2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) | 978-1-6654-5661-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ASONAM55673.2022.10068627

Dynamic Healthcare Embeddings for Improving Patient Care

Hankyu Jang
Department of Computer Science
University of Iowa
Iowa City, Iowa, USA
hankyu-jang@uiowa.edu

Sulyun Lee
Interdisciplinary Graduate Program in Informatics
University of Iowa
Iowa City, Iowa, USA
sulyun-lee@uiowa.edu

D. M. Hasibul Hasan
Department of Computer Science
University of Iowa
Iowa City, Iowa, USA
dmhasibul-hasan@uiowa.edu

Philip M. Polgreen
Department of Internal Medicine
University of Iowa
Iowa City, Iowa, USA
philip-polgreen@uiowa.edu

Sriram V. Pemmaraju
Department of Computer Science
University of Iowa
Iowa City, Iowa, USA
sriram-pemmaraju@uiowa.edu
*For the CDC MInD Healthcare Network

Bijaya Adhikari
Department of Computer Science
University of Iowa
Iowa City, Iowa, USA
bijaya-adhikari@uiowa.edu

***Abstract*—As hospitals move towards automating and integrating their computing systems, more fine-grained hospital operations data are becoming available. These data include hospital architectural drawings, logs of interactions between patients and healthcare professionals, prescription data, procedures data, and data on patient admission, discharge, and transfers. This has opened up many fascinating avenues for healthcare-related prediction tasks for improving patient care. However, in order to leverage off-the-shelf machine learning software for these tasks, one needs to learn structured representations of entities involved from heterogeneous, dynamic data streams. Here, we propose DECENT, an auto-encoding heterogeneous co-evolving dynamic neural network, for learning heterogeneous dynamic embeddings of patients, doctors, rooms, and medications from diverse data streams. These embeddings capture similarities among doctors, rooms, patients, and medications based on static attributes and dynamic interactions.**

DECENT enables several applications in healthcare prediction, such as predicting mortality risk and case severity of patients, adverse events (e.g., transfer back into an intensive care unit), and future healthcare-associated infections. The results of using the learned patient embeddings in predictive modeling show that DECENT has a gain of up to 48.1% on the mortality risk prediction task, 12.6% on the case severity prediction task, 6.4% on the medical intensive care unit transfer task, and 3.8% on the Clostridioides difficile (C.diff) Infection (CDI) prediction task over the state-of-the-art baselines. In addition, case studies on the learned doctor, medication, and room embeddings show that our approach learns meaningful and interpretable embeddings.

***Index Terms*—dynamic embedding, heterogeneous networks, patient embedding, healthcare analytics**

I. INTRODUCTION

The availability of large-scale, high-resolution hospital operations data has opened up many avenues for the use of predictive modeling to improve patient care. Questions such as “How likely is a patient at risk for an adverse event or complication that might require transfer into a Medical Intensive Care Unit (MICU) from another unit?”, “Is a particular patient at risk of developing a *healthcare-associated infection* (HAI) during their visit?”, “Does a patient have a high risk of mortality?”. These are just a few examples of key predictive tasks which could help healthcare practitioners provide informed and personalized patient care.

Recent advances in machine learning have enabled predictions in numerous domains. However, machine learning models such as recurrent neural networks, graph convolution networks, and transformers, often leveraged for predictive tasks, assume that the input data is structured. Unfortunately, hospital operations data consists of diverse data types, including architectural diagrams, admission-discharge-transfer logs, inpatient-doctor interactions, room visits, prescriptions, clinical notes, etc. These data tend to be unstructured and high-dimensional. Hence, learning structured representations of entities such as patients, doctors, rooms, and medications from hospital operations data is a key step in enabling the use of off-the-shelf machine learning algorithms for predictive tasks in healthcare settings.

There has been some recent interest in learning embeddings in a healthcare setting, primarily focusing on patient embeddings, such as MiME [1]. A major drawback of these approaches is that they all learn *static* embeddings. In a healthcare setting, events that occur over time (e.g., prescription of an antibiotic, transfer to the MICU, and exposure to infection) play a crucial role, and a single static embedding is not representative enough. For example, a patient with a low risk of HAI at admission time could be considered high risk at a later day if there are patients with HAI in her unit. However, she could be considered low risk again in the future if she is treated with appropriate medications and shows no symptoms. Thus, learning a single embedding for the entire duration of the patient’s stay does not capture the evolving nature of the risks involved and the care patient has received.

IEEE/ACM ASONAM 2022, November 10-13, 2022
978-1-6654-5661-6/22/$31.00 © 2022 IEEE

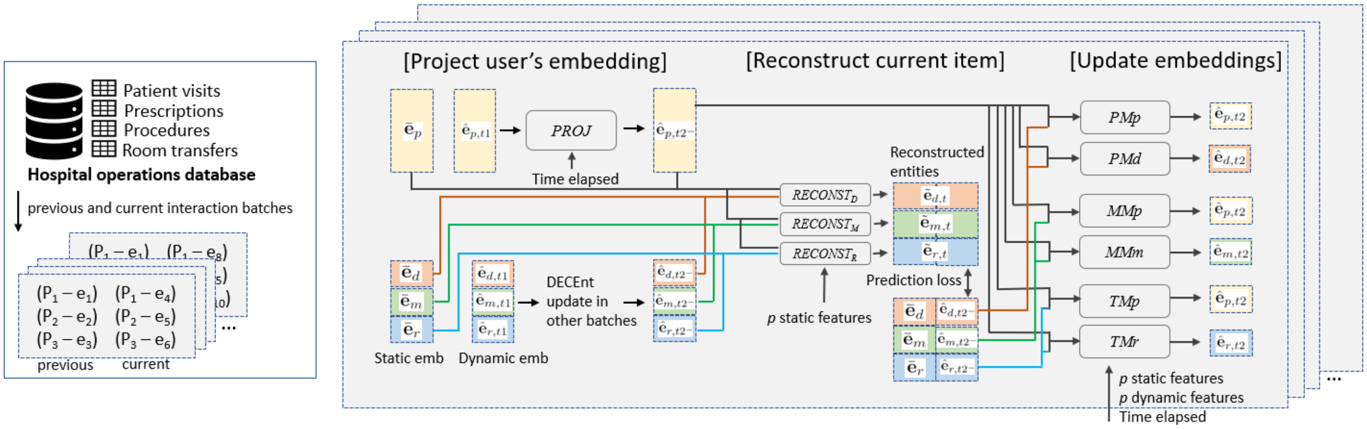
Patient visits
Prescriptions
Procedures
Room transfers
Hospital operations database
previous and current interaction batches
$(P_1 - e_1)$ $(P_1 - e_8)$
$(P_1 - e_1)$ $(P_1 - e_4)$
$(P_2 - e_2)$ $(P_2 - e_5)$
$(P_3 - e_3)$ $(P_3 - e_6)$
previous
current
...
[Project user's embedding]
[Reconstruct current item]
[Update embeddings]
$\bar{\mathbf{e}}_p$
$\hat{\mathbf{e}}_{p,t1}$
PROJ
$\hat{\mathbf{e}}_{p,t2-}$
Time elapsed
$\bar{\mathbf{e}}_d$
$\bar{\mathbf{e}}_m$
$\bar{\mathbf{e}}_r$
$\hat{\mathbf{e}}_{d,t1}$
$\hat{\mathbf{e}}_{m,t1}$
$\hat{\mathbf{e}}_{r,t1}$
DECEnt update in other batches
$\hat{\mathbf{e}}_{d,t2-}$
$\hat{\mathbf{e}}_{m,t2-}$
$\hat{\mathbf{e}}_{r,t2-}$
Static emb
Dynamic emb
$RECONST_D$
$RECONST_M$
$RECONST_R$
p static features
Reconstructed entities
$\tilde{\mathbf{e}}_{d,t}$
$\tilde{\mathbf{e}}_{m,t}$
$\tilde{\mathbf{e}}_{r,t}$
Prediction loss
PMp
PMd
MMp
MMm
TMp
TMr
$\hat{\mathbf{e}}_{p,t2}$
$\hat{\mathbf{e}}_{d,t2}$
$\hat{\mathbf{e}}_{p,t2}$
$\hat{\mathbf{e}}_{m,t2}$
$\hat{\mathbf{e}}_{p,t2}$
$\hat{\mathbf{e}}_{r,t2}$
p static features
p dynamic features
Time elapsed
...

Fig. 1: Our overall framework. Our approach DECEnt learns dynamic embeddings of healthcare entities from heterogeneous interaction log and other hospital operations data. For an interaction between a patient p and an entity e at time t_2, p's dynamic embedding at time t_1 (time of p's previous interaction) is projected to time t_2^- (right before t_2). Then, the static and dynamic embeddings of p and e at time t_2^- and p's static features are used to reconstruct static and dynamic embeddings of entity e. Finally, dynamic embeddings of p and e get updated to time t_2 via update modules. We train DECEnt in batches of interactions in parallel, while maintaining the temporal order of interactions across batches.

Furthermore, none of the existing research on patient embeddings takes interactions between patients and other healthcare entities (e.g., physicians, hospital rooms, and medications) into account. However, capturing these interactions in learned embeddings is critical for numerous healthcare-associated predictive tasks. As an example, consider two interactions that happen in quick succession: (i) a patient p_1 in hospital room r_1 is prescribed the antibiotic vancomycin, which is commonly used as a treatment for suspected CDI[1], (ii) a patient p_2 transfers into hospital room r_2 that is in the same unit as room r_1. Together, these interactions indicate an elevated risk of C. diff infection for patient p_2, and we want the embeddings we learn to capture this.

There is also a separate thread of research on learning general-purpose dynamic embeddings (e.g., [3], [4]). However, these approaches learn embeddings from *homogeneous* interactions between a user and a predefined item type. Such an approach is not readily applicable in a healthcare setting, where a patient may interact with heterogeneous entities, including physicians, medications, and rooms, which have different impacts on patients. Hence lumping all of these together into a single entity type will limit the discriminative power of the learned embeddings. Another line of related approaches includes dynamic network embeddings [5]. However, these approaches are not readily applicable in our setting as they usually require a coarse snapshot representation of the network.

To address the gap between existing approaches and the requirements of healthcare predictive modeling, we propose DECEnt to learn dynamic embedding of entities associated with healthcare based on heterogeneous interactions. DECEnt learns general-purpose embedding in an unsupervised way which can be used in various predictive modeling tasks that can not be obtained via supervised training. Specifically, DECEnt jointly learns dynamic embeddings of patients, doctors, medications, and rooms while preserving hierarchical relationships between medications, doctor specializations, and physical proximity between rooms. In order to do so, DECEnt maintains and updates weights for each specific interaction type and minimizes intra-entity similarity loss while performing an auto-encoder training scheme so that the embeddings of patient-entity interactions can re-construct their original features. Our overall framework is presented in Figure 1. Our contributions in this paper are as follows:

- We propose DECEnt, a novel approach for learning dynamic embeddings of healthcare entities from heterogeneous interactions.
- DECEnt enables several healthcare predictive modeling applications, including adverse event prediction, such as transfer to MICU, case severity and mortality prediction.
- We perform extensive experiments for evaluating patient embeddings. Our results show that DECEnt outperforms state-of-the-art baselines in all the healthcare predictive modeling tasks we consider. Moreover, our embeddings are interpretable and meaningful.

II. Preliminaries

In this section, we give our setup, introduce the notations used throughout the paper, and finally state the problem formally.

A. Setup and Notation

Assume we are given a hospital operations database with a record of events on healthcare entities. The set of all healthcare

[1] CDI is shorthand for *Clostridioides difficile infection*, a healthcare-associated infection that affects gastro-intestinal regions leading to inflammation of the colon and severe diarrhea due to disruption of normal healthy bacteria in the colon [2].

entities τ includes the set of Doctors $\mathcal{D}$, the set of Patients $\mathcal{P}$, the set of medications $\mathcal{M}$, and the set of hospital rooms $\mathcal{R}$.

The database has a record of time-stamped interactions between entities for each medical event that occurred in the hospital. In this paper, we consider three types of interactions. A **physician** interaction (p, d, t), with $t \in \mathbb{R}^+$ and $0 \leq t \leq T$, represents that a doctor $d \in \mathcal{D}$ performed a medical procedure on patient $p \in \mathcal{P}$ at time t. Similarly, a **medication** interaction (p, m, t) indicates that medicine $m \in \mathcal{M}$ was prescribed to patient $p \in \mathcal{P}$ at time t, and finally a **spatial** interaction (p, r, t) represents that a patient $p \in \mathcal{P}$ was transferred to a hospital room $r \in \mathcal{R}$ at time t. We represent sets of physician, medication, and spatial interactions as $\mathcal{PR}$, $\mathcal{MD}$, and $\mathcal{TR}$, respectively.

The database also consists of relationships among entities of the same type, represented as static graphs $G_{\text{entitytype}}$. These are as follows:

- **Room graph:** The architectural layout of the hospital can be represented as a graph $G_{\text{room}}(\mathcal{R}, E_{\text{room}})$ between hospital rooms. Two rooms r_1 and r_2 are connected by an edge (r_1, r_2) in G_{room} if they are adjacent to each other.
- **Medication graph:** The medication hierarchy can be represented as a tree $G_{\text{med}}(\mathcal{M} \cup M, E_{\text{med}})$. Note that each leaf m in the tree is a medication, i.e., $m \in \mathcal{M}$. The intermediate nodes M represent medication sub-types.
- **Doctor graph:** Each doctor $d \in \mathcal{D}$ has a specialty. We create a graph $G_{\text{doc}}(\mathcal{D}, E_{\text{doc}})$, where the edges (d_1, d_2) are based on the proximity of the specialty of doctors d_1 and d_2.

Finally, for each patient $p \in \mathcal{P}$, we are given static attributes $\mathbf{p}_p$ with demographic and medical information. We are also given dynamic attributes $\hat{\mathbf{p}}_{p,t}$ at time t that include information on the length of hospital stay, cumulative antibiotics count, gastric acid suppressors, and others.

In what follows, the lower case bold letters such as $\mathbf{v}$ represent vectors, the uppercase bold letters such as $\mathbf{W}$ represent matrices, and calligraphic symbols such as $\mathcal{S}$ represent sets. The lower case bold letters with hat such as $\hat{\mathbf{v}}$ represent dynamic vectors, and $\hat{\mathbf{v}}_t$ represents the vector at time t. Functions are represented by lowercase letters followed by braces, for example, $f(\cdot)$.

B. Problem Statement

Having defined the notations, we can now state our problem. Our goal is to learn dynamic embeddings for all patients, medications, rooms, and doctors based on the set of all interactions $\mathcal{S} = \mathcal{PR} \cup \mathcal{MD} \cup \mathcal{TR}$. Without the loss of generality, we assume that the interactions in $\mathcal{S}$ are sorted by time. We also aim to preserve the relationship among the entities of the same type represented by the static graphs in the embedding space. Formally, the problem can be stated as follows:

Problem 1: DYNAMIC HEALTHCARE EMBEDDINGS PROBLEM

Given: A set $\mathcal{S}$ of time-stamped interactions among healthcare entities, static networks G_{room}, G_{med}, G_{doc}, and dynamic and static attributes of patients $\hat{\mathbf{p}}$ and $\mathbf{p}$.
Learn: Dynamic embeddings $\hat{\mathbf{e}}_{u,t}$ for each entity $u \in \mathcal{P}$ and $\hat{\mathbf{e}}_{v,t}$ for each entity $v \in \mathcal{D} \cup \mathcal{M} \cup \mathcal{R}$ and for each time t.
Such that: A function $f(\hat{\mathbf{e}}_{u,t}, \hat{\mathbf{e}}_{v,t})$ encodes information to be predictive of v, and the $dist(\hat{\mathbf{e}}_{v,t}, \hat{\mathbf{e}}_{v',t})$ between the embeddings $\hat{\mathbf{e}}_{v,t}$ and $\hat{\mathbf{e}}_{v',t}$ is of two entities of the same type v and v' reflects the distance between the two in $G_{\text{entitytype}}$.

III. METHOD

Here we propose our method DECENT (short for **D**YNAMIC **E**MBEDDING OF HEALTH**C**ARE **ENT**ITIES), a novel auto-encoding heterogeneous co-evolving dynamic neural network, to solve Problem 1. DECENT models heterogeneous interactions using a set of jointly learned co-evolving networks. It has three main components at a high level, each of which models one of the interaction types. As the interactions occur, the dynamic embeddings of the entities involved are updated simultaneously. These dynamic embeddings are jointly trained to reconstruct the entity that the patients interacts with while preserving similarities imposed by $G_{\text{entitytype}}$. We describe each of the components in detail next.

A. Physician Module

Consider a physician interaction $(p, d, t) \in \mathcal{PR}$ between a patient p and a doctor d at time t. To reflect that p and d interacted with each other, our method simultaneously updates the maintained dynamic embeddings $\hat{\mathbf{e}}_{p,t}$ and $\hat{\mathbf{e}}_{d,t}$ of both p and d. To do so, we design a pair of co-evolving deep neural networks PM_p and PM_d to update the patient's and doctor's embedding, respectively.

Following the literature in the co-evolutionary neural networks [3], [4], we design PM_p and PM_d to be mutually recursive, i.e., the patient's embedding at time t, $\hat{\mathbf{e}}_{p,t}$ generated by the PM_p depends on both the patient's embedding at time t^- (just before time t) $\hat{\mathbf{e}}_{p,t^-}$ and the doctor's embedding $\hat{\mathbf{e}}_{d,t^-}$ prior to the interaction as well as the time elapsed between the patient's previous and the current interaction $\Delta_{p,t}$.

However, relying only on interactions is not enough for patient care predictive modeling applications. While the interactions capture the medical events of a particular patient, they do not capture the inherent risk the patient is at due to the patient's own underlying health conditions and demographic features. Hence, we ensure that the patient's dynamic embedding $\hat{\mathbf{e}}_{p,t}$ also depends on the patient's static feature $\mathbf{p}_p$ and the dynamic feature $\hat{\mathbf{p}}_{p,t}$ at time t. We ensure that the doctor's dynamic embedding $\hat{\mathbf{e}}_{d,t}$ depends on $\mathbf{p}_p$ and $\hat{\mathbf{p}}_{p,t}$ to better track all the patients the doctor d has interacted with.

Following are the update equations for PM_p and PM_d respectively.

$$\begin{aligned}
\hat{\mathbf{e}}_{p,t} &= \sigma\left[\mathbf{W}_p^{PM}[\hat{\mathbf{e}}_{p,t^-} \mid \hat{\mathbf{e}}_{d,t^-} \mid \Delta_{p,t} \mid \mathbf{p}_p \mid \hat{\mathbf{p}}_{p,t}] + \mathbf{B}_p^{PM}\right] \\
\hat{\mathbf{e}}_{d,t} &= \sigma\left[\mathbf{W}_d^{PM}[\hat{\mathbf{e}}_{d,t^-} \mid \hat{\mathbf{e}}_{p,t^-} \mid \Delta_{d,t} \mid \mathbf{p}_p \mid \hat{\mathbf{p}}_{p,t}] + \mathbf{B}_d^{PM}\right]
\end{aligned} \tag{1}$$

Here, $\mathbf{W}_p^{PM}$ is the weight matrix parameterizing PM_p and $\mathbf{W}_d^{PM}$ is the weight matrix for PM_d. $\mathbf{B}_p^{PM}$ and $\mathbf{B}_d^{PM}$ are bias. $[\mathbf{a} \mid \mathbf{b}]$ denotes concatenation of the two vectors $\mathbf{a}$ and $\mathbf{b}$. Finally σ is a non-linear activation function, and in our experiments, we set σ to be the tanh activation.

Note that $\hat{\mathbf{e}}_{p,t^-}$ in Equation 1 is the patient embedding immediately prior to the interaction (p, d, t). In earlier related approaches, $\hat{\mathbf{e}}_{p,t^-}$ is taken to be the embedding generated by PM_p for patient p's last interaction before time t. However, in cases where the gap between consecutive interactions are too large, this approach is sub-optimal. Hence, in this paper, we use the embedding projection operation [4]. Specifically, a patient p's embedding after time Δ is projected to be,

$$\hat{\mathbf{e}}_{p,t+\Delta} = (1 + \mathbf{W} \times \Delta) + \hat{\mathbf{e}}_{p,t} \tag{2}$$

where $\mathbf{W}$ is a linear weight matrix. We use Equation 2 to generate $\hat{\mathbf{e}}_{p,t^-}$ by projecting the embedding generated by PM_p for p's previous interaction.

B. Medication Module

Similar to the Physician module, we maintain a pair of co-evolving deep neural networks MM_p and MM_m to update the embeddings $\hat{\mathbf{e}}_{p,t}$ and $\hat{\mathbf{e}}_{m,t}$ post a medication interaction (p, m, t). Recall that a medication interaction $(p, m, t) \in \mathcal{MD}$ represents that a medication m was prescribed to a patient p at time t. Here, MM_p updates the dynamic embedding of p at time t and MM_m does the same for medicine m. The update equation for the medication module are similar to the physician module,

$$\begin{aligned}
\hat{\mathbf{e}}_{p,t} &= \sigma\left[\mathbf{W}_p^{MM}[\hat{\mathbf{e}}_{p,t^-} \mid \hat{\mathbf{e}}_{m,t^-} \mid \Delta_{p,t} \mid \mathbf{p}_p \mid \hat{\mathbf{p}}_{p,t}] + \mathbf{B}_p^{MM}\right] \\
\hat{\mathbf{e}}_{m,t} &= \sigma\left[\mathbf{W}_m^{MM}[\hat{\mathbf{e}}_{m,t^-} \mid \hat{\mathbf{e}}_{p,t^-} \mid \Delta_{m,t} \mid \mathbf{p}_p \mid \hat{\mathbf{p}}_{p,t}] + \mathbf{B}_m^{MM}\right]
\end{aligned} \tag{3}$$

where $\mathbf{W}_p^{MM}$ and $\mathbf{W}_m^{MM}$ are the weight matrices for MM_p and MM_m, respectively. $\mathbf{B}_p^{MM}$ and $\mathbf{B}_m^{MM}$ are bias.

C. Transfer Module

Similar to the previous two modules, the transfer module also consist of two co-evolving neural networks. TM_p updates the patient embedding $\hat{\mathbf{e}}_{p,t}$ and TM_r updates the room embedding post a transfer interaction $(p, r, t) \in \mathcal{TR}$. The update equations for the transfer module are as follows:

$$\begin{aligned}
\hat{\mathbf{e}}_{p,t} &= \sigma\left[\mathbf{W}_p^{TM}[\hat{\mathbf{e}}_{p,t^-} \mid \hat{\mathbf{e}}_{r,t^-} \mid \Delta_{p,t} \mid \mathbf{p}_p \mid \hat{\mathbf{p}}_{p,t}] + \mathbf{B}_p^{TM}\right] \\
\hat{\mathbf{e}}_{r,t} &= \sigma\left[\mathbf{W}_r^{TM}[\hat{\mathbf{e}}_{r,t^-} \mid \hat{\mathbf{e}}_{p,t^-} \mid \Delta_{r,t} \mid \mathbf{p}_p \mid \hat{\mathbf{p}}_{p,t}] + \mathbf{B}_r^{TM}\right]
\end{aligned} \tag{4}$$

where $\mathbf{W}_p^{TM}$ and $\mathbf{W}_r^{TM}$ are the weight matrices for TM_p and TM_r, respectively. $\mathbf{B}_p^{TM}$ and $\mathbf{B}_r^{TM}$ are bias.

D. Reconstruction Module

Consider the interaction (p, e, t_2) such that a patient p interacted with an entity e at time t_2 and that p's previous interaction occurred at time t_1 such that $t_1 < t_2$. To learn a meaningful representations of patient p and an entity e in the latent space, we train DECENT to update embeddings of both p and e such that they are capable of reconstructing the embedding of e at time t_2. Specifically, while processing the interaction (p, e, t_2), our goal is to learn dynamic embeddings $\hat{\mathbf{e}}_{p,t_2^-}$ and $\hat{\mathbf{e}}_{e,t_2^-}$ immediately before time t_2 such that they are useful in reconstructing both dynamic embedding $\hat{\mathbf{e}}_{e,t_2^-}$ and the static embedding $\bar{\mathbf{e}}_e$ of entity e.

In order to reconstruct the dynamic and the static embeddings of entity e, we feed in static embeddings $\bar{\mathbf{e}}_p$ and $\bar{\mathbf{e}}_e$ of patient p and entity e along with the projection of the generated dynamic embeddings $\hat{\mathbf{e}}_{p,t_2^-}$ and dynamic embeddings of e that is updated via consecutive interactions with other patients $\hat{\mathbf{e}}_{e,t_2^-}$ to the reconstruction module. Each entity type (e.g., doctor, medication, room) has its own reconstruction module (e.g., *RECONST*$_{\mathrm{D}}$, *RECONST*$_{\mathrm{M}}$, *RECONST*$_{\mathrm{R}}$), which is defined as follows:

$$\begin{aligned}
\tilde{\mathbf{e}}_{d,t_2^-} &= \mathbf{W}_d\left[\hat{\mathbf{e}}_{p,t_2^-} \mid \bar{\mathbf{e}}_p \mid \mathbf{p}_p \mid \hat{\mathbf{e}}_{d,t_2^-} \mid \bar{\mathbf{e}}_d\right] + \mathbf{B}_d \\
\tilde{\mathbf{e}}_{m,t_2^-} &= \mathbf{W}_m\left[\hat{\mathbf{e}}_{p,t_2^-} \mid \bar{\mathbf{e}}_p \mid \mathbf{p}_p \mid \hat{\mathbf{e}}_{m,t_2^-} \mid \bar{\mathbf{e}}_m\right] + \mathbf{B}_m \\
\tilde{\mathbf{e}}_{r,t_2^-} &= \mathbf{W}_r\left[\hat{\mathbf{e}}_{p,t_2^-} \mid \bar{\mathbf{e}}_p \mid \mathbf{p}_p \mid \hat{\mathbf{e}}_{r,t_2^-} \mid \bar{\mathbf{e}}_r\right] + \mathbf{B}_r
\end{aligned} \tag{5}$$

where $\mathbf{W}_d$, $\mathbf{W}_m$, and $\mathbf{W}_r$ are the learnable weight matrices for reconstructing embeddings of doctor, medication, and room and $\mathbf{B}_d$, $\mathbf{B}_m$, and $\mathbf{B}_r$ are bias. Note that $\tilde{\mathbf{e}}_{e,t_2^-}$ is the predicted embedding of size $|\hat{\mathbf{e}}_{e,t_2^-}| + |\bar{\mathbf{e}}_e|$.

DECENT uses onehot vector for static embeddings. We have another model DECENT+, which we use Bourgain embeddings [6] for static embeddings of $v \in \mathcal{D} \cup \mathcal{M} \cup \mathcal{R}$, which we compute from static graphs $G_{\text{entitytype}}$.

E. Overall Framework

While our primary goal is to train DECENT to encode the entity that a patient interacts with, we also want to enforce additional losses to ensure that the generated embeddings are interpretable. We analyze our learned embeddings with the help of a domain expert (See Section IV). We describe the losses used to train DECENT next.

Reconstruction Loss: *Reconstruction loss* is encoded as the error between the predicted and the ground truth embeddings of the entity a patient interacts with. Continuing with our example from Section III-D, we want to minimize the difference between the reconstructed embedding $\tilde{\mathbf{e}}_{e,t_2^-}$ and the ground truth embedding $[\hat{\mathbf{e}}_{e,t_2^-} \mid \bar{\mathbf{e}}_{e_2}]$, where $\mid$ is the concatenation operation. We enforce the reconstruction loss on all interactions including procedure interactions $\mathcal{PR}$, medication interactions $\mathcal{MD}$, and transfer interactions $\mathcal{TR}$.

Formally, reconstruction loss is defined as follows:

$$\begin{aligned}
L_{reconst} = &\sum_{(p,d,t) \in \mathcal{PR}} ||\tilde{\mathbf{e}}_{d,t^-} - [\hat{\mathbf{e}}_{d,t^-} | \bar{\mathbf{e}}_d]||_2 \\
+ &\sum_{(p,m,t) \in \mathcal{MD}} ||\tilde{\mathbf{e}}_{m,t^-} - [\hat{\mathbf{e}}_{m,t^-} | \bar{\mathbf{e}}_m]||_2 \\
+ &\sum_{(p,r,t) \in \mathcal{TR}} ||\tilde{\mathbf{e}}_{r,t^-} - [\hat{\mathbf{e}}_{r,t^-} | \bar{\mathbf{e}}_r]||_2
\end{aligned} \tag{6}$$

where $||\mathbf{v}||_2$ is the L_2 norm of the vector $\mathbf{v}$.

Temporal Consistency Loss: We want to ensure that the embeddings of entities do not vary dramatically between consecutive interactions. To this end, we define *temporal consistency* loss as the L_2 norm of the difference between the embeddings of each entity between each consecutive interaction. Formally, it is defined as follows:

$$L_{temp} = \sum_{(p,e,t)\in\mathcal{S}} ||\hat{\mathbf{e}}_{p,t} - \hat{\mathbf{e}}_{p,t-}||_2 + ||\hat{\mathbf{e}}_{e,t} - \hat{\mathbf{e}}_{e,t-}||_2 \quad (7)$$

As defined earlier, $\mathcal{S}$ is the set of all interactions, i.e., $\mathcal{S} = \mathcal{PR} \cup \mathcal{MD} \cup \mathcal{TR}$.

Domain Specific Loss: Furthermore, we also implement a set of losses to ensure that the entities known to be similar as per domain knowledge have similar embeddings. To this end, we first compute the Laplacian matrices, $\mathbf{L}_{\text{room}}$, $\mathbf{L}_{\text{med}}$, and $\mathbf{L}_{\text{doc}}$ corresponding to the static graphs, G_{room}, G_{med}, and G_{doc} representing similarities between the entities (see Section II-A for details on the graphs). We then compute the graph Laplacian based *domain specific loss* as follows:

$$\begin{aligned} L_{dom} = \lambda_{dom}^{D} & \sum_{t\in[0,T],d\in\mathcal{D}} \hat{\mathbf{e}}_{d,t}^{T}\mathbf{L}_{\text{doc}}\hat{\mathbf{e}}_{d,t} \quad (8) \\ + \lambda_{dom}^{M} & \sum_{t\in[0,T],m\in\mathcal{M}} \hat{\mathbf{e}}_{m,t}^{T}\mathbf{L}_{\text{med}}\hat{\mathbf{e}}_{m,t} \\ + \lambda_{dom}^{R} & \sum_{t\in[0,T],r\in\mathcal{R}} \hat{\mathbf{e}}_{r,t}^{T}\mathbf{L}_{\text{room}}\hat{\mathbf{e}}_{r,t} \end{aligned}$$

Note that $\hat{\mathbf{e}}_{r,t}^{T}$ is a transpose of $\hat{\mathbf{e}}_{r,t}$ and λ_{dom}^{D}, λ_{dom}^{M} and λ_{dom}^{R} are scaling constants in the equation above. Note that the L_{dom} decreases in magnitude if entities connected by an edge have similar embeddings.

Overall Loss and Training: Our overall loss is the weighted aggregation of the previous three losses. Note that we jointly train all three modules along with the reconstruction module after pre-training each individual module. We optimize the overall loss using the Adam optimization algorithm [7]. We use Adam optimizer with the learning rate of 1e-3 and the weight decay of 1e-5. The size of the dynamic embeddings of DECEnt and DECEnt+ are set to 128. We train DECEnt and DECEnt+ for 1000 epochs with an early stopping on the training loss with the patience of 10 epochs.

IV. Experiment

We describe our experimental setup next. We provide code for academic purposes [2] Experiments were conducted on Intel(R) Xeon(R) machine with 528GB memory and 4 GPUs (GeForce GTX 1080 Ti).

Data: We evaluate DECEnt on the real world hospitals operation data collected from University of Iowa Hospitals and Clinics (UIHC). UIHC is a large (800-bed) tertiary care teaching hospital located in Iowa City, Iowa. The dataset consists of de-identified electronic medical records (EMR), and admission-discharge-transfer (ADT) records on 6,496 patients between January 01, 2010 and March 31, 2010. Each patient visit has a set of diagnoses, a timestamped record on a set of medications prescribed, and a procedures performed by physicians. For each patient room transfer event during the visit, the source room and the destination room are recorded with a timestamp. The patients in our dataset interacted with 575 doctors where 23,085 physician interactions were observed during the time-frame. 686 unique medicines were prescribed to patients, with a total of 349,345 medication interactions. Moreover, patients visited 557 rooms, with a total of 16,771 spatial interactions.

Baselines: We compare performance of DECEnt and DECEnt+ against natural and state-of-the-art baselines in all of our applications. The first group of baselines include popular static network embeddings Node2Vec [8] and DeepWalk [9] and dynamic network embedding CTDNE [10]. After extensive literature review [11], [12], we find that predictive modelling tasks in healthcare analytics consist of an off-the-shelf classifier and a set of handcrafted feature. To this end, we design Domain baselines by augmenting classifiers with feature selection targeted for each predictive task. We also compare against time-nested deep recurrent neural models LSTM and RNN. Our final baseline is the state-of-the-art co-evolutionary neural network JODIE [4].

A. Application 1: MICU Transfer Prediction

The first application we consider seeks to forecast whether a patient is at risk of transfer to a Medical Intensive Care Unit (MICU). The MICU provides care for patients at a critical stage. A patient is only transferred to the MICU when there is a necessity for constant monitoring and intensive care. Therefore, an early indication of the high risk of transfer to a MICU can guide healthcare professionals to increase patient care. On the other hand, MICU beds are a scarce resource, and hence, an early indication of potential MICU transfer helps hospital officials allocate resources better.

Here we pose MICU transfer prediction as a binary classification problem. The input to a classifier is the embedding generated by DECEnt at time t. The output is a label representing whether a patient will be transferred to a MICU at time $t+1$. Here we only consider the cases where a Non-MICU to MICU transfer occurs at least three days from admission. We construct the positive instances (+) based on the actual MICU transfer events. If there is more than one transfer for a patient, we only consider the latest event. We randomly sample one day of their embedding for patients with no such event and use it as negative instances (-). Note that the MICU transfers are rare events. In order to remain faithful to the problem, we ensure that there is a significant class imbalance of greater than 100:1.

We train three independent off-the-shelf classifiers, logistic regression (LR), random forest (RF), and multi-layer perceptrons (MLP), to predict MICU transfers. We perform 5-fold cross-validation. Due to the extreme class imbalance, we randomly undersample the majority class training data to match the size of the minority class. However, we retain the class imbalance in the test set. We use the area under

[2] https://github.com/HankyuJang/DECEnt-dynamic-healthcare-embeddings

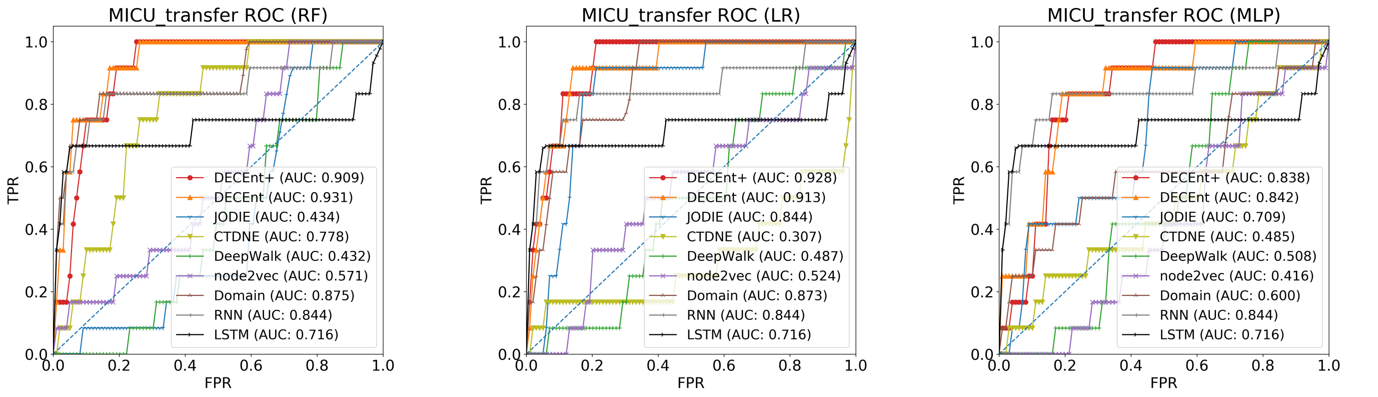

Fig. 2: ROC curves for DECEnt (in orange), DECEnt+ (in red) and the baselines for MICU transfer prediction task. Each figure corresponds to a different classifier: random forest (left), logistic regression (middle) and multi-layer preceptron (right). Both variants of DECEnt outperform all other methods consistently.

the ROC curve, AUC, as an evaluation metric robust under class imbalance. Finally, we report the average AUC over 30 repetitions and the standard deviation. We repeat the entire process with the baseline methods. Results are presented in Fig 2.

As seen in the figure, DECEnt outperforms all the baselines regardless of the underlying classifier. The gain of DECEnt over the most competitive baseline Domain is 6.4%. A significant advantage DECEnt has over the baselines, is that it is able to distinguish entity types from each other and it is specifically designed to model the heterogeneous dynamic interactions. The results show that DECEnt indeed learns embeddings that are useful for predictive modelling. This highlights the importance of principally encoding the heterogeneous nature of interactions that occur in a healthcare setting.

Another interesting observation we make is that DECEnt outperforms DECEnt+ in two out of three classifiers implying that restrictively enforcing the embeddings to respect the domain specific distances may lead to poorer performance in some cases while being useful in other. Impressively, even in the classifier where DECEnt performs the poorest, it still outperforms the most competitive baseline.

B. Application 2: CDI Prediction

CDI spreads in a healthcare setting, infecting patients who are already in a weakened state. Acquiring CDI increases a patient's mortality risk and prolongs hospital stay. Hence, early identification of patients at risk of CDI gives healthcare providers valuable lead time to combat the infection. It also helps prevent infection spread as practitioners can employ contact precaution and additional sanitary measures with special attention to the area around a patient at risk. Note that according to current practice, a patient is tested for CDI after three days of symptoms [13]. This means that pathogen may have spread even before positive test results.

Here we pose CDI prediction as a classification problem. The goal here is to accurately predict a label indicating if a patient would get infected within the next three days, given the embeddings learned by DECEnt and baselines. For the patients who get CDI, we use their embeddings three days before the positive report as positive class instances. We randomly sample one day of their visit for each of the rest of the patients and use it as negative class instances. We have a class imbalance of 89:1 as CDI cases are rare. Our experimental setup is the same as Application 1. We present our results in Table I.

TABLE I: Average AUC and the corresponding s.t.dev. for various approaches for the CDI prediction task. DECEnt and DECEnt+ outperform all the baselines.

Method	AUC		
RNN	0.56 (0.119)		
LSTM	0.585 (0.103)		
-	LR	RF	MLP
Domain	0.655 (0.123)	0.709 (0.104)	0.582 (0.137)
DeepWalk	0.494 (0.087)	0.487 (0.093)	0.492 (0.103)
Node2Vec	0.453 (0.098)	0.43 (0.106)	0.478 (0.1)
CTDNE	0.463 (0.101)	0.528 (0.079)	0.483 (0.116)
JODIE	0.552 (0.192)	0.377 (0.177)	0.469 (0.176)
DECEnt	0.732 (0.069)	0.711 (0.08)	0.668 (0.082)
DECEnt +	**0.736 (0.064)**	0.717 (0.078)	0.664 (0.091)

[a] The value in bold denotes best performance

As observed in the table DECEnt and DECEnt+ outperform all the baselines with the gain over the best performing baseline Domain of 3.81%. As in the previous applications static embedding approaches perform the worst, followed by the dynamic embedding baseline CTDNE, and finally JODIE.

C. Application 3: Mortality and Case Severity Risk Prediction

The Agency for Health Research and Quality (AHRQ) is a federal agency that performs mortality and case severity analysis on inpatient visits across hospitals in the US. Hospitals submit records on each patient visit to the agency. The agency then reports the severity and mortality along with the "expected" severity and mortality risk for each patient. The agency's report acts as a quality control metric for hospitals via retrospective analysis. Predicting case severity and mortality while the patient is in the hospital has many applications, ranging from personalized patient care to resource allocation.

TABLE II: Average F1 Macro and the corresponding s.t.dev. for various methods for mortality and severity prediction. DECEnt and DECEnt+ outperform all the baseline methods.

Method	**Mortality**	**Severity**
RNN	0.276 (0.039)	0.31 (0.032)
LSTM	0.289 (0.033)	0.308 (0.026)
Domain	0.22 (0.017)	0.258 (0.007)
DeepWalk	0.172 (0.034)	0.192 (0.019)
Node2Vec	0.172 (0.02)	0.196 (0.009)
CTDNE	0.184 (0.019)	0.199 (0.007)
JODIE	0.143 (0.039)	0.193 (0.014)
DECEnt	0.421 (0.027)	0.34 (0.014)
DECEnt+	**0.428 (0.022)**	**0.349 (0.015)**

[a] The value in bold denotes best performance

AHRQ classifies each case into four categories, namely *Minor*, *Moderate*, *Major*, and *Extreme* for both mortality and case severity. Hence, we model both these tasks as multi-label classification problem. The input to a classifier is the embedding generated by DECEnt by the time of discharge. We report the results on logistic regression classifier on F1-Macro scores in Table II. The results show that DECEnt and DECEnt+ consistently outperform all the baselines for both mortality and case severity prediction, with a gain of up to 48.1% over LSTM and 12.58% over RNN, respectively, compared to the best performing baselines for each prediction task. This result reinforces our conclusion from previous applications that DECEnt is superior to state-of-the-art baselines in predictive modeling tasks in the healthcare setting.

D. Evaluating the Embeddings

Results described earlier in the paper show that embeddings obtained via DECEnt are excellent for a variety of predictive tasks. In this subsection, we present findings from a qualitative exploration of the embeddings. This was done in collaboration with a medical doctor at the hospital who specializes in infectious diseases.

We start our exploration by computing *dispersions* for subsets of healthcare entities. Suppose that the set of doctors $\mathcal{D}$ is partitioned into subsets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_d$. For any $1 \leq i < j \leq d$ and time t we define the pairwise *doctor dispersion* between $\mathcal{D}_i$ and $\mathcal{D}_j$ as

$$disp_{\mathcal{D},t}(i,j) = \frac{\sum_{d \in \mathcal{D}_i, d' \in \mathcal{D}_j} ||\hat{\mathrm{e}}_{d,t} - \hat{\mathrm{e}}_{d',t}||_2}{|\mathcal{D}_i| \cdot |\mathcal{D}_j|} \tag{9}$$

Recall that $\hat{\mathrm{e}}_{d,t}$ denotes the embedding of a doctor d at time t. If the partition $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_d$ represents a grouping of doctors by specialty (e.g., pediatrics, cardiology, anesthesia, neurology, etc.) then the pairwise doctor dispersions provide a measure of the average distance between doctors from different specialties in the computed embeddings.

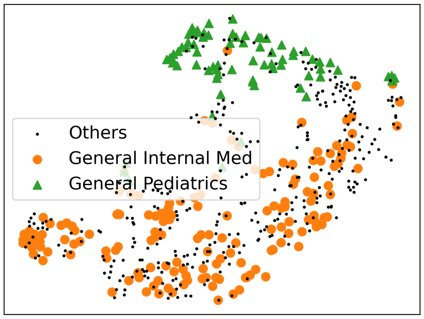

Fig. 3: Doctor embeddings learned by DECEnt. On average pediatricians are further away from the rest of the doctors compared to doctors in Internal Medicine.

One of the findings from the pairwise dispersion also confirms what is generally known about hospital operations. For example, doctors in general internal medicine and anesthesia interact with a wide variety of patients, and thus doctors in these specialties are close to those in other specialties. On the other hand, patients of doctors in Family Practice and Otolaryngology (ears, nose, and throat) tend not to require referral to other specialties. As a result, doctors in these specialties are relatively far away from doctors in other specialties. In Figure 3, we plot the doctor embeddings learned by DECEnt projecting them to 2-d using t-SNE [14].

V. Related Work

Network Embeddings for Healthcare Analytics. Network embedding [8], [9] has gained much research interest lately. Recently, several approaches to learn embeddings of nodes in a dynamic network have been proposed. These approaches aim to capture both structural similarity and temporal evolution [15].

Similar approaches have been explored for medical data. A class of approaches [1] preserve the similarity of the medical codes in consecutive hospital visits by the same patient. eNRBM [16] uses restricted Boltzmann Machines, and [17] uses convolutional neural network to represent abstract medical concepts.

Coevolving Networks. User-item interaction based embedding has gained a lot of attention recently. It has become a powerful tool for representing the evolution of users and items based on dynamic interactions [18]. DeepCoevolve [3] uses RNN to learn the user and item embedding through the complex mutual influence in any interaction over the time. JODIE [4] extends [3] by adding a new projection operator that can predict user-item interaction at any future point of the time.

Healthcare Analytics. An area closely related to the current paper is that of Healthcare Analytics. Li et al. explored applicability of machine learning for CDI prediction using manual feature engineering [11]. Several other approaches have been proposed for case detection tasks [19]. A separate line of work focuses on mortality prediction [20]. Other loosely related works include outbreak detection [21], missing infection inference [22], and architectural analysis [23].

Autoencoders for Representation Learning. SDNE [24] uses autoencoders to preserve second-order proximity of the network. NetRAs [25] uses graph encoder-decoder framework using LSTM networks and rooted random walks. Deep Patient [26] uses stacked autoencoders to encode patients.

VI. Discussion and Limitations

This paper proposes DECEnt, a novel approach for learning heterogeneous dynamic embeddings of patients, doctors, rooms, and medications from diverse hospital operation data streams. These embeddings capture similarities among entities based on static attributes and dynamic interactions. Consequently, these embeddings can serve as input to a variety of prediction tasks to improve clinical decision-making and patient care. Our results show that on a variety of prediction tasks DECEnt substantially outperforms baselines and produces embeddings meaningful to clinical experts.

As we see it, our work has limitations. All our patient-physician interactions are associated with procedures. While such procedure-related interactions are essential, we need to consider a more extensive, richer set of patient-physician interactions. Such interactions can be extracted from the clinical notes from hospitals. Each clinical note provides much context for each interaction, and one possible way to label the interaction with this context is to compute embeddings of clinical notes and attach these as weights or features to the interactions. In the longer term, we are interested in deploying DECEnt on top of the existing electronic medical record system at the hospital to provide clinical support. To reach this goal, we need a way to make the embeddings learned by DECEnt interpretable to healthcare professionals. Specifically, we need to identify and explain the factors that cause two healthcare entities to be close to (or far from) each other. We propose to do so by introducing prototypes [27] and adding an explainability module for feature ablation.

Besides the future work mentioned above, we see a direction to expand our work. We can apply DECEnt to other prediction tasks in the healthcare setting. For example, predicting the risk of readmission (see [12]) prior to discharge and predicting the length of stay in the hospital early during a patient visit (see [28]) are both tasks that can enable additional clinical resources for high-risk patients.

Acknowledgements: The authors acknowledge funding from the CDC MInD Healthcare Network grant U01CK000594 and NSF grant 1955939. The authors thank feedback from other University of Iowa CompEpi group members.

References

[1] E. Choi, C. Xiao, W. Stewart, and J. Sun, "Mime: Multilevel medical embedding of electronic health records for predictive healthcare," in *NIPS*, 2018.

[2] C. for Disease Control and Prevention, *Clostridioides difficile Infection*, Reviewed Nov 13, 2019 (accessed June 10, 2020).

[3] H. Dai, Y. Wang, R. Trivedi, and L. Song, "Deep coevolutionary network: Embedding user and item features for recommendation," in *DLRS*, 2017.

[4] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *ACM SIGKDD*, 2019.

[5] L. Zhou, Y. Yang, X. Ren, F. Wu, and Y. Zhuang, "Dynamic network embedding by modeling triadic closure process," in *AAAI*, 2018.

[6] J. Bourgain, "The metrical interpretation of superreflexivity in banach spaces," *Israel Journal of Mathematics*, vol. 56, pp. 222–230, 1986.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[8] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *ACM SIGKDD*, 2016.

[9] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *ACM SIGKDD*, 2014.

[10] G. H. Nguyen, J. B. Lee, R. A. Rossi, N. K. Ahmed, E. Koh, and S. Kim, "Continuous-time dynamic network embeddings," in *IW3C2 WWW*, 2018.

[11] B. Y. Li, J. Oh, V. B. Young, K. Rao, and J. Wiens, "Using machine learning and the electronic health record to predict complicated clostridium difficile infection," in *OFID*, 2019.

[12] X. Min, B. Yu, and F. Wang, "Predictive modeling of the hospital readmission risk from patients' claims data using machine learning: a case study on copd," *Sci. Rep.*, 2019.

[13] M. Monsalve, S. Pemmaraju, S. Johnson, and P. M. Polgreen, "Improving risk prediction of clostridium difficile infection using temporal event-pairs," in *IEEE ICHI*, 2015.

[14] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *JMLR*, 2008.

[15] M. Torricelli, M. Karsai, and L. Gauvin, "weg2vec: Event embedding for temporal networks," *Sci. Rep.*, 2020.

[16] T. Tran, T. D. Nguyen, D. Phung, and S. Venkatesh, "Learning vector representation of medical objects via EMR-driven nonnegative restricted Boltzmann machines (eNRBM)," *J Biomed Inform*, 2015.

[17] Z. Zhu, C. Yin, B. Qian, Y. Cheng, J. Wei, and F. Wang, "Measuring patient similarities via a deep architecture with medical concept embedding," in *IEEE ICDM*, 2016.

[18] M. Farajtabar, Y. Wang, M. Gomez Rodriguez, S. Li, H. Zha, and L. Song, "Coevolve: A joint point process model for information diffusion and network co-evolution," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[19] M. Makar, J. Guttag, and J. Wiens, "Learning the probability of activation in the presence of latent spreaders," in *AAAI*, 2018.

[20] E. Sherman, H. Gurm, U. Balis, S. Owens, and J. Wiens, "Leveraging clinical time-series data for prediction: a cautionary tale," in *AMIA*, 2017.

[21] B. Adhikari, B. Lewis, A. Vullikanti, J. M. Jiménez, and B. A. Prakash, "Fast and near-optimal monitoring for healthcare acquired infection outbreaks," *PLoS CompBio*, 2019.

[22] S. Sundareisan, J. Vreeken, and B. A. Prakash, "Hidden hazards: Finding missing nodes in large graph epidemics," in *SDM*, 2015.

[23] H. Jang, S. Justice, P. M. Polgreen, A. M. Segre, D. K. Sewell, and S. V. Pemmaraju, "Evaluating architectural changes to alter pathogen dynamics in a dialysis unit," in *IEEE/ACM ASONAM*, 2019.

[24] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *ACM SIGKDD*, 2016.

[25] W. Yu, C. Zheng, W. Cheng, C. C. Aggarwal, D. Song, B. Zong, H. Chen, and W. Wang, "Learning deep network representations with adversarially regularized autoencoders," in *ACM SIGKDD*, 2018.

[26] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep patient: An unsupervised representation to predict the future of patients from the electronic health records," *Sci. Rep.*, 2016.

[27] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *MIPRO*. IEEE, 2018.

[28] L. Turgeman, J. H. May, and R. Sciulli, "Insights from a machine learning model for predicting the hospital length of stay (los) at the time of admission," *Expert Syst. Appl.*, 2017.