Generative Adversarial Networks Enhanced Pre-training for Insufficient Electronic Health Records Modeling

Houxing Ren
School of Computer Science and Engineering, Beihang University, Beijing, China
Peng Cheng Laboratory, Shenzhen, China
renhouxing@buaa.edu.cn

Jingyuan Wang*
School of Computer Science and Engineering, Beihang University, Beijing, China
Peng Cheng Laboratory, Shenzhen, China
jywang@buaa.edu.cn

Wayne Xin Zhao
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China
batmanfly@gmail.com

ABSTRACT

In recent years, automatic computational systems based on deep learning are widely used in medical fields, such as automatic diagnosing and disease prediction. Most of these systems are designed for data sufficient scenarios. However, due to the disease rarity or privacy, the medical data are always insufficient. When applying these data-hungry deep learning models with insufficient data, it is likely to lead to issues of over-fitting and cause serious performance problems. Many data augmentation methods have been proposed to solve the data insufficiency problem, such as using GAN (Generative Adversarial Networks) to generate training data. However, the augmented data usually contains lots of noise. Directly using them to train sensitive medical models is very difficult to achieve satisfactory results.

To overcome this problem, we propose a novel deep model learning method for insufficient EHR (Electronic Health Record) data modeling, namely GRACE, which stands GeneRative Adversarial networks enhanCed prE-training. In the method, we propose an item-relation-aware GAN to capture changing trends and correlations among data for generating high-quality EHR records. Furthermore, we design a pre-training mechanism consisting of a masked records prediction task and a real-fake contrastive learning task to learn representations for EHR data using both generated and real data. After the pre-training, only the representations of real data is used to train the final prediction model. In this way, we can fully exploit useful information in generated data through pre-training, and also avoid the problems caused by directly using noisy generated data to train the final prediction model. The effectiveness of the proposed method is evaluated using extensive experiments on three healthcare-related real-world datasets. We also deploy our method in a maternal and child health care hospital for the online test. Both offline and online experimental results demonstrate the effectiveness of the proposed method. We believe doctors and patients can benefit from our effective learning method in various healthcare-related applications.

CCS CONCEPTS

• **Applied computing → Health informatics**.

KEYWORDS

Healthcare Informatics, Representation Learning, Pre-training

ACM Reference Format:
Houxing Ren, Jingyuan Wang, and Wayne Xin Zhao. 2022. Generative Adversarial Networks Enhanced Pre-training for Insufficient Electronic Health Records Modeling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539020

1 INTRODUCTION

Nowadays, *Electronic Health Records (EHRs)* are widely available from Hospital Information Systems (HIS). Many computational systems have been developed to leverage EHR data for important medical applications, such as automatic disease prediction. As a typical approach, these EHR-based computational systems take as input the historical EHR data of a patient and then predict the target output, *e.g.,* probability of suffering some disease [7, 22] or physical features in a near future [25], which can help doctors to identify the potential health risk at the early stage and provide better treatments.

As EHR data are usually represented in a sequence form, most of disease prediction systems are developed based on sequence neural networks, such as recurrent neural networks (RNNs) [3, 7, 22] and Transformers [21, 28]. In the literature, existing methods mainly focus on devising suitable architecture of sequence neural networks to handle some unique characteristics of EHR data, *e.g.,* irregular time intervals [3, 21] and incompleteness [28, 33]. Most of these works rely on relatively sufficient training datasets to learn their model parameters. However, EHR data is very different from ordinary sequence data. Due to the disease rarity or privacy, the number of available samples is insufficient [4, 40]. When applying these data-hungry models, *i.e.,* deep learning models, in an insufficient data setting, it is likely to lead to the over-fitting problem. To overcome the insufficient data problem, some methods propose to employ meta-learning [40] or knowledge graph [6, 35]. While,

[*] Corresponding author.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539020

these methods still require auxiliary information to resist the data sparsity, and such auxiliary information is not always accessible.

As a promising approach, generative adversarial networks (GANs) [12] have been demonstrated as an effective data augmentation approach to solving the insufficient data problem for deep learning model training. For example, RTSGAN [25] proposed to generate time series data via GANs and further utilize the generated data to train the sequence prediction models. However, it is still with some challenges when we directly use GAN-based data augmentation methods to solve the data insufficiency problem in EHR applications, *i.e.,* directly using generated EHR data to train EHR prediction models. Firstly, EHR data is more complicated in essence than other types of sequence data, *e.g.,* containing temporal dependency and correlation among physiological indices. It is difficult to synthesize reasonable data with insufficient real data using a commonly used GAN model. Secondly, the real EHR data is very delicate, *i.e.,* a small change in a record can reflect a large change in a patient's physical condition. Augmented data generated by GAN models always contain some noises, which may mislead diagnostic models if we directly use the generated EHR data to train prediction models.

Inspired by recent advances of pre-training methods [26, 28], we propose to leverage the pre-training technique to better use GAN-generated data in EHR-based prediction models. Specifically, we propose a novel method which combines GAN and pre-training to model insufficient EHR data, namely GRACE, which stands GeneRative Adversarial networks enhanCed prE-training. Our approach contains two major components: 1) A item-relation-aware GAN architecture which can model the relations among data records for generating reasonable high-quality EHR data; and 2) A pre-training based representation learning mechanism consists of a masked records prediction task and a real-fake contrastive learning task. In the representation learning mechanism, the augmented data generated by the item-relation-aware GAN are only used to pre-train the representations of EHR, while only the representations of real data are used to train the final prediction models. In this way, we can effectively solve the issues of over-fitting to noise in the generated EHR data for the final prediction model.

The contribution of the paper can be summarized as follows:

- To the best of our knowledge, it is the first work that combines GAN and pre-training in representation learning of EHR data. The proposed method can fully exploit useful information in generated data through pre-training, and avoid the problems caused by directly using noisy generated data to train the final prediction model. We believe this method can be applied in applications which suffer from the insufficient data problem.
- We designed a novel item-relation-aware GAN architecture that can generate reasonable EHR data. This architecture can be applied to other healthcare-related data generation applications, such as medical education.
- We designed elaborate pre-training tasks, especially the real-fake contrastive learning task, to exploit self-supervised information in generated EHR data. This method is instructive for self-supervising applications on EHR data.
- We construct extensive experiments on three healthcare datasets and experimental results show the effectiveness of the proposed learning paradigm. Moreover, we also deploy our method in the HIS of a maternal and child health care hospital for gestational diabetes prediction. Both offline and online experiments proved that GRACE can greatly improve the utilization efficiency of limited EHR data. We believe doctors and patients can benefit from our method in various EHR-based healthcare applications.

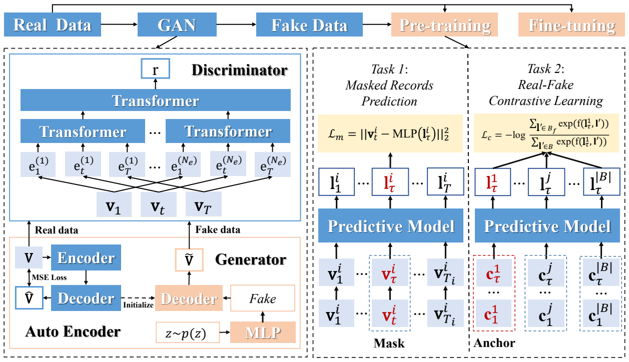

Real Data
GAN
Fake Data
Pre-training
Fine-tuning
r
Discriminator
Transformer
Transformer
Transformer
$e_1^{(1)}$ $e_t^{(1)}$ $e_T^{(1)}$ $e_1^{(N_e)}$ $e_t^{(N_e)}$ $e_T^{(N_e)}$
$\mathbf{v}_1$ $\mathbf{v}_t$ $\mathbf{v}_T$
Real data
Fake data
V
Encoder
MSE Loss
$\hat{V}$
Decoder
Initialize
$\tilde{V}$
Generator
Decoder
Fake
Auto Encoder
$z \sim p(z)$
MLP
Task 1:
Masked Records
Prediction
$\mathcal{L}_m = ||\mathbf{v}_t^i - \text{MLP}(\mathbf{l}_\tau^i)||_2^2$
$\mathbf{l}_1^i$ $\mathbf{l}_\tau^i$ $\mathbf{l}_T^i$
Predictive Model
$\mathbf{v}_1^i$ $\mathbf{v}_\tau^i$ $\mathbf{v}_{T_i}^i$
$\mathbf{v}_1^i$ $\mathbf{v}_t^i$ $\mathbf{v}_{T_i}^i$
Mask
Task 2:
Real-Fake
Contrastive Learning
$\mathcal{L}_c = -\log \frac{\sum_{\mathbf{l}' \in B_f} \exp(f(\mathbf{l}_\tau^1, \mathbf{l}'))}{\sum_{\mathbf{l}' \in B} \exp(f(\mathbf{l}_\tau^1, \mathbf{l}'))}$
$\mathbf{l}_\tau^1$ $\mathbf{l}_\tau^j$ $\mathbf{l}_\tau^{|B|}$
Predictive Model
$\mathbf{c}_\tau^1$ $\mathbf{c}_\tau^j$ $\mathbf{c}_\tau^{|B|}$
$\mathbf{c}_1^1$ $\mathbf{c}_1^j$ $\mathbf{c}_1^{|B|}$
Anchor

Figure 1: Overview of the GRACE method.

2 OVERVIEW

In this section, we formally define the dataset, the problem and introduce our framework.

2.1 EHR Data and Problem Definition

EHR Examination Records. During disease progression, patients need to visit the hospital multiple times for examination. For each visit, a patient underwent a number of physical examinations. We denote the *examination record* for the t-th visit as $\boldsymbol{v}_t = \left(e_t^{(1)}, e_t^{(2)}, \ldots, e_t^{(N_e)}\right)$, where the item $e_t^{(i)}$ is the i-th *examination item* of the examination record $\boldsymbol{v}_t$ (*i.e.,* diastolic pressure), and N_e denotes the number of examination item of the each visit.

Visit Sequences in EHR. We represent the examination records for a patient during his disease progression as a visit sequence of chronologically ordered events. For the i-th patient, his visit sequence is denoted as $\boldsymbol{V}^{(i)}$, where $\boldsymbol{V}^{(i)} = \left(\boldsymbol{v}_1^{(i)}, \boldsymbol{v}_2^{(i)}, \ldots, \boldsymbol{v}_{T_i}^{(i)}\right)$ is the visit sequence of the i-th patient.

Problem Definition. The tasks in this paper aim to predict the probability of a patient suffering from a certain disease. Given the visit sequence $\boldsymbol{V}^{(i)}$ of a patients, we define the model as a prediction function that is with $\boldsymbol{V}_t^{(i)}$ as inputs and gives the output as

$$\hat{y}^{(i)} = f\left(\boldsymbol{V}^{(i)}\right), \tag{1}$$

where $y_t^{(i)} \in \{0, 1\}$ is a binary diagnosis label indicating whether the patient will suffer from a disease.

2.2 Framework

In this part, we present our framework. Our core idea is to use GAN (Generative Adversarial Network) to generate fake data to alleviate the insufficiency of EHR data. Similar ideas have been used to solve the data insufficiency problem of time series classification [25, 38], which directly uses fake data generated by GAN to train classification models. However, due to the characteristics of EHR data, we can't follow the same way to solve the insufficiency problem in automated disease diagnosis. First, EHR contains rich semantic information, it is challenging work for a general GAN model to generate semantically sound fake data, so we need to design specialized mechanisms to ensure that the generated data has satisfactory semantics. Second, due to insufficient data and containing rich semantics, no matter what GAN model we design, it is still very hard to generate perfect fake EHR data. The fake EHR generated by GAN always contains many noises, which is very easy to mislead models if we directly use them to train our model.

As shown in Figure 1, to overcome above problems, the proposed GRACE model adopts an item-relation-aware GAN to generate the highest quality fake data possible, and then indirectly exploits information of the fake data through a pre-training approach. Specifically, the model uses the fake data to pre-train a representation learning model for EHR representation generating, and then, uses the EHR representations of real data to fine-tune the final diagnostic models. In this way, the proposed GRACE model avoids the defect of directly using fake data to train the diagnostic model, and meanwhile, fully exploits the valuable information in the fake data through pre-training the representation learning model, which solves the data insufficiency problem of EHR.

3 METHODS

In this section, we present details of the proposed GRACE model, which includes an item-relation-aware GAN component and a representation pre-training component. The overall architecture for the proposed GRACE is illustrated in Figure 1.

3.1 Item-relation-aware GAN

As shown in Figure 1, the proposed item-relation-aware GAN consists of a generator and a discriminator.

3.1.1 *EHR Generator.*

The generator aims to generate fake visit sequences from random vectors. Since the target are visit sequences, our model first generates a hidden state vector $\tilde{\boldsymbol{s}}_\tau$ from a random vector $\boldsymbol{z}$ as

$$\tilde{\boldsymbol{s}}_\tau = \boldsymbol{W}_z \times \boldsymbol{z} + \boldsymbol{b}_z,\ \boldsymbol{z} \sim p(\boldsymbol{z}), \tag{2}$$

where $\boldsymbol{W}_z \in \mathbb{R}^{h \times h}$ and $\boldsymbol{b}_z \in \mathbb{R}^h$ are learnable parameters, $p(\boldsymbol{z})$ denotes a random distribution (*i.e.,* Gaussian distribution). Then, the GAN model uses a Transformer decoder to decode the hidden state vector as a visit sequences, *i.e.,*

$$(\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \ldots, \tilde{\boldsymbol{v}_T}) = \text{TransformerDecoder}\left(\tilde{\boldsymbol{s}}_\tau\right). \tag{3}$$

When generating fake visit sequences, we cannot know the target length of the generated sequence. We use a simple method to define the stop signal. We find that when the length of generated sequence saturates, the timestamp tends to be convergence. Then, we stop the generating process when the time interval between the two successive visits is less than the threshold, which is defined as the smallest time interval in real visit sequences.

In Eq. (3), the Transformer decoder is responsible for converting a dense hidden state to an EHR visit sequence. Since the EHR data is very complicated and contains rich semantic information, if we direct use random parameters to initialize the Transformer decoder, it is very hard to accomplish the sequence generation task. To overcome this issue, we design an AutoEncoder initialization mechanism to set the parameter of the Transformer decoder in Eq. (3).

The AutoEncoder contains an encoder and a decoder. The encoder is responsible for encoding a real visit sequence data as a dense vector. Given a real visit sequence $\boldsymbol{V}^{(i)}$, we add a virtual visit $\boldsymbol{v}_\tau$ to the sequence, *i.e.,* letting $\boldsymbol{V}^{(i)} = \left(\boldsymbol{v}_1^{(i)}, \ldots, \boldsymbol{v}_{T_i}^{(i)}, \boldsymbol{v}_\tau^{(i)}\right)$. For each visit vector $\boldsymbol{v}_t$ in $\boldsymbol{V}^{(i)}$, the encoder convert it as a dense vector $\boldsymbol{x}_t \in \mathbb{R}^h$ using a fully connected layer as

$$\boldsymbol{x}_t = \boldsymbol{W}_v \times \boldsymbol{v}_t + \boldsymbol{b}_v, \tag{4}$$

where $\boldsymbol{W}_v \in \mathbb{R}^{h \times N_e}$ and $\boldsymbol{b}_v \in \mathbb{R}^h$ are learnable parameters and h are a hyper-parameter. Next, the encoder adopts a Transformer to convert the dense vector sequence, *i.e.,* $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T, \boldsymbol{x}_\tau)$, as a hidden state vector sequence:

$$(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T, \boldsymbol{s}_\tau) = \text{TransformerEncoder}\left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T, \boldsymbol{x}_\tau\right). \tag{5}$$

Here, the vector $\boldsymbol{s}_\tau$ is a representation of the whole visit sequence $\boldsymbol{V}^{(i)}$.

In the decoder, we adopt a Transformer to decode the representation vector $\boldsymbol{V}^{(i)}$ as the other dense vector sequence, *i.e.,*

$$(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t) = \text{TransformerDecoder}(\boldsymbol{s}_\tau). \tag{6}$$

Next, the decoder a multilayer perceptron (MLP) to convert each dense vector $\boldsymbol{o}_t$ as a corresponding reconstructed EHR visit, *i.e.,*

$$\hat{\boldsymbol{v}}_t = \text{MLP}(\boldsymbol{o}_t). \tag{7}$$

To train the AutoEncoder, we employ a mean square error (MSE) an objective function as

$$\mathcal{L}_a = \frac{1}{N} \sum_{i=0}^{N} \sum_{t<T_i} \left\| \hat{\boldsymbol{v}}_t^{(i)} - \boldsymbol{v}_t^{(i)} \right\|_2^2, \tag{8}$$

where N denotes the number of samples and T_i denotes the length of the i-th visit sequence. When training the AutoEncoder, we use the teacher-forcing policy [30], *i.e.,* always using ground truth data $\boldsymbol{v}_t$ as the next-step input.

Once the AutoEncoder has been trained, we initialize the TansformerDecoder in Eq. (3) with the TansformerDecoder of the AutoEncoder (in Eq. (6)). In this way, the data characteristics of the real visit sequence are memorized by the TansformerDecoder, and therefore it can be used to generate higher quality fake data.

3.1.2 *Item-relation-aware Discriminator.*

The discriminator aims to distinguish the real data and the generated fake data. A high-performance discriminator can push the generator to produce better fake data. To improve the performance of our discriminator, we design a hierarchical item-relation aware mechanism to capture characteristics of EHR visit sequences. The item-relation aware mechanism contains an item-trend-aware transformer to model the changing trend for each item, and an item-correlation-aware transformer to model correlations among items. The two transformers hierarchically stack as a whole.

Item-trend-aware Transformer. During the disease progression, each examination item has a changing trend. For example, blood pressure is growing gradually during the full disease progression for patients that suffer from hypertension. We hope the discriminator can model the changing trend to encourage the generator to produce data with the changing trends. Based on this idea, we employ a Transformer to model each item along the time axis. Specifically, for the j-th item at the t-th visit, *i.e.,* $e_t^{(j)}$, we covert it as a representation vector through

$$\boldsymbol{p}_t^{(j)} = \boldsymbol{W}_e^{(j)} \times e_t^{(j)} + \boldsymbol{b}_e^{(j)}, \tag{9}$$

where $\boldsymbol{W}_e^{(j)} \in \mathbb{R}^{h\times 1}$ and $\boldsymbol{b}_e^{(j)} \in \mathbb{R}^h$ are learnable parameters. Next, a Transformer encodes the sequence of $\boldsymbol{p}_t^{(j)}$ as

$$\begin{aligned} \left(\boldsymbol{q}_0^{(j)}, \dots, \boldsymbol{q}_T^{(j)}\right) &= \text{Transformer}\left(\boldsymbol{p}_0^{(j)}, \dots, \boldsymbol{p}_T^{(j)}\right), \\ \boldsymbol{q}^{(j)} &= \text{Average Pooling}\left(\boldsymbol{q}_0^{(j)}, \dots, \boldsymbol{q}_T^{(j)}\right), \end{aligned} \tag{10}$$

We use $\boldsymbol{q}^{(j)}$ to represent the changing trend of the j-th item.

Item-correlation-aware Transformer. During the disease progression, different EHR items have some correlations with each other. For example, when diastolic pressure grows, systolic pressure usually grows too. We also hope the discriminator can model the correlations among items to further improve the quality of generated fake data. Therefore, we employ another Transformer to model correlation among different items. Specifically, given the representations $\boldsymbol{q}^{(j)}$ of each item, we model them as

$$\begin{aligned} \left(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(N_e)}\right) &= \text{Transformer}\left(\boldsymbol{q}^{(1)}, \dots, \boldsymbol{q}^{(N_e)}\right), \\ \boldsymbol{u} &= \text{Average Pooling}\left(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(N_e)}\right), \end{aligned} \tag{11}$$

where N_e denotes the number of examination items.

Then we use the final representation $\boldsymbol{u}$ of a visit sequence to classify whether the input data is real as

$$r = \boldsymbol{W}_u \times \boldsymbol{u} + \boldsymbol{b}_u, \tag{12}$$

where $\boldsymbol{W}_u \in \mathbb{R}^{1\times h}$ and $\boldsymbol{b}_u \in \mathbb{R}^1$ are learnable parameters, r denotes the score of the input visit sequence. The larger the value, the higher the probability that inputs are real data.

3.1.3 *Training.*

We employ the WGAN [14] method to train the generator and discriminator. The generator in WGAN aims to minimize the 1-Wasserstein distance between real data distribution and fake data distribution, which is easier to convergence in training. The optimization objective is defined as

$$\min_{Gen} \max_{Dis} \mathbb{E}_{\boldsymbol{V}\sim p(\boldsymbol{V})}\left[Dis(\boldsymbol{V})\right] - \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[Dis\left(Gen(\boldsymbol{z})\right)\right], \tag{13}$$

where "Dis" denotes the 1-Lipschitz discriminator and "Gen" denotes the Generator.

In the whole item-relation-aware GAN training, we first train the AutoEncoder with Eq. (8) until it convergence. Then, we jointly train the decoder, the generator, and the discriminator with Eq. (13).

3.2 Pre-training for Representation Learning

In this section, we propose a pre-training mechanism to learn representations of EHR visit sequences using both real data and fake sequences generated by the item-relation-aware GAN.

Deep Model for EHR data. The function of the model is to generate representation vectors of EHR visit sequences. The input of the representation learning model is a visit sequence of a patient, and the output is another representation vector sequence, *i.e.,*

$$(\boldsymbol{l}_1, \boldsymbol{l}_2, \dots, \boldsymbol{l}_T, \boldsymbol{l}_\tau) = \text{Model}\left(\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_T, \boldsymbol{v}_\tau\right), \tag{14}$$

where $(\boldsymbol{l}_1, \boldsymbol{l}_2, \dots, \boldsymbol{l}_T)$ is the representation vector sequence, and $\boldsymbol{l}_\tau$ is a vector that can represent the whole visit sequence. The $Model(\cdot)$ function denotes a sequence-to-sequence model. We use different models (*e.g.,* Transformer) in experiments to evaluate the proposed representation learning framework.

To train the representation learning model using both real and fake data, we design two pre-training tasks. One for exploiting inner sequence correlations using the masked records prediction, and the other for exploiting cross sequence correlation using the real-fake data contrastive learning.

Pre-training task #1: Masked Records Prediction. In natural language processing, the masked language model (MLM) [9] task achieved great success in pre-training, which can mine correlations between words effectively. In the EHR sequence, the correlation between visit records also has close correlations. To model correlation among examination records, we propose a masked records prediction pre-training task.

Given a visit sequence $\boldsymbol{V}$, we randomly mask some visits using a virtual visit $\boldsymbol{v}_\tau$ (in practice, we set a masked ration δ, *e.g.,* 0.15, *i.e.,* randomly select δ visits to mask), and use the corresponding representation vector to predict the masked visit records. We denote $\boldsymbol{v}'$ as a masked visit, and $\boldsymbol{l}'$ denotes the corresponding representation vector. A multilayer perceptron (MLP) is employed to predict the masked visit records as

$$\hat{\boldsymbol{v}}' = \text{MLP}\left(\boldsymbol{l}'\right). \tag{15}$$

We employ mean squared error (MSE) as the optimized objective of the masked records prediction task, *i.e.,*

$$\mathcal{L}_m = \frac{1}{|\boldsymbol{V}'|} \sum_{\boldsymbol{v}'\in\boldsymbol{V}'} \left(r_v' + \beta \times (1 - r_v')\right) \times ||\hat{\boldsymbol{v}}' - \boldsymbol{v}'||_2^2, \tag{16}$$

where $\boldsymbol{V}'$ is the set that consists of all masked visits, $|\boldsymbol{V}'|$ denotes the size of $\boldsymbol{V}'$, r_v' indicates whether $\boldsymbol{v}'$ is from real data ($r_v' = 1$ when $\boldsymbol{v}'$ is from real data, otherwise $r_v' = 0$), and β is a hyper-parameter. Here, we incorporate the weight β to adjust the weight of real and generated samples. The basic idea is that the confidence of the generated data is less than the real data, so $\beta < 1$.

Pre-training task #2: Real-Fake Contrastive Learning. Although the item-relation-aware GAN can effectively capture the characteristics of EHR data and generate reasonable data, there is also a gap between the fake data and the real data. We believe a representation that can distinguish real and fake samples is more effective for real data modeling. Based on this insight, we design a real-fake contrastive task to train the representation learning model.

Specifically, for a fake sample in a mini-batch, we use the representations of sequences from the fake data as positive samples and the rest representations as negative samples. We employ a contrastive loss function, called InfoNCE [31], to classify positive and negative samples. When we use the representation $\boldsymbol{l}^{(j)}$ of a sample, *i.e.,* the corresponding representation vector of the virtual visit $\boldsymbol{v}_\tau^{(j)}$, from the fake data as the anchor sample, the InfoNCE loss function is defined as

$$\mathcal{L}_r^{(j)} = -\log \frac{\sum_{\boldsymbol{l}'\in\mathcal{B}_{fake}} \exp\left(f\left(\boldsymbol{l}^{(j)}, \boldsymbol{l}'\right)\right)}{\sum_{\boldsymbol{l}'\in\mathcal{B}} \exp\left(f\left(\boldsymbol{l}^{(j)}, \boldsymbol{l}'\right)\right)}, \tag{17}$$

where j denotes the j-th sample in the mini-batch, $\mathcal{B}$ is the set that consists of the representations of samples in the mini-batch, $\mathcal{B}_{fake}$ is the set that consists of the representations of fake samples in the mini-batch, and $f(\cdot,\cdot)$ is a similarity function. We simply use dot-product to measure the similarity, *i.e.,* $f(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}\cdot\boldsymbol{y}$, which cares both angle and magnitude. Similarly, if we use representation of a sample from real data as the anchor sample, we just need to replace $\mathcal{B}_{fake}$ with $\mathcal{B}_{real}$ that is the set consisting of the representations of real samples in the mini-batch.

The loss of a mini-batch is the average of loss when we use all samples as the anchor sample, *i.e.,*

$$\mathcal{L}_r = \frac{1}{|\mathcal{B}|}\sum_{j=1}^{|\mathcal{B}|}\mathcal{L}_c^{(j)}, \tag{18}$$

where $|\mathcal{B}|$ denotes the size of the mini-batch.

Pre-training Loss. When we pre-train the whole representation model, we combine the pre-training loss functions with a balance hyper-parameter α as

$$\mathcal{L}_p = \alpha \times \mathcal{L}_m + (1-\alpha)\times\mathcal{L}_r, \tag{19}$$

where $\mathcal{L}_m$ (Eq. (16)) is the pre-training loss of the masked records prediction task and $\mathcal{L}_r$ (Eq. (18)) is the pre-training loss of the real-fake contrastive learning task.

3.3 Diagnosis Model and Discussion

When we got the pre-trained representation learning model, we apply its generated representations to various downstream diagnosis applications. For example, for a binary diagnosis classification task, *e.g.,* hypertension prediction, we input the representation vector generated by Eq. (14) to a fully connected layer with Sigmoid classifier as

$$\hat{y}_i = \text{Sigmoid}\left(\boldsymbol{W}_y \times \boldsymbol{l}_\tau^{(i)} + \boldsymbol{b}_y\right), \tag{20}$$

where $\boldsymbol{W}_y \in \mathbb{R}^{1\times h}$ and $\boldsymbol{b}_y \in \mathbb{R}^1$ are learnable parameters, $\boldsymbol{l}_\tau^{(i)}$ are the corresponding representation vector which represents a visit sequence. Let y denote ground truth labels, then we use the cross-entropy loss to fine tune the model as

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i\log\left(\hat{y}_i\right) + (1-y_i)\log\left(1-\hat{y}_i\right)\right). \tag{21}$$

For training with the whole GRACE model, we first train the AutoEncoder with Eq. (8) until it convergence. Then, we only keep the decoder of the AutoEncoder, and jointly train the decoder, generator, and discriminator, *i.e.,*, the Item-relation-aware GAN, with Eq. (13). After that, we use the trained item-relation-aware GAN to generate N_G fake samples. Finally, we use both real and generated fake data to pre-train the representation learning model with Eq. (19).

Compared with existing studies on insufficient EHR data [4, 6, 40], the proposed GRACE has the following merits. First, our method can effectively improve the performance of models but does not need any additional information. Most previous methods need additional information which is hard to access in rare diseases, *e.g.,* GRAM [6] needs relations among events to build the tree and MetaPred [40] needs additional datasets to perform meta-learning. Second, our method avoids directly training the predictive model using noisy fake data which may mislead the model fitting. Finally, our method takes full advantage of the fake data, *i.e.,* we encourage the model to capture the gap between the real and fake data using a real-fake contrastive learning pre-training task. Note that the previous methods usually treat both the real and fake data equally and they ignore the gap between the real and fake data.

Table 1: Datasets statistics.

Dataset	Diabetes	Hypertension	Mortality
# of samples	48,586	48,586	20,378
# of positive	10,290	3,583	2,610
# of negative	38,296	45,003	17,768
# of visits	316,398	316,398	1,549,300
Avg. # of visits	6.51	6.51	76.03

Compared with previous studies on pre-training for EHR data [19, 20, 26, 27], the proposed GRACE has the following merits. On the one hand, our method can perform pre-training on insufficient data, but previous methods such as RAPT [28] need a huge amount of data to perform pre-training. On the other hand, in addition to capturing the characteristics of EHR data, our method can further effectively model the real EHR data by the proposed real-fake contrastive learning pre-training task.

4 EXPERIMENTS

In this section, we construct experiments to demonstrate the effectiveness of our methods.

4.1 Experimental Setup

4.1.1 Construction of the Datasets. The pregnant dataset was collected from the prenatal care examination records of a hospital in Beijing spanning from 2008 to 2018, which contains 48,586 samples. All user identity information was removed for anonymization. All experiments were carried out within the hospital with strict regularizations on privacy protection. For this dataset, we select five items as input features of our model, namely gestational week, diastolic pressure, systolic pressure, fundal height, and weight.

The in-hospital mortality dataset is collected from Medical Information Mart for Intensive Care (MIMIC) database [17], which is a large (with 20,378 samples), single-center database comprising information relating to patients admitted to critical care units at a large tertiary care hospital[1]. We follow the MIMIC-based benchmark [15] proposed by Harutyunyan *et al.* to construct the in-hospital mortality prediction dataset[2]. For this data, we use 15 numerical items as input features.

We summarized the detailed dataset statistics in Table 1.

4.1.2 Evaluated Tasks. We use three healthcare-related tasks to test the effectiveness of our methods.

Diabetes Prediction. The task aims to diagnose gestational diabetes. Both baselines and our method take prenatal care examination records before 30 weeks as inputs and generate the probability of suffering gestational diabetes.

[1] https://mimic.mit.edu

[2] https://github.com/YerevaNN/mimic3-benchmarks

Table 2: Experiments results in percent for three tasks. The best results except for the last group are in bold. We repeat the dataset splitting process by five times and report the result as average performance (standard deviation).

Task		Diabetes Prediction			Hypertension Prediction			Mortality Prediction			AVG
Metric		F1	AUC-PR	AUC-ROC	F1	AUC-PR	AUC-ROC	F1	AUC-PR	AUC-ROC	
Vanilla Training	LSTM	36.5 (0.7)	29.2 (1.0)	60.5 (1.0)	31.0 (1.7)	25.4 (2.2)	76.6 (1.2)	31.4 (0.4)	26.4 (1.0)	69.1 (1.0)	42.9 (0.6)
	GRU	36.1 (0.5)	28.5 (0.6)	59.7 (0.7)	32.3 (0.6)	25.9 (0.7)	77.0 (0.3)	32.4 (1.1)	27.8 (0.8)	70.6 (1.1)	43.4 (0.6)
	Dipole	40.2 (2.8)	33.3 (4.1)	65.4 (2.9)	32.0 (1.6)	25.0 (2.4)	76.9 (1.1)	32.7 (1.0)	27.2 (1.3)	67.7 (1.1)	44.7 (0.6)
	RETAIN	44.3 (1.5)	39.9 (2.1)	70.1 (1.9)	32.0 (2.1)	27.2 (0.9)	77.5 (0.4)	32.2 (0.9)	26.2 (1.0)	69.1 (0.7)	46.6 (0.5)
	RAPT	49.4 (8.0)	47.0 (12.2)	75.5 (9.5)	29.5 (7.7)	25.7 (3.8)	76.7 (2.8)	34.9 (1.6)	31.3 (1.9)	73.2 (1.1)	49.3 (2.5)
	Trans.	49.1 (7.9)	44.6 (12.2)	74.1 (9.5)	31.9 (1.9)	26.4 (1.5)	77.5 (0.7)	36.2 (0.8)	31.6 (1.2)	73.3 (0.7)	49.4 (2.9)
	HiTANet	49.4 (7.5)	48.0 (11.0)	75.1 (7.1)	33.3 (1.1)	26.9 (1.5)	78.2 (0.7)	34.0 (2.0)	29.1 (1.6)	70.9 (1.2)	49.4 (3.0)
	T-LSTM	49.3 (2.1)	47.8 (3.1)	76.6 (1.7)	34.3 (0.7)	29.4 (1.3)	78.3 (0.4)	35.3 (0.7)	30.0 (0.7)	72.8 (0.4)	50.4 (0.6)
w/ GRACE	Trans.	61.0 (0.8)	63.9 (1.0)	84.5 (0.3)	35.0 (1.4)	30.1 (0.9)	**79.5** (0.4)	**37.6** (1.8)	32.6 (1.6)	**75.1** (1.8)	55.5 (1.0)
	RAPT	**62.7** (2.3)	**65.0** (1.8)	**84.6** (1.0)	**35.4** (1.0)	**31.0** (1.5)	**79.5** (0.3)	36.8 (1.9)	**33.0** (1.3)	74.6 (1.8)	**55.9** (0.6)
w/ All Data	Trans.	59.6 (1.1)	61.6 (1.2)	83.7 (0.5)	33.1 (4.4)	29.3 (1.1)	79.5 (0.4)	38.1 (1.8)	33.9 (2.3)	75.3 (1.7)	54.9 (1.1)
	RAPT	62.6 (1.1)	64.3 (1.7)	84.4 (0.8)	36.3 (1.1)	33.2 (0.7)	80.1 (0.3)	39.2 (1.1)	35.3 (1.5)	75.5 (1.4)	56.8 (0.6)

Hypertension Prediction. The task aims to diagnose gestational hypertension. Similar to diabetes prediction, both baselines and our method take examination records before 30 weeks as input and generate the probability of suffering gestational hypertension.

In-hospital Mortality Prediction. The task aims to predict in-hospital mortality. Both baselines and our method take examination records of the first 48 hours of an ICU stay as input and generate the probability of in-hospital mortality.

4.1.3 Comparison Methods. We consider the following methods as baselines for comparison:

- LSTM [16]. This is the original long short-term memory neural network with visit sequences as inputs.
- GRU [5]. This is a gating mechanism in the recurrent neural networks, which has fewer parameters than LSTM.
- Transformer [32]. This uses an attention mechanism to model sequence data, which deals with long-term dependencies.
- RETAIN [7]. This is the REverse Time AttentIoN model, employing two RNNs to generate attention weights.
- T-LSTM [3]. This is the time-aware LSTM, which adopts a decaying function to handle irregular time between visits.
- Dipole [22]. This is a sequence neural network that is specifically designed for medical visit sequence data. Dipole adopts three attention mechanisms to handle long-term medical code dependencies and provide interpretability.
- HiTANet [21]. This is a hierarchical attention-based model that generates visit representations with local time and proposes a novel attention mechanism to associate timestamps with visits.
- RAPT [28]. This is a pre-training method for EHR data, which proposed a time-aware transformer and three pre-training tasks. In our experiments, we only use the time-aware transformer.
- GRACE. This is our method. Because our method is model-agnostic, we use Transformer and RAPT as the representation learning model, *i.e.,* the $Model(\cdot)$ function in Eq. (14).

4.1.4 Evaluation Metrics. For the tasks, we use *Area Under Receiver Operating Characteristic Curve (AUC-ROC), Area Under Precision-Recall Curve (AUC-PR)* and *F1-score (F1)* as the evaluation metrics.

We split all datasets into three parts, namely the training set, the validation set, and the test set. For evaluating the performance of our method on insufficient datasets, we randomly extract 3,000 samples as the training set, 1,000 samples as the validation set, and use the rest samples as the test set. We trained the model with the training set, tuned the hyper-parameters with the validation set, and then computed the performance on the test set. In addition, we repeat the above dataset splitting process by five times and report the average performance and the standard deviation for both baselines and our method.

4.1.5 Implementation Details. Our software environment contains ubuntu 20.04, PyTorch v1.7.0, and python 3.8.8. All of the experiments are conducted on a machine with four GPUs (NVIDIA GeForce GTX 2080 Ti) and 64GB memory.

For training models, we used RMsprop [13] with a batch size of 64 in the GAN training stage, and Adam [18] with a batch size of 64 in the pre-training stage and the fine-tuning stage. For the learning rate, we set it as 5e-4 in the pre-training stage and 1e-4 in other stages. In the GAN training process, we set the clip weight as $[-0.01, 0.01]$ for the 1-Lipschitz discriminator. For the experiments, we set the hidden state dimension as $h = 128$ for both baselines and our approach. We set masked ratio $\delta = 0.3$, $\beta = 0.2$ (Eq. (16)), $\alpha = 0.1$ (Eq. (19)), and the number of generated data $N_G = 10,000$. Finally, we employed dropout [29] with dropout rate=0.5 for the classification layer of all models on classification tasks. These hyper-parameters were selected based on the performance on the validation set.

4.2 Result and Analysis

Table 2 shows the result on three tasks. The first group is the models without any pre-training, *i.e.,* initialize with random parameters. The second group is the models pre-trained with our method. The third group is the models pre-trained over the whole dataset using the masked records prediction task, which is used as a reference to compare our model with large size dataset pre-training (our model was pre-trained over a small size dataset with only 3,000 samples). From the results, we have the following findings.

1) Comparing the performance of models in the first group, we can find that LSTM and GRU perform worse among all the baselines. Because they do not consider any characteristics of EHR data. Dipole and RETAIN perform a few better than LSTM and GRU but perform worse than other models, due to that they mainly introduced attention to model correlation among items in a sequence. Besides, the models which consider irregular time intervals perform better, such as RAPT, HiTANet, and T-LSTM. This shows that modeling time information is helpful for EHR modeling. The performance of T-LSTM is better than two Transformer-based models. A possible reason is that Transformer-based models suffer from over-fitting problems for the limited size of the training set. Considering RAPT has more parameters than Transformer, that is also the reason why RAPT performs worse than Transformer.

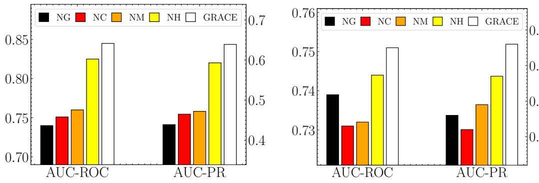
2) Comparing the first group and the second group, we can find that the models pre-trained with our method are better than all baselines with a large margin in all cases. That demonstrates that the proposed pre-training method can effectively improve the performance of models on insufficient data. Here, the pre-trained RAPT performs better than the pre-trained Transformer. This shows our method can effective training models with more parameters.

3) Comparing the second group and the third group, we can find that the performance of models pre-trained with our methods is closed to the performance of the models pre-trained with all data. On some tasks, our methods even surpass the performance of the models pre-trained with all data, *e.g.,* Transformer and RAPT on diabetes prediction task. That indicates that our methods can help the models with insufficient data achieve comparable performance to the model trained over a huge amount of data.

4.3 Method Analysis

4.3.1 Ablation Study. In our method, we have incorporated generated data, item-relation aware mechanisms in the GAN discriminator, and two pre-training tasks, *i.e.,* masked records prediction, and real-fake contrastive learning. Here, we determine how each component contributes to the final performance. We compare four variants of the proposed method: (1) NG without generated data, *i.e.,* only using the real data in the masked records task over the training set, (2) NC without the real-fake contrastive learning task, (3) NM without the masked records prediction task, (4) NH without the hierarchical item-relation aware mechanisms in the discriminator, *i.e.,* using vanilla Transformer as the discriminator. The results of AUC-ROC and AUC-PR scores of the diabetes prediction task and the mortality prediction task for the ablative models are reported.

Figure 2 presents all the comparison results of the four variants. First, our method outperforms all variants. These results indicate that all parts are essential to improve the performance of our model. Second, NM performs better than NC. That indicates capturing the gap between the real data and the fake data is more useful. Third, for in-hospital mortality prediction task, NG performs better than NC and NM, but for diabetes prediction tasks, NG performs worse than NC and NM. A possible reason is that the in-hospital-mortality data has many missing items, which results in the generated data being noisier, which affects the quality of the generated data. Finally, our method performs better than NH. That indicates the proposed item-relation-aware discriminator can effectively handle the correlations among EHR records and encourage the generator to generate more reasonable data.



(a) Diabetes prediction. (b) Mortality prediction.

Figure 2: Ablation study of our method on two datasets.

Table 3: Performance comparison with self-training [34]. All results are based on Transformer.

Model	Diabetes		Hypertension		Mortality	
	PR	ROC	PR	ROC	PR	ROC
Only real data	44.6	74.1	26.4	77.5	31.6	73.3
w/ Self-training	37.2	69.1	23.2	75.2	28.8	70.9
w/ Our Method	**63.9**	**84.5**	**30.1**	**79.5**	**32.6**	**75.1**

4.3.2 Effect of Pre-training. In this part, we compare our method with the baseline of directly training on the generated data. The experiment aims to demonstrate the effectiveness of our novel pre-training-based generated data application mode. For the baseline, we apply the self-training method in [34] to label the generated data. The method first uses the real data to train a predictive model (*e.g.,* Transformer), and use the trained model to label the generated data. After that, it uses both real data and labeled generated data to train a new predictive model.

We show the results in Table 3. As we can see, self-training performs worse than our method and even perform worse than training with only the source data. This indicates that, for EHR, the generated data and the labels contain a lot of noise, which is not suitable to train a predictive model directly. The proposed method indirectly uses the generated data to pre-train data representations, which avoids this shortage of the generated data.

4.3.3 Generated Samples Analysis. In this part, we qualitatively analyze why the proposed item-relation-aware GAN is useful to improve performance. As we mentioned above, each item has its changing trend and different items have correlations with each other. Here, we use the correlation coefficient of the changing rates for different items to qualitatively analyze how realistic the generated data with different discriminators. Specifically, we calculate the changing rate of each item as

$$\eta_t^{(i)} = \left(e_t^{(i)} - e_{t-1}^{(i)}\right) / \left(e_{t-1}^{(i)}\right),$$

where i denotes the i-th item, and t denotes the t-th visit. Based on the changing rate, we calculate the correlation coefficient of $\eta_t^{(i)}$ among all items.

The correlation coefficients of the changing rate for the items of the diabetes dataset are shown in Figure 3. Here, Figure 3(a), Figure 3(b) and Figure 3(c) are the correlation coefficients for source

	DP	SP	FH	W
DP	1.000	0.833	0.044	0.379
SP	0.833	1.000	0.038	0.435
FH	0.044	0.038	1.000	0.040
W	0.379	0.435	0.040	1.000

(a) Source.

	DP	SP	FH	W
DP	1.000	0.959	-0.003	0.927
SP	0.959	1.000	-0.001	0.883
FH	-0.003	-0.001	1.000	-0.003
W	0.927	0.883	-0.003	1.000

(b) GRACE.

	DP	SP	FH	W
DP	1.000	0.982	-0.326	0.964
SP	0.982	1.000	-0.158	0.985
FH	-0.326	-0.158	1.000	-0.151
W	0.964	0.985	-0.151	1.000

(c) Transformer.

Figure 3: The correlation coefficient of different data. Here, "DP" denotes diastolic pressure, "SP" denotes systolic pressure, "FH" denotes fundal height, and "W" denotes weight.

data, generated data with GRACE, and generated data with a vanilla GAN, respectively. We calculate the distance between the two generated data to the source data, which are 0.155 for GRACE and 0.256 for the vanilla GAN. The distance of the source data to the data generated by GRACE is more close to the generated by the vanilla GAN. The results indicate that the proposed GRACE can effectively capture the characteristics of EHR data and generate more reasonable data.

4.4 Real World Deployment and Online Tests

In this part, we deploy the proposed GRACE in a maternal and child health care institution which is one of the largest maternal and child health care hospital in Beijing, and test its online performance for gestational diabetes diagnosis. Specifically, for each patient, once new examination records are obtained, we input the whole visit sequence into the model and get the output of the model. Once the model output "diseased" (*i.e.,* $\hat{y} = 1$), we stop the process and output the diagnosis to a doctor. We counted two metrics: accuracy and average diagnosis time. The diagnosis time is the time to output "diseased". Here we incorporate the Transformer and RAPT without pre-training and the best baseline T-LSTM for comparison.

As shown in Table 4, models trained with the proposed GRACE are consistently better than baselines. In addition, we extensively refer to the literature of medical study and find that gestational diabetes mainly occurs in 24-28 weeks of pregnancy [1]. As we can see, the diagnosis times of GRACE are more reasonable. Based on this online test performance, we believe doctors and patients can benefit from GRACE in various healthcare-related tasks.

5 RELATED WORK

Deep Learning on EHR Data. Since healthcare became an important research domain, various deep learning models have been proposed for modeling EHR data. Usually, the EHR data can be formed as sequences, so sequential deep learning models, such as Recurrent neural network (RNN) based models [2, 3, 7, 11, 22], and Transformer based models [21, 28, 37, 39], are widely used to model EHR data. These models were proposed to handle some characteristics of EHR-based applications, such as interpretability [7, 22, 39], irregular time intervals [3, 21, 28], incompleteness [28, 33], insufficiency [6, 35, 40], medical knowledge [23, 36], and so on. Our methods focus on the insufficiency of EHR data, existing methods usually leverage additional information to enhance the model. For example, GRAM [6] employed medical knowledge using graph-based attention, MetaPred [40] introduced meta-learning, and MedPath [35] employed knowledge graph. Oppositely our method does not require this additional information.

Table 4: Performance comparison of online applications on diabetes prediction task. Here, "ACC" denotes accuracy, "ADT" denotes average diagnosis time.

Model		ACC	ADT
w/o Pre-training	T-LSTM	21.2	15.3
	Transformer	21.6	15.3
	RAPT	24.4	16.6
w/ GRACE	Transformer	73.7	24.5
	RAPT	**76.1**	26.3

With the development of pre-training in natural language processing [9], many works try to migrate this technology to the medical field. For model architecture, some works [19, 20, 26, 27] proposed to modify the BERT architecture for EHR data. For example, Med-BERT [20] extended the architecture to create generalized embedding with a large vocabulary, and RareBERT [26] presented a novel architecture for learning robust representation on a highly imbalanced dataset. For pre-training tasks, RAPT [28] proposed three pre-training tasks which are suitable for EHR data. Our method is model-agnostic. It can adopt these existing model architectures in Eq. (14) to implement the representation learning model.

GAN-based Model for Sequential Data. For insufficient data applications, many studies propose on generate sequential data via GAN-based models as data augmentation [4, 8, 25]. In the early stage, GAN models were migrated to generate time series data [10, 24]. They used RNNs as both the generator and discriminator to generate data from random vector sequences. However, these methods are not sufficient for time series data, because they directly generated the data by random sequence vector and neglect the temporal dependencies.

Then, many works leveraged AutoEncoder to enhance the GANs model. These models can be divided into two main categories: generating in latent space [4, 25, 38] and generating in feature space [8]. For methods that generate in latent space, EHRGAN [4] generates fake representations by simply mixing real data representations and random noise vector by a random binary mask vector, and RTSGAN [25] synthesized fake representations in the latent space by MLP layer. Then, they use the decoder to output the sequence data in the feature space. For methods that generate in feature space, MedGAN [8] first trained an AutoEncoder, and then jointly fine-tuned the decoder with the generator. The GRACE model also adopts an AutoEncoder architecture to generate fake data from latent space. Compared with the existing methods, our method contains specialized mechanisms for EHR characteristic modeling, and therefore is more suitable for medical applications.

6 CONCLUSION

In this paper, we propose a novel deep learning model training method, namely GRACE, to solve the data insufficiency problem in

EHR-based applications. In GRACE, we proposed an item-relation-aware GAN to generate high-quality fake data through capturing characteristics of EHR sequences using a real-data autoencoder initialized generator and a hierarchical item-relation aware discriminator. We further use the GAN-generated data to pre-train an EHR representation learning model through a masked records prediction task and a real-fake contrastive learning task. Finally, we use the EHR representations of real data produced by the pre-training model to fine train the final prediction model. The proposed method can generate high-quality augment data to solve the data insufficiency problem of EHR, and can also avoid introducing noises of fake data into the final prediction model. Extensive experiments on three healthcare-related real-world datasets demonstrated the effectiveness of our method. We also deployed our method in a maternal and child health care hospital for the online test, which further evaluated the performance of the proposed method.

As future work, we will test our approach on more kinds of EHR data, and enhance the generalizability of our approach. We also consider to apply the proposed framework over other applications that suffer the similar data insufficiency problem as EHR applications.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2019YFB2101804), the National Natural Science Foundation of China (Grant No. 82161148011, 72171013), and the Fundamental Research Funds for the Central Universities (Grant No. YWF-22-L-838).

REFERENCES

[1] American Diabetes Association et al. 2004. Gestational diabetes mellitus. *Diabetes care* 27, suppl 1 (2004), s88–s90.

[2] Tian Bai, Shanshan Zhang, Brian L. Egleston, and Slobodan Vucetic. 2018. Interpretable Representation Learning for Healthcare via Capturing Disease Progression through Time. In *KDD'18*. ACM, 43–51.

[3] Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, and Jiayu Zhou. 2017. Patient Subtyping via Time-Aware LSTM Networks. In *KDD'17*. ACM, 65–74.

[4] Zhengping Che, Yu Cheng, Shuangfei Zhai, Zhaonan Sun, and Yan Liu. 2017. Boosting Deep Learning Risk Prediction with Generative Adversarial Networks for Electronic Health Records. In *ICDM'17*. IEEE Computer Society, 787–792.

[5] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP'14*. ACL, 1724–1734.

[6] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, and Jimeng Sun. 2017. GRAM: Graph-based Attention Model for Healthcare Representation Learning. In *KDD'17*. ACM, 787–795.

[7] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter F. Stewart. 2016. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. In *NIPS'16*. 3504–3512.

[8] Edward Choi, Siddharth Biswal, Bradley A. Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. 2017. Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. In *MLHC'17 (Proceedings of Machine Learning Research, Vol. 68)*. PMLR, 286–305.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT'19 (1)*. Association for Computational Linguistics, 4171–4186.

[10] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. 2017. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. *CoRR* abs/1706.02633 (2017).

[11] Junyi Gao, Cao Xiao, Yasha Wang, Wen Tang, Lucas M. Glass, and Jimeng Sun. 2020. StageNet: Stage-Aware Neural Networks for Health Risk Prediction. In *WWW'20*. ACM / IW3C2, 530–540.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. *CoRR* abs/1406.2661 (2014).

[13] Alex Graves. 2013. Generating Sequences With Recurrent Neural Networks. *CoRR* abs/1308.0850 (2013).

[14] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved Training of Wasserstein GANs. In *NIPS'17*. 5767–5777.

[15] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, and Aram Galstyan. 2017. Multitask Learning and Benchmarking with Clinical Time Series Data. *CoRR* abs/1703.07771 (2017).

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (1997), 1735–1780.

[17] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3, 1 (2016), 1–9.

[18] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR'15 (Poster)*.

[19] Yikuan Li, Mohammad Mamouei, Gholamreza Salimi Khorshidi, Shishir Rao, Abdelaali Hassaïne, Dexter Canoy, Thomas Lukasiewicz, and Kazem Rahimi. 2021. Hi-BEHRT: Hierarchical Transformer-based model for accurate prediction of clinical events using multimodal longitudinal electronic health records. *CoRR* abs/2106.11360 (2021).

[20] Yikuan Li, Shishir Rao, Jose Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. 2020. BEHRT: transformer for electronic health records. *Scientific reports* 10, 1 (2020), 1–12.

[21] Junyu Luo, Muchao Ye, Cao Xiao, and Fenglong Ma. 2020. HiTANet: Hierarchical Time-Aware Attention Networks for Risk Prediction on Electronic Health Records. In *KDD'20*. ACM, 647–656.

[22] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis Prediction in Healthcare via Attention-based Bidirectional Recurrent Neural Networks. In *KDD'17*. ACM, 1903–1911.

[23] Fenglong Ma, Jing Gao, Qiuling Suo, Quanzeng You, Jing Zhou, and Aidong Zhang. 2018. Risk Prediction on Electronic Health Records with Prior Medical Knowledge. In *KDD'18*. ACM, 1910–1919.

[24] Olof Mogren. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *CoRR* abs/1611.09904 (2016).

[25] Hengzhi Pei, Kan Ren, Yuqing Yang, Chang Liu, Tao Qin, and Dongsheng Li. 2021. Towards Generating Real-World Time Series Data. *CoRR* abs/2111.08386 (2021).

[26] P. K. S. Prakash, Srinivas Chilukuri, Nikhil Ranade, and Shankar Viswanathan. 2021. RareBERT: Transformer Architecture for Rare Disease Patient Identification using Administrative Claims. In *AAAI21*. AAAI Press, 453–460.

[27] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. 2021. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine* 4, 1 (2021), 1–13.

[28] Houxing Ren, Jingyuan Wang, Wayne Xin Zhao, and Ning Wu. 2021. RAPT: Pre-training of Time-Aware Transformer for Learning Robust Healthcare Representation. In *KDD'21*. ACM, 3503–3511.

[29] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958.

[30] Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. Generating Text with Recurrent Neural Networks. In *ICML'11*. Omnipress, 1017–1024.

[31] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018).

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS'17*. 5998–6008.

[33] Jianing Xi, Liping Ye, Qinghua Huang, and Xuelong Li. 2021. Tolerating Data Missing in Breast Cancer Diagnosis from Clinical Ultrasound Reports via Knowledge Graph Inference. In *KDD'21*. ACM, 3756–3764.

[34] David Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *ACL'95*. Morgan Kaufmann Publishers / ACL, 189–196.

[35] Muchao Ye, Suhan Cui, Yaqing Wang, Junyu Luo, Cao Xiao, and Fenglong Ma. 2021. MedPath: Augmenting Health Risk Prediction via Medical Knowledge Paths. In *WWW'21*. ACM / IW3C2, 1397–1409.

[36] Muchao Ye, Suhan Cui, Yaqing Wang, Junyu Luo, Cao Xiao, and Fenglong Ma. 2021. MedRetriever: Target-Driven Interpretable Health Risk Prediction via Retrieving Unstructured Medical Text. In *CIKM'21*. ACM, 2414–2423.

[37] Muchao Ye, Junyu Luo, Cao Xiao, and Fenglong Ma. 2020. LSAN: Modeling Long-term Dependencies and Short-term Correlations with Hierarchical Attention for Risk Prediction. In *CIKM'20*. ACM, 1753–1762.

[38] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. 2019. Time-series Generative Adversarial Networks. In *NeurIPS'19*. 5509–5519.

[39] Xianli Zhang, Buyue Qian, Shilei Cao, Yang Li, Hang Chen, Yefeng Zheng, and Ian Davidson. 2020. INPREM: An Interpretable and Trustworthy Predictive Model for Healthcare. In *KDD'20*. ACM, 450–460.

[40] Xi Sheryl Zhang, Fengyi Tang, Hiroko H. Dodge, Jiayu Zhou, and Fei Wang. 2019. MetaPred: Meta-Learning for Clinical Risk Prediction with Limited Patient Electronic Health Records. In *KDD'19*. ACM, 2487–2495.