RSD: A Reinforced Siamese Network with Domain Knowledge Guidance for Early Diagnosis

Houxing Ren
School of Computer Science and Engineering, Beihang University, Beijing, 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
batmanfly@gmail.com

ABSTRACT

The availability of electronic health record data makes it possible to develop automatic disease diagnosis approaches. In this paper, we study the early diagnosis of diseases. As being a difficult task (even for experienced doctors), early diagnosis of diseases poses several challenges that are not well solved by prior studies, including insufficient training data, dynamic and complex signs of complications and trade-off between earliness and accuracy.

To address these challenges, we propose a *Reinforced Siamese network with Domain knowledge regularization* approach, namely *RSD*, to achieve high performance for early diagnosis. The RSD approach consists of a diagnosis module and a control module. The diagnosis module adopts any EHR Encoder as a basic framework to extract representations, and introduces two improved training strategies. To overcome the insufficient sample problem, we design a Siamese network architecture to enhance the model learning. Furthermore, we propose a domain knowledge regularization strategy to guide the model learning with domain knowledge. Based on the diagnosis module, our control module learns to automatically determine whether making a disease alert to the patients based on the diagnosis results. Through carefully designed architecture, rewards and policies, it is able to effectively balance earliness and accuracy for diagnosis. Experimental results have demonstrated the effectiveness of our approach on both diagnosis prediction and early diagnosis. We also perform extensive analysis experiments to verify the robustness of the proposed approach.

CCS CONCEPTS

• **Applied computing** → **Health informatics**.

KEYWORDS

Early Diagnosis, Siamese Network, Reinforcement Learning

*Corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557440

ACM Reference Format:
Houxing Ren, Jingyuan Wang, and Wayne Xin Zhao. 2022. RSD: A Reinforced Siamese Network with Domain Knowledge Guidance for Early Diagnosis. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557440

1 INTRODUCTION

Various chronic diseases seriously threaten the health of humans, for example, according to the World Health Organization (WHO), seven of the top ten causes of death in 2019 were chronic non-communicable diseases [35]. It also has been shown that medical interventions at an early stage of disease progression can effectively prevent the potential health threats [13]. However, even for experienced doctors, it is not easy to accurately diagnose diseases at an early stage. Firstly, for most diseases (especially the diseases with severe harm), evident symptoms usually appear at the final stage of disease progression. It is difficult to capture faint evidence from complex examination record data. Secondly, the physical examination indicators of patients dynamically change in the disease progression, and the sign of complications are usually hidden in the change trends of indicators. Therefore, early diagnosis of diseases requires the ability to capture diagnostic features from time-varying, complex electronic health records.

In recent years, with the revival of neural networks, deep learning provides a promising computational framework for solving complicated health care tasks [5, 7]. Many studies try to utilize the excellent modeling capacity for automatically learning effective diagnostic features or representations from electronic health records [7]. Especially, Recurrent Neural Networks (RNN) are widely used in health care studies, since many electronic examination records can be formed into sequences [3, 25].

However, there are three major challenges to adapt existing neural network models to early diagnosis. The first challenge is the insufficiency of positive cases in training data. The incidence rates of diseases are relatively low. For example, the incidence rate is about 1.0% for hypertension [44], and is about 1.8% for diabetes [11]. Usually, it is more difficult to collect positive samples due to privacy or other constraints. The lack of positive training samples may cause the over-fitting in training deep learning models for the diagnosis task, which leads to performance decreasing in practice. The second challenge is the dynamic and complex signs of complications. During the disease progression, substantial changes would

take place in the physical indexes of patients [43], such as body weights. It is difficult to effectively extract and learn distinguishing features from dynamic, irregular, and unstable examination records for automatically diagnosing diseases. The third challenge is the trade-off problem between *earliness* and *accuracy* in diagnosis. On one hand, to reduce the risk to patients, diseases should be diagnosed as early as possible, so that their harm can be effectively prevented or controlled. On the other hand, automatic diagnosis requires sufficient examination records for learning, and incomplete information in early examinations is likely to cause improper treatment. We need to balance the two factors in practice.

To address the above three challenges, in this paper, we propose a *Reinforced Siamese networks with Domain knowledge regularization* approach, namely *RSD*, to achieving high performance for early diagnosis. The RSD approach consists of a diagnosis module and a control module.

The diagnosis module adopts any EHR Encoder as a basic framework to extract feature representations from the visit sequence data of examination records and introduces two improved training strategies. To overcome the insufficient sample problem, we design a Siamese network architecture to utilize pairwise sample relations to enhance the model learning. Such a learning strategy is effective to increase the potential use of limited positive samples. In addition, the Siamese network can also provide more information (*i.e.,* mean encodings of healthy samples and diseased samples) to the control module which can help the control module determine whether to make a disease alert to the patients. Furthermore, we propose a domain knowledge regularization strategy that leverages prior medical knowledge to guide the model to learn more stable representations.

Based on the diagnosis module, our control module determines whether to make a disease alert to the patients based on the diagnosis results obtained from the diagnosis module. We carefully design the model architecture, rewards, and policies in the reinforcement learning framework, so that our control module is effective to balance earliness and accuracy for diagnosis. Our solution integrates the above techniques in a joint approach, which can simultaneously address the aforementioned three challenges.

Overall, our contributions are summarized as follows:

- We design two improvement techniques to obtain more stable representations for early diagnosis, namely Siamese network training and domain knowledge guidance.
- We design a reinforcement learning framework as control module which is effective to balance earliness and accuracy for diagnosis.
- Extensive experiments on three real-world EHR datasets demonstrate the effectiveness of our method. The proposed RSD approach achieves superior performance for both of the diagnosis prediction and the early diagnosing tasks compared with several state-of-the-art baselines.

2 PROBLEM FORMULATION

In this section, we introduce the background for this work and formally define our task.

EHR Sequence Data. In the disease progression, patients need to visit the hospital multiple times for physical index examination. We

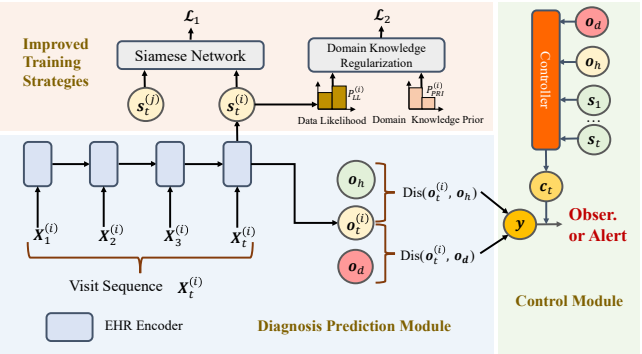


Figure 1: Overview of the proposed approach RSD. In our approach, the diagnosis module learns to predict the confidence score for the disease outcome, and another control module determines whether making the diagnosis alert or continuing observation.

represent the examination records for a patient during his disease progression as a *visit sequence* of chronologically ordered events with irregular time intervals. For the patient i, his visit sequence is denoted as

$$\boldsymbol{X}^{(i)} = \left(\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \ldots, \boldsymbol{x}_t^{(i)}, \ldots, \boldsymbol{x}_{T_i}^{(i)}\right), \tag{1}$$

where $\boldsymbol{x}_t^{(i)}$ is the examination records (*i.e.,* a feature vector consisting multiple examination results) for the t-th visit. The visits are unevenly distributed over the duration of disease progression. For the t-th visit, we denote the visit time as τ_t, which is the timestamp for the t-th visit.

EHR Encoder. As mentioned before, our method is a general methods which can adopt any existing model as its EHR encoder to extract representations of EHR data [2, 7, 28, 29, 45]. We formally define these methods as sequence-to-vector model which output a representation vector of EHR visit sequences, *i.e.,*

$$\boldsymbol{s}^{(i)} = \text{Model}\left(\boldsymbol{X}^{(i)}\right), \tag{2}$$

where $\boldsymbol{s}^{(i)}$ is the representation that represents the whole visit sequence, *e.g.,* the last hidden state of LSTM, and the average hidden state of Transformer.

Diagnosis Prediction. Since not all the EHR data can be observed in its entirety, it is needed to adaptively make the diagnosis based on current data. Given the visit sequence $\boldsymbol{X}^{(i)}$ of a patient, we define the early partial observation of $\boldsymbol{X}^{(i)}$ as

$$\boldsymbol{X}_t^{(i)} = \left(\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \ldots, \boldsymbol{x}_t^{(i)}\right), \tag{3}$$

which is a sub-sequence of $\boldsymbol{X}^{(i)}$, representing the available examination data at the t-th visit. With available records, we would like to automate the diagnosis by learning a diagnosis model. For the t-th visit, we define our diagnosis model as a prediction function that is with $\boldsymbol{X}_t^{(i)}$ as inputs and gives the output:

$$\boldsymbol{s}_t^{(i)} = \text{Model}\left(\boldsymbol{X}_t^{(i)}\right), \; y_t^{(i)} = f\left(\boldsymbol{s}_t^{(i)}\right), \tag{4}$$

where $y_t^{(i)} \in \{healthy, diseased\}$ is a binary diagnosis label indicating whether the patient will suffer from a disease or not.

Early Diagnosis. To reduce the risk of diseases to patients, the diagnosis for diseases should be given as early as possible, so that the possible risks could be prevented at an early stage. However, early diagnosis is difficult based on given limited symptoms or signs. It is likely to lead to inappropriate or even wrong diagnosis and treatment. Therefore, our diagnosis model should be as *accurate* as possible, meanwhile, as *early* as possible. In order to balance earliness and accuracy, we equip our diagnosis model $f(\cdot)$ in Eq. (4) with a *control module*. The role of the control module is to monitor the health condition of a patient, and makes the alert when it was confident with the diagnosis results. Formally, it associates the diagnosis label $y_t^{(i)}$ with a learned confidence score $c_t^{(i)} \in [0, 1]$ at each visit. The controller makes the alert when the confidence score is above a threshold. As will be shown in Section 3.2, we automate the progress with a reinforcement learning framework.

To develop our approach, we design two core components, namely *diagnosis module* and *control module*, which makes the diagnosis and determines whether to make the disease alert, respectively. Figure 1 presents the overview of our proposed approach. Next, we describe the two modules in detail.

3 METHODS

In this section, we present the proposed method. Our core idea is to perform dynamic early diagnosis by extending traditional EHR Encoder with two modules: diagnosis module and control module. We start with the diagnosis module, then present how to construct the control module based on the diagnosis module and finally discuss how to train the entire network.

3.1 Diagnosis Module

The diagnosis module aims to generate the diagnosis results according to the available EHR data. With the EHR Encoder, it is straightforward to feed diagnostic features into a prediction unit (*e.g.,* MLP) for diagnosis. However, insufficient training data and complex signs problems make the model learning particularly difficult. In order to better optimize the diagnosis prediction model, we propose two strategies to tackle the issues, namely *Siamese network training* and *domain knowledge guidance*. After that, we present how to predict with the trained model in the inference stage.

3.1.1 Siamese Network Training. Instead of considering individual training samples, Siamese networks [8] model the relations between sample pairs. Given a dataset of N samples, a Siamese network can induce $N \times (N-1)/2$ sample pairs as its training data, while a traditional supervised model has only N training samples. Therefore, Siamese networks are likely to be more suitable for application scenarios with insufficient samples.

Specifically, Siamese network learns to map samples into a latent space, where the samples with the same label have close semantic distance. Given the encoded diagnostic features of a sample pair, denoted by $\boldsymbol{s}^{(i)}$ and $\boldsymbol{s}^{(j)}$ (see Eq. (2)), we first employ a projection head (*e.g.,* MLP with batch normalization [21]) to generate more qualitative representation for Siamese network [4]:

$$\boldsymbol{o}^{(i)} = \mathrm{MLP}(\boldsymbol{s}^{(i)}). \tag{5}$$

Then, we use the Euclidean distance to measure their semantic distance as

$$d_{ij} = \|\boldsymbol{o}^{(i)} - \boldsymbol{o}^{(j)}\|_2. \tag{6}$$

In our model, we consider three types of sequence pairs in Eq. (7), which are $\{\boldsymbol{o}_{T_i}^{(i)}, \boldsymbol{o}_{T_j}^{(j)}\}$, $\{\boldsymbol{o}_t^{(i)}, \boldsymbol{o}_{T_i}^{(i)}\}$ and $\{\boldsymbol{o}_t^{(i)}, \boldsymbol{o}_{T_j}^{(j)}\}$. Besides the pairs consisting of two entire sequences ($\{\boldsymbol{o}_{T_i}^{(i)}, \boldsymbol{o}_{T_j}^{(j)}\}$), we also pair an entire sequence with its subsequence ($\{\boldsymbol{o}_t^{(i)}, \boldsymbol{o}_{T_i}^{(i)}\}$), and pair a subsequence with another entire sequence ($\{\boldsymbol{o}_t^{(i)}, \boldsymbol{o}_{T_j}^{(j)}\}$), where T_i and T_j are the lengths of the two sequences, respectively. There are two major purposes for considering subsequences. First, it can augment the training data with more supervision signals. Second, since our focus is to make the prediction based on partial observations, such a strategy can enhance the capacity of learning from subsequences. Given a subsequence, we use the final health status corresponding to the entire sequence as its ground-truth label.

Formally, the loss of the Siamese network over N visit sequences is defined as:

$$\mathcal{L}_1 = \frac{1}{K} \sum_{i=1}^{N} \sum_{i \neq j} w_i \left(1 - z_{ij}\right) \max(\Delta - d_{ij}, 0)^2 + z_{ij} d_{ij}^2, \tag{7}$$

where $K = N \times (N-1)$, z_{ij} indicates whether the labels of the samples i and j are the same ($z_{ij} = 1$ when $y^{(i)} = y^{(j)}$, otherwise $z_{ij} = 0$), the threshold Δ in Eq. (7) is a preset parameter, and $w_i = \frac{t}{T_i}$ denotes the instance weight for the i-th subsequence (t denotes the t-th visit out of the total T_i visits). Here, we incorporate the instance weight to adaptively set the instance weight of a subsequence[1]. The basic idea is that the fewer data we have observed, the less certain we make the right predictions. Besides, subsequences are usually not stable and easy to contain noise. Such an adaptive weight strategy is also to reduce the influence of noise. Minimizing the loss of $\mathcal{L}_1$ enforces the EHR Encoder (*e.g.,* Model in Eq. (2)) to extract similar features for prenatal care sequences with the same outcome.

3.1.2 Domain Knowledge Guidance. In the medical field, some physical examination indicators have been found to be closely related to diseases. For example, diastolic pressure and systolic pressure are strong indicators of hypertension [42], and Body Mass Indexes (BMI) of patient are strong indicators of diabetes [18]. These distinguishing indicators (*i.e.,* features) in the examination data of patients are more important to consider in diagnosis prediction.

Specially, we consider the normal ranges of these distinguishing features suggested by doctors as *domain knowledge*. To incorporate such domain knowledge, we adopt a posterior regularization method [9] to model the effect of these distinguishing features in diagnosis prediction. Given a distinguishing feature g, such as diastolic pressure at a certain visit time, we assume its values follow different Gaussian distributions for healthy cases and for diseased cases:

$$P_h(g) \sim \mathcal{N}\left(\mu_h, \sigma_h^2\right),\ P_d(g) \sim \mathcal{N}\left(\mu_d, \sigma_d^2\right), \tag{8}$$

where $P_h(\cdot)$ and $P_d(\cdot)$ denote the value distributions of feature g for healthy and diseased cases, respectively, and μ_h and σ_h are mean and standard deviation of g for healthy cases, as well as μ_d and σ_d are for diseased cases. Note that normal distribution is a

[1]Note that the second sequence in our equation is always complete, so that we do not incorporate a similar weight for it.

widely adopted distribution for characterizing examination records, and we also can use other distributions depending on the situation. In our experiments, we use the normal distribution because these features are basically in line with the normal distribution.

Given a feature g for the i-th patient, denoted $g^{(i)}$, we express the domain knowledge for discriminating between healthy and diseased conditions as a Bernoulli distribution as

$$\begin{aligned} P_{PRI}^{(i)} &= \Big(P(y^{(i)} = \text{healthy}), P(y^{(i)} = \text{diseased})\Big) \\ &= \text{Normalization}\left(\frac{g_t^{(i)} - \mu_h}{\sigma_h}, \frac{g_t^{(i)} - \mu_d}{\sigma_d}\right) \end{aligned}, \tag{9}$$

which defines the prior distribution of the diagnosis label $y^{(i)}$, $\text{Normalization}(p_1, p_2) = (\frac{p_1}{p_1+p_2}, \frac{p_2}{p_1+p_2})$ and $g_t^{(i)}$ is actual value of feature g for the i-th patients at the t-th visit.

Using the feature $\boldsymbol{o}_t^i$ extracted by EHR Encoder for the i-th patient at the t-th visit, we compute the likelihood probability distribution of $y^{(i)}$ given the examination data as

$$P_{LL}^{(i)} = \text{Softmax}\Big(\text{Dis}(\boldsymbol{o}^{(i)}, \boldsymbol{o}_d), \text{Dis}(\boldsymbol{o}^{(i)}, \boldsymbol{o}_h)\Big), \tag{10}$$

where $\text{Dis}(\cdot,\cdot)$ is the Euclidean distance, and $\text{Softmax}(p_1, p_2) = (\frac{e^{p_1}}{e^{p_1}+e^{p_2}}, \frac{e^{p_2}}{e^{p_1}+e^{p_2}})$. $\boldsymbol{o}_h$ and $\boldsymbol{o}_d$ denote the mean of feature encodings for all cases with healthy and diseased outcomes, defined as

$$\boldsymbol{o}_h = \frac{1}{N_h}\sum_{i=1}^{N_h} \boldsymbol{o}_h^{(i)}, \ \boldsymbol{o}_d = \frac{1}{N_d}\sum_{i'=1}^{N_d} \boldsymbol{o}_d^{(i')} \tag{11}$$

where $\boldsymbol{o}_h^{(i)}$ ($\boldsymbol{o}_d^{(i')}$) denotes the feature encoding of the i-th healthy sample (i'-th diseased sample) by Siamese network (Eq. (5)), and N_h and N_d are the numbers of healthy and diseased samples, respectively, in training set.

We incorporate domain knowledge using a posterior regularization approach. The core idea is to pull the model-dependent diagnosis towards the prior estimation with domain knowledge, so that domain knowledge can be leveraged to guide the learning of our diagnosis model. Formally, we define the posterior regularization loss as:

$$\mathcal{L}_2 = \frac{1}{N}\sum_{i=1}^{N} \min_{0<g<G} \Big(\text{KL}(P_{PRI}^{(i,g)} \parallel P_{LL}^{(i)})\Big), \tag{12}$$

where G is the number of features used as domain knowledge, $P_{PRI}^{(i,g)}$ is the prior distribution of being diseased given the g-th domain knowledge feature for the i-th sample, $P_{LL}^{(i)}$ is the likelihood of being diseased for the i-th sample, and $\text{KL}(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence which measures the difference between two distributions.

Finally, the loss of diagnosing module is defined as

$$\mathcal{L}_D = \mathcal{L}_1 + \alpha \times \mathcal{L}_2, \tag{13}$$

where α is hyper-parameter to balance the weights for the Siamese network ($\mathcal{L}_1$) and posterior regularization ($\mathcal{L}_2$).

3.1.3 Diagnosis Generation as Classification. After training the loss in Eq. (13), we can learn the parameters of EHR Encoder. In the diagnoses generation step, we calculate the mean of features for all cases with health outcomes and with diseased outcomes as $\boldsymbol{o}_h$ and $\boldsymbol{o}_d$ (following Eq. (11)), respectively. For a case to be diagnosed, we first apply the Encoder to obtain his feature encoding $\boldsymbol{o}^{(i)}$. Then, we compare its distances to both $\boldsymbol{o}_h$ and $\boldsymbol{o}_d$, and use the label with closer distance as the diagnosis results:

$$y^{(i)} = \begin{cases} \text{healthy} & \text{if } Dis(\boldsymbol{o}^{(i)}, \boldsymbol{o}_h) < Dis(\boldsymbol{o}^{(i)}, \boldsymbol{o}_d), \\ \text{diseased} & \text{if } Dis(\boldsymbol{o}^{(i)}, \boldsymbol{o}_h) \geq Dis(\boldsymbol{o}^{(i)}, \boldsymbol{o}_d). \end{cases} \tag{14}$$

Such a method is simple yet non-parametric, which does not rely on manual selection of a classification threshold.

3.2 Control Module

The function of the control module is to determine whether making the diagnosis alert or continuing observation. It needs to make a balance between earliness and accuracy. We formulate such a decision problem in a reinforcement learning (RL) framework, where the key point is how to train an effective policy based on appropriate rewards for the control module.

3.2.1 Reinforcement Learning Framework. The framework contains four major parts: *State, Policy, Action,* and *Reward.*

State. For the t-th visit of the i-th patient, the state observed by the control module is the feature encoding $\boldsymbol{s}_t^{(i)}$ extracted from $\boldsymbol{x}_t^{(i)}$ by the EHR Encoder (Eq. (2)).

Action and Policy. Our control module considers two kinds of actions, namely { Observation, Alert }. Once it has adopted the Alert action, it will halt the monitor process, and meanwhile make the diagnosis alert. At each time step, we draw an action $a_t^{(i)}$ from a stochastic policy conditioned on current state.

As mentioned above, there are limited symptoms or signs for early diagnosis. To effectively capture limited signs, we carefully design a symptom-trend-aware network, namely STAN. Instead of directly employing a simple network (*e.g.,* MLP) to learn the stochastic policy, we introduce the mean encodings of health and diseased samples $\boldsymbol{s}_h$ and $\boldsymbol{s}_d$ to provide more information. We first employ multi-head attention (*MHA*) and residual connection [17] to learn symptom-aware representations, *i.e.,*

$$\begin{aligned} \boldsymbol{u}_t^{(i)} &= \text{MHA}\left(\boldsymbol{s}_t^{(i)}, \boldsymbol{O}_m, \boldsymbol{O}_m\right), \\ \boldsymbol{g}_t^{(i)} &= \boldsymbol{u}_t^{(i)} + \boldsymbol{s}_t^{(i)}, \end{aligned} \tag{15}$$

where $\boldsymbol{O}_m$ are constructed by stacking the mean encodings of health and diseased samples $\boldsymbol{o}_h$ and $\boldsymbol{o}_d$. And multi-head attention is defined as

$$\begin{aligned} \text{Attn}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V}) &= \text{softmax}\left(\frac{\boldsymbol{q}\boldsymbol{K}^\top}{\sqrt{h}}\right)\boldsymbol{V}, \\ \text{MHA}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V}) &= \boldsymbol{W}_o \text{Concat}(head_1, \ldots, head_n), \\ \text{where } head_j &= \text{Attn}\left(\boldsymbol{q}\boldsymbol{W}_q^{(j)}, \boldsymbol{K}\boldsymbol{W}_k^{(j)}, \boldsymbol{V}\boldsymbol{W}_v^{(j)}\right), \end{aligned} \tag{16}$$

where $\boldsymbol{W}_q^{(i)}, \boldsymbol{W}_k^{(i)}, \boldsymbol{W}_v^{(i)} \in \mathbb{R}^{h\times h}$, $\boldsymbol{W}_o \in \mathbb{R}^{h\times nh}$ are learnable parameters and n denotes the numbers of heads. Then, we employ a sequence model to capture the trends of symptom-aware representations. The basic idea is that if the case suffer from the disease, the symptom would become clear with time. Here, we employ LSTM to capture the trend, *i.e.,*

$$\boldsymbol{r}_t^{(i)} = \text{LSTM}(\boldsymbol{g}_1^{(i)}, \ldots, \boldsymbol{g}_t^{(i)}), \tag{17}$$

where $\boldsymbol{r}_t^{(i)}$ is the last hidden state and we employ a fully-connected neural network to learn the stochastic policy as

$$\boldsymbol{p}_t^{(i)} = \left(1 - c_t^{(i)}, c_t^{(i)}\right) = \text{Softmax}\left(\text{MLP}\left(\boldsymbol{r}_t^{(i)}\right)\right), \tag{18}$$

Table 1: Rewards of the control module.

Reward / Prediction / Action	Right	Wrong
Alert	$\rho \cdot \left(1 - \frac{\tau_t}{\tau_{\max}}\right)$	$-\left(1 - \frac{\tau_t}{\tau_{\max}}\right)$
Observation	$-\gamma \times \frac{\tau_t}{\tau_{\max}}$	$\gamma \times \frac{\tau_t}{\tau_{\max}}$

where $c_t^{(i)}$ is the confidence score of the diagnosis results (*i.e.,* the diseased outcome) by our model based on $\boldsymbol{s}_t^{(i)}$.

Reward. As shown in Table 1, we consider four cases to set the reward based on the adopted action and the correctness of diagnosis prediction, where τ_t is the timestamp of the t-th visit of the i-th patient, $\tau_{\max}$ is set as the maximum timestamp, and γ is a positive scaling coefficient.

We can observe that the two cases in the diagonal line receive non-negative rewards, while the rest two anti-diagonal cases receive negative rewards. This setting encourages the control module to alert patients if predictions are correct, while suggest the control module to collect more information if the predictions are wrong.

Note that the rewards of the four cases involve a factor of $\frac{\tau_t}{\tau_{\max}}$, which means that time directly affects the reward setting. If the agent made a right or wrong alert at an earlier stage, it would receive a more positive or negative reward with the absolute value of $(1 - \frac{\tau_t}{\tau_{\max}})$ (with a small τ_t). Contrastively, if the agent missed the diagnosing chance or avoided false alert at a later stage, it would receive a reward with a larger absolute value of $\gamma \times \frac{\tau_t}{\tau_{\max}}$. These settings confirm to previous discussions and our intuition.

In practice, we find it is better to use a small ρ (*e.g.,* $\rho \leq 0.1$), which makes it less aggressive to adopt the action of Alert in a very early stage. A major reason is that early diagnosis of diseases is difficult, and the model should be able to collect sufficient evidence before making the diagnosis. Our reward setting largely follows the suggestions of experienced doctors, and we also extensively refer to the literature of medical study.

3.2.2 Training. Given a set of training data, the expected reward of all examination records for the i-th patient is expressed as

$$\mathcal{J}(\theta_C) = \mathbb{E}\left[r_t^{(i)}\right], \tag{19}$$

where $r_t^{(i)}$ is the reward received by the i-th sample at the t-th visit, and θ_C is the parameter of the control module. We can maximize this term by directly applying gradient based optimization methods. The gradient of $\mathcal{J}$ is given by

$$\nabla_{\theta_C} \mathcal{J}(\theta_C) = \mathbb{E}\left[\nabla_{\theta_C} \log P\left(a_t^{(i)} \mid \boldsymbol{s}_t^{(i)}\right) r_t^{(i)}\right], \tag{20}$$

where $P(a_t^{(i)} \mid \boldsymbol{s}_t^{(i)})$ is the action probability controlled by the policy:

$$P(a_t^{(i)} \mid \boldsymbol{s}_t^{(i)}) = \begin{cases} 1 - c_t^{(i)} & \text{if } a_t^{(i)} = \text{Observation}, \\ c_t^{(i)} & \text{if } a_t^{(i)} = \text{Alert}. \end{cases} \tag{21}$$

Algorithm 1 The training algorithm for the RSD model.

Input: An electronic health dataset $\boldsymbol{X}$.
Output: Model parameters θ_D (Diagnosis module), θ_C (Control module).

```
Randomly initialize θ_D, θ_C.
for episode = 1 to epoch do
    Calculate state of visit sequence s_t^(i) by Eq. (2).
    Calculate loss of diagnosis prediction by Eq. (13).
    Perform stochastic gradient descent on Eq. (13) w.r.t. θ_D.
end for
for episode = 1 to epoch do
    Generate state by Eq. (2) and Policy by Eq. (18).
    Sample action from policy and calculate the reward.
    Calculate gradient of control module by Eq. (22).
    Perform stochastic gradient ascent on Eq. (22) w.r.t. θ_C.
end for
return θ_D, θ_C.
```

We calculate $\nabla_{\theta_C} J(\theta_C)$ by REINFORCE [46] algorithm as

$$\nabla_{\theta_C} \mathcal{J}(\theta_C) = \frac{1}{K} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \nabla_{\theta_C} \log P\left(a_t^{(i)} \mid \boldsymbol{s}_t^{(i)}\right) \left(r_t^{(i)} - b\right), \tag{22}$$

where $b = \mathbb{E}\left[r_t^{(i)}\right]$ is a reward baseline.

3.3 Learning and Discussion

Model Training. To optimize the entire approach, we need to first learn the loss of $\mathcal{L}_D$ in Eq. (13) for the diagnosis module (Section 3.1). Once the parameters θ_D for the diagnosis module have been optimized, we can optimize the control module (Section 3.2) by maximizing the reward function in Eq. (19) through a mini-batch gradient ascent method. We present the entire learning algorithm for our approach in Algorithm 1. Intuitively, we can also train the two parts in an alternative way. However, the performance of the control module depends on the accuracy of the diagnosis module. It will incur performance loss with unconverged diagnosis module. As a result, we first train the diagnosis module, then fix the parameters of the diagnosis module and train the control module.

Difference. To our knowledge, there are very few studies on the automatic diagnosis of diseases. Although the discussed studies are closely related to our work, they do not fully consider the three challenges raised in Section 1. Our work makes an important technical contribution with respect to the three challenges. First, we propose the Siamese network to handle the data insufficiency problem. Second, we propose domain knowledge regularization to improve the training of the diagnosis module. As a comparison, most existing studies [1, 2, 28] require graph-structured domain knowledge, which is difficult to obtain in practice. Finally, we design the model architecture, rewards and policies in the reinforcement learning framework. In the literature, there are seldom studies that consider both earliness and accuracy for the diagnosis of diseases.

4 EXPERIMENTS

In this section, we construct the experiments to verify the effectiveness of our model. We first evaluate the performance of our improved strategies by constructing experiments of diagnosis prediction, where part (or all) of a visit sequence is given for prediction. Then, we evaluate the early diagnosis performance of our approach,

Table 2: Statistics of three datasets.

Dataset	Diabetes	Hypertension	Mortality
# of healthy case	3,663	1,201	2,717
# of diseased case	16,177	18,639	17,819
Avg. # of visits	11.86	11.86	20.16
# of feature	5	5	15

which integrates the diagnosis module with the control module. In this task, the data can be assumed to come in a stream, where the model needs to determine when to stop for the disease alert. Finally, we construct the ablation study and performance tuning to test how each part actually contributes to the final performance and how the parameters influence the final performance.

4.1 Datasets

In the experiments, we first evaluate the performance of the proposed model for diagnosing two types of pregnancy complications, namely gestational hypertension and gestational diabetes. Our data was collected from the prenatal care examination records of a hospital in Beijing spanning from 2016 to 2018. Each visit of a patient has five numeric examination records, *e.g.,* systolic pressure. All the identity information about users has been removed or anonymized. In addition, we collect the in-hospital mortality dataset from Medical Information Mart for Intensive Care (MIMIC) database [22], which is a large (with 20,536 samples) database comprising information relating to patients admitted to critical care units at a large tertiary care hospital. We follow the benchmark [15] proposed by Harutyunyan *et al.* to construct the in-hospital mortality prediction dataset which has a similar format to the pregnancy complications dataset. We summarize the detailed dataset statistics in Table 2.

To evaluate the performance of models, we split the three datasets into three parts with a ratio of 7:1:2 (by visit sequences rather than visits), namely the training set, the validation set and the test set. To carry out our experiments, we train the model with the training set, tune the parameters with the validation set, and then compute the performance on the test set. In addition, we employ five-fold cross-validation to evaluate the performance of models and report the average performance and the standard deviation for both baselines and our model. In the real-world applications, the positive samples and negative samples are imbalance. Learning with imbalanced data is a long-standing issue in machine learning, which is beyond our focus in this work. Currently, to deal with this issue, we adopt a widely-used strategy for balancing the data distribution of the training set, i.e., undersampling. In order to make the experiment results more convincing, we recover the original data distribution (more negative samples) for positive and negative samples in the validation and the test sets.

Next, we conduct the evaluation experiments with the above datasets on two kinds of tasks, namely *diagnosis prediction* and *early diagnosis*. The major difference is as follows: diagnosis prediction aims to make the diagnosis based on all the available examination records, while early diagnosis needs to dynamically monitor the visit sequence and determines to alert the patient (followed by a halting of the monitor) when confident diagnosis can be drawn.

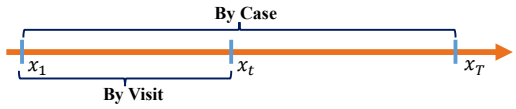


Figure 2: Illustration for the "*ByVisit*" setup and "*ByCase*" setup for diagnosis prediction.

4.2 Evaluation on Diagnosis Prediction

We first evaluate the performance of diagnosis prediction, where part (or all) of a visit sequence is given for prediction.

4.2.1 Comparison Methods. We compare the diagnosis prediction module with LSTM [19], RETAIN [7], Dipole [29], T-LSTM [2], Transformer [45], HiTANet [28]. Among these models, LSTM and Transformer are representatives of the widely used sequential deep learning models. RETAIN, T-LSTM, Dipole, and HiTANet are representatives of the sequential deep learning models designed for EHR data. Some traditional machine learning methods such as Random Forest [27] cannot be integrated with our improved training strategies, our method is not directly comparable.

To compare other training methods with our improved training strategies, given all the comparison methods, we feed the feature vectors extracted by different neural networks into a Sigmoid function, which classifies the input visit sequences as healthy or diseased outcomes. The method optimizes models with the cross-entropy loss function. Note that in this task we only utilize the diagnosis module of our approach for comparison, since the task is to make the diagnosis based on the available data.

4.2.2 Parameter Setting. Our software environment contains Pytorch v1.7.0 and python 3.8.8. All of the experiments are conducted on a machine with one RTX 2080 Ti. For training models, we use Adam [23] optimizer. In the experiments, we set the hidden state dimension as $h = 128$ for all baseline and our approach. We set the threshold $\Delta = 40$, and set the hyper-parameter $\alpha = 0.3$ in Eq. (13). These hyper-parameters are selected based on the performance of the comparison methods on the validation set.

4.2.3 Evaluation Metrics. Since the task is a classification problem, we use *Area Under Receiver Operating Characteristic Curve (AUC-ROC)* as the evaluation metrics. The definition of the evaluation metric is the area under the *ROC Curve* (AUC-ROC), where the area is spanned by taking FPR and TPR as X and Y axes, respectively. Here, we have $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{FP}{TN+FP}$.

4.2.4 Experimental Results. Table 3 presents the performance comparison of our diagnosis prediction module and the baselines. We consider two kinds of setups in the experiments. In the first setup, we consider each visit to the hospital as a check-point to be diagnosed: the examination data from the first visit to the checked visit is available as input, *i.e.,* all $\boldsymbol{X}_t^{(i)}$ are used as test samples. For this setup, if an entire sequence contains n visits, we will generate n samples. We name this experiment setup as "*ByVisit*". In the second setup, we directly use the entire visit sequence of a patient, *i.e.,* $\boldsymbol{X}_{T_i}^{(i)}$, as a test sample. We name this setup as "*ByCase*". We present an illustration for the two kinds of setups in Fig. 2. In addition, we

Table 3: Performance comparison for diagnosis prediction task on the three datasets. $*$ denotes that the proposed model significantly outperforms the same model trained with cross-entropy loss at the level of 0.01.

Methods		Diabetes Dataset		Hypertension Dataset		Mortality Dataset	
		Cross-Entropy	Improved Strategies	Cross-Entropy	Improved Strategies	Cross-Entropy	Improved Strategies
ByVisit	LSTM	0.626 ± 0.013	**0.684 ± 0.029***	0.679 ± 0.014	**0.745 ± 0.010***	0.681 ± 0.016	**0.694 ± 0.016***
	Transformer	0.678 ± 0.010	**0.706 ± 0.008***	0.680 ± 0.006	**0.714 ± 0.014***	0.686 ± 0.015	**0.701 ± 0.017***
	RETAIN	0.652 ± 0.011	**0.692 ± 0.009***	0.701 ± 0.010	**0.726 ± 0.024***	0.648 ± 0.019	**0.675 ± 0.019***
	TLSTM	0.669 ± 0.011	**0.762 ± 0.021***	0.716 ± 0.023	**0.750 ± 0.014***	0.674 ± 0.016	**0.688 ± 0.014***
	Dipole	0.647 ± 0.012	**0.683 ± 0.006***	0.684 ± 0.010	**0.709 ± 0.015***	**0.688 ± 0.024**	0.676 ± 0.017
	HiTANet	0.683 ± 0.023	**0.749 ± 0.016***	0.727 ± 0.008	**0.754 ± 0.012***	0.678 ± 0.020	**0.684 ± 0.022***
ByCase	LSTM	0.648 ± 0.013	**0.735 ± 0.036***	0.791 ± 0.018	**0.811 ± 0.020***	0.710 ± 0.013	**0.728 ± 0.020***
	Transformer	0.724 ± 0.012	**0.772 ± 0.009***	0.776 ± 0.012	**0.792 ± 0.013***	0.708 ± 0.011	**0.731 ± 0.016***
	RETAIN	0.685 ± 0.013	**0.748 ± 0.011***	0.782 ± 0.016	**0.803 ± 0.017***	0.670 ± 0.015	**0.709 ± 0.017***
	TLSTM	0.726 ± 0.016	**0.855 ± 0.014***	0.800 ± 0.019	**0.812 ± 0.020***	0.704 ± 0.014	**0.723 ± 0.014***
	Dipole	0.673 ± 0.015	**0.739 ± 0.009***	0.793 ± 0.015	**0.795 ± 0.018***	0.722 ± 0.017	**0.725 ± 0.016***
	HiTANet	0.737 ± 0.029	**0.834 ± 0.023***	0.797 ± 0.012	**0.805 ± 0.018***	0.696 ± 0.022	**0.716 ± 0.025***

conduct the paired t-test[20] between our method and baseline on the performance to test the significance of improvement.

In Table 3, we present the diagnosis prediction results. Compared the performance of the cross-entropy baseline and our improved strategies, we can observe the effectiveness of our improved strategies. We can see that in almost all the settings, the improved strategies achieve better performance and significantly outperform cross-entropy. Our method incorporates two special techniques to improve the training, namely Siamese network training and domain knowledge guidance and we will further examine the effect of the two techniques in Section 4.4.

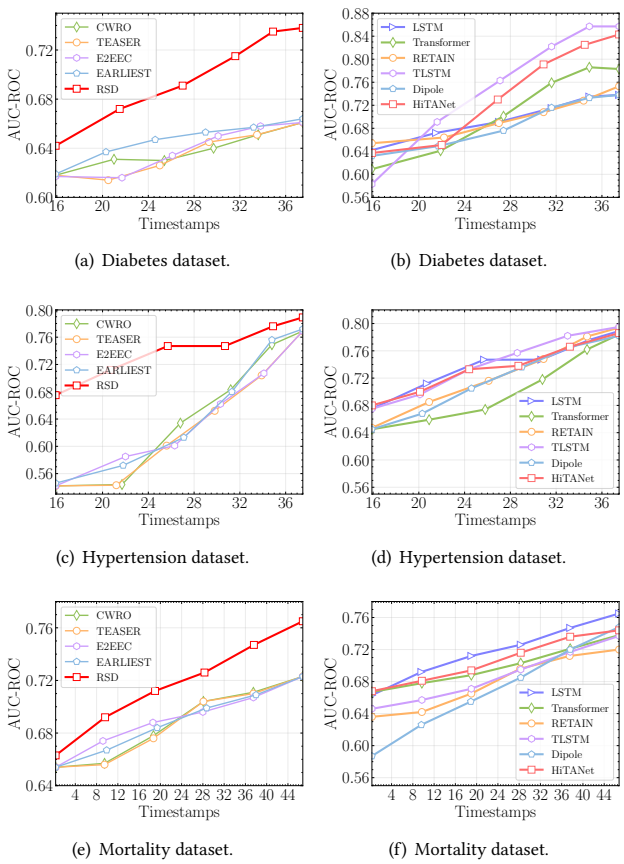
4.3 Evaluation on Early Diagnosis

Next, we evaluate the early diagnosis performance of our approach, which integrates the diagnosis module with the control module. In this task, the data can be assumed to come in a stream, where the model needs to determine when to stop for the disease alert.

4.3.1 Baselines. We consider the following comparison methods of early prediction for comparison: (1) CWRO [16]. It is an ensemble model consisting of a set of classifiers. It uses the "agreement" of the set to decide whether to alert or observe. (2) TEASER [40]. It uses another classifier to analyze the output of probabilistic main classifier and decide whether to alert or observe. (3) E2EEC [39]. It is a deep learning based early classification model, and uses a supervised learning method to train a control module by designing a novel loss function that optimizes for both diagnosis prediction task and early classification. (4) EARLIEST [14]. It is an adaptive model for early classification. EARLIEST directly models the multiple objectives of early classification (namely accuracy and earliness), and enables the joint optimization despite conflicting tendencies. It employs the cross-entropy loss to training the classification module and Reinforcement Learning (RL) approach to training the control module, respectively. Among these baselines, CWRO is a rule-based baseline, that define some rules to perform early classification based on pre-trained classification models. TEASER and E2EEC are two model-based baselines, they define a controller architecture to output the probability of Alert action. And EARLIEST is a recently proposed reinforcement learning based method. For fairness, when comparing our methods with these methods, we use LSTM as the EHR Encoder for all methods.

4.3.2 Experiment Results. Following prior studies [14], we consider early diagnosis as early classification of time series for evaluation. Since there is a trade-off between earliness and accuracy, we should compare the overall performance of different methods on varying advance rates. For each early classification method, we can adjust their related hyper-parameters to control the average advance rate over the test set. For example, the hyper-parameter for our model is γ of rewards in Table 1. We tune the parameter γ in the value sets of $\{0.7, 0.75, 0.8, 0.85\}$, $\{0.3, 0.35, 0.4, 0.45\}$ and $\{2.1, 2.2, 2.3, 2.4\}$, respectively. A larger value of γ leads to a larger advance rate. For other baselines, there are also similar hyper-parameters to be tuned for deriving different advance rates. We consider only the first visit and full visits as the first and last check-points for comparison. For each method, we further generate four intermediate check-points by tuning the hyper-parameters. Then we plot the performance curve under different advance rates. Note that the advance rates for different methods might be different, since they are controlled by hyper-parameters and actually depend on both model and data. Therefore, to compare the performance of the two methods, we mainly focus on whether a curve is above the other one spanned by these check-points.

Figure 3 (a) (c) (e) presents the results of early diagnosis for the comparison methods. From the figure, we can make the following observations: First, TEASER performs worse than CWRO on hypertension dataset but better on the other two datasets. A possible reason is that TEASER is designed for multi-class tasks but diagnosis is a binary classification task. So, the inputs information of TEASER is limited and its performance is inconsistent. Second, E2EEC, EARLIEST and our model outperform rule-based methods. A major reason that the three methods have involved explicit supervision information or reward to guide the learning of model parameters, so that their performance can be optimized in a more effective way. Third, The reinforcement learning based methods, *i.e.,* EARLIEST and our model, outperform the supervised learning



(a) Diabetes dataset. (b) Diabetes dataset.

(c) Hypertension dataset. (d) Hypertension dataset.

(e) Mortality dataset. (f) Mortality dataset.

Figure 3: Performance comparison for early diagnosis task. (a) (c) (e) are performance comparison among different methods with LSTM as EHR Encoder. (b) (d) (f) are performance comparison among different EHR Encoder with our method.

based methods. Reinforcement learning can utilize rewards to inform the control modules about the earliness that it should advance to, *e.g.,* a larger reward for an earlier diagnosis. Therefore, the reinforcement learning methods have an advantage over the supervised learning methods by adaptively learning to control the earliness. Finally, the proposed RSD model achieves the best performance among all the compared methods, *i.e.,* having a performance curve above all the other curves. It indicates that our approach is effective for early diagnosis of diseases.

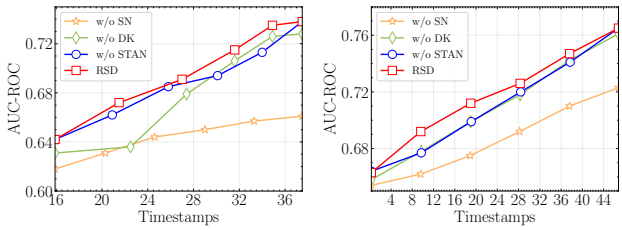
Furthermore, we evaluate our methods on all EHR encoders and present the result in Figure 3 (b) (d) (f). As we can see, the performances of all models have the same varying trend with decreasing advance rate. It further verifies the robustness of our approach.

4.4 Methods Analysis

Diagnosis module. In our approach, the diagnosis prediction module consists of two parts: Siamese network and domain knowledge regularization. Here, we check how each part actually contributes to the final performance. We construct the ablation study experiment based on the LSTM model on the three experiment datasets.

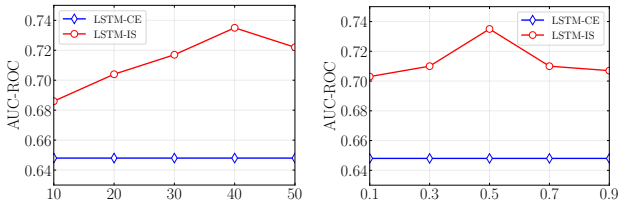
Table 4: Ablation study of the diagnosis module.

Methods	Diabetes		Hypertension		Mortality	
	Visit	Case	Visit	Case	Visit	Case
Complete	**0.684**	**0.735**	**0.745**	**0.811**	**0.694**	**0.728**
$\neg DK$	0.664	0.711	0.745	0.808	0.694	0.726
$\neg SN$	0.627	0.652	0.680	0.792	0.683	0.710
$\neg DK + SN$	0.626	0.648	0.679	0.791	0.681	0.710



(a) Diabetes dataset. (b) Mortality dataset.

Figure 4: Ablation study of the control module.



(a) Siamese network (Δ). (b) Diagnosis loss (α).

Figure 5: Performance tuning on diabetes dataset.

We compare three variants of the proposed RSD model: $\underline{\neg DK}$ removing the domain knowledge regularization; $\underline{\neg SN}$ removing the Siamese Network; $\underline{\neg DK + SN}$ removing both the two parts, which degenerates to the standard LSTM model. Table 4 presents the diagnosis prediction performance of the complete model and three variants. As we can see, the performance order of the four methods is as follows: $\neg DK + SN < \neg SN < \neg DK < Complete$. These results indicate that the two components are essential to improve the performance of our approach. Comparing the performance of $\neg DK$, $\neg SN$, we can find that the Siamese network is more useful for the final performance. A possible reason is that the effect of the Siamese network seems to be more significant on small datasets.

Control module. In addition, there are three parts that influence the control module: Siamese network, domain knowledge and the proposed STAN network. Here, we check how each part actually contributes to the final performance. We construct the ablation study experiment based on the LSTM model on the two experiment datasets. We compare three variants of the proposed RSD model: $\underline{\neg DK}$ removing the domain knowledge regularization; $\underline{\neg SN}$ removing the Siamese Network and replacing the proposed STAN with a fully connected layer that takes the state as input. $\underline{\neg STAN}$ replacing

Table 5: Online test on hypertension prediction task.

Methods	False Positive	False Negative	Diagnosis time
EARLIEST	0.238	0.652	26.1
RSD	**0.213**	**0.406**	**25.4**

the proposed STAN with a fully connected layer that takes the state and the mean encodings of health and diseased samples as input. Figure 4 presents the early diagnosis performance of the complete model and three variants. As we can see, the performance of the proposed RSD is better than $\neg DK$, $\neg SN$ and $\neg STAN$. These results indicate that the three components are essential to improve the performance of our approach.

Performance Tuning. In addition to the model components, there are several parameters to tune in our model. Here, we incorporate the LSTM trained with cross-entropy loss for comparison. We report the tuning results with AUC-ROC score of "*ByCase*" on the diabetes dataset. We tune the parameters of Siamese network (Δ in Eq. (7)) and diagnoses loss (α in Eq. (13)), respectively. We vary the threshold (Δ) of Siamese network in the set $\{10, 20, 30, 40, 50\}$, and the balancing parameter (α) of diagnosis loss in the set $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Figure 5 presents the performance of varying parameters the on diabetes dataset. As we can see, $\Delta = 40$ and $\alpha = 0.3$ lead to the optimal performance for the three datasets. Overall, our model is relatively stable when varying the two parameters, consistently better than LSTM trained with cross-entropy loss. These results have verified the robustness of our approach in our task.

Real World Deployment and Online Test. In this part, we deploy the proposed RSD in a hospital which is one of the largest maternal and child healthcare hospital in Beijing and test its online performance for gestational hypertension diagnosis which has a lower incidence rate. Specifically, for each patient, once new examination records are obtained, we input the whole visit sequence into the model. Once the model output Alert, we give the diagnosis results to the doctor. We calculate three metrics: false positive rate, false negative rate and average diagnosis time. And we incorporate the best baseline EARLIEST for comparison. Here, we collect 671 samples for test. Table 5 present the comparison between our method and EARLIEST. As we can see, our method is consistently better than EARLIEST. That indicates that the proposed RSD can accurately and early make Alert for the case that suffers from the disease. Based on this online test performance, we believe doctors and patients can benefit from RSD in various tasks.

5 RELATED WORK

Deep Learning for Diagnosis Prediction. Traditional machine learning models used in diagnoses prediction include machine learning techniques [12, 41], Convolutional Neural Networks (CNNs) [5] and Recurrent Neural Networks (RNNs) are widely adopted in diagnosis prediction applications, since most of electronic medical record data can be formed as a sequence of examination records of a patient at multiple visits to hospital. Dipole [29] employed the attention mechanism to capture the temporal dependencies. T-LSTM [2] and HiTANet [28] dealt with irregular times between the successive elements of sequential data. For interpretability, RETAIN [7] to retain the prediction accuracy of the RNN with better interpretation. And RetainVis [24] improved the RETAIN model by employing a bidirectional structure considering time decays. To address the data insufficiency problem, previous studies [6, 31] leveraged medical knowledge using graph-based attentions, MetaPred [50] employed meta-learning, and PRIME [30] leveraged prior medical knowledge. Furthermore, some studies [26, 36–38] proposed to use pre-training to help learn effective repesentations.

Early Classification. Our task can be considered as a special application in early classification of time series. The early classification is usually implemented through three approaches: MPL based approaches, shapelet based approaches, and model based approaches.

MPL based approaches [47, 48] usually learned a Minimum Prediction Length (MPL). These methods classified the testing cases when they were confident about the results of MPL. In addition, some researchers use posterior probabilities [34] to generate discriminative MPL for each label. Shapelet based approaches [10, 49] selected a set of shapelets by evaluating the quality of all cases in training dataset, and developed a distance threshold for sequence matching. They classified the testing cases when the matching degree between the testing sequence and the shapelet is above a preset threshold. Model based approaches generate a score according to different models for each subsequence to determine whether to make the classification. These approaches include joint-optimized methods and reinforcement learning methods. Joint-optimized methods [33, 39, 40] usually designed suitable loss functions to optimize earliness. As a comparison, reinforcement learning approaches [14, 32] automatically learned such a decision process, which are more inconsistent with real-world scenarios.

6 CONCLUSION

In this paper, we presented a reinforced Siamese network for early diagnosis of diseases, called RSD. We designed a diagnosis agent consisting of a diagnosis module and a control module. For the diagnosis module, we designed two improvement techniques, namely Siamese network training and domain knowledge guidance. For the control module, we implemented it under the reinforcement learning framework with carefully designed architecture, rewards and policies. By combining the two modules, our approach presents an effective solution to early diagnosis of diseases, which fully considers the challenges including insufficient training data, dynamic and complex signs of complications, and trade-off between earliness and accuracy. The experimental results on three real-world datasets have shown the superiority of our approach on early diagnosis of diseases against a number of competitive baselines. As future work, we will consider extending our approach to the diagnosis of other diseases. We believe the proposed solutions can be generalized or applied to more kinds of diagnosis tasks.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2019YFB2103203), the National Natural Science Foundation of China (Grant No. 82161148011, 72171013, 72222022), and the Fundamental Research Funds for the Central Universities (Grant No. YWF-22-L-838).

REFERENCES

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

[2] 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*. ACM, 65–74.

[3] Ling Chen, Xue Li, Quan Z. Sheng, Wen-Chih Peng, John Bennett, Hsiao-Yun Hu, and Nicole Huang. 2016. Mining Health Examination Records - A Graph-Based Approach. *IEEE Trans. Knowl. Data Eng.* 28, 9 (2016), 2423–2437.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 1597–1607. http://proceedings.mlr.press/v119/chen20j.html

[5] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. 2016. Risk Prediction with Electronic Health Records: A Deep Learning Approach. In *SDM*. SIAM, 432–440.

[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*. 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*. 3504–3512.

[8] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *CVPR (1)*. IEEE Computer Society, 539–546.

[9] Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior Regularization for Structured Latent Variable Models. *J. Mach. Learn. Res.* 11 (2010), 2001–2049.

[10] Mohamed F. Ghalwash, Vladan Radosavljevic, and Zoran Obradovic. 2014. Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In *KDD*. ACM, 402–411.

[11] Edward W Gregg, Qiuping Gu, Yiling J Cheng, KM Venkat Narayan, and Catherine C Cowie. 2007. Mortality trends in men and women with diabetes, 1971 to 2000. *Annals of internal medicine* 147, 3 (2007), 149–155.

[12] Shiying Hao, Karl G. Sylvester, Xuefeng Bruce Ling, Andrew Young Shin, Zhongkai Hu, Bo Jin, Chunqing Zhu, Dorothy Dai, Frank Stearns, Eric Widen, Devore S. Culver, Shaun T. Alfreds, and Todd Rogow. 2015. Risk prediction for future 6-month healthcare resource utilization in Maine. In *BIBM*. IEEE Computer Society, 863–866.

[13] Wendy W Harrison and Vladimir Yevseyenkov. 2015. Early interventions to prevent retinal vasculopathy in diabetes: a review. *Clinical Optometry* 7 (2015), 71.

[14] Thomas Hartvigsen, Cansu Sen, Xiangnan Kong, and Elke A. Rundensteiner. 2019. Adaptive-Halting Policy Network for Early Classification. In *KDD*. ACM, 101–110.

[15] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific Data* 6, 1 (2019), 96. https://doi.org/10.1038/s41597-019-0103-9

[16] Nima Hatami and Camelia Chira. 2013. Classifiers with a reject option for early time-series classification. In *Proceedings of the IEEE Symposium on Computational Intelligence and Ensemble Learning, CIEL 2013, IEEE Symposium Series on Computational Intelligence (SSCI), 16-19 April 2013, Singapore*. IEEE, 9–16. https://doi.org/10.1109/CIEL.2013.6613134

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. IEEE Computer Society, 770–778.

[18] Barbara Heude, Olivier Thiébaugeorges, Valérie Goua, Anne Forhan, Monique Kaminski, Bernard Foliguet, Michel Schweitzer, Guillaume Magnin, Marie-Aline Charles, EDEN Mother-Child Cohort Study Group, et al. 2012. Pre-pregnancy body mass index and weight gain during pregnancy: relations with gestational diabetes and Hypertension, and birth outcomes. *Maternal and child health journal* 16, 2 (2012), 355–363.

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

[20] Henry Hsu and Peter A Lachenbruch. 2014. Paired t test. *Wiley StatsRef: statistics reference online* (2014).

[21] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015 (JMLR Workshop and Conference Proceedings, Vol. 37)*, Francis R. Bach and David M. Blei (Eds.). JMLR.org, 448–456. http://proceedings.mlr.press/v37/ioffe15.html

[22] 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.

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

[24] Bum Chul Kwon, Min-Je Choi, Joanne Taery Kim, Edward Choi, Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo. 2019. RetainVis: Visual Analytics with Interpretable and Interactive Recurrent Neural Networks on Electronic Medical Records. *IEEE Trans. Vis. Comput. Graph.* 25, 1 (2019), 299–309.

[25] Zhaoqian Lan, Guopeng Zhou, Yichun Duan, and Wei Yan. 2018. AI-Assisted Prediction on Potential Health Risks with Regular Physical Examination Records. In *DSC*. IEEE, 346–352.

[26] 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.

[27] Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.

[28] 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*. ACM, 647–656.

[29] 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*. ACM, 1903–1911.

[30] 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*. ACM, 1910–1919.

[31] Fenglong Ma, Quanzeng You, Houping Xiao, Radha Chitta, Jing Zhou, and Jing Gao. 2018. KAME: Knowledge-based Attention Model for Diagnosis Prediction in Healthcare. In *CIKM*. ACM, 743–752.

[32] Coralie Martinez, Emmanuel Ramasso, Guillaume Perrin, and Michèle Rombaut. 2020. Adaptive early classification of temporal sequences using deep reinforcement learning. *Knowl. Based Syst.* 190 (2020), 105290.

[33] Usue Mori, Alexander Mendiburu, Sanjoy Dasgupta, and José Antonio Lozano. 2018. Early Classification of Time Series by Simultaneously Optimizing the Accuracy and Earliness. *IEEE Trans. Neural Networks Learn. Syst.* 29, 10 (2018), 4569–4578.

[34] Usue Mori, Alexander Mendiburu, Eamonn J. Keogh, and José Antonio Lozano. 2017. Reliable early classification of time series based on discriminating the classes over time. *Data Min. Knowl. Discov.* 31, 1 (2017), 233–263.

[35] World Health Organization et al. 2019. *World health statistics overview 2019: monitoring health for the SDGs, sustainable development goals*. Technical Report. World Health Organization.

[36] P. K. S. Prakash, Srinivas Chilukuri, Nikhil Ranade, and Shankar Viswanathan. 2021. RareBERT: Transformer Architecture for Rare Disease Patient Identification using Administrative Claims. In *AAA21*. AAAI Press, 453–460.

[37] Houxing Ren, Jingyuan Wang, and Wayne Xin Zhao. 2022. Generative Adversarial Networks Enhanced Pre-Training for Insufficient Electronic Health Records Modeling. In *KDD*. ACM, 3810–3818.

[38] 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*. ACM, 3503–3511.

[39] Marc Rußwurm, Sébastien Lefèvre, Nicolas Courty, Rémi Emonet, Marco Körner, and Romain Tavenard. 2019. End-to-end Learning for Early Classification of Time Series. *CoRR* abs/1901.10681 (2019).

[40] Patrick Schäfer and Ulf Leser. 2020. TEASER: early and accurate time series classification. *Data Min. Knowl. Discov.* 34, 5 (2020), 1336–1362.

[41] Akihiro Shimoda, Daisuke Ichikawa, and Hiroshi Oyama. 2018. Using machine-learning approaches to predict non-participation in a nationwide general health check-up scheme. *Comput. Methods Programs Biomed.* 163 (2018), 39–46.

[42] Baha M Sibai. 2003. Diagnosis and management of gestational Hypertension and preeclampsia. *Obstetrics & Gynecology* 102, 1 (2003), 181–192.

[43] Priya Soma-Pillay, Nelson-Piercy Catherine, Heli Tolppanen, Alexandre Mebazaa, Heli Tolppanen, and Alexandre Mebazaa. 2016. Physiological changes in pregnancy. *Cardiovascular journal of Africa* 27, 2 (2016), 89.

[44] Karen Tu, Zhongliang Chen, Lorraine L Lipscombe, et al. 2008. Mortality among patients with hypertension from 1995 to 2005: a population-based study. *Cmaj* 178, 11 (2008), 1436–1440.

[45] 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*. 5998–6008.

[46] Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* 8 (1992), 229–256.

[47] Zhengzheng Xing, Jian Pei, Guozhu Dong, and Philip S. Yu. 2008. Mining Sequence Classifiers for Early Prediction. In *SDM*. SIAM, 644–655.

[48] Zhengzheng Xing, Jian Pei, and Philip S. Yu. 2009. Early Prediction on Time Series: A Nearest Neighbor Approach. In *IJCAI*. 1297–1302.

[49] Liuyi Yao, Yaliang Li, Yezheng Li, Hengtong Zhang, Mengdi Huai, Jing Gao, and Aidong Zhang. 2019. DTEC: Distance Transformation Based Early Time Series Classification. In *SDM*. SIAM, 486–494.

[50] 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*. ACM, 2487–2495.