HiTANet: Hierarchical Time-Aware Attention Networks for Risk Prediction on Electronic Health Records

Junyu Luo*
Penn State University
junyu@psu.edu

Muchao Ye
Penn State University
muchao@psu.edu

Cao Xiao
IQVIA
cao.xiao@iqvia.com

Fenglong Ma†
Penn State University
fenglong@psu.edu

ABSTRACT

Deep learning methods especially recurrent neural network based models have demonstrated early success in disease risk prediction on longitudinal patient data. Existing works follow a strong assumption to implicitly assume the stationary disease progression during each time period, and thus, take a homogeneous way to decay the information from previous time steps for all patients. However, in reality, disease progression is non-stationary. Besides, the key time steps for a target disease vary among patients. To leverage time information for risk prediction in a more reasonable way, we propose a new hierarchical time-aware attention network, named HiTANet, which imitates the decision making process of doctors in risk prediction. Particularly, HiTANet models time information in local and global stages. The local evaluation stage has a time-aware Transformer that embeds time information into visit-level embedding and generates local attention weight for each visit. The global synthesis stage further adopts a time-aware key-query attention mechanism to assign global weights to different time steps. Finally, the two types of attention weights are dynamically combined to generate the patient representations for further risk prediction. We evaluate HiTANet on three real-world datasets. Compared with the best results among twelve competing baselines, HiTANet achieves over 7% in terms of F1 score on all datasets, which demonstrates the effectiveness of the proposed model and the necessity of modeling time information in risk prediction task.

KEYWORDS

Risk prediction, healthcare informatics, attention mechanism, transformer

1 INTRODUCTION

With the collection of massive electronic health records (EHRs) data, deep learning models, especially recurrent neural networks (RNNs), have demonstrated early successes in risk prediction tasks, which use historical EHR data to forecast the future health status of patients. RNN-based models are impressively powerful in modeling complex EHR data, especially for patients with chronic and progressing conditions such as heart diseases and Parkinson's disease. Most existing studies focus on extracting temporal disease progression patterns from longitudinal patient data [3, 29, 32]. For example, Pham *et al.* [32] combined RNN and multi-scale pooling to integrate temporal disease patterns from different time scales. Baytas *et al.* [3] and Ma *et al.* [29] simulated progression of patients' status by using temporal information to decay the information collected from patients' historical visits.

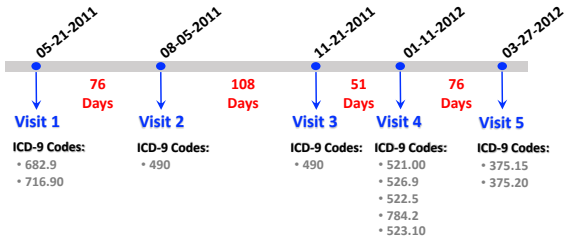

Figure 1: An example of time-ordered patient EHR data that includes five visits. Each visit records a set of diagnosis codes

However, existing studies implicitly assume the stationary progression during each time period, and thus take a homogeneous way to decay the information from previous time steps for all patients in their models. This assumption does not leverage time information for risk prediction in a reasonable way, which leaves two open challenges to be solved.

C1. The importance of historical patient information with respect to current health risk does not decay monotonically. First, disease progression is non-stationary, which can be faster, more slower or even recurrent. The example in Figure 1 shows that the diagnoses of the second and the third visits are the same despite a big time interval between them. This indicates that the patient's health information stays almost unchanged between these two visits, and thus previous information should not be decayed. Moreover, the importance of diagnosis associated with each visit should not be decayed in a monotonic way.

C2. The importance of previous timestamps varies among patients. When assessing patient risks, doctors will first learn about the current health status of the patient, such as symptoms, preconditions and their duration. Based on these information, doctors can roughly infer some key timestamps related to disease progression. Due to the differences among patients, these important timestamps for different patients should be varied. However, this process is ignored by existing work.

*Also with Sichuan University.
†Corresponding author.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403107

To tackle the two challenges, we design a new **Hi**erarchical **T**ime-aware **A**ttention **Net**work (shorten for HiTANet). HiTANet consists of two stages, namely the local evaluation stage and the global synthesis stage. The *local evaluation stage* deals with the challenge **C1** and has a **time-aware Transformer** that can learn time-aware attention weights for individual visits. It combines time information and visit information and learns *local attention weights* for visits and a overall representation for each patient. Via embedding time information into vectors, it effectively avoids the drawbacks introduced by a monotonic time decay function. For solving the challenge **C2**, we design a *global synthesis stage*, where we propose the **time-aware key-query attention mechanism**. It utilizes the overall representation outputted by the time-aware Transformer as the query vector, and generates the *global attention weight* of each visit from the global view by using the temporal information again. Finally, we fuse these two attention mechanisms together to obtain patient representations for disease risk prediction. Compared with state-of-the-art approaches, HiTANet not only utilizes time information in both local and global levels but also remains as a flexible yet robust approach for risk prediction.

The proposed HiTANet has the following technical contributions:

- We design a time-aware Transformer to embed time information. It first embeds time information into visit representations instead of introducing a monotonic time decay function, and then learns a local attention weight for each visit.
- We propose a new time-aware key-query attention mechanism to identify the key time steps among patients' historical visits in their EHR data. It uses a patient's overall representation as a query vector and the time embedding of each visit as key vector.
- We conduct experiments on three real-world datasets to show the effectiveness of the proposed HiTANet. Ablation studies and model analysis confirm the reasonableness and interpretability of the proposed model.

2 RELATED WORK

In recent years, various deep learning models [7, 44, 45] have been proposed for risk prediction on EHR data. In addition to multi-layer perception [8, 12] or convolutional neural networks [10, 39], RNNs are the most widely used architecture due to their expressive power in capturing temporal patterns [3, 13, 14, 23, 24, 29, 37, 38]. We review these studies from three perspectives.

Attention Mechanisms for Risk Prediction Attention-based models aim to learn a weight for each visit and obtain the patient representation by conducting a weighted sum operation. Retain [14] is an interpretable model for risk prediction, which not only learns visit-level weights but also assigns a weight for each diagnosis code within a visit. Retain uses two RNNs to learn the weights separately. Though Retain provides explanations for predictions, there exists a balance between performance and interpretability. Thus, the performance of Retain may be not better than other models with attention mechanisms, such as Dipole [23]. Dipole tries to model longitudinal EHR data using bidirectional RNNs as well as employ three attention mechanisms. Bidirectional RNNs can learn satisfactory representations for patients from both directions and guarantee achieving good performance. Different from using transitional attention approaches [22], some studies such as SAnD [35] apply self-attention [40] to improve the prediction performance. One benefit of applying the self-attention mechanism is to use contextual visits to generate hidden states, which avoids the drawback of RNN-based models. In the proposed HiTANet, we also employ self-attention mechanism to learn visit representations.

Time-aware Models for Risk Prediction Most RNN-based methods focus on modeling the sequential characteristics of EHR data. However, EHR data are not only sequential but also temporal because each visit is associated with time information, which is highly related to prediction tasks. To take time information into consideration, T-LSTM [3] is proposed, which assumes that the patient information may decay if there is a time interval between two consecutive visits. T-LSTM uses a information decay function working with modified gates of LSTM (Long short-term memory networks) [17] to make risk predictions. Based on this assumption, RetainEX [19], TimeLine [2], and ConCare [27] are proposed. RetainEX [19] is built upon Retain by considering information decay, which also uses traditional attention mechanisms to learn weights for visits and diagnosis codes. Both TimeLine [2] and ConCare [27] apply self-attention mechanisms to improve the performance, as well as add the information decay function when learning patient representations. However, as we discussed in Introduction (see Figure 1), the information may not only decay in a monotonic way. Thus, we propose a new solution to model the importance of time information in HiTANet.

External Knowledge for Risk Prediction Some studies try to improve the performance of the risk prediction task by incorporating external information, such as medical knowledge graph, prior medical knowledge, or data from other related tasks. Medical knowledge graph contains the relationships between the target disease and other diseases, which is used in [13, 15, 25, 26, 42, 43]. Beside, prior knowledge provided by doctors and authoritative sources is another kind of useful information for the improvement of performance. Ma *et al.* [24] propose a risk prediction framework to model different disease-related knowledge with a log-linear model. Even with enough medical knowledge, if the number of patient data is limited, we still cannot obtain a satisfactory prediction model. To hand the issue of small data, MetaPred [45] is proposed, which aims to transfer domain knowledge from other related prediction task. Different from the work in this category, we do not incorporate any external information to further improve the performance.

3 TASK DEFINITION

In longitudinal EHR data, each patient data can be considered as a time-ordered visit sequence, and within each visit, there are several diagnosis codes.

Definition 1 (Diagnosis Codes). Let $C = \{c_1, c_2, \cdots, c_N\}$ denote all unique diagnosis codes, and let c_* abstractly represent the whole patient data, which is appended to the end of each patient data.

Definition 2 (Binary EHR Data). Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T, \mathbf{x}_*]$ represent a patient's visit information, where the t-th visit $\mathbf{x}_t \in \{0, 1\}^{N+1}$ is a binary vector, and $\mathbf{x}_* \in \mathbb{R}^{N+1}$ is an one-hot vector. If the i-th diagnosis code $c_i \in \{c_1, \cdots, c_N\}$ appears in the t-th visit, then $\mathbf{x}_{ti} = 1$, otherwise $\mathbf{x}_{ti} = 0$. $\mathbf{x}_*$ only contains the special code c_*. For all the patient data, they all have the same $\mathbf{x}_*$.

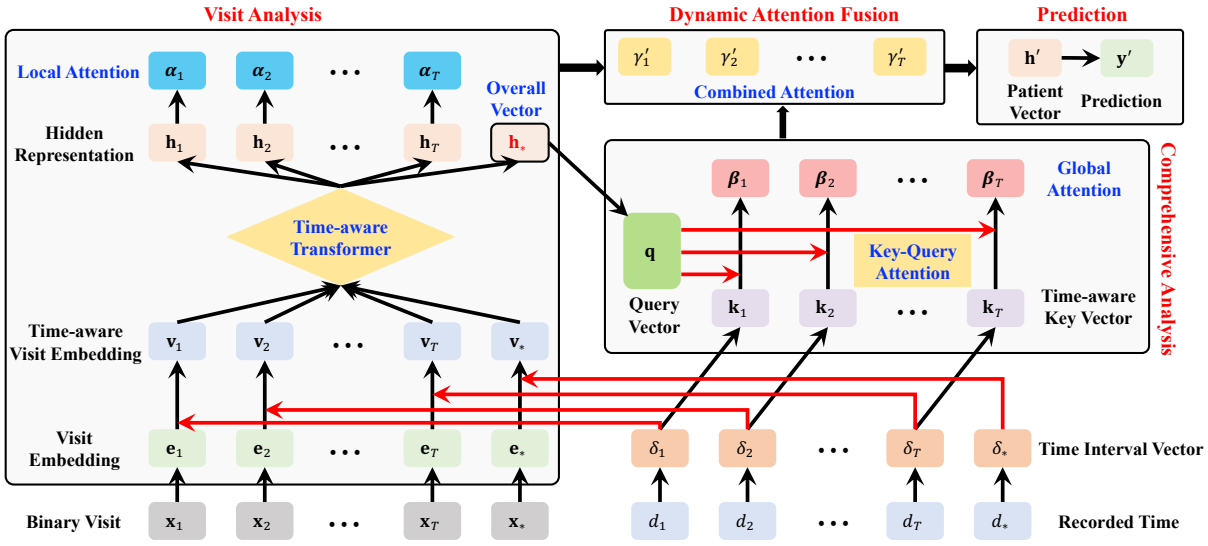

Figure 2: The HiTANet Model. The model consists of three major components. The first component is the local level visit analysis. A time-aware Transformer is used to model each EHR visit and generate the hidden state of each visit, which is then used to produce the local attention weight. The second key component is the global level comprehensive analysis. An overall diagnosis representation is used as a query vector, and the embedded time information is considered as key vectors. A time-aware key-query attention is applied to generate global attention weights. The dynamic attention fusion component is used to combine these two attention weights. Finally, a new patient representation is obtained based on combined attention weights and hidden representations, which is further used to make a prediction.

Definition 3 (Time Interval). Let d_t denote the corresponding time information of the visit $\mathbf{x}_t$. For the special visit $\mathbf{x}_*$, let $d_* = d_T$. Let $\delta_t = d_T - d_t$ represent the interval (in days) between the last visit and the t-th visit.

Problem 1 (Risk Prediction). Given a patient visit data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T, \mathbf{x}_*]$ and the time vector $\mathbf{\Delta} = [\delta_1, \delta_2, \cdots, \delta_T, \delta_*]$, the goal of risk prediction task is to forecast whether the patient will suffer the target disease in the further.

4 THE PROPOSED HITANET MODEL

This section presents our HiTANet model (Figure 2). HiTANet is designed as a hierarchical structure that comprises three key components: visit analysis, comprehensive analysis, and time-aware dynamic attention fusion. (1) Visit Analysis. For the t-th visit, HiTANet learns a vector representation $\mathbf{v}_t$ by considering both corresponding diagnosis code $\mathbf{x}_t$ and time interval δ_t. By employing the basic Transformer [40], HiTANet can learn a hidden state $\mathbf{h}_t$ based on $[\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_T, \mathbf{v}_*]$, and then generate a local-level attention score α_t using $\mathbf{h}_t$ except for the special visit $\mathbf{x}_*$. (2) Comprehensive Analysis. To model the disease progression with time changes, we first embed $\mathbf{h}_*$ (i.e., the representation of the abstract whole patient data $\mathbf{x}_*$) into a "query" vector $\mathbf{q}$, and then embed each time interval δ_t into a "key" vector $\mathbf{k}_t$. Thus, HiTANet can produce a global-level attention score β_t for each visit $\mathbf{x}_t$ using the designed key-query attention. (3) Time-aware Dynamic Attention Fusion. To obtain an overall attention score γ'_t for each visit, HiTANet takes both attention scores α_t and β_t and the abstract representation $\mathbf{h}_*$ into consideration. According to the learned overall attention score γ'_t, HiTANet generates the final presentation $\mathbf{h}'$ to be used in prediction.

4.1 Local Level: Visit Analysis

Given a sparse binary visit vector $\mathbf{x}_t$, we first encode it to a relatively dense space $\mathbf{e}_t \in \mathbb{R}^m$ using a linear function as follows:

$$\mathbf{e}_t = \mathbf{W}_e \mathbf{x}_t + \mathbf{b}_e, \tag{1}$$

where $\mathbf{W}_e \in \mathbb{R}^{m \times (N+1)}$ is the weight matrix and $\mathbf{b}_e \in \mathbb{R}^m$ is the bias vector. Thus, the data of each patient can be represented by $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_T, \mathbf{e}_*]$. Though most state-of-the-art risk prediction models [2, 3, 9, 19, 28, 32] are built upon RNNs that take $\mathbf{E}$ as the inputs and can achieve good performance, the interactions among different visits are all calculated in a black box. To explicitly model those interactions, we propose to use the Transformer structure [40]. The benefits of employing Transformer are two-fold. On the one hand, Transformer allows that each visit interacts with the remaining ones using the self-attention mechanism. Compared with RNN-based models, it largely reduces the important information decay. On the other hand, the structure of Transformer provides us a interpretable way for the visit fusion. Besides, it can successfully persevere the independence of each visit.

However, existing Transformer-based models are mainly used for tasks in natural language processing [4], and only a few studies investigate the use of Transformer in healthcare domain [20, 34]. But all of these models ignore the importance of time information, which is a key clue for disease progression. For example, there is

an ICD-9 diagnosis code 786.05 (shortness of breath)[1] appearing in a patient's visit. After a few days, another code 427.9 (Cardiac dysrhythmia)[2] also exists in a patient's visit, which indicates that the patient may have a risk to suffer heart failure disease and the illness becomes worse. Without modeling time information, it is hard for existing models to capture the progression of diseases with time changes. Thus, taking time information into considering when modeling EHR data is essential.

Though there are several RNN-based models [2, 3, 19] considering the importance of time information, they all implicitly assume that diseases have stationary decay. A information decay function is used to learn with RNN-based models. In reality, this assumption may not always correct. For some chronic diseases, the progression may be very slow, and the interval of two consecutive visits may be more than one year. If the two visits contain similar diagnosis codes, the doctor then knows that the disease does not get worse. In such a case, the information should not be decayed too much. To address these issues, we propose a novel time-aware Transformer, which embeds time information first and then takes the time vectors as a part of the inputs.

Specifically, we combine time vector Δ and visit vector $\mathbf{E}$ as the input of the proposed time-aware Transformer. However, Δ and $\mathbf{E}$ are not in the same latent space. Thus, we need to embed the time vector Δ to the latent visit space as follows:

$$\begin{aligned} \mathbf{f}_t &= \mathbf{1} - \tanh((\mathbf{W}_f \frac{\delta_t}{180} + \mathbf{b}_f)^2), \\ \mathbf{r}_t &= \mathbf{W}_r \mathbf{f}_t + \mathbf{b}_r, \end{aligned} \tag{2}$$

where $\mathbf{W}_f \in \mathbb{R}^a$, $\mathbf{b}_f \in \mathbb{R}^a$, $\mathbf{W}_r \in \mathbb{R}^{m \times a}$, and $\mathbf{b}_r \in \mathbb{R}^m$ are all parameters. A common assumption for risk prediction task is that the more recent visits, the more important. Thus, the visits close to the last one should be activated. To achieve this goal, we use the square operation, which is the element-wise square. Only when the activation of $\mathbf{W}_f$ and $\mathbf{b}_f$ is close to zero, the corresponding position will be activated. In such a way, different positions of $\mathbf{f}_t$ can represent different preference of temporal distances. The advantage of using such an operation is to prevent the influence of values far from 0, i.e., δ_T. Using the embedded time vector $\mathbf{r}_t \in \mathbb{R}^m$, we can obtain the input vector of the designed time-aware Transformer, which is $\mathbf{v}_t = \mathbf{e}_t + \mathbf{r}_t$.

Given the input matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_T, \mathbf{v}_*]$, a standard one-layer Transformer (denoted as F) is applied to learn the long-term dependencies among each visit with the emphasis on time information:

$$[\mathbf{h}_1, \mathbf{h}_2 \cdots, \mathbf{h}_T, \mathbf{h}_*] = F([\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_T, \mathbf{v}_*]), \tag{3}$$

where $\mathbf{h}_t \in \mathbb{R}^l$ is the hidden representation for each visit by aggregating all the other visit information with self-attention mechanism in Transformer[3].

When doctors diagnose, instead of only focusing on current visit, they will review historical medical records and search for the ones that are highly related to the target disease. To simulate such a diagnosis procedure, we calculate an attention score η_t for each visit (except for $\mathbf{h}_*$) using local-based attention mechanism [23].

$$\eta_t = \mathbf{W}_\eta^\top \mathbf{h}_t + b_\eta,$$

where $\mathbf{W}_\eta \in \mathbb{R}^l$ and $b_\eta \in \mathbb{R}$ are parameters to be learned. After obtaining an attention vector $\boldsymbol{\eta} = [\eta_1, \eta_2, \cdots, \eta_T]$, a softmax layer is employed to generate *local attention weights*, i.e.,

$$\boldsymbol{\alpha} = \text{Softmax}(\boldsymbol{\eta}) = [\alpha_1, \alpha_2, \cdots, \alpha_T]. \tag{4}$$

4.2 Global Level: Comprehensive Analysis

Using the designed time-aware Transformer, we can learn a local attention weight for each visit, which reflects the importance of each individual visit. In fact, doctors focus on not only individual visits but also the disease progression by analyzing the overall diagnosis (i.e., $\mathbf{x}_*$) to make the final judgement. To imitate this step, we propose a novel time-aware key-query attention mechanism.

Since $\mathbf{h}_*$ obtained by Eq. (3) represents the latent state of the overall diagnosis, we first convert $\mathbf{h}_*$ as a query vector $\mathbf{q} \in \mathbb{R}^s$:

$$\mathbf{q} = \text{ReLU}(\mathbf{W}_q \mathbf{h}_* + \mathbf{b}_q), \tag{5}$$

where $\mathbf{W}_q \in \mathbb{R}^{s \times l}$ and $\mathbf{b}_q \in \mathbb{R}^s$ are parameters. Here, we use a nonlinear activation function ReLU to only keep the positive values. Compared with negative values, these positive ones may be more valuable to summarize the characteristics of the overall diagnosis.

When analyzing the overall diagnosis, doctors also want to know which time points are vital for the disease. To this end, we embed each time information δ_t into a latent space as follows:

$$\begin{aligned} \mathbf{o}_t &= \mathbf{1} - \tanh((\mathbf{W}_o \frac{\delta_t}{180} + \mathbf{b}_o)^2), \\ \mathbf{k}_t &= \tanh(\mathbf{W}_k \mathbf{o}_t + \mathbf{b}_k), \end{aligned} \tag{6}$$

where $\mathbf{W}_o \in \mathbb{R}^n$, $\mathbf{b}_o \in \mathbb{R}^n$, $\mathbf{W}_k \in \mathbb{R}^{s \times n}$, and $\mathbf{b}_k \in \mathbb{R}^s$ are parameters, and $\mathbf{k}_t \in \mathbb{R}^s$ is called time-aware key vector. Eq. (6) is similar to Eq. (2) but is different in target. The purpose of activation form is the same. Eq. (2) focuses on capturing the importance of diagnosis codes appearing associated with the time information. However, Eq. (6) tries to characterize the importance of time information itself during the disease progression without considering any diagnosis codes. Besides, we use the ReLU activation function to keep the key information introduced by the positive values.

To learn the significance of each time interval during the risk prediction, we put key and query vectors together and calculate the attention scores. Following the key-query attention mechanism in Transformer [40], we can obtain an attention weight as follows:

$$\phi_t = \frac{\mathbf{q}^\top \mathbf{k}_t}{\sqrt{s}}. \tag{7}$$

We then apply a softmax layer to normalize the attention weights, and finally, the *global attention weights* can be represented by:

$$\boldsymbol{\beta} = \text{Softmax}(\boldsymbol{\phi}) = [\beta_1, \beta_2, \cdots, \beta_T]. \tag{8}$$

4.3 Time-aware Dynamic Attention Fusion

Through analyzing different visits and the overall diagnosis, we obtain two attention vectors: the local attention vector $\boldsymbol{\alpha}$ that pays attention to each visit representation and the global attention vector $\boldsymbol{\beta}$ that focuses on each time representation. The local attention mechanism can be seen as using a "forward" operation to imitate

[1] http://www.icd9data.com/2015/Volume1/780-799/780-789/786/786.05.htm
[2] http://www.icd9data.com/2015/Volume1/390-459/420-429/427/427.9.htm
[3] The details of Transformer is introduced in Appendix A.

doctors' diagnosis procedure, and the global attention mechanism is similar to a "backward" operation, which is retrospectively analyze the importance of time information. Since they focus on different perspectives for predicting the risk of diseases, we need to consider both of them together. Besides, to capture the preference of visit representation and time representation for different cases, we propose a dynamic attention fusion mechanism. In particular, we first embed the overall representation $\mathbf{h}_*$ into a new space and then normalize it with a softmax layer as follows:

$$\mathbf{z} = \text{Softmax}(\mathbf{W}_z\mathbf{h}_* + \mathbf{b}_z) = [z_\alpha, z_\beta], \tag{9}$$

where $\mathbf{W}_z \in \mathbb{R}^{2\times l}$ and $\mathbf{b}_z \in \mathbb{R}^2$ are parameters. We then generate an overall attention weight for each visit based on both attention weights and the embedded overall representation $\mathbf{z}$ as follows:

$$\gamma_t = \alpha_t * z_\alpha + \beta_t * z_\beta. \tag{10}$$

Finally, we normalize the fused attention weights and obtained the final attention score γ'_t for each visit as follows:

$$\gamma'_t = \frac{\gamma_t}{\sum_{j=1}^{T}\gamma_j}. \tag{11}$$

4.4 Prediction

Based on the generated attention weights and the hidden state of each visit, we can finally obtain the representation of a patient data as follows:

$$\mathbf{h}' = \sum_{t=1}^{T}\gamma'_t\mathbf{h}_t. \tag{12}$$

A simple linear layer with a softmax layer can be used to make a binary prediction as follows:

$$\mathbf{y}' = \text{Softmax}(\mathbf{W}_u\mathbf{h}' + \mathbf{b}_u), \tag{13}$$

where $\mathbf{W}_u \in \mathbb{R}^{2\times l}$ and $\mathbf{b}_u \in \mathbb{R}^2$ are parameters. Let $\boldsymbol{\theta}$ represent all the parameters and $\mathbf{y}$ denote the ground truth. The cross-entropy between the ground truth $\mathbf{y}$ and the prediction probabilities $\mathbf{y}'$ is used to calculate the loss. Thus, the objective function of risk prediction is the average of cross-entropy:

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{|\mathcal{P}|}\sum_{p=1}^{|\mathcal{P}|}\left(\mathbf{y}_p^\top \log(\mathbf{y}'_p) + (\mathbf{1}-\mathbf{y}_p)^\top \log(\mathbf{1}-\mathbf{y}'_p)\right), \tag{14}$$

where $|\mathcal{P}|$ is the total number of patient data. Algorithm 1 describes the overall training procedure of the proposed HiTANet.

Algorithm 1 Training Procedure

Input: Training set $\mathcal{D}_t$, and validation set $\mathcal{D}_v$
Output: Trained model parameter $\boldsymbol{\theta}_{best}$

```
Randomly initialize the parameter θ of HiTANet;
for epoch = 1 to EPOCH do
    Randomly initialize the sample order of training set D_t;
    for (X, Δ, y) ∈ D_t do
        Obtain dense embeddings e according to Eq. (1);
        Obtain time embeddings r according to Eq. (2);
        Calculate the hybrid input v = e + r;
        Encode v to obtain h using transformer F according to Eq. (3);
        Calculate the local attention vector α using Eq. (4);
        Calculate the global attention vector β using Eq. (5)-(8);
        Calculate the final attention γ' using Eq. (9)-(11);
        Calculate prediction y' using Eq. (12)-(13);
        Calculate the prediction loss L using Eq. (14);
        Update parameters θ according to the gradient of L;
    end for
    Calculate the average validation loss L_v on validation set D_v.
    if L_v < L_v^min then
        θ_best = θ;
        L_v^min = L_v;
    end if
end for
```

5 EXPERIMENTS

In this section, we first introduce the experimental settings, including data, baselines, and implementation details, and then discuss the experimental results, as well as detailed analysis to demonstrate the effectiveness of HiTANet [4].

5.1 Experimental Setup

Datasets. We consider three disease cohorts extracted from a real world EHR database: Chronic Obstructive Pulmonary Disease (COPD), Heart Failure and Kidney Disease. The data statistics are listed in Table 1. We formulate the risk prediction task as a binary classification problem to predict whether a patient has a specific disease onset. For each dataset, we first select a set of optional case patients according to the general medical diagnosis guidelines from doctors, and then with the help of domain experts, we further confirm whether the patients really suffer from these diseases. For each positive case patient, we track back from the date of confirmation of disease and hold off the visits within the prediction window (180 days). Finally, we use the remaining visits before the prediction window as input data. For each negative control patient, we hold off the last one year's visits and use the remaining visits as the input data. The max length of each patient's record is set to 50 visits.

Table 1: Dataset Details.

Dataset	COPD	Heart Failure	Kidney Disease
Case (Positive)	7,314	3,080	2,810
Control (Negative)	21,942	9,240	8,430
Avg visits per patient	30.39	38.74	39.09
Avg codes per visit	3.50	4.24	4.40
Unique ICD-9 codes	10,053	8,692	8,802

Baseline Methods. We evaluate HiTANet against state-of-the-art models in the following categories.

(1) **Traditional methods** considered here include support vector machine (SVM) [41], Linear Regression (LR) [33], and Random Forest (RF) [21]. They serve as the fundamental foot-stones for comparison.
(2) **Plain RNNs** include Long Short-Term Memory (LSTM) [17] and Gate Recurrent Unit (GRU) [11]. They are the basic frameworks of most risk prediction models.
(3) **Attention-based Models** include the four approaches below. Dipole− [23] uses a GRU as the backbone and assigns an attention weight for each visit. Dipole [23] is developed upon Dipole−, which applies the bidirectional GRU as the backbone.

[4] Code is available at https://github.com/HiTANet2020/HiTANet

Table 2: Average Performance on Three Disease Prediction Tasks

Method		COPD					Heart Failure					Kidney Diseases				
		Acc	Pre	Recall	F1	Auc	Acc	Pre	Recall	F1	Auc	Acc	Pre	Recall	F1	Auc
Classical Methods	SVM	0.804	0.713	0.319	0.441	0.639	0.784	**0.757**	0.327	0.457	0.644	0.840	**0.777**	0.545	0.641	0.745
	LR	0.678	0.328	0.319	0.324	0.556	0.716	0.489	0.466	0.477	0.639	0.772	0.558	0.636	0.594	0.728
	RF	0.798	0.664	0.334	0.444	0.640	0.779	0.746	0.310	0.438	0.635	0.819	0.758	0.452	0.567	0.701
Plain RNNs	LSTM	0.807	0.680	0.461	0.548	0.693	0.812	0.640	0.510	0.561	0.708	0.823	0.680	0.572	0.616	0.739
	GRU	0.820	0.694	0.462	0.553	0.698	0.794	0.679	0.490	0.567	0.700	0.818	0.678	0.591	0.629	0.745
Attention-based Models	Dipole–	0.818	0.699	0.440	0.538	0.690	0.795	0.689	0.481	0.565	0.698	0.826	0.679	0.635	0.656	0.764
	Dipole	0.821	0.687	0.477	0.562	0.704	0.794	0.713	0.445	0.542	0.687	0.843	0.771	0.571	0.656	0.755
	Retain	0.821	0.696	0.463	0.555	0.699	0.784	0.655	0.474	0.549	0.689	0.821	0.706	0.544	0.614	0.732
	SAnD	0.810	0.653	0.462	0.539	0.692	0.785	0.661	0.466	0.544	0.686	0.823	0.690	0.592	0.636	0.748
Time-based Models	RetainEx	0.829	**0.728**	0.470	0.570	0.707	0.799	0.730	0.438	0.546	0.688	0.827	0.745	0.520	0.612	0.728
	T-LSTM	0.818	0.687	0.525	0.595	0.722	**0.831**	0.695	0.527	0.598	0.727	0.832	0.728	0.524	0.608	0.729
	TimeLine	0.812	0.654	0.478	0.550	0.698	0.792	0.661	0.510	0.574	0.705	0.827	0.697	0.607	0.648	0.756
Ours	HiTANet	**0.840**	0.707	**0.583**	**0.637**	**0.752**	0.823	0.724	**0.587**	**0.647**	**0.750**	**0.851**	0.743	**0.668**	**0.702**	**0.792**
	(*std*)	0.002	0.024	0.030	0.009	0.009	0.001	0.023	0.033	0.013	0.010	0.005	0.014	0.027	0.015	0.011

Also, we select Retain[5] [14], an interpretable LSTM based model with attention improvement, as a baseline. SAnD [35] borrows the main idea of Transformer and uses the hierarchical aggregation mechanism.

(4) **Time-based Models**. We use three time-based models as baselines. ReatinEx[6] [19] is an improved version of Retain, which uses a bidirectional structure and considers time decays between two visits. T-LSTM[7] [3] is an improved LSTM based approach by modifying gate information to model the time decay. TimeLine [2] is also a Transformer-based interpretable deep learning model with time decaying for each visit.

Note that we do not compare the proposed HiTANet with PRIME [24] and ConCare [28] in the experiments. PRIME extracts a prior knowledge vector for each patient as auxiliary information to improve the performance, which is different from our setting. For ConCare, it requires an RNN encoder for each input code, which leads to high computation complexity, since the number of input codes is around 10,000 for each dataset as shown in Table 1. Thus, they are not considered as baselines.

Metrics. We used Accuracy (Acc), Precision (Pre), Recall, F1, and Area Under Curve (Auc) scores in evaluation. Precision can reflect the false alarm rate, while Recall can reflect the rate of missing report, and F1 is the average score of Precision and Recall. Auc is a popular comprehensive score for binary classifier.

Evaluation Strategy. For all methods, we randomly partition the whole datasets into three parts, training data, validation data, and testing data, in a ratio 0.75:0.10:0.15. We fix the best model on the validation set and report the performance on the test set. We perform five random runs and report both mean and standard deviation for testing performance.

Implementation Details. For all the deep learning-based models, we implement them in PyTorch [30] and train them on an Ubuntu 16.04 with 64GB memory and a Tesla V100 GPU. For traditional machine learning methods implemented with scikit-learn [31], we select the top 256 frequent diagnosis codes and use their frequency to represent the whole EHR record. Batch size is set to 50 for all the methods. The dimension of the final hidden state for prediction is set to 256, i.e., $l = 256$. The layer of RNN or Transformer is set to 1 for all the methods, unless there is a hierarchical structure. Dropout [36] methods are used for all the models in the final prediction layer, unless there is a default setting. The dropout rate is set to 0.5. Adam [18] optimizer is used for all the methods. For learning rate, we use grid search approach to select the best one for each method according to validation set. For the proposed HiTANet, the learning rate is set to 1e-4, $m = 256$, $a = 64$, $s = 64$, and $n = 64$. For the hyper-parameters of Transformer, we set the dimension size of attention embedding as 64, the multi-head number as 4, and the size of middle feed-forward network as 1024.

5.2 Experimental Results

We report the average performance of the proposed HiTANet model and other baseline models on three datasets, including COPD, Heart Failure and Kidney Diseases in Table 2 [8], where *std* represents the value of standard deviation. As we can observe from Table 2, HiTANet shows stable and outstanding performance and achieves the state-of-the-art scores on most metrics.

We first zoom into the classic machine learning methods, including SVM, LR and RF. Compared with deep learning methods, the overall performance of classic methods is lower. The reason is that they lack the sequence modeling ability and cannot distinguish each visit. On the COPD and Heart Failure datasets, the scores of F1 and Auc are 10% lower compared with deep learning baselines like LSTM and GRU. However, on the Kidney Disease dataset, the classic methods perform equally or slightly better than many deep learning-based methods. One possible explanation is that, for kidney disease, the individual signal may have a more important role. Models like recurrent neural networks may have over-fitting problem in this situation. HiTANet is designed to keep the independence of each visit, which can further avoid this issue.

[5] https://github.com/easyfan327/Pytorch-RETAIN
[6] https://github.com/minjechoi/RetainVis
[7] https://github.com/illidanlab/T-LSTM

[8] The experimental results of two additional datasets are listed in Appendix B.

As plain deep learning models, the performance of LSTM and GRU models is stable but not outstanding. However, on the Heart Failure dataset, they perform better than several attention-based models. The reason is that heart failure disease is a chronic disease, and there are several risk factors. Even plain recurrent neural networks, they can easy capture those important characteristics. However, for other two datasets, they do not perform as good as on the Heart Failure dataset.

Compared with basic RNN methods, the improvement of all attention-based models, including Retain, Dipole−, Dipole, and SAnD, is significant except for the results on the Heart Failure dataset. As discussed above, this dataset contains a lot of risk factors, which is easily captured by plain RNN models. However, using attention mechanism can force models to only focus on the visits that contain risk factors and ignore the rest visits. Besides, aggregating all visits together may further induce noise and hurt final performance. To tackle this issue, we design an abstract representation $\mathbf{h}_*$ to synthesize overall information in HiTANet.

Finally, we examine the influence of time information for risk prediction task. Comparing time-based models with attention-based models, we can observe that the overall performance is comparable (on the Kidney Diseases dataset) or better (on the COPD and Heart Failure datasets). This can prove that modeling time information of each visit is meaningful. However, they all assume that the information monotonously decays in accord with the length of time interval and directly use a fixed function, (e.g., $\frac{1}{\log(\delta+e)}$), to generate the time parameter. As a result, the generated time information may not be suitable for all the scenarios, which leads to a slight drop on the performance in some cases. However, in HiTANet, time parameter is generated by a complicated self-learned nonlinear layer that ensure to learn the best time-level attention to avoid this problem.

5.3 Ablation Study

In this section, we focus on the comparison between HiTANet and its variants that change parts of the full HiTANet model. Doing such an ablation study can clearly make us know how each individual module of HiTANet contributes to the final performance. Table 3 shows the average performance on the COPD, Heart Failure, and Kidney diseases, respectively. The settings are the same with the previous experiments, and we still run five times to obtain the average performance.

Table 3: Average Performance for HiTANet's Variants

Method	COPD		Heart Failure		Kidney Diseases	
	F1	Auc	F1	Auc	F1	Auc
HiTANet	**0.637**	**0.752**	**0.645**	**0.750**	0.702	**0.792**
HiTANet-LT	0.624	0.742	0.633	0.740	**0.707**	0.789
HiTANet-GT	0.589	0.718	0.599	0.718	0.699	0.786
HiTANet-GLT	0.547	0.694	0.616	0.730	0.661	0.761
HiTANet-GLT∗	0.415	0.626	0.463	0.644	0.558	0.697

In the proposed HiTANet, there are two components using time information. The one is embedding time information in local-level visit analysis, i.e., Eq. (2), the other is to identify the importance of time information during the disease progression with Eq. (6). Here, we use two variant models to validate the influence of time information. HiTANet-LT means removing time embedding from local-level visit analysis component, i.e., directly using $[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_T, \mathbf{e}_*]$ as the inputs of F to learn the hidden states $[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_T, \mathbf{h}_*]$ with Eq. (3). All other components are the same as HiTANet. HiTANet-GT represents removing the whole comprehensive analysis and directly using attention weights learned by Eq. (4) to calculate the final patient representation with Eq. (12), which can be considered as a flat structure of HiTANet. As shown in Table 3, the performance of both HiTANet-LT and HiTANet-GT drops, which validates that modeling time information is essential for risk prediction task. Especially for HiTANet-GT, its performance drops dramatically, which further confirms the importance of designing a hierarchical structure for modeling time information.

Next, we continue reducing the proposed model by removing the local time embedding from HiTANet-GT, i.e., remaining the structure of Transformer to learn hidden states and the local attention mechanism to learn patient representation, which is named HiTANet-GLT. Compared with HiTANet-GT, its performance drops a lot on both the COPD and Kidney Diseases datasets. This again demonstrates the effectiveness of modeling time information. Since the proposed model HiTANet introduces an overall diagnosis representation $\mathbf{h}_*$, we aim to check whether this vector is useful for the prediction. We use HiTANet-GLT∗ to represent this approach and find that its performance is the worst among all the variants of HiTANet as shown in Table 3, which confirms that only using the overall representation will lose a lot of key information and further hurt the performance.

5.4 Attention Analysis

From this ablation study and experimental results listed in Table 2, we can safely conclude that using a hierarchical structure to model both local and global time information can significantly improve the performance of risk prediction task. To further illustrate the reasonableness of the proposed HiTANet, we conduct case studies to interpret the learned local-level attention weights and visualize the learned global-level attention weights.

5.4.1 Local Attention Analysis. Local attention weights obtained by Eq. (4) represent the importance of different visits. Next, we analyze one positive case (with heart failure) and one negative case (without heart failure) to show the interpretability of the proposed HiTANet. In particular, we print out the local attention weight of each visit and remove part of diagnosis codes to check the prediction changes.

Figure 3 shows the data of a positive patient, which has five time-ordered visits and the learned local attention weight for each visit. Using the proposed HiTANet, the predicted probability of suffering heart failure disease is 0.890. We can observe that the first visit can be assigned the largest weight, because it contains an ICD9 diagnosis code *780.53*[9], which refers to *Hypersomnia with sleep apnea, unspecified*. It is commonly seen as a late manifestation of heart failure [5, 6]. However, if we remove the record of *780.53*, the probability of being positive drops to 0.853. This observation proves that HiTANet is able to learn the correct local attention weights as a human doctor picking the important visits.

[9] http://www.icd9data.com/2015/Volume1/780-799/780-789/780/780.53.htm

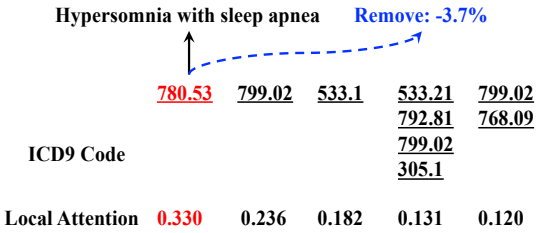

Figure 3: A positive example from the Heart Failure testing set. HiTANet **assigns a higher attention to the first visit, which contains Hypersomnia, a common signal of Heart Failure problems. If we remove this record, then the probability of predicting as a positive case will drop 3.7%.**

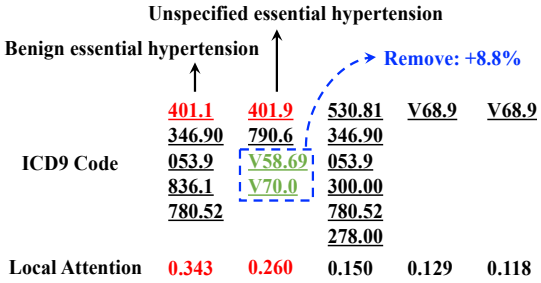

Figure 4: A negative example from the Heart Failure testing set. HiTANet **assigns high attention weights to the first two visits. They both contain hypertension related diagnosis codes marked in red, which are the risk factors for Heart Failure. Codes marked in green means the adopted treatments. If we remove the treatment codes, the probability of being positive will increase 8.8%.**

Figure 4 shows the local attention weights learned by the proposed HiTANet on a negative example. Code *401.xx*[10] refers to *Hypertension*, which is the most prevalent modifiable risk factor for the development of Heart Failure [16]. Code *V58.69*[11] and *V70.0*[12] are the treatment procedures of *Hypertension*. HiTANet pays high attention to these two visits, because they contain risk factors. If we remove the records of treatment procedures *V58.69* and *V70.0*, the probability of being positive will increase from 0.251 to 0.339, which means that HiTANet also takes into account the influence of treatment procedures in predicting the risk of Heart Failure. Thus, HiTANet can effectively capture important disease relations and reflect them in the form of attention weights. In the meanwhile, it can also capture the effectiveness of treatment procedures.

5.4.2 Global Attention Analysis. Since HiTANet uses a dynamic system to generate the global time attention according to the overall diagnosis representation $\mathbf{h}_*$, even in the same time stamp, the global time attention weight may be different. Hence, we plot the average of the time attention weights for the same time stamps (in weeks) to explore the macroscopic tendency as shown in Figure 5.

[10]http://www.icd9data.com/2015/Volume1/390-459/401-405/401/default.htm
[11]http://www.icd9data.com/2015/Volume1/V01-V91/V50-V59/V58/V58.69.htm
[12]http://www.icd9data.com/2015/Volume1/V01-V91/V70-V82/V70/default.htm

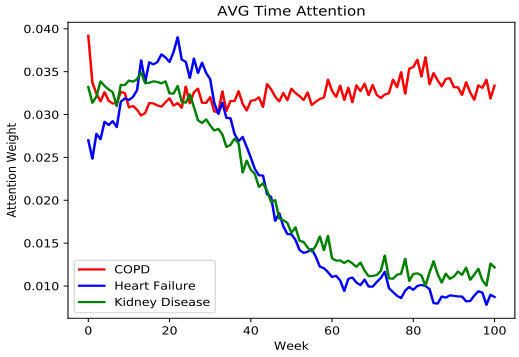

Figure 5: Average global attention weights learned by HiTANet **on the three validation sets.**

We can observe that none of the attention weights follows a strictly monotonic decreasing order. Some distant time stamps even get greater weights on the COPD dataset. At the first glance, it may conflict with the common assumption that the recent visits are more important because they can reveal more current information of the patient. However, the progression of disease is in a gradual way. Let us consider the Heart Failure disease. It focuses more on 20~30 weeks before making a definite diagnosis. One possible explanation is that the high risk causes of Heart Failure like high blood pressure, diabetes, obesity and smoking do not directly cause the target disease, but they will greatly increase the risks as the time increases. As described in Arnold's work [1], early treatments of those risk factors can effectively prevent Heart Failure. In other words, a person who had high blood pressure half-year ago is more likely to have heart disease now, compared with a man who just found high blood pressure a month ago. In the previous situation, his/her heart suffers more from the high blood pressure. This finding is interesting because it may provide a different aspect for disease prediction and alarm.

6 CONCLUSIONS

Risk prediction from EHR data is one of the key challenges in predictive healthcare. Current studies on risk prediction either still rely on RNN-based models for feature modeling or ignore the full use of time information in feature aggregation. In this paper, a hierarchical self-attention-based model named HiTANet is proposed to address these problems. HiTANet aggregates visit representations with local time embeddings with the designed time-aware Transformer, and it then recognizes key timestamps associated with visits by a novel self-attention mechanism using the synthesized global embeddings as query vectors and time embeddings as key vectors. We then use the dynamically fused attention of these two time-aware attention weights to learn the final attention weight for each visit. We evaluated HiTANeton real world EHR data and show that the proposed HiTANet outperforms state-of-the-art deep neural network models and achieves stable improvements in risk prediction tasks on three large-scale real-world disease cohorts. In the meanwhile, the case analysis results demonstrate that the inference process of HiTANet in risk prediction is highly interpretable.

REFERENCES

[1] J Malcolm O Arnold, Salim Yusuf, James Young, James Mathew, David Johnstone, Alvaro Avezum, Eva Lonn, Janice Pogue, and Jackie Bosch. 2003. Prevention of heart failure in patients in the Heart Outcomes Prevention Evaluation (HOPE) study. *Circulation* 107, 9 (2003), 1284–1290.

[2] Tian Bai, Shanshan Zhang, Brian L Egleston, and Slobodan Vucetic. 2018. Interpretable representation learning for healthcare via capturing disease progression through time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 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 *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 65–74.

[4] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. 2019. Meta-learning with differentiable closed-form solvers. In *The International Conference on Learning Representations (ICLR)*.

[5] T Douglas Bradley and John S Floras. 2003. Sleep apnea and heart failure: Part I: obstructive sleep apnea. *Circulation* 107, 12 (2003), 1671–1678.

[6] T Douglas Bradley and John S Floras. 2003. Sleep apnea and heart failure: Part II: central sleep apnea. *Circulation* 107, 13 (2003), 1822–1826.

[7] Prithwish Chakraborty and Faisal Farooq. 2019. A Robust Framework for Accelerated Outcome-driven Risk Factor Identification from EHR. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1800–1808.

[8] Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. 2015. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 507–516.

[9] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8, 1 (2018), 1–12.

[10] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. 2016. Risk prediction with electronic health records: A deep learning approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 432–440.

[11] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[12] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. 2016. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1495–1504.

[13] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. 2017. GRAM: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 787–795.

[14] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*. 3504–3512.

[15] Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. 2018. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In *Advances in Neural Information Processing Systems*. 4547–4557.

[16] Shannon M Dunlay, Susan A Weston, Steven J Jacobsen, and Véronique L Roger. 2009. Risk factors for heart failure: a population-based case-control study. *The American journal of medicine* 122, 11 (2009), 1023–1028.

[17] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[18] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Bum Chul Kwon, Min-Je Choi, Joanne Taery Kim, Edward Choi, Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo. 2018. Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 299–309.

[20] Yikuan Li, Shishir Rao, Jose Roberto Ayala Solares, Abdelaali Hassaine, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. 2019. BEHRT: Transformer for Electronic Health Records. *arXiv preprint arXiv:1907.09538* (2019).

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

[22] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[23] 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 *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1903–1911.

[24] 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 *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1910–1919.

[25] Fenglong Ma, Yaqing Wang, Houping Xiao, Ye Yuan, Radha Chitta, Jing Zhou, and Jing Gao. 2018. A General Framework for Diagnosis Prediction via Incorporating Medical Code Descriptions. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 1070–1075.

[26] 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 *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 743–752.

[27] Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiantao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. 2019. Concare: Personalized clinical feature embedding via capturing the healthcare context. *arXiv preprint arXiv:1911.12216* (2019).

[28] Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiantao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. 2020. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

[29] Tengfei Ma, Cao Xiao, and Fei Wang. 2018. Health-atm: A deep architecture for multifaceted patient health record representation and risk prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 261–269.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 8024–8035.

[31] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[32] Trang Pham, Truyen Tran, Dinh Phung, and Svetha Venkatesh. 2016. Deepcare: A deep dynamic memory model for predictive medicine. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 30–41.

[33] George AF Seber and Alan J Lee. 2012. *Linear regression analysis*. Vol. 329. John Wiley & Sons.

[34] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. 2019. Pre-training of graph augmented transformers for medication recommendation. *arXiv preprint arXiv:1906.00346* (2019).

[35] Huan Song, Deepta Rajan, Jayaraman J Thiagarajan, and Andreas Spanias. 2018. Attend and diagnose: Clinical time series analysis using attention models. In *Thirty-second AAAI conference on artificial intelligence*.

[36] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[37] Qiuling Suo, Fenglong Ma, Giovanni Canino, Jing Gao, Aidong Zhang, Pierangelo Veltri, and Gnasso Agostino. 2017. A multi-task framework for monitoring health conditions via attention-based recurrent neural networks. In *AMIA annual symposium proceedings*, Vol. 2017. American Medical Informatics Association, 1665.

[38] Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Jing Gao, and Aidong Zhang. 2018. Deep patient similarity learning for personalized healthcare. *IEEE transactions on nanobioscience* 17, 3 (2018), 219–227.

[39] Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Aidong Zhang, and Jing Gao. 2017. Personalized disease prediction using a cnn-based similarity learning method. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 811–816.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[41] Lipo Wang. 2005. *Support vector machines: theory and applications*. Vol. 177. Springer Science & Business Media.

[42] Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. 2019. EHR Coding with Multi-scale Feature Attention and Structured Knowledge Graph Propagation. In *CIKM*. 649–658.

[43] Changchang Yin, Rongjian Zhao, Buyue Qian, Xin Lv, and Ping Zhang. 2019. Domain Knowledge Guided Deep Learning with Electronic Health Records. In *2019 IEEE International Conference on Data Mining (ICDM)*.

[44] Xianli Zhang, Buyue Qian, Yang Li, Changchang Yin, Xudong Wang, and Qinghua Zheng. 2019. KnowRisk: An Interpretable Knowledge-Guided Model for Disease Risk Prediction. In *ICDM*. IEEE, 1492–1497.

[45] 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 *(KDD '19)*. ACM, New York, NY, USA, 2487–2495. http://doi.acm.org/10.1145/3292500.3330779

A TRANSFORMER DETAILS

The structure of a Transformer is defined as follows. First, we add position embedding to the original input to capture the order information. We then apply the scaled dot-product attention to each input for modeling the interaction. Finally, we pass the generated embedding through a position-wise feed-forward network to enhance the expression ability of each embedding position.

A.1 Positional Encoding

Besides the time embedding mentioned in Section 4.1, the Transformer also contains a inner positional encoding procedure to capture the basic input order.

$$PE_{(t,2i)} = sin(t/10000^{2i/m}), \tag{15}$$

$$PE_{(t,2i+1)} = cos(t/10000^{2i/m}), \tag{16}$$

where m is the dimension size of the hidden space, and i is the detention of the position embedding PE. The generated the position embedding will be added to the original input $\mathbf{v}_t$.

A.2 Scaled Dot-Product Attention

For each input, we use three fully connection layers to generate three additional representations as $\mathbf{q}', \mathbf{k}', \mathbf{v}'$. By combing them, we can further build three two-dimension matrix $\mathbf{Q}', \mathbf{K}', \mathbf{V}'$, and the final attention fusion can be described as:

$$\text{Attention}(\mathbf{Q}', \mathbf{K}', \mathbf{V}') = \text{Softmax}(\frac{\mathbf{Q}'\mathbf{K}'^T}{\sqrt{d_k}})\mathbf{V}', \tag{17}$$

where d_k is the dimension of attention embedding. In our experiments, it is set to 64. The new input will be the aggregated embedding from each input in the ratio of attention weight.

A.3 Multi-Head Attention

Since different self-attention operations may have their own focus, to improve the prediction performance, in our experiments, the number of attention group is set to 4.

A.4 Position-wise Feed-Forward Networks

A feed-forward layer is applied to each position separately and identically.

$$\text{FFN}(\mathbf{x}') = max(0, \mathbf{x}'\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2. \tag{18}$$

The dimension size of the middle feed-forward space is 1024, and $\mathbf{x}'$ is the middle input embedding.

B ADDITIONAL EXPERIMENTAL RESULTS

Here, we use two additional datasets to validate the proposed HiTANet. The data statistics are listed in Table 4. The experimental results are shown in Tables 5 and 6. The settings of these two experiments are the same as the main part. From the results, we can observe that the proposed HiTANet also achieves the best performance on these two datasets in terms of accuracy, F1 score, and Auc measures. This again confirms that the improvement of HiTANet is stable and repeatable for most diseases prediction tasks. We can observe similar patterns for baselines as the previous three diseases. Classical methods are less robust compared with deep learning-based methods, but sometimes they show better performance. The Plain RNNs show a stable performance. The improvement of attention-based models is also significant. However, the improvement of time-based models is not obvious, possibly due to the same reason as we discussed in the previous main part.

Table 4: Dataset Details.

Dataset	Amnesia	Dementias
Case (Positive)	2,982	2,385
Control (Negative)	8,946	7,155
Avg visits per patient	39.00	41.05
Avg codes per visit	4.70	4.71
Unique ICD-9 codes	9,032	7,813

Table 5: Average Performance on the Amnesia Dataset

Method		Amnesia				
		Acc	Pre	Recall	F1	Auc
Classical Methods	SVM	0.835	0.694	0.558	0.619	0.740
	LR	0.763	0.506	0.593	0.546	0.705
	RF	0.823	0.730	0.421	0.534	0.686
Plain RNNs	LSTM	0.811	0.706	0.455	0.548	0.694
	GRU	0.819	0.713	0.484	0.576	0.709
Attention-based Models	Dipole	0.827	0.695	0.522	0.589	0.723
	Dipole–	0.840	0.711	0.572	0.632	0.749
	Retain	0.822	0.646	0.581	0.610	0.739
	SAnD	0.828	0.677	0.557	0.605	0.735
Time-based Models	RetainEx	0.824	0.690	0.489	0.572	0.710
	TA-LSTM	0.826	**0.765**	0.420	0.542	0.689
	TimeLine	0.814	0.615	**0.610**	0.612	0.744
Ours	HiTANet	**0.848**	0.727	0.597	**0.654**	**0.762**

Table 6: Average Performance on the Dementias Dataset

Method		Dementias				
		Acc	Pre	Recall	F1	Auc
Classical Methods	SVM	0.783	0.757	0.153	0.255	0.569
	LR	0.714	0.433	0.590	0.499	0.672
	RF	0.800	0.661	0.355	0.462	0.649
Plain RNNs	LSTM	0.793	0.607	0.552	0.573	0.713
	GRU	0.790	0.609	0.473	0.527	0.685
Attention-based Models	Dipole	0.803	0.644	0.422	0.507	0.673
	Dipole–	0.803	0.635	0.444	0.519	0.681
	Retain	0.810	**0.654**	0.463	0.539	0.692
	SAnD	0.782	0.613	0.280	0.377	0.611
Time-based Models	RetainEx	0.803	0.630	0.469	0.535	0.690
	TA-LSTM	0.798	0.643	0.450	0.521	0.682
	TimeLine	0.787	0.583	0.426	0.488	0.664
Ours	HiTANet	**0.810**	0.622	**0.553**	**0.584**	**0.723**