24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

MedGraph: Structural and Temporal Representation Learning of Electronic Medical Records

Bhagya Hettige[1] and **Weiqing Wang**[2] and **Yuan-Fang Li**[3] and **Suong Le**[4] and **Wray Buntine**[5]

Abstract. Electronic medical record (EMR) data contains historical sequences of visits of patients, and each visit contains rich information, such as patient demographics, hospital utilisation and medical codes, including diagnosis, procedure and medication codes. Most existing EMR embedding methods capture visit-code associations by constructing input visit representations as binary vectors with a static vocabulary of medical codes. With this limited representation, they fail in encapsulating rich attribute information of visits (demographics and utilisation information) and/or codes (e.g., medical code descriptions). Furthermore, current work considers visits of the same patient as discrete-time events and ignores time gaps between them. However, the time gaps between visits depict dynamics of the patient's medical history inducing varying influences on future visits. To address these limitations, we present MedGraph, a supervised EMR embedding method that captures two types of information: (1) the visit-code associations in an attributed bipartite graph, and (2) the temporal sequencing of visits through a point process. MedGraph produces Gaussian embeddings for visits and codes to model the uncertainty. We evaluate the performance of MedGraph through an extensive experimental study and show that MedGraph outperforms state-of-the-art EMR embedding methods in several medical risk prediction tasks.

1 Introduction

Electronic medical records (EMR) contain rich clinical data from a patient's stays in hospital. A high volume of EMRs is collected by hospitals that can be used in medical risk prediction to improve the quality of personalised healthcare. EMR data forms a unique and complex data structure. An EMR represents a hospital visit, and it typically contains patient demographics (e.g. age, gender) and hospital utilisation information (e.g. duration/ward of stay). Also, an unordered set of medical concepts (e.g. diagnosis, procedure and medication codes) are associated with each visit. These medical concepts are usually taken from pre-defined standards in healthcare such as International Classification of Diseases (ICD) and National Drug Codes (NDC). Moreover, EMRs accumulated over a period of time naturally form a sequence of visits of a patient's hospitalisation history. Transforming EMRs into low-dimensional vectors has been an active research topic recently, as it enables these complex data in downstream machine learning algorithms to perform predictive healthcare tasks [1, 10, 23, 26, 28, 34]. Learning these *structural* visit-code associations and *temporal* visit-sequence influences are two important aspects of EMR embedding.

Considering structural visit-code associations, a patient's visit contains a set of unordered medical codes. Existing methods, such as Med2Vec [6], RETAIN [8] and Dipole [23], propose sophisticated deep learning models to derive latent representations for visits using multi-hot-encoded medical codes as inputs. The major limitations of these methods are three-fold. First, the visits are represented using a fixed vocabulary of medical codes, but in the real-world EMR systems, new or previously unseen medical codes can be introduced (due to revised versions of medical codes, e.g. ICD-9 and ICD-10). Second, some medical codes are rarely reported and the existing methods does not deal with the sparsity of the data. Third, they do not capture demographics and utilisation information attached with visits (except for Med2Vec [6]), and side information found in medical codes such as ICD text code descriptions. These additional attribute information are important in determining similar visits and similar medical codes.

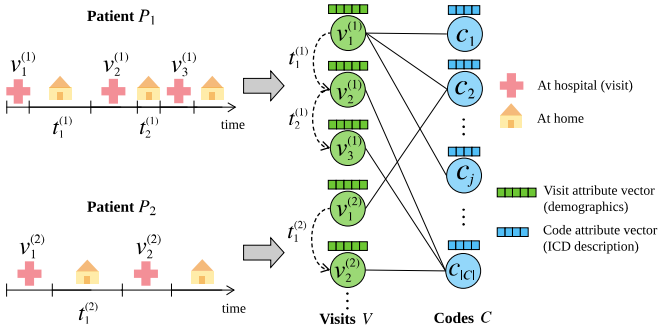


Figure 1: The MedGraph data structure. $v_j^{(i)}$ is the j-th visit of i-th patient.

Considering temporal visit sequences, EMRs are longitudinal medical events which are time-stamped. Each patient has a temporal sequence of visits, and these visits possess time-dependent relationships among them, i.e. previous visits in a patient's history can have an influence on the next visit. The influence of historical visits on the next visit also degrades with time, so that more recent visits may have higher influence than older visits. Moreover, the time gap between consecutive visits is not fixed and the larger the time gap between two visits, the less related they are. Recurrent neural network (RNN)-based architectures [8, 10, 23, 29] have been proposed in previous work to learn the temporality of visits, but they consider the visit sequences as time-series by treating the time as indexes (i.e. events are ordered periodically) and do not account for the varying time gaps. On the contrary, visits are continuous-time events and there are varying influences of historical visits on the current visit, based on the time gaps between them.

In this paper, we propose MedGraph, a novel EMR embedding method that leverages both *structural* and *temporal* information in EMRs to improve the embedding quality. MedGraph encompasses a novel graph-based data structure to represent EMR, in which we represent visits and codes as nodes, and their different interactions as edges. In Figure 1, each patient has a temporal visit sequence connected via dashed, directed edges, and each visit has a set of codes connected via thick, undirected

[1] Monash University, Australia, email: bhagya.hettige@monash.edu
[2] Monash University, Australia, email: teresa.wang@monash.edu
[3] Monash University, Australia, email: yuanfang.li@monash.edu
[4] Monash Health, Australia, email: suong.le@monashhealth.org
[5] Monash University, Australia, email: wray.buntine@monash.edu

edges. Denoting the sets of visits and codes as V and C respectively, V-C relationships form a bipartite graph with V and C node partitions. Each node carries supplementary information such as demographics and utilisation information in visits, and textual descriptions in codes, which makes V-C an attributed bipartite graph. The $V \xrightarrow{t} V$ relationships form temporal sequences of nodes where t is the time gap between two consecutive visits of a patient. Since the graph is an open data structure, it is extensible for new or unseen medical codes.

Taking advantage of the graph structure, `MedGraph` effectively learns low-dimensional representations for visits and codes by capturing both visit-code associations (based on the graph structure) and hospitalisation history (based on the temporal visit sequences) under one unified framework. Structurally, `MedGraph` learns representations for visits and codes by considering them in an attributed bipartite graph. `MedGraph` proposes to model temporal visit sequence information as a temporal point process, such as the Hawkes process [14], to effectively capture and account for the varying influence of historical visits of different time gaps. These theoretical point process models often make strong assumptions about the generative process of the sequential event data which do not necessarily depict the real-world dynamics and also limit the expressive power of the model [12, 20]. To automatically learn a more expressive representation for the varying historical visit influence when a real parametric model is unknown, `MedGraph` uses an RNN-based architecture, as in RMTPP [12], to model the conditional intensity function. We have designed `MedGraph` to be accountable for the uncertainty of the embeddings by learning node embeddings (i.e. visits and codes) as probability distributions (e.g. Gaussians). To the best of our knowledge, `MedGraph` is the first EMR embedding method that models uncertainty of the visit and code embeddings as Gaussian distributions. The contributions of our proposed EMR embedding framework, `MedGraph`, are threefold:

1. A customised graph-based data structure designed for EMR data, that naturally captures both the visit-code co-location information (V-C) *structurally* as an attributed bipartite graph and the visit sequence information ($V \xrightarrow{t} V$) *temporally* as a continuous event sequence.
2. A novel approach to effectively and efficiently learn $V \xrightarrow{t} V$ relationships using a point process based sequence learning method with RNN-based conditional intensity function, considering varying influence of historical visits on the next visit.
3. An extensive experimental study on two real-world EMR datasets that shows `MedGraph`'s superiority over state-of-the-art embedding methods on a number of tasks: 30-day readmission prediction, mortality prediction, medical code classification, medical code visualisation and uncertainty modelling.

2 Related Work

Representation learning of EMRs aims at learning low-dimensional vectors for hospital visits and medical codes through their interactions in terms of visit-code relationship and visit-visit sequencing information.

Several embedding models have been proposed to learn from visit-code relationships in EMR data [33]. Most of these works, including Med2Vec [6], RETAIN [8], Deepr [28], GRAM [7] and Dipole [23], capture the visit-code associations by constructing the input visit vector as a multi-hot encoded medical code vector. MiME [10] assumes the hierarchical structure of EMR data and represents patients, visits, diagnoses, procedures and medications in a hierarchy in the stated order. In the real-world EMRs, though, this hierarchical granularity of medical codes is often not found. GCT [11] attempts to address this challenge with a graph data structure to learn the implicit hierarchical relations of codes using Transformers. However, none of these works consider the rich attributes of visits/codes. An exception to this are Med2Vec [6], which considers visit demographics, and MNN [31], which incorporates clinical notes. Still, they do not consider code attributes. In contrast, GRAM [7] models EMR with a convex combination of the embeddings of the code and its ancestors on the ontology tree, and it strictly depends on this ontology structure. But not all the medical codes form such rich ontology graphs.

Existing EMR representation learning work captures the temporality of visit sequences using either Skip-gram [25] with multi-layer perceptron (MLP)-based architectures [2, 6] or RNN-based architectures [5, 8, 9, 10, 17, 23, 29, 30]. However, Skip-gram is only capable of capturing neighbouring visits within a predefined number of time steps without any particular order in the context window. RNN models assume that the events are recorded periodically, and cannot effectively capture the varying time gaps between visits. A recent work, PacRNN [31], proposes a continuous-time model using a point process for a specialised healthcare task, i.e. ranked diagnosis code recommendation and time prediction of the next visit.

`MedGraph` uses a graph-based data structure to model EMR data, and we show that the graph can capture much more useful and meaningful data compared to the existing approaches. Our approach also effectively captures the temporality of historical visits of the patients via the proposed temporal point process model. Experimental results show that `MedGraph` produces more effective EMR embeddings.

3 `MedGraph`: Medical Data Graph Embedding

In this section, we describe the algorithm for `MedGraph`. Without loss of generality, the algorithm will be discussed for a single patient for simplicity of the notations. Figure 1 is a heterogeneous graph with two types of nodes and two types of edges. These two edge types denote two distinguishing types of information about the visits, i.e. codes in the visits and temporal visit sequences. Therefore, we dismantle the two edge types. Accordingly, we extract a subgraph from Fig. 1 for a single patient, and decompose it into an attributed bipartite graph (Fig. 2a) and a temporal (timestamped) sequence graph (Fig. 2b).

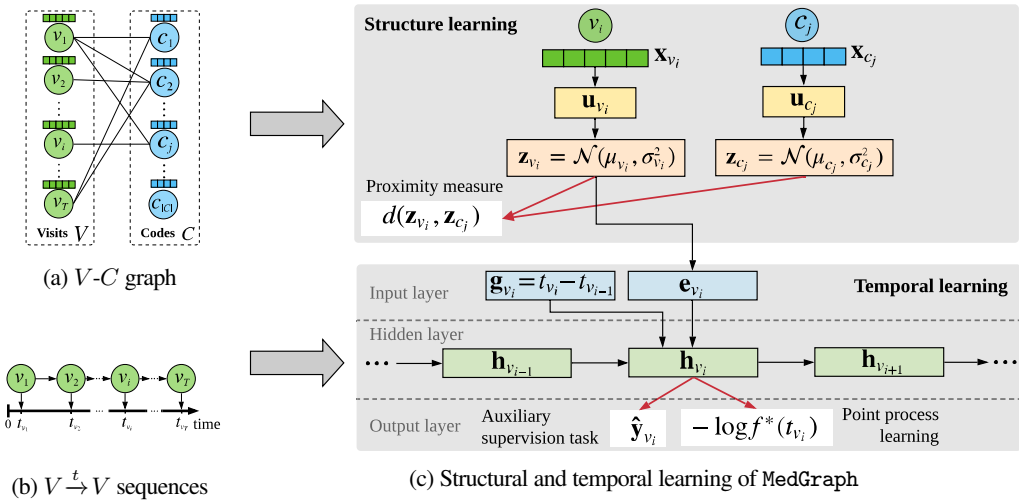
3.1 Notations of `MedGraph`

As denoted in Fig. 2, assume a patient has a time ordered sequence of visits $v_1,v_2,...,v_T$, where $T \geq 1$ is the length of the visit sequence and each visit v_i has an unordered set of medical codes $C_{v_i} \subseteq C$ with the code set $C = \{c_1,c_2,...,c_{|C|}\}$. C denotes the set of code nodes, such as diagnosis codes. We construct the graph data structure for this data as follows.

V-C attributed bipartite graph: This graph has two node partitions, visits $V = \{v_1,v_2,...v_T\}$ and codes C, and a set of edges, E_{vc}, where $(v_i,c_j) \in E_{vc}$ is an edge denoting the link between visit v_i and code c_j. Each visit v_i has a D_v-dimensional visit attribute vector $\mathbf{x}_{v_i} \in \mathbb{R}^{D_v}$ with visit demographic and utilisation information, such as age, gender and length of stay. Each code c_j has a D_c-dimensional code attribute vector $\mathbf{x}_{c_j} \in \mathbb{R}^{D_c}$ with code supplementary medical information, such as ICD code description as text or multi-hot ICD ontology ancestors.

$V \xrightarrow{t} V$ temporal sequence: The temporally ordered sequence of visits of a patient is denoted as $S = \{(v_i,t_{v_i},\mathbf{y}_{v_i})\}_{i=1}^{T}$, where for visit v_i, $t_{v_i} \geq 0$ is the timestamp and $\mathbf{y}_{v_i} \in \{0,1\}^s$ is the ground-truth for the underlying auxiliary task (optional) with s classes.

Each visit and code, $k \in V \cup C$, is represented as a low-dimensional Gaussian embedding in a shared embedding space, $\mathbf{z}_k = \mathcal{N}(\mu_k, \sigma_k^2)$ where $\mu_k \in \mathbb{R}^L$, $\sigma_k^2 \in \mathbb{R}^{L \times L}$ with embedding dimension $L \ll |V|,|C|,D_v,D_c$ capturing visit-code associations and time-gap-based influence of historical visits. We learn σ_k as a diagonal covariance vector, $\sigma_k^2 \in \mathbb{R}^L$, instead of a covariance matrix to reduce the number of parameters to learn.



(a) V-C graph

(b) $V \xrightarrow{t} V$ sequences

(c) Structural and temporal learning of `MedGraph`

Figure 2: `MedGraph` architecture

3.2 `MedGraph` Architecture

Figure 2 shows the architecture of our proposed algorithm, `MedGraph`. Given an EMR dataset, we transform these data into a graph-based representation. Then, we consider the V-C attributed bipartite graph and learn visit and code similarities based on the graph structure proximity. We further improve the visit embeddings by learning temporal visit sequences, $V \xrightarrow{t} V$, as events occurring in continuous-time modelled through a temporal point process which captures the mutual excitation phenomenon among temporally ordered events [14], so that we can learn the varying influence (due to time gaps between visits) of the historical visits. `MedGraph` is an end-to-end risk prediction tool with a supervised task plugged into the output layer of the RNN. Alternatively, if there is no specific supervision task, `MedGraph` can also be learned in an unsupervised manner with only structure learning and temporal sequence learning.

3.2.1 *Structural learning for V-C graph (Fig. 2a)*

In contrast to previous EMR embedding methods which collect codes in a visit as a multi-hot vector, `MedGraph` employs a bipartite graph to denote edges between V and C. Due to this versatile structure, we can plug in auxiliary attribute data at nodes resulting in an attributed bipartite graph.

Let $(v_i, c_j) \in E_{vc}$ be an edge between visit v_i and code c_j (i.e. $c_j \in C_{v_i}$) with attributes $\mathbf{x}_{v_i} \in \mathbb{R}^{D_v}$ and $\mathbf{x}_{c_j} \in \mathbb{R}^{D_c}$, respectively. These attribute vectors can be numerical values and/or one-hot encoded categorical values. We project the two node types to a uniform semantic latent space using two transformation matrices denoted by, $\mathbf{W}_v \in \mathbb{R}^{D_v \times m}$ and $\mathbf{W}_c \in \mathbb{R}^{D_c \times m}$, where m is the intermediate vector dimension, for visit and code domains, respectively.

$$\mathbf{u}_{v_i} = \mathbf{W}_v \mathbf{x}_{v_i} \text{ and } \mathbf{u}_{c_j} = \mathbf{W}_c \mathbf{x}_{c_j} \tag{1}$$

Then, we apply another layer of linear transformations to the m-dimensional vectors to obtain Gaussian embeddings in a common embedding space for the two node types denoted by $\mathbf{z}_{v_i} = \mathcal{N}(\mu_{v_i}, \sigma^2_{v_i})$ and $\mathbf{z}_{c_j} = \mathcal{N}(\mu_{c_j}, \sigma^2_{c_j})$:

$$\mu_{v_i} = \mathbf{W}_\mu \mathbf{u}_{v_i} + \mathbf{b}_\mu \tag{2}$$
$$\sigma^2_{v_i} = ELU(\mathbf{W}_\sigma \mathbf{u}_{v_i} + \mathbf{b}_\sigma) + 1 \tag{3}$$
$$\mu_{c_j} = \mathbf{W}_\mu \mathbf{u}_{c_j} + \mathbf{b}_\mu \tag{4}$$
$$\sigma^2_{c_j} = ELU(\mathbf{W}_\sigma \mathbf{u}_{c_j} + \mathbf{b}_\sigma) + 1 \tag{5}$$

where $\mathbf{W}_\mu \in \mathbb{R}^{m \times L}$, $\mathbf{b}_\mu \in \mathbb{R}^L$, $\mathbf{W}_\sigma \in \mathbb{R}^{m \times L}$ and $\mathbf{b}_\sigma \in \mathbb{R}^L$ denote the shared mean and variance encoders for both node types. We adopted exponential linear unit (ELU) for the activation function in σ as it drives the mean of the activation outputs be closer to zero which makes learning and convergence much faster. To obtain positive covariance for interpretability of uncertainty, we add one in σ^2 functions.

Embedding into a common L-dimensional embedding space enables similarity computation between two heterogeneous nodes. Since the embeddings are Gaussians, we measure the Wasserstein distance as in DVNE [36] and RASE [16], specifically 2-nd Wasserstein distance (W_2) between the embeddings. By computing W_2 distance, we can preserve transitivity property in the embedding space [36]. As a result, when we model the explicit visit-code relations, visit-visit and code-code associations are also implicitly modelled in the embedding space. For example, if both v_1 and v_2 are linked to c_2, then it is highly likely that v_1 and v_2 are similar, and we implicitly capture this similarity by preserving triangle inequality property in the embedding space. We define $d(\mathbf{z}_{v_i}, \mathbf{z}_{c_j})$ as the W_2 distance for our embeddings of visit v_i and code c_j in the embedding space. Modelling only the diagonal covariance vectors results in $\sigma^2_{v_i}\sigma^2_{c_j} = \sigma^2_{c_j}\sigma^2_{v_i}$ [36]. Therefore, the distance computation [13] simplifies to:

$$d(\mathbf{z}_{v_i}, \mathbf{z}_{c_j}) = W_2(\mathbf{z}_{v_i}, \mathbf{z}_{c_j}) = (\| \mu_{v_i} - \mu_{c_j} \|_2^2 + \| \sigma_{v_i} - \sigma_{c_j} \|_F^2)^{1/2} \tag{6}$$

Then, we define joint probability between the two node distribution representations as the likelihood of the existence of a link between them by: $P(v_i, c_j) = Sigmoid(-d(\mathbf{z}_{v_i}, \mathbf{z}_{c_j}))$ similarly to the first-order proximity measure in GLACE [15]. Since our graph is unweighted, we can define the prior probability using the structural information observed in the graph as: $\hat{P}(v_i, c_j) = \frac{1}{|E_{vc}|}$. To preserve this proximity measure in the embedding space, we minimise the distance between the prior and observed probability distributions for all edges observed in V-C graph. Since $\hat{P}$ and P are discrete probability distributions, we define the structural loss function with Kullback–Leibler divergence (D_{KL}) as:

$$\begin{aligned} \mathcal{L}_{struc} = D_{KL}(\hat{P} || P) &= \sum_{(k,l) \in E_{vc}} \hat{P}(k,l) \log \left(\frac{\hat{P}(k,l)}{P(k,l)} \right) \\ &\propto - \sum_{(k,l) \in E_{vc}} \hat{P}(k,l) \log P(k,l) \\ &\propto - \sum_{(k,l) \in E_{vc}} \log P(k,l) \end{aligned} \tag{7}$$

3.2.2 Temporal learning for $V \xrightarrow{t} V$ sequences (Fig. 2b)

The objective of our temporal sequence modelling is to learn time gap-based influence of historical visits on the next visit of a patient. Thus, we model the visits as continuous-time events with a point process model [14, 19, 20] to capture the varying historical visit influence. Typical parametric point process models establish strict assumptions on the generative process of the events, so that these models have restricted expressive power and do not guarantee to reflect the real-world data. Hence, following the idea of RMTPP [12], we learn the visit influence using an RNN-based architecture to model a flexible and expressive marked temporal point process and capture the visit sequence dynamics automatically without being restricted to strict parametric assumptions.

RNNs use output from the hidden units of the current time step as inputs for the next time step. Consequently, the network can memorise the influence of each past data event through the hidden state vectors $\mathbf{h}_{v_i}$. Thus, we use $\mathbf{h}_{v_i}$ to represent the influence of the history up to the v_i-th visit and model conditional intensity function of a temporal point process. We construct a latent vector $\mathbf{e}_{v_i} \in \mathbb{R}^L$ to model the markers of the marked temporal point process using the learned visit embedding from the previous section (structural learning) $\mathbf{z}_{v_i} = \mathcal{N}(\mu_{v_i}, \sigma^2_{v_i})$ where we treat the covariance vector as a noise:

$$\mathbf{e}_{v_i} = \mu_{v_i} + \varepsilon_{v_i} \cdot \sigma^2_{v_i} \text{ where } \varepsilon_{v_i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{8}$$

For the input layer of the RNN cell (bottom block in Fig. 2c), we feed the event information (i.e. $\mathbf{e}_{v_i}$) and the timing information (i.e. $\mathbf{g}_{v_i}$) about the current visit event. Since we are interested in modelling time gaps between the visits, we set $\mathbf{g}_{v_i} = (t_{v_i} - t_{v_{i-1}})$ as the time gap between the previous and the current visit. We update the RNN cell to output an effective hidden state vector at the current time step, $\mathbf{h}_{v_i}$, using the current visit event $(\mathbf{e}_{v_i}, \mathbf{g}_{v_i})$ and the influence from the memory carried out from the past visit events $(\mathbf{h}_{v_{i-1}})$:

$$\mathbf{h}_{v_i} = ReLU(\mathbf{W}_{tv}\mathbf{e}_{v_i} + \mathbf{W}_g\mathbf{g}_{v_i} + \mathbf{W}_h\mathbf{h}_{v_{i-1}} + \mathbf{b}_h) \tag{9}$$

where $\mathbf{W}_{tv} \in \mathbb{R}^{L \times m'}$, $\mathbf{W}_g \in \mathbb{R}^{D_t \times m'}$, $\mathbf{W}_h \in \mathbb{R}^{m' \times m'}$, $\mathbf{b}_h \in \mathbb{R}^{m'}$, D_t is the time vector dimension and m' is the RNN hidden state dimension. We define the conditional intensity function to model the point process by:

$$\lambda^*(t) = \exp(\mathbf{v}_t^\top \cdot \mathbf{h}_{v_i} + w_t(t - t_{v_i}) + b_t) \tag{10}$$

where $\mathbf{v}_t \in \mathbb{R}^{m'}$ and $w_t, b_t \in \mathbb{R}$. The exponential function is a non-linear transformation which guarantees positive intensity values. The first term $\mathbf{v}_t^\top \cdot \mathbf{h}_{v_i}$ comprises the *historical influence*, the second term $w_t(t - t_{v_i})$ denotes the *current influence*, and the final term b_t emphasises *base intensity* in the intensity function. Then, we define the likelihood of the next visit occurring at time t given the historical visit sequence up to time t_{v_i} by:

$$\begin{aligned} f(t - t_{v_i} | \mathbf{h}_{v_i}) &= f^*(t) = \lambda^*(t) \exp\left(- \int_{t_{v_i}}^{t} \lambda^*(\tau) d\tau \right) \\ &= \exp\Bigg\{ \mathbf{v}_t^\top \cdot \mathbf{h}_{v_i} + w_t(t - t_{v_i}) + b_t \\ &\quad + \frac{1}{w_t} \left(\exp(\mathbf{v}_t^\top \cdot \mathbf{h}_{v_i} + b_t) - \exp(\mathbf{v}_t^\top \cdot \mathbf{h}_{v_i} + w_t(t - t_{v_i}) + b_t) \right) \Bigg\} \end{aligned} \tag{11}$$

Accordingly, given a temporal visit sequence S for a patient with T visits, we can define the temporal loss function as the negative log-likelihood of observing the visit sequence (i.e. maximise the likelihood of observing the sequence by minimising the negation):

$$\mathcal{L}_{temp} = -\sum_{i=1}^{T} \log f(t_{v_{i+1}} - t_{v_i} | \mathbf{h}_{v_i}) \tag{12}$$

3.2.3 Auxiliary supervision task

We can train `MedGraph` to predict medical risk in the future. Medical risk prediction is an important task for personalised healthcare [8, 10]. Therefore, we incorporate an auxiliary medical risk prediction task. This allows prediction of future outcomes of a patient at a given point in time to supplement predictive personalised healthcare. We assume that the hidden state of the current visit, $\mathbf{h}_{v_i}$, not only carries the information from the current visit itself, but also memorises the time-gap-based influence of past visits through point process modelling. For simplicity, we describe a classification task in which the outcome for the visit is $\mathbf{y}_{v_i} \in \{0,1\}^s$ where s is the number of classes. We use $\mathbf{h}_{v_i}$ to predict the class label (i.e. future medical risk outcome) as follows:

$$\hat{\mathbf{y}}_{v_i} = Softmax(\mathbf{W}_s\mathbf{h}_{v_i} + \mathbf{b}_s) \tag{13}$$

where $\mathbf{W}_s \in \mathbb{R}^{s \times m'}$ and $\mathbf{b}_s \in \mathbb{R}^s$. Then, we compute the classification loss for a visit sequence S of a patient using cross-entropy:

$$\mathcal{L}_{tsk} = -\frac{1}{T}\sum_{i=1}^{T} \left(\mathbf{y}_{v_i}^\top \log(\hat{\mathbf{y}}_{v_i}) + (1 - \mathbf{y}_{v_i})^\top \log(\mathbf{1} - \hat{\mathbf{y}}_{v_i}) \right) \tag{14}$$

3.3 Unified Training and Model Optimisation

`MedGraph` is an end-to-end medical risk prediction model, which exploits V-C graph structure and $V \xrightarrow{t} V$ temporal sequences in improving predictive performance of an underlying medical risk prediction task. For each patient, the unified loss of the predictive model is defined as:

$$\mathcal{L} = \alpha\mathcal{L}_{struc} + \beta\mathcal{L}_{temp} + \gamma\mathcal{L}_{tsk} \tag{15}$$

where $\alpha, \beta, \gamma \geq 0$ are hyperparameters which control learning from structural, temporal and underlying healthcare prediction task, respectively. `MedGraph` can be trained in an unsupervised manner (setting $\gamma = 0$) when there is no auxiliary supervision task, e.g. when learning general-purpose embeddings for an exploratory analysis of EMR data [6].

To optimise the structural loss computation, we use the negative sampling [25, 35] approach, which selects K number of negative V-C edges for each positive edge.

4 Experiments

In this section, we evaluate the performance of `MedGraph` on two real-world EMR datasets and compare its performance against several state-of-the-art embedding methods along with two variant versions of `MedGraph` in the ablation study. We perform several medical risk prediction tasks, qualitative analysis of the medical code representations and the uncertainty modelling capability of the embeddings produced by `MedGraph`. Source code for `MedGraph` is available at `https://github.com/bhagya-hettige/MedGraph`.

4.1 Datasets

We use two real-world, cohort-specific proprietary EMR datasets in the experimental study: heart failure (HF) and chronic liver disease (CL). We remove patients with less than two visits. Brief statistics of the two datasets after removing the shorter visit sequences are shown in Table 1. Both EMR datasets are extracted from the same hospital, in which *ICD-10-CM diagnosis codes* and *in-house procedure codes* are used. For the visits we extract patient demographics (e.g. age, gender, ethnicity, birth country, etc.) and hospitalisation utilisation information (e.g. length of stay, admission source, etc.) as visit attributes. For the medical codes,

we use tf-idf vectors of ICD-10-CM code descriptions for the diagnoses, and tf-idf vectors of code descriptions provided by the hospital for the in-house procedures as code attributes.

Table 1: Statistics of the real-world cohort-specific EMR datasets.

Dataset	HF	CL
Data collection time period	2010-2017	2000-2017
Total # of patients	10,713	3,830
Total # of visits	204,753	122,733
Avg. # of visits per patient	18.51	29.84
Total # of unique medical codes	8,541	8,382
# of unique diagnosis codes	6,278	6,010
# of unique procedure codes	2,263	2,372
Avg. # of medical codes per visit	5.27	5.02
Max # of medical codes per visit	98	100

4.2 Baselines

We compare `MedGraph` to several state-of-the-art EMR embedding methods to evaluate the performance on several risk prediction tasks. We choose Skip-gram based (Med2Vec [6]) and RNN-based (Dipole [23] and RETAIN [8]) EMR embedding methods for comparison[6]. We also choose a state-of-the-art general-purpose graph embedding model, GCN [21], to learn V-C attributed bipartite graph.

Med2Vec [6] is a Skip-gram based EMR embedding method that produces both code- and visit-level representations by predicting medical codes appearing in neighbouring visits. It captures visit demographics, and we feed the visit attribute vector $\mathbf{x}_{v_i}$ for each visit.

Dipole [23] is an attention-based bidirectional RNN framework, which takes the influence of historical visits via the trained attention weights.

RETAIN [8] is an end-to-end RNN-based healthcare prediction model with a reverse-time attention mechanism, which models the influence of previous visits and important medical codes in them.

GCN [21], the graph convolutional networks (GCN) model, is a recent state-of-the-art semi-supervised graph embedding approach which learns by aggregated neighbourhood information. We use GCN layers (unsupervised) to model the V-C bipartite relations. Since GCN only supports homogeneous graphs, we ignore the node heterogeneity and construct attribute vectors of visits (i.e. $\mathbf{x}_{v_i}$) and codes (i.e. $\mathbf{x}_{c_j}$) by pre- and post-padding with zeros respectively.

4.3 Our Approaches

In addition to these baseline methods, we conduct a comprehensive ablation study to evaluate the effectiveness and importance of the two important components of `MedGraph`, namely structure learning and point process based temporal learning. We denote each variant model with a negation (¬) in front of the ablated component in learning. We summarise the variants in Table 2.

Table 2: Our approaches trained with Eq. 15: $\mathcal{L} = \alpha\mathcal{L}_{struc} + \beta\mathcal{L}_{temp} + \gamma\mathcal{L}_{tsk}$.

Notation	α	β	γ
`MedGraph`	>0	>0	>0 for the predictive model; $=0$ for the unsupervised model
`MedGraph(S,¬T)`	>0	$=0$	
`MedGraph(¬S,T)`	$=0$	>0	

[6] GRAM [7] is not chosen a baseline as the in-house procedure codes in our proprietary datasets do not form an ontology on which GRAM depends.

`MedGraph` is our full model that incorporates both structural and temporal aspects (Fig. 2) by modelling V-C bipartite relations and time-gap-based temporal point process of visit sequences.

`MedGraph(S,¬T)` is a simple RNN model which treats visit events as an equally spaced time-series and thus does not model temporal point processes. It learns from V-C bipartite associations.

`MedGraph(¬S,T)` does not learn any structural information in the V-C graph, so it does not learn medical codes of visits. It implements the proposed temporal point process based RNN model to learn time-gap-based influence of historical visits.

4.4 Hyperparameter Settings

For all baseline models, we use $L = 256$ as the visit and code embedding dimension. Since `MedGraph` produces mean and covariance vectors, we halve the embedding dimension to $L = 128$ in `MedGraph` for a fair comparison by learning the same number of parameters per node (i.e. $\mu_i \in \mathbb{R}^{128}$ and $\sigma_i^2 \in \mathbb{R}^{128}$). For all the methods, we use a batch size of 128 for visits and 32 for visit sequences. For `MedGraph`, we set the number of negative edges as 10, and α,β,γ are tuned to be optimal on a validation set. We use Adam optimiser with a learning rate fixed at 0.001. Other parameters for baseline models are referred from the papers and tuned to be optimal. `MedGraph`'s RNN layer uses LSTM. All the experiments are conducted on a modest hardware setting with a MacBook Pro laptop with 16GB memory and a 2.6 GHz Intel Core i7 processor.

4.5 Medical Risk Prediction

We perform two real-world medical risk prediction tasks, 30-day readmission prediction and mortality prediction, to assess the model's effectiveness in predictive healthcare. For `MedGraph`, we use each risk prediction task as the auxiliary supervision task (cf Section 3.2.3) in this set of experiments in an end-to-end manner ($\gamma > 0$). Since Med2Vec, Dipole and GCN are not designed for medical risk prediction tasks, we first learn visit embeddings using these methods, and then use these fixed visit vectors as inputs to the prediction model (i.e. XGBoost classifier [4]). For both tasks, we randomly split the visit sequences into 2 parts with a 4:1 ratio. The former is used to train the embedding models, while the latter is held out for evaluation. We randomly sample 25% of visit sequences from the held-off dataset as the validation set, and the rest as the test set.

4.5.1 30-day readmission risk prediction

Accurate readmission prediction allows hospitals to target intervention strategies on high-risk patients to prevent readmissions. Given a visit, we train the models to predict 30-day readmission after a patient's discharge from the current visit/admission [18, 28], thus casting it as a binary classification task. Here, we consider all the readmissions and not only the unplanned readmissions. Both cohort datasets are imbalanced and the majority class is positive, with a prevalent 30-day readmission rate of 69.8% and 76.2% respectively. Therefore, area under the receiver operating curve (AUC) is a better measure to compare the models [8]. We plot AUC scores for HF and CL datasets in Figure 3.

As can be seen from the plot, `MedGraph`, as well as its two variants, clearly and consistently outperform all compared state-of-the-art baseline methods by a significant margin. This shows the effectiveness of each component in our model. The contribution of each type of information is experimentally validated by the performance improvement gains of the full model over the non-structural (`MedGraph(¬S,T)`) and time-series (`MedGraph(S,¬T)` with no time gaps) variants of our method. Our observation is also supported by a previous study on readmissions [3],

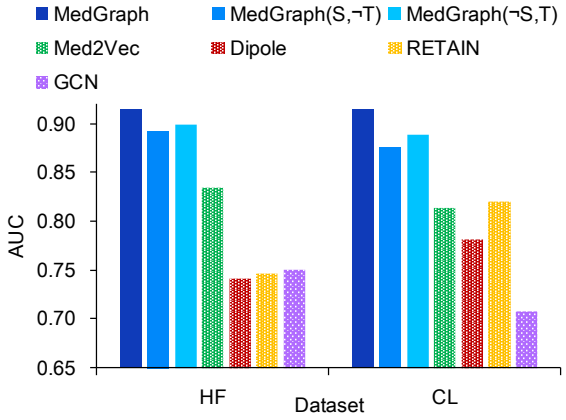


Figure 3: Performance on 30-day readmission risk prediction task.

which shows that readmission of a patient is dependent on various factors including patient demographics and past medical incidents.

Among the EMR baseline models, Med2Vec performs the best on HF, and second best on CL even though it does not model temporality of the visits, which can be attributed to its involvement of visit demographics in embedding. Capturing visit-code associations is shown to be effective, as GCN's performance is competitive compared with several EMR embedding baselines in this task.

From the performance gaps between RNN-based methods (Dipole, RETAIN and MedGraph(S,¬T)) and our time-gap-based point process models (MedGraph and MedGraph(¬S,T)), we can see that time gap information modelling is important in making an accurate prediction of the readmission risk in both datasets. Actually, our proposed point process model learns time gaps between historical visits via point process modelling (Eq. 12). Consequently, we can successfully predict a patient's readmission with a higher prediction accuracy at the 30-day time threshold.

4.5.2 *Mortality prediction*

Early identification of patients who are at high risk of death can assist hospitals to appropriately allocate resources to these patients and mitigate the mortality risk. Given a patient's visit sequence, we train the models to predict the patient's death in the next admission [22], thus casting it as a binary classification task. We report area under receiver operating curve (AUC) and average precision (AP) on both HF and CL datasets (with a prevalent mortality rate of 27.0% and 40.7% respectively) in Table 3.

From the table, we can see that MedGraph shows superior performance over all the other methods across both datasets. This demonstrates the effectiveness of the visit representations learned by MedGraph in predicting the death of patients in their next visit. The possible reasons for the superiority of our method include: the incorporation of attribute information in visits and codes and adopting temporal point processes in modelling visit-visit sequences. Intuitively, according to a related study [27], the outcome of a patient's death depends on their demographics (i.e. $\mathbf{x}_{v_i}$), current health conditions (i.e. V-C structure) and historical health conditions (i.e. $V \xrightarrow{t} V$ sequence), all of which are captured by our model.

Moreover, MedGraph outperforms both its variant versions, which showcases the effectiveness of the proposed structural and temporal learning in improving predictive performance of the visit representations in the mortality prediction task.

The poor performance of GCN and the performance gain of the RNN-based methods (i.e. Dipole and RETAIN) over GCN, suggest that the structural learning is not sufficient to capture visit similarities in this task and temporality is important. However, Med2Vec performs better due to Skip-gram based code co-occurrence learning and incorporation of visit attributes, though it ignores temporality.

Table 3: Performance on the mortality prediction task. Best result is **bolded** and second-best is underlined.

	Method	HF AUC	HF AP	CL AUC	CL AP
Baselines	Med2Vec	0.7091	0.4752	0.7224	0.6255
	Dipole	0.6464	0.4145	0.6411	0.5487
	RETAIN	0.6908	0.5402	0.7034	0.6836
	GCN	0.5581	0.3177	0.6172	0.4956
Ours	MedGraph(S,¬T)	0.7002	0.6548	0.7385	0.7123
	MedGraph(¬S,T)	0.7131	0.6751	0.7306	0.7118
	MedGraph	**0.7205**	**0.6853**	**0.7415**	**0.7143**

4.6 Medical Code Representation Analysis

In this section, we explore the medical code embeddings to evaluate their descriptiveness. We propose a novel evaluation task to quantitatively analyse the informativeness of the code embeddings and we qualitatively study the interpretability of the learned code embeddings by MedGraph. We learn general-purpose embeddings with MedGraph with $\gamma = 0$ in Eq. 15.

4.6.1 *Multi-class code classification*

The Clinical Classifications Software (CCS)[7] divides the ICD codes into a number of clinically meaningful categories. Thus, if the code embeddings are predictive of their CCS categories, then the embeddings have learnt useful latent information. With this hypothesis, we perform multi-class code classification to predict the corresponding medical concept classes of medical codes produced by CCS.

First, each method learns code embeddings, and then a logistic regression (LR) classifier is trained on the code embeddings to classify each code into their associated CCS class. We select the 10 most common CCS classes in the datasets, since some CCSs are rare. We randomly sample different percentages of diagnosis codes $(10\%, 20\%, ... 90\%)$ as the training set for the classifier, and use the rest for evaluation. We report micro- and macro-F1 scores which have been widely used in the evaluation of multi-class classification tasks [35, 36].

MedGraph is capable of learning supplementary attributes of codes. Thus, it should be able to produce more expressive code embeddings compared to the methods which do not use attributes. To conduct a fair evaluation with these baselines, we train MedGraph without code attributes (i.e. let $\mathbf{X}_C = \mathbf{I}_{|C|}$). This variant is denoted MedGraph(¬A). We exclude MedGraph(¬S,T) in this task as it does not learn code embeddings. Micro-F1 scores for HF are presented in Figure 4. The trend is similar for macro-F1 scores, and in the CL dataset, which we omit for brevity reasons.

As can be seen in Fig. 4, MedGraph and MedGraph(S,¬T) produce embeddings that are highly descriptive of their medical context, consistently and substantially outperforming the non-attributed version, MedGraph(¬A), by a large margin of improvements in the code classification task. GCN also showcases a significant performance improvement over non-attributed models. This demonstrates the effectiveness of incorporating code-level attributes and structural visit-code relations in producing high quality code embeddings. Thus, superiority of our method over MedGraph(¬A) and the rest of the baselines can be attributed to two factors: (1) the use of standard code descriptions as supplementary

[7] https://www.hcup-us.ahrq.gov/toolssoftware/ccsr/ccs_refined.jsp

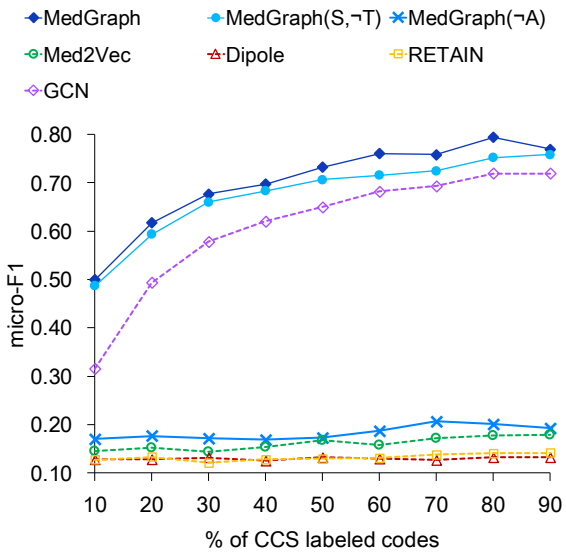


Figure 4: Multi-class code classification task. Improvements are statistically significant for $p < 0.01$ in a paired t-test.

code attributes, and (2) the learning of code-visit associations through the graph-based data structure, as opposed to multi-hot medical code vectors.

We also see that `MedGraph`(¬`A`) consistently outperforms all the three evaluated EMR embedding methods, producing more meaningful embeddings. The difference between our `MedGraph`(¬`A`) method and these methods is the way we learn visit-code associations. These baseline methods construct multi-hot medical code vectors to represent visits, and then these are used as features in their models to learn visit-code relations ignoring the inherent graph structure. On the contrary, our models learn visit-code associations through the structural information in the V-C bipartite graph. Thus, ours is capable of learning not only the code co-location in visits through local neighbourhood, but also the similar code neighbourhoods through global connectivity due to transitivity property in W_2 distance. This shows that the structural learning of visit-code associations we proposed is effective in producing meaningful code embeddings. Among the baselines, Med2Vec shows superiority over Dipole and RETAIN, which is attributed to its neighbouring code learning technique (similar to Skip-gram) within a predefined context window.

`MedGraph`(`S`,¬`T`), which learns V-C structure and model time-series visits with no time gaps, is slightly surpassed by `MedGraph`, since injecting time-gap-based temporal learning of medical history enables to learn additional useful latent patterns in EMRs. For example, when a patient is diagnosed with cancer at an earlier visit, `MedGraph` learns that the patient may revisit the hospital for chemotherapy or other related procedures via the patient's medical history.

4.6.2 *Interpretation of code embeddings*

Interpretability of the learned code embeddings is important in various applications in medical domain, including healthcare analysis tasks. In Figure 5, we project the 128-dimensional mean vector of embeddings of ICD-10-CM diagnosis codes trained with `MedGraph` on HF dataset (∼500 codes belonging to the top 10 CCS classes) into 2 dimensions using t-SNE, for visualisation [32]. We also publish an interactive plot for further analysis[8]. Colour of a node indicates the associated CCS class.

Overall, `MedGraph` clusters codes belonging to most CCS clinical concepts with clear boundaries. Moreover, there are several overlapped

[8] `https://bhagya-hettige.github.io/MedGraph`

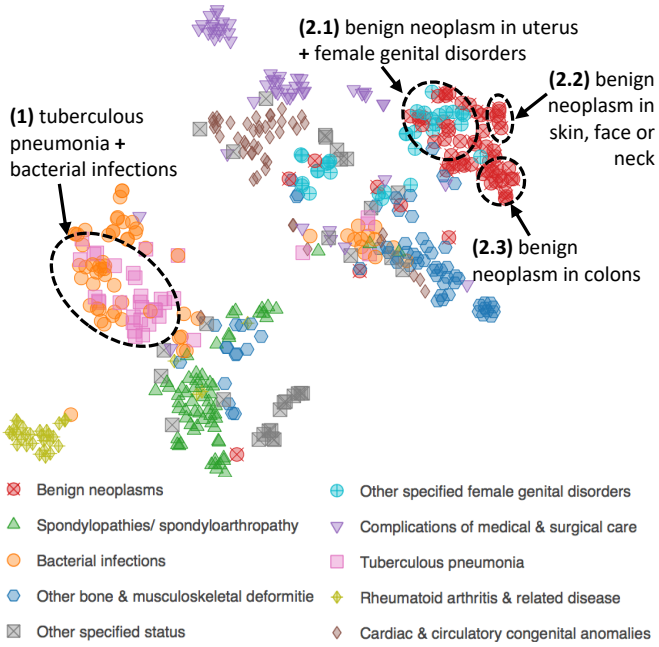


Figure 5: 2-D visualisation of codes. Code's colour is its CCS class.

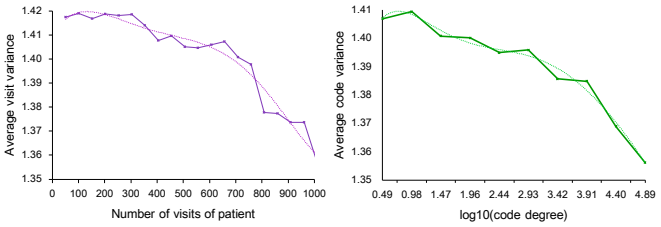
CCS classes due to broad definitions of CCS. `MedGraph` learns interesting latent relations between codes, especially in these overlapped CCS regions. For brevity reasons, we only analyse two such scenarios: (1) codes of "tuberculosis pneumonia" and "bacterial infections" are overlapping, forming a cluster showing their clinically closer relationships [24], and (2) "benign neoplasm" related codes further separates into more granular classes (focussed on different organs) within a broader CCS class, identifying the inherent differences in these sub classes.

4.7 Uncertainty Modelling of EMR

Different from the existing EMR embedding methods, `MedGraph` learns the uncertainty of visit and code embeddings as Gaussians. In this task, we study the nature of learned uncertainty terms and its intuition in the real-world EMRs. We learn interpretable diagonal covariances with non-negative values (cf Section 3). We define the average variance across the 10 largest dimensions as a node's variance [36]. We conduct the following analysis on the HF dataset. To obtain general-purpose embeddings in this exploratory analysis, we learn the embeddings in an unsupervised manner, setting $\gamma = 0$ in Eq. 15.

Visit Embedding Uncertainty (Fig. 6a): We divide all the patients into 20 buckets based on their visit counts. For each bucket, we compute the average visit variance and plot it against the number of visits. When the number of visits of a patient increases the average variance of visit embeddings decreases (Fig. 6a). Intuitively, when a patient has a longer medical history, their visit embeddings are more comprehensive and descriptive, thus less uncertain.

Code Embedding Uncertainty (Fig. 6b): We divide the ICD-10 codes into 10 buckets based on their degrees (i.e. number of visits a code is connected to). We compute the average variance of each bucket and plot it against the $\log_{10}(degree(code))$. We observe that the average variance decreases, when the code degree increases in Fig. 6b. Intuitively, lower degree codes (i.e. when the code rarely occurs) have less structural information to learn, hence their embeddings have a higher degree of uncertainty. In contrast, higher degree codes occur more frequently, so they possess a lower embedding uncertainty as they are more expressive in terms of the structure.



(a) Visit uncertainty. (b) Code uncertainty.

Figure 6: Analysis of uncertainty of the embeddings. Trendlines show the trend of the results.

5 Conclusion

Learning low-dimensional representations for EMRs is essential in improving personalised healthcare. In this work, we propose `MedGraph`, an effective EMR embedding framework for visits and codes. `MedGraph` introduces a graph-based data structure to naturally capture both visit-code co-location information *structurally*, and visit sequencing information *temporally*. Based on this structure `MedGraph` learns from the visit-code bipartite graph and exploits temporal point processes to capture medical history in an end-to-end manner. `MedGraph` supports visit- and code-level attributes. We further improve the expressive power of `MedGraph` by modelling uncertainty of the embeddings. Results on two real-world EMR datasets demonstrate that `MedGraph` produces meaningful representations for EMRs, significantly outperforming state-of-the-art EMR embedding methods on a number of medical risk prediction tasks.

Acknowledgements

This work has been supported by the Monash Institute of Medical Engineering (MIME), Australia.

REFERENCES

[1] Tian Bai, Brian L. Egleston, Richard Bleicher, and Slobodan Vucetic, 'Medical Concept Representation Learning from Multi-source Data', in *IJCAI*, (2019).
[2] Xiangrui Cai, Jinyang Gao, Kee Yuan Ngiam, Beng Chin Ooi, Ying Zhang, and Xiaojie Yuan, 'Medical Concept Embedding with Time-Aware Attention', in *IJCAI*, (2018).
[3] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad, 'Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission', in *ACM SIGKDD*, (2015).
[4] Tianqi Chen and Carlos Guestrin, 'XGBoost: A Scalable Tree Boosting System', in *ACM SIGKDD*, (2016).
[5] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun, 'Doctor AI: Predicting Clinical Events via Recurrent Neural Networks', in *MLHC*, (2016).
[6] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun, 'Multi-layer Representation Learning for Medical Concepts', in *ACM SIGKDD*, (2016).
[7] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, and Jimeng Sun, 'GRAM: Graph-based Attention Model for Healthcare Representation Learning', in *ACM SIGKDD*, (2017).
[8] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter F. Stewart, 'RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism', in *NIPS*, (2016).
[9] Edward Choi, Andy Schuetz, Walter F. Stewart, and Jimeng Sun, 'Using recurrent neural network models for early detection of heart failure onset', *JAMIA*, **24**(2), 361–370, (2017).
[10] Edward Choi, Cao Xiao, Walter F. Stewart, and Jimeng Sun, 'MiME: Multilevel Medical Embedding of Electronic Health Records for Predictive Healthcare', in *NeurIPS*, (2018).
[11] Edward Choi, Zhen Xu, Yujia Li, Michael W. Dusenberry, Gerardo Flores, Yuan Xue, and Andrew M. Dai, 'Learning the Graphical Structure of Electronic Health Records with Graph Convolutional Transformer', (2020).
[12] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song, 'Recurrent Marked Temporal Point Processes: Embedding Event History to Vector', in *ACM SIGKDD*, (2016).
[13] Clark R Givens, Rae Michael Shortt, et al., 'A class of Wasserstein metrics for probability distributions', *The Michigan Mathematical Journal*, **31**(2), 231–240, (1984).
[14] Alan G. Hawkes, 'Spectra of Some Self-Exciting and Mutually Exciting Point Processes', *Biometrika*, **58**(1), 83–90, (1971).
[15] Bhagya Hettige, Yuan-Fang Li, Weiqing Wang, and Wray L. Buntine, 'Gaussian Embedding of Large-Scale Attributed Graphs', in *ADC*, (2020).
[16] Bhagya Hettige, Weiqing Wang, Yuan-Fang Li, Suong Le, and Wray L. Buntine, 'Robust Attribute and Structure Preserving Graph Embedding', in *PAKDD*, (2020).
[17] Abhyuday N. Jagannatha and Hong Yu, 'Bidirectional RNN for Medical Event Detection in Electronic Health Records', in *NAACL HLT*, (2016).
[18] Devan Kansagara, Honora Englander, Amanda Salanitro, David Kagen, Cecelia Theobald, Michele Freeman, and Sunil Kripalani, 'Risk Prediction Models for Hospital Readmission: A Systematic Review', *JAMA*, **306**(15), 1688–1698, (2011).
[19] John Frank Charles Kingman, *Poisson Processes*, volume 3, Oxford University Press, 1992.
[20] John Frank Charles Kingman, *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*, volume 2, Springer, 2007.
[21] Thomas N. Kipf and Max Welling, 'Semi-Supervised Classification with Graph Convolutional Networks', in *ICLR*, (2017).
[22] William A Knaus, Douglas P Wagner, Elizabeth A Draper, Jack E Zimmerman, Marilyn Bergner, Paulo G Bastos, Carl A Sirio, Donald J Murphy, Ted Lotring, Anne Damiano, et al., 'The APACHE III prognostic system: risk prediction of hospital mortality for critically III hospitalized adults', *Chest*, **100**(6), 1619–1636, (1991).
[23] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao, 'Dipole: Diagnosis Prediction in Healthcare via Attention-based Bidirectional Recurrent Neural Networks', in *ACM SIGKDD*, (2017).
[24] H Matsuura and Y Yamaji, 'Tuberculous pneumonia', *QJM: An International Journal of Medicine*, **111**(2), 131–131, (10 2017).
[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean, 'Distributed Representations of Words and Phrases and their Compositionality', in *NIPS*, (2013).
[26] Riccardo Miotto, Li Li, Brian Kidd, and Joel T. Dudley, 'Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records', *Scientific Reports*, **6**, 26094, (05 2016).
[27] Mohammad Amin Morid, Olivia R. Liu Sheng, and Samir E. AbdelRahman, 'Leveraging Time Series Data in Similarity Based Healthcare Predictive Models: The Case of Early ICU Mortality Prediction', in *AMCIS*, (2017).
[28] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh, 'Deepr: A Convolutional Net for Mdical Records', *IEEE Journal of Biomedical and Health Informatics*, **21**(1), 22–30, (2016).
[29] Trang Pham, Truyen Tran, Dinh Q. Phung, and Svetha Venkatesh, 'DeepCare: A Deep Dynamic Memory Model for Predictive Medicine', in *PAKDD*, (2016).
[30] Zhi Qiao, Xian Wu, Shen Ge, and Wei Fan, 'MNN: Multimodal Attentional Neural Networks for Diagnosis Prediction', in *IJCAI*, (2019).
[31] Zhi Qiao, Shiwan Zhao, Cao Xiao, Xiang Li, Yong Qin, and Fei Wang, 'Pairwise-Ranking based Collaborative Recurrent Neural Networks for Clinical Event Prediction', in *IJCAI*, (2018).
[32] P. E. Rauber, A. X. Falcão, and A. C. Telea, 'Visualizing Time-Dependent Data Using Dynamic t-SNE', in *Eurographics Conference on Visualization*, (2016).
[33] Benjamin Shickel, Patrick Tighe, Azra Bihorac, and Parisa Rashidi, 'Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis', *IEEE Journal of Biomedical and Health Informatics*, **22**(5), 1589–1604, (2018).
[34] Lihong Song, Chin Wang Cheong, Kejing Yin, William K. Cheung, Benjamin C. M. Fung, and Jonathan Poon, 'Medical Concept Embedding with Multiple Ontological Representations', in *IJCAI*, pp. 4613–4619, (2019).
[35] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei, 'LINE: Large-scale Information Network Embedding', in *WWW*, (2015).
[36] Dingyuan Zhu, Peng Cui, Daixin Wang, and Wenwu Zhu, 'Deep Variational Network Embedding in Wasserstein Space', in *ACM SIGKDD*, (2018).