multi-task prediction of disease onsets from longitudinal laboratory tests," in *Proceedings of the 1st Machine Learning in Health Care, MLHC*, 2016.

[16] C. Yin, B. Qian, S. Cao *et al.*, "Deep similarity-based batch mode active learning with exploration-exploitation," in *2017 IEEE International Conference on Data Mining, ICDM*, 2017.

[17] Z. Che, S. Purushotham, K. Cho *et al.*, "Recurrent neural networks for multivariate time series with missing values," *CoRR*, 2016.

[18] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Using recurrent neural network models for early detection of heart failure onset," *JAMIA*, 2017.

[19] Z. Che, D. C. Kale, W. Li, M. T. Bahadori, and Y. Liu, "Deep computational phenotyping," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[20] A. Rajkomar, E. Oren, K. Chen *et al.*, "Scalable and accurate deep learning for electronic health records," *CoRR*, 2018.

[21] T. Pham, T. Tran, D. Q. Phung, and S. Venkatesh, "Deepcare: A deep dynamic memory model for predictive medicine," in *Advances in Knowledge Discovery and Data Mining*, 2016.

[22] Z. C. Lipton, D. C. Kale, and R. C. Wetzel, "Directly modeling missing data in sequences with rnns: Improved classification of clinical time series," in *Proceedings of the 1st Machine Learning in Health Care, MLHC*, 2016.

[23] F. Ma, R. Chitta, J. Zhou *et al.*, "Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[24] B. C. Kwon, M. Choi, J. T. Kim *et al.*, "Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records," *IEEE Trans. Vis. Comput. Graph.*, no. 1, 2019.