LSAN: Modeling Long-term Dependencies and Short-term Correlations with Hierarchical Attention for Risk Prediction

Muchao Ye
Pennsylvania State University
muchao@psu.edu

Junyu Luo
Pennsylvania State University
junyu@psu.edu

Cao Xiao
IQVIA
cao.xiao@iqvia.com

Fenglong Ma*
Pennsylvania State University
fenglong@psu.edu

ABSTRACT

Risk prediction using electronic health records (EHR) is a challenging data mining task due to the two-level hierarchical structure of EHR data. EHR data consist of a set of time-ordered visits, and within each visit, there is a set of unordered diagnosis codes. Existing approaches focus on modeling temporal visits with deep neural network (DNN) techniques. However, they ignore the importance of modeling diagnosis codes within visits, and a lot of task-unrelated information within visits usually leads to unsatisfactory performance of existing approaches. To minimize the effect caused by noise information of EHR data, in this paper, we propose a novel DNN for risk prediction termed as LSAN, which consists of a Hierarchical Attention Module (HAM) and a Temporal Aggregation Module (TAM). Particularly, LSAN applies HAM to model the hierarchical structure of EHR data. Using the attention mechanism in the hierarchy of diagnosis code, HAM is able to retain diagnosis details and assign flexible attention weights to different diagnosis codes by their relevance to corresponding diseases. Moreover, the attention mechanism in the hierarchy of visit learns a comprehensive feature throughout the visit history by paying greater attention to visits with higher relevance. Based on the foundation laying by HAM, TAM uses a two-pathway structure to learn a robust temporal aggregation mechanism among all visits for LSAN. It extracts long-term dependencies by a Transformer encoder and short-term correlations by a parallel convolutional layer among different visits. With the construction of HAM and TAM, LSAN achieves the state-of-the-art performance on three real-world datasets with larger AUCs, recalls and F1 scores. Furthermore, the model analysis results demonstrate the effectiveness of the network construction with good interpretability and robustness of decision making by LSAN[1].

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Applied computing** → **Health informatics**.

KEYWORDS

data mining; electronic health records; temporal modeling; attention mechanism

ACM Reference Format:
Muchao Ye, Junyu Luo, Cao Xiao, and Fenglong Ma. 2020. LSAN: Modeling Long-term Dependencies and Short-term Correlations with Hierarchical Attention for Risk Prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411864

1 INTRODUCTION

With the wide utilization and tremendous improvements in digital healthcare systems, electronic health records (EHRs) have become valuable data that can be used to enhance decision making and healthcare delivery. One of the core EHR based health analytic problems is predicting the future health status of patients based on their historical EHR data, which is referred to as the **risk prediction** in data mining [4, 6, 17, 18]. Towards this task, we first need to solve a challenge stemming from EHR data.

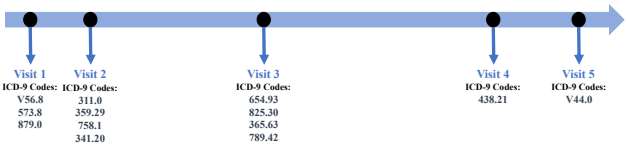


Figure 1: Example data of a heart failure patient for illustrating the properties of EHRs. The hierarchical structure in the example EHRs is that the patient has 5 visits, and each visit is diagnosed with multiple ICD-9 codes.

As shown in Figure 1, EHR data use a two-level hierarchical structure to capture the medical journey of a patient encoded by medical codes from a high-dimensional coding system, *e.g.*, the International Classification of Diseases (ICD) codes[2]. The hierarchy begins with the patient, followed by multiple visits of the patient in temporal order. Then within each visit, there are ICD codes that describe the symptoms of patients for that visit. Existing studies mostly focus on modeling the visit-level hierarchy [2–4, 6, 15, 17, 20, 21, 27, 31]. They usually rely on recurrent neural networks (RNNs), attention mechanisms, and time-aware mechanism to learn the context information from all visits and then make predictions. Although these methods have achieved early success on risk prediction, they often over-simplify the within-visit level learning. For example, they only use a simple embedding layer to learn the representation for each

*Corresponding author.
[1]The implementation code is available at https://github.com/dmmlprojs/lsan.

[2]http://www.icd9data.com/

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411864

visit without fully considering the relatedness of diagnosis codes to the target disease in the within-visit level.

When predicting the risk of getting the target disease, we aim to automatically learn the disease progression process from EHR data. However, many medical codes are irrelevant and should be treated as noise for risk prediction. In Figure 1, the patient will suffer from heart failure disease after six months. We can observe that during her first three visits, she is diagnosed with symptoms that are not closely related to heart failure, such as "open wound of breast" (879.0), "acute transverse myelitis" (341.20), "uterine scar" (654.93) and "fracture of other tarsal and metatarsal bone" (825.30). When we directly apply existing approaches, including RNN-based models and attention mechanisms, such noisy information within each visit may significantly hurt their performance due to the "error accumulation" issue [36]. Thus, in order to predict her health status correctly, a good risk prediction method should be able to minimize the effect introduced by the noise as much as possible. Towards this aim, we take the following two aspects into consideration.

- *Distinguishing the importance of diagnosis codes within each visit.* Within each visit, there may exist diagnosis codes that are unrelated to the target task. Existing models normally treat each diagnosis code equally, but it is better to distinguish their relevance to the disease and assign different contribution scores to each code when learning the embedding of the corresponding visit. Therefore, one remaining challenge is how to automatically learn such differences among codes within each visit.
- *Filtering out noise by extracting local temporal correlations among neighboring visits.* To capture the temporal patterns of disease changes, we need to mine the EHR data by consecutive segments, *i.e.*, neighboring visits, instead of a single visit. The benefit of extracting local temporal patterns within neighboring visits is that such a consideration can help us find out the symptoms unrelated to the disease progression and filter out the noise. Thus, a new challenge we should face is how to automatically extract the latent local temporal patterns of disease progression among neighboring visits.

To tackle the aforementioned challenges, we propose a new framework (shorten for LSAN) by modeling the Long-term dependencies and Short-term correlations with the utilization of a hierarchical Attention Network. Specifically, LSAN comprises two modules, *Hierarchical Attention Module* (HAM) and *Temporal Aggregation Module* (TAM), as described below.

- *Hierarchical Attention Module* depicts the hierarchical nature of EHR data and summarizes the diagnosis results by their relevance to the target disease. The attention mechanism in the **hierarchy of diagnosis code** allows LSAN to learn different weights to the codes within each visit according to their relevance to the target disease. Then, a weighted sum operation on diagnosis code embeddings is used to obtain the embedding of each visit. The weighted sum representation can retain the details of each diagnosis code while highlighting their relevance by corresponding weights. Similarly, the weighted sum representation in the **hierarchy of visit** summarizes details of each visit with weights highlighting their importance to the final decision making.
- *Temporal Aggregation Module* learns two complementary temporal information by better utilization of **long-term dependencies** and **short-term correlations**. For one thing, long-term dependencies are temporal information on how each visit relates to the rest visits in the complete medical journey. They help overcome individual bias inside each visit by integrating diagnosis results from all visits. For another, short-term correlations are how every visit relates to each other in a short period. They are also vital to removing irrelevant information in EHR data because patients will have correlated symptoms with neighboring visits in each disease development stage. As for TAM, it selects two appropriate building blocks to model them in the hierarchy of visit: it learns the long-term dependencies between all visits by a Transformer and short-term correlation by a convolutional layer (Conv). The pathway of Transformer is good at utilizing the long-term temporal structure to acquire a comprehensive representation, and the pathway of Conv can sharpen salient symptoms within neighboring visits in each stage. Thus, these two complementary pathways in TAM help LSAN learn better relevance between diagnosis results and target diseases.

LSAN is an end-to-end model, which first learns visit embeddings with the designed diagnosis-code-level attention in HAM and then applies TAM to capture both long-term dependencies and short-term correlations among visits. The outputs of TAM are used to learn the final comprehensive patient representation by the visit-level attention in HAM. Lastly, the comprehensive representation is used to make a prediction. Thus, they tightly work together and significantly enhance each other. To summarize, our contributions are listed as follows:

- We develop a new framework termed as LSAN for health risk prediction, which not only fully exploits the internal hierarchical structure of EHR but also captures different granularity of disease progressions among temporal visits.
- We design a hierarchical attention mechanism named HAM which corresponds to the two-hierarchy structure of EHR data. HAM provides more expressive details and flexibility for patient representation learning, and lays the foundation for reducing irrelevant diagnosis codes in EHR data.
- We propose a novel module named TAM to model the process of disease progression by considering both long-term dependencies and short-term correlations among visits. Such a design can remove noise information among visits and further lead to a robust risk prediction model.
- We evaluate LSAN against both shallow methods and deep learning methods on three real-world datasets, and it surpasses the state-of-the-art models in AUCs, recalls and F1 scores. We also justify the effectiveness, interpretability and robustness of our method by ablation studies and case studies.

2 RELATED WORK

Recent years have witnessed the great progress achieved by the deep learning models in risk prediction due to the rich representations that can be learned by DNNs [23, 33]. In this section, we mainly discuss how existing models employ RNNs, attention mechanisms and time-aware mechanisms for architecture design and their limitations in dealing with noise information. Though there

are some approaches incorporating external information to improve the prediction performance [5, 7, 18–20, 35], they have different problem settings from this paper.

2.1 RNNs with Attention Mechanism

Recent methods usually construct DNNs by combining RNNs and attention mechanisms together and mainly model the temporal characteristic of EHR data in the hierarchy of visit [6, 17, 27, 31]. They regard EHR data as sequence data like video [9] and text [34], so they use RNNs like Long Short-Term Memory (LSTM) [12] and Gated Recurrent Units (GRUs) [8] to propagate the medical information throughout the whole medical history. Attention mechanism is built in the hierarchy of visit [6, 17, 20, 27] to assign greater importance to relevant visits. Exemplary pioneering work is conducted by Choi *et al.* [6], which adapts RNNs into an end-to-end interpretable network for risk prediction task. Later, Ma *et al.* [17] utilize a bidirectional RNN to aggregate the input EHR data and explore three ways to attend features in the hierarchy of visit for risk prediction. Their success makes combining RNNs with an attention mechanism module become the mainstream method in EHR data mining, and they inspire the design principle of later works like Timeline [2] and Health-ATM [21].

However, only working in the hierarchy of visit is not enough to release the noise information for risk prediction, since they ignore the fact that different diagnosis codes may contribute differently to the target disease even they are in the same visit. Different from existing EHR data modeling, our proposed HAM starts from the bottom hierarchy of diagnosis code, and it learns the visit embedding by assigning higher weights to the relevant diagnosis codes in each visit. The extra processing in the hierarchy of diagnosis code can prevent irrelevant information from propagating to the topper hierarchy, which is useful in removing noise information.

2.2 Temporal Modeling for EHR Data

Modeling temporal information is a useful technique to aggregate diagnosis results in the hierarchy of visits. A representative line of works to model temporal information for EHR data is to use the time-aware mechanism. That is, they fuse the date of each visit to learn its corresponding embedding. Some approaches [2, 21] use the interval between each visit and the final one, while others [3, 15] integrate the interval information between each visit and their previous one for embedding learning.

We would like to point out that current methods mostly pay attention to learning the long-term dependencies for temporal aggregation. Compared to them, the proposed LSAN tries to filter out the noise information of irrelevant symptoms by aggregating both long-term dependencies and short-term correlations in TAM. We employ Transformer and a convolutional layer to model them separately, which can extract more abstract temporal relations in the temporal structure of EHR data. As a consequence, our model can depress the noise information in each diagnostic result with more robust temporal information.

3 PROPOSED METHOD

In this section, we introduce the proposed LSAN by discussing the basic notations of risk prediction task first. Next, we present the overview of our LSAN. Finally, we detail two important modules of LSAN, namely HAM and TAM.

3.1 Basic Notations and Target Task

Notations. For each patient p, the historical diagnostic results of p can be denoted as a sequential list $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n]$, where $\boldsymbol{h}_i$ is the diagnostic results of the i-th visit, and n is the number of visits. For each visit $\boldsymbol{h}_i$, it consists of a subset of ICD-9 codes $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m\}$, where m is the number of unique diagnosis codes in the dataset. Each code $\boldsymbol{c}_j \in \boldsymbol{C}$ represents a unique symptom or abnormal finding of certain diseases, and $\boldsymbol{c}_j = [0, \ldots, 1, \ldots, 0]^\top \in \mathbb{R}^m$, where 1 appears in the j-th row. Thus, each visit $\boldsymbol{h}_i$ is expressed by a sparse column vector $\{0, 1\}^m \in \mathbb{R}^m$, where the j-th element $\boldsymbol{h}_{ij}$ is 1 if the diagnostic results contain $\boldsymbol{c}_j$, otherwise $\boldsymbol{h}_{ij} = 0$.

Target Task. The task of risk prediction is to find a function f that can accurately predict the health status of patient p with the longitudinal data $\boldsymbol{H} \in \mathbb{R}^{m \times n}$. The main concerns of function f are to extract the hidden disease progression information from patient data $\boldsymbol{H}$ and to deal with the issue of noise information. The prediction result given by f is

$$\hat{y} = f(\boldsymbol{H}). \tag{1}$$

In the task setting, the ground truth of the health status of p is given, which is denoted as y. Since the risk prediction problem can be regarded as a binary classification problem, y is set to 1 when the patient will suffer from the target disease in the future and to 0 otherwise. The aim of function f is to provide an accurate prediction such that $\hat{y}$ is as close as y.

3.2 Framework

Architecture Overview. Our major contribution lies in the design of function f, namely LSAN. As shown in Figure 2, LSAN consists of Hierarchical Attention Module (HAM) and Temporal Aggregation Module (TAM). For one thing, HAM has a hierarchical attention mechanism in the hierarchies of **diagnosis code** and **visit**. In the hierarchy of diagnosis code, it gets a single dense diagnosis embedding $\bar{\boldsymbol{h}}_i$ for each visit by summing up the diagnosis code embeddings with code-level attention weights. In the hierarchy of visit, it attends the aggregated visit embeddings by their relevance to target disease and attains a comprehensive representation for risk prediction.

For another, the key of TAM is to aggregate the visit embeddings with two kinds of temporal information from global and local temporal structures. When the features of all visits $\left[\bar{\boldsymbol{h}}_1, \ldots, \bar{\boldsymbol{h}}_n\right]$ are put into TAM, it models **long-term dependencies** in the global structure by Transformer and **short-term correlations** in the local structure by a convolutional layer. As mentioned before, long-term dependencies include global context throughout all visits, and short-term correlations among neighboring visits pay attention to diagnosis results in the same stage to sharpen the correlated results and filter out irrelevant ones. Therefore, the proposed approach of learning long-term and short-term temporal information separately can release the bias from noise information.

Next, we discuss the details of HAM and TAM in the order of how inputs $\boldsymbol{H}$ flow in our architecture as Figure 2 shows.

HAM in Hierarchy of Diagnosis Code. The input of patient p is a sparse matrix $\boldsymbol{H} \in \mathbb{R}^{m \times n}$. Since the original sparse expression of

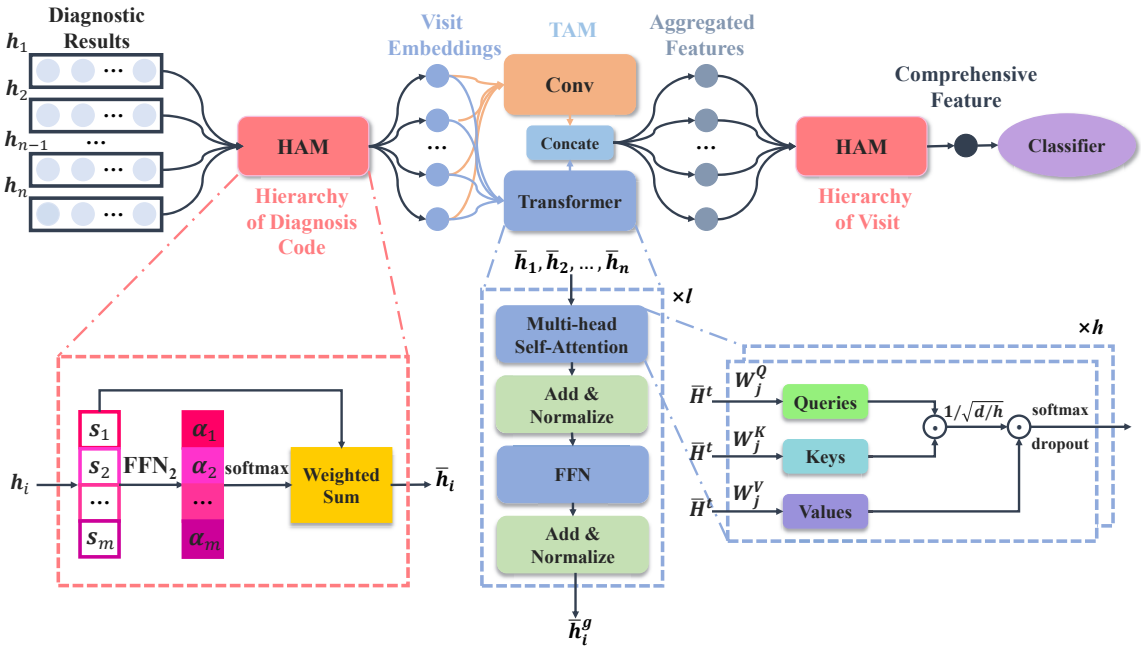


Figure 2: The proposed framework LSAN is equipped with two key modules: Hierarchical Attention Module (HAM) and Temporal Aggregation Module (TAM). Given the diagnostic results represented by ICD-9 codes in each visit, LSAN first uses the weighted sum of diagnosis code embeddings as the single embedding of each visit via the HAM module. Then LSAN learns a temporally aggregated feature for each visit by utilizing the long-term dependencies and short-term correlations of visit embeddings, which are achieved by our TAM module. Finally, we attend all aggregated features by the same attention mechanism in the hierarchy of visit to get a comprehensive feature for risk prediction. The details of computation in the hierarchy of visit are not depicted in the figure for they are similar to previous attention calculation.

diagnosis code is not good for representation learning, we should learn a dense diagnosis code embedding in the following way. In the hierarchy of diagnosis code, HAM first encodes each diagnosis code $\boldsymbol{c}_i$ into a dense embedding $\boldsymbol{e}_i \in \mathbb{R}^d$ through a 1-layer feedforward network FFN_1,

$$\boldsymbol{e}_i = \text{FFN}_1(\boldsymbol{c}_i) = \text{ReLU}(\boldsymbol{W}_1\boldsymbol{c}_i + \boldsymbol{b}_1), \tag{2}$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{d\times m}$, $\boldsymbol{b}_1 \in \mathbb{R}^d$, and ReLU is the rectified linear unit.

Now for the diagnosis code set $\boldsymbol{C}$, it is represented by a collection of dense embeddings $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m] \in \mathbb{R}^{d\times m}$. For the i-th visit, we obtain a dense embedding set $\boldsymbol{S}_i = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m]$ where $\boldsymbol{s}_j = \boldsymbol{e}_j$ if $\boldsymbol{h}_{ij} = 1$ to reflect the existence of a certain symptom or disease, otherwise $\boldsymbol{s}_j = \boldsymbol{0}$.

The 2-dimensional representation $\boldsymbol{S}_i \in \mathbb{R}^{d\times m}$ for each diagnostic result is still redundant for the learning process, so the next step that HAM takes is to attain a single feature $\bar{\boldsymbol{h}}_i \in \mathbb{R}^d$ to represent the latent information of the i-th visit. Particularly, in the hierarchy of diagnosis code, HAM uses a 3-layer feedforward network FFN_2 to learn the attention weight of every diagnosis code $\boldsymbol{s}_i$ in the set $\boldsymbol{S}_i$ and then attends them together for each visit. As such, we first get an attention score $\alpha_i \in \mathbb{R}$ for each diagnosis code embedding by

$$\alpha_i = \text{FFN}_2(\boldsymbol{s}_i). \tag{3}$$

Note that if $\boldsymbol{s}_i = \boldsymbol{0}$, we set $\alpha_i = -\infty$. We then normalize the attention scores with softmax function to get the normalized weights,

$$a_i = \frac{\exp(\alpha_i)}{\sum_{j=1}^{m}\exp(\alpha_j)}. \tag{4}$$

After that, we can obtain a single embedding $\bar{\boldsymbol{h}}_i$ for the i-th visit,

$$\bar{\boldsymbol{h}}_i = \sum_{i=1}^{m} a_i \cdot \boldsymbol{s}_i. \tag{5}$$

As a result, in the hierarchy of diagnosis code, we gain a set of attended features $\bar{\boldsymbol{H}} = \left[\bar{\boldsymbol{h}}_1, \ldots, \bar{\boldsymbol{h}}_n\right] \in \mathbb{R}^{d\times n}$ for a patient p.

TAM in Long-term Dependencies Modeling. TAM learns the long-term dependencies by Transformer. Transformer attends all visit features in parallel and does not obscure the details of each feature, so it suffices to overcome the problem that RNNs have. In practice, we use multi-head self-attention mechanism in Transformer for feature attending, and the Transformer encoder in TAM has l layers, where the computations are the same in each layer. Without loss of generality, we take the first layer to illustrate the computations. First, we add positional encoding into the i-th input visit,

$$\bar{\boldsymbol{h}}_i^t = \bar{\boldsymbol{h}}_i + \boldsymbol{t}_i, \tag{6}$$

where $\boldsymbol{t}_i$ is the positional encoding, $\boldsymbol{t}_{i,2k-1} = \cos(i/10000^{\frac{2k-1}{d}})$ and $\boldsymbol{t}_{i,2k} = \sin(i/10000^{\frac{2k}{d}})$ for $1 \le 2k-1 < 2k \le d$.

Each layer of Transformer has h heads. In the j-th head, the new features $\bar{\boldsymbol{h}}_i^t \in \mathbb{R}^d$ are packed into a matrix $\bar{\boldsymbol{H}}^t \in \mathbb{R}^{d\times n}$, and it is used to generate a set of queries $\boldsymbol{Q}_j = [\boldsymbol{q}_1^j, \ldots, \boldsymbol{q}_n^j]$, keys $\boldsymbol{K}_j = [\boldsymbol{k}_1^j, \ldots, \boldsymbol{k}_n^j]$ and values $\boldsymbol{V}_j = [\boldsymbol{v}_1^j, \ldots, \boldsymbol{v}_n^j]$ by linear transformations,

$$\boldsymbol{Q}_j = \boldsymbol{W}_j^Q \bar{\boldsymbol{H}}^t, \boldsymbol{K}_j = \boldsymbol{W}_j^K \bar{\boldsymbol{H}}^t, \boldsymbol{V}_j = \boldsymbol{W}_j^V \bar{\boldsymbol{H}}^t, \tag{7}$$

where $\boldsymbol{Q}_j, \boldsymbol{K}_j, \boldsymbol{V}_j \in \mathbb{R}^{\frac{d}{h}\times n}$, and $\boldsymbol{W}_j^Q, \boldsymbol{W}_j^K, \boldsymbol{W}_j^V \in \mathbb{R}^{\frac{d}{h}\times d}$ are learnable projection matrices.

Then with the self-attention mechanism, we generate the features that are aggregated with long-term dependencies by

$$\text{head}_j = \text{Attention}(\boldsymbol{Q}_j, \boldsymbol{K}_j, \boldsymbol{V}_j) = \text{softmax}(\frac{\boldsymbol{Q}_j \boldsymbol{K}_j^{\mathrm{T}}}{\sqrt{d/h}} \boldsymbol{V}_j). \tag{8}$$

Later, we concatenate all features in h heads and conduct a linear transformation to project it back to the original space,

$$\boldsymbol{O} = \text{Concate}(\text{head}_1, ..., \text{head}_h) \cdot \boldsymbol{W}^O, \tag{9}$$

where Concate denotes feature concatenation operation, and $\boldsymbol{W}^O \in \mathbb{R}^{d\times d}$ is a linear transformation matrix. The output $\boldsymbol{O} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_n] \in \mathbb{R}^{d\times n}$, and for each feature $\boldsymbol{o}_i$, it is an aggregated feature with long-term dependency information for $\bar{\boldsymbol{h}}_i^t$. We put $\boldsymbol{o}_i$ into an additional dropout layer [29], a normalization layer [13] with a residual connection [11]. Then we have

$$\bar{\boldsymbol{h}}_i^o = \bar{\boldsymbol{h}}_i^t + \text{dropout}(\text{norm}(\boldsymbol{o}_i)), \tag{10}$$

where dropout$(\cdot)$ is the dropout layer and norm$(\cdot)$ is the normalization layer. Before being put into the next layer, $\bar{\boldsymbol{h}}_i^o$ is applied into a 2-layer feedforward network, which outputs

$$\text{FFN}_3(\bar{\boldsymbol{h}}_i^o) = \boldsymbol{W}_3(\text{ReLU}(\boldsymbol{W}_2 \bar{\boldsymbol{h}}_i^o + \boldsymbol{b}_2)) + \boldsymbol{b}_3, \tag{11}$$

where $\boldsymbol{W}_2, \boldsymbol{W}_3 \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}_2, \boldsymbol{b}_3 \in \mathbb{R}^d$ are learnable parameters.

The output $\text{FFN}_3(\bar{\boldsymbol{h}}_i^o)$ will again be put into dropout and normalization layer with a residual connection similar to Eq. 10. As Figure 2 shows, the final output in the first layer will be treated as the input of the next $l-1$ layers. With the same calculations in the remaining layers, we can finally have the features $\bar{\boldsymbol{H}}^g = \left[\bar{\boldsymbol{h}}_1^g, ..., \bar{\boldsymbol{h}}_n^g\right] \in \mathbb{R}^{d\times n}$ aggregated with long-term dependencies by Transformer.

TAM in Short-term Correlations Modeling. As discussed before, another way to filter out the noise coming from irrelevant diagnosis codes is to extract the correlated disease progression information in each stage for temporal aggregation. In this case, the temporal information we should use is the short-term correlations between neighboring visits. In the designed TAM, it learns the short-term correlations between neighboring visits by employing a 1-dimensional convolutional layer Conv with kernel size w to sharpen the correlated diagnostic results. Within the window $\left[i - \frac{w-1}{2}, i + \frac{w-1}{2}\right]$, we aggregate $\bar{\boldsymbol{h}}_i$ by

$$\bar{\boldsymbol{h}}_i^l = \text{conv1d}(\left[\bar{\boldsymbol{h}}_{i-\frac{w-1}{2}}, \ldots, \bar{\boldsymbol{h}}_{i+\frac{w-1}{2}}\right]), \tag{12}$$

where conv1d is the learnable 1-dimensional convolutional operation, and the input $\left[\bar{\boldsymbol{h}}_{i-\frac{w-1}{2}}, \ldots, \bar{\boldsymbol{h}}_{i+\frac{w-1}{2}}\right]$ are the features within the range of kernel. The size of the output $\bar{\boldsymbol{h}}_i^l \in \mathbb{R}^d$ is still the same as that of $\bar{\boldsymbol{h}}_i$. The outputs of Conv aggregate the short-term correlations between visit embeddings within the filter window, which are

$$\bar{\boldsymbol{H}}^l = \text{Conv}(\bar{\boldsymbol{H}}) = \left[\bar{\boldsymbol{h}}_1^l, ..., \bar{\boldsymbol{h}}_n^l\right]. \tag{13}$$

The above procedure of aggregating features by learning from both global and local temporal structures have two-folded strengths. The first one is that the long-term dependencies information aggregated by Transformer enables each feature to learn the context of all diagnoses and avoid the biased information due to the irregular visits. The other advantage is that the short-term correlations learned by the convolutional layer sharpen the information related to neighboring visits in the same progression stage inside each visit. These two temporal information are both beneficial to the robustness of learned features, so we concatenate $\bar{\boldsymbol{h}}_i^g$ and $\bar{\boldsymbol{h}}_i^l$ to get a feature $\widetilde{\boldsymbol{h}}_i \in \mathbb{R}^{2d}$ for risk prediction,

$$\widetilde{\boldsymbol{h}}_i = \text{Concate}(\bar{\boldsymbol{h}}_i^g, \bar{\boldsymbol{h}}_i^l). \tag{14}$$

Finally, TAM outputs a matrix $\widetilde{\boldsymbol{H}} = \left[\widetilde{\boldsymbol{h}}_1, \ldots, \widetilde{\boldsymbol{h}}_n\right] \in \mathbb{R}^{2d\times n}$.

HAM in the Hierarchy of Visit. To learn a comprehensive feature with these temporally aggregated features for decision making in risk prediction, we should extract overall semantics from all visits. In the hierarchy of visit, we use HAM to attend the features in $\widetilde{\boldsymbol{H}}$ for risk prediction feature learning. Similar to what HAM does in the hierarchy of diagnosis code, it first employs a 3-layer feedforward network FFN_4 to learn attention scores $\beta_i \in \mathbb{R}$,

$$\beta_i = \text{FFN}_4(\widetilde{\boldsymbol{h}}_i). \tag{15}$$

We then get the normalized attention weights $b_i \in \mathbb{R}$ with softmax function

$$b_i = \frac{\exp(\beta_i)}{\sum_{j=1}^n \exp(\beta_j)}. \tag{16}$$

The comprehensive feature $\boldsymbol{x} \in \mathbb{R}^{2d}$ for risk prediction is learned by the attention mechanism, where

$$\boldsymbol{x} = \sum_{i=1}^n b_i \cdot \widetilde{\boldsymbol{h}}_i. \tag{17}$$

Risk Prediction. Finally, we utilize $\boldsymbol{x}$ for risk prediction,

$$\hat{y} = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b), \tag{18}$$

where $\sigma(x) = 1/(1+\exp(-x))$ is the Sigmoid function, and $\boldsymbol{w} \in \mathbb{R}^{2d}$, $b \in \mathbb{R}$ are learnable weights. Therefore, we attain the risk prediction result $\hat{y}$ as we want.

With the training set $\mathcal{T}$, we use binary cross-entropy loss $\mathcal{L}$ to train the model and get the learned parameters $\boldsymbol{\theta}$,

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \left(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right). \tag{19}$$

4 EXPERIMENTS

In this section, we report the experimental results on three real-world EHR datasets to demonstrate the effectiveness of the proposed method. After the discussion of the experimental setting, we first compare our method with shallow and deep methods. In addition, we analyze the effectiveness, interpretability and robustness of our model by an ablation study and case studies.

Table 1: Statistics of the used datasets.

Dataset	Heart Failure	Kidney Disease	Dementia
Total Cases	12,320	11,240	9,540
Positive Cases	3,080	2,810	2,385
Negative Cases	9,240	8,430	7,155
Average Visits per Patient	38.74	39.09	41.05
Minimum Number of Visits	5	5	5
Maximum Number of Visits	458	469	532
Average Codes per Visit	4.24	4.40	4.71
Unique ICD-9 Codes	8,692	8,802	7,813

4.1 Experimental Setup

Dataset. Our experiments are conducted on three real-world EHR datasets from the EHR database of IQVIA, and the diseases that we are concerned about are Heart Failure, Kidney Disease and Dementia. These diseases share one common characteristic: they tend to develop slowly under the influence of detrimental factors. The statistic information of these three datasets is shown in Table 1. Positives cases refer to the cases where patients suffer from the target disease, and the rest are referred as negative ones.

Baselines. The baselines for performance comparison fall into the following categories: (1) Shallow methods, including SVM [30], Linear Regression [28], and Random Forest [16]; (2) Vanilla RNNs, including LSTM [12] and GRU [8]; (3) RNNs with attention [6, 17] and (4) Time-aware DNNs [2, 3, 15]. For the convenience of illustrating the comparison results, we would like to briefly discuss some recent deep learning baselines.

- Retain [6] is an early successful work for EHR mining in the design of RNNs with attention. It introduces a reverse RNN that processes EHR data in reverse time order to mimic physicians in real life who pay more attention to recent visit results. It has an attention mechanism that try to detect significant past visits and clinical variables, which increases the interpretability of deep models.
- Dipole [17] is another significant representative of modeling temporality in EHR data by RNNs with attention. Dipole uses bidirectional RNN to process EHR data in two directions, and the attention module is built on the top of bidirectional RNN to interpret the importance of each visit result. Similar to Retain, Dipole finds that the employment of attention mechanism is beneficial to the improvement of performance and interpretability for decision making.
- RetainEx [15] is an improved version of Retain by taking into consideration the temporal information. One difference between RetainEX and Retain is that the representation of temporal interal between current visit and previous one is appended to the representation of diagnostic result during each visit for temporal aggregation in bidirectional RNNs. And this modification can enhance the interactivity of data mining tools with humans and obtain better interpretability in predictive results.
- T-LSTM [3] is a time-aware LSTM that is designed to handle the irregularity problem in EHR data by taking the length of visit intervals into consideration. It makes modification on the memory cell of LSTM, and its modification can decrease the effects of previous visit results with long intervals and reduce the influence of irregular sampling in EHR data.
- Timeline [2] designs a time-aware mechanism that learns the time decaying factor of each diagnosis code to improve the representation learning for EHR data. The time intervals are used to control how much information each diagnosis code embedding flows into the representation for each visit, and the visit results are then processed step by step within the RNN built in the top.

Implementation. LSAN is implemented in PyTorch [25] framework with Python in a NVIDIA Tesla P100 GPU and Intel Xeon E5-2680 CPUs. The parameters are trained by Adam optimizer [14] with the learning rate of 1×10^{-4} and the mini-batch size is 64. The size of each diagnosis code embedding d is 128. The number of Transformer layers l is 3, and in each layer the number of heads h is 8. The dropout rate is set to 0.1 in Transformer. For Conv component, w is set to 3 with the padding size 1. Shallow methods are implemented with scikit-learn [26] and EHRs are represented by the frequencies of 256 most frequent diagnosis codes. The numbers of layer in RNNs are set to 1 with the same hidden state size. As for the rest methods, we follow their instructions for implementation.

Evaluation Metrics. Since risk prediction task can be regarded as a binary classification problem, we mainly rely on the following two kinds of statistics for evaluation. The first statistic for evaluation is the area under the Receiver Operating Characteristic (ROC) curve, *i.e.* AUC, which is the probability that a model ranks a randomly chosen positive case higher than a randomly chosen negative case [10]. The higher the AUC is, the better ability the model has in distinguishing positive cases and negative ones. The other statistics for evaluation include precision, recall and F1 score. In risk prediction task, precision is the fraction of correctly classified positive cases over cases classified as positive, while recall is the fraction of correctly classified positive cases over all positive cases. Since our datasets have imbalanced distributions with larger portion of negative cases, we should count on the metric of F1 score, which is the harmonic mean of precision and recall. The higher F1 score is, the better classification performance is.

4.2 Performance Evaluation

As illustrated in Table 2, the proposed LSAN achieves superior performance over the listed baselines in the metrics of AUC, recall and F1 score, and what we can learn from this observation is illustrated as follows.

Analysis of AUC. First of all, an impressive observation that we have from Table 2 is that despite the variability among the EHR data of different diseases, LSAN manages to consistently outperform all listed methods on three datasets in terms of AUC. It should be noted that LSAN can obtain a margin of 11.9%, 10.3% and 11.8% in AUC over the second best method on the Heart Failure, Kidney Disease and Dementia datasets, respectively. Such notable advantages of LSAN in terms of AUC indicate that the designed LSAN is effective and powerful in separating positive cases and negative ones.

We owe the strength that LSAN has in AUC to the hierarchical attention in HAM and robustness of temporal modeling in TAM. There are some downsides of existing methods: (1) They build features for each visit directly from the hierarchy of visit and they treat the diagnosis code of the same visit equally. (2) They only learn the temporal information of long-term dependencies either

Table 2: Comparison of risk prediction performance on different EHR datasets.

Dataset		Heart Failure				Kidney Disease				Dementia			
Metrics		AUC (%)	Precision (%)	Recall (%)	F1 Score (%)	AUC (%)	Precision (%)	Recall (%)	F1 Score (%)	AUC (%)	Precision (%)	Recall (%)	F1 Score (%)
Shallow Methods	SVM [30]	64.4	**75.7**	32.7	45.7	74.5	**77.7**	54.5	64.1	56.9	**75.7**	15.3	25.5
	Linear Regression [28]	63.9	48.9	46.6	47.7	72.8	55.8	63.6	59.4	67.2	43.3	59.0	49.9
	Random Forest [16]	63.5	74.6	31.0	43.8	70.1	75.8	45.2	56.7	64.9	66.1	35.5	46.2
RNN Variants	LSTM [12]	70.8	64.0	51.0	56.1	73.9	68.0	57.2	61.6	71.3	60.7	55.2	57.3
	GRU [8]	70.0	67.9	49.0	56.7	74.5	67.8	59.1	62.9	68.5	60.9	47.3	52.7
RNNs + Attention	Dipole– [17]	69.8	68.9	48.1	56.5	76.4	67.9	63.5	65.6	68.1	63.5	44.4	51.9
	Dipole [17]	68.7	71.3	44.5	54.2	75.5	77.1	57.1	65.6	67.3	64.4	42.2	50.7
	Retain [6]	68.9	65.5	47.4	54.9	73.2	70.6	54.4	61.4	69.2	65.4	46.3	53.9
Time-aware DNNs	RetainEx [15]	68.8	73.0	43.8	54.6	72.8	74.5	52.0	61.2	69.0	63.0	46.9	53.5
	T-LSTM [3]	72.7	69.5	52.7	59.8	72.9	72.8	52.4	60.8	68.2	64.3	45.0	52.1
	Timeline [2]	70.5	66.1	51.0	57.4	75.6	69.7	60.7	64.8	66.4	58.3	42.6	48.8
Ours	LSAN	**84.6**	62.1	**62.6**	**62.3**	**86.7**	65.1	**67.2**	**66.1**	**83.1**	58.4	**61.6**	**59.9**

by RNNs or time-aware mechanism, and this sole pathway of learning the interaction among different visits cannot aggregate more complex temporal information. As for our method, firstly, LSAN has an extra attention mechanism in the hierarchy of diagnosis code to downplay the irrelevant diagnosis codes to the target disease. Such a consideration reduces the propagation of irrelevant results in the hierarchy of visit. Secondly, LSAN pays attention to modeling the temporal interactions in both short and long term between the visit embeddings by TAM, which is designed to model both the long-term dependencies and short-term correlations. It is the ability to model more complex temporal interactions that leads to the improvement of LSAN over existing methods.

Analysis of Precision, Recall and F1 score. In addition to AUC, our method also achieves better performance than existing methods in the metrics of recall and F1 score. As Table 2 shows, LSAN has enhancement of 9.9%, 3.6% and 2.6% in recall over the second best method on the Heart Failure, Kidney Disease and Dementia datasets, respectively. Recall is a metric that penalizes false negative in risk prediction. In the real life, patients are "false negative" if they are diagnosed negative but actually have the diseases, and it is life-threatening for them if they cannot receive imperative treatments due to the false prediction. Therefore, the high recalls yielded by our method show that the proposed LSAN has lower probability of diagnosing positive patients as negative, and it is more applicable for risk prediction task.

F1 score is the harmonic mean of precision and recall, and it is usually applied in datasets with imbalanced distribution such as our EHR datasets. We can see that LSAN has best F1 scores on the Heart Failure, Kidney Disease and Dementia datasets, despite the fact that SVM has the highest precisions. Such good performance again manifests that LSAN is a better deep learning model for the application of risk prediction.

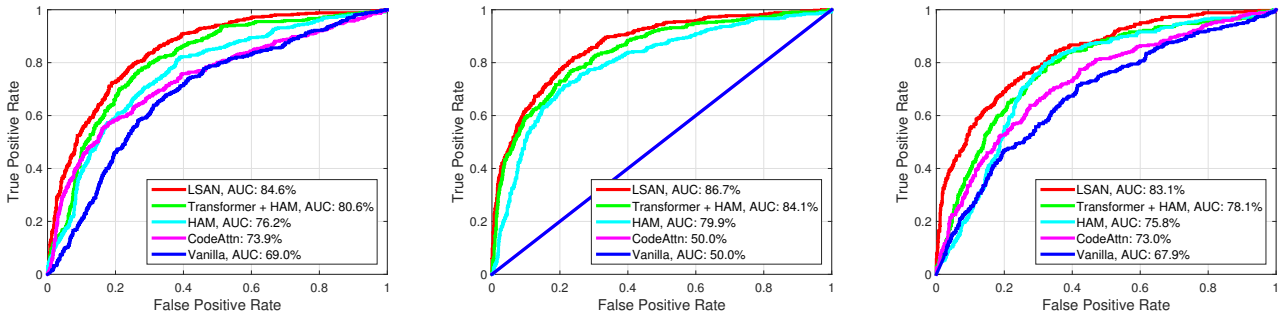
4.3 Ablation Study

We now need to examine the effectiveness of different components of our network. To this end, we conduct an ablation study on the datasets that we have mentioned. To determine whether the designed components improve the performance, we add them one by one from scratch and verify their performance by ROC-AUC statistic. The models that are used in our ablation study include:

- **Vanilla.** A simple method for risk prediction is to summarize all diagnosis code embeddings as the diagnostic representation of each visit, and then use the summarized embedding of these diagnostic representations over all visits for risk prediction. We refer this operation to Vanilla.
- **CodeAttn.** The basic framework of LSAN is laid down by the hierarchical attention mechanism we use in the hierarchy of diagnosis code and the hierarchy of visit. In this baseline, we replace the simple summarization operations of Vanilla in the hierarchy of diagnosis code by the attention operations as Eq. 3, Eq. 4 and Eq. 5 show, and we call this baseline as CodeAttn.
- **HAM.** We then replace the summarization operations of CodeAttn in the hierarchy of visit by the attention operations as Eq. 15, 16 and 17 show. This baseline is our HAM.
- **Transformer + HAM.** We also need to verify the effectiveness of TAM. Since TAM consists of Transformer and Conv, in this baseline, we only add Transformer on HAM in order to investigate the utility of Transformer.
- **LSAN.** Finally, we attach the Conv component to previous baseline and attain the complete model of LSAN.

We present the ablation study results on Figure 3 with ROC curves and list the AUC values in Table 3. In Table 3, the performance of the models described above is shown from row 1 to row 5 successively. Now we have a detailed analysis on the results.

Effect of HAM. Evidenced by the results in Figure 3 and the first 3 rows of Table 3, we can see that the HAM modules acts as the cornerstone of LSAN. As we can see from Table 3, Vanilla which simply summarizes diagnosis code embeddings for risk prediction has the lowest AUC values, so simple summarization operations cannot make full use of the dense representations. However, when these summarization operations are replaced with attention mechanism in the hierarchy of diagnosis code and visit by HAM, HAM achieves an 7.2%, 29.9% and 7.9% improvement in terms of AUC over Vanilla



(a) Heart Failure (b) Kidney Disease (c) Dementia

Figure 3: Comparison of ROC curve in different datasets in the ablation study.

Table 3: Ablation study results on AUC.

Model	HAM		TAM		Datasets		
	Diagnosis Code	Visit	Transformer	Conv	Heart Failure (%)	Kidney Disease (%)	Dementia (%)
Vanilla	✗	✗	✗	✗	69.0	50.0	67.9
CodeAttn	✓	✗	✗	✗	73.9	50.0	73.0
HAM	✓	✓	✗	✗	76.2	79.9	75.8
Transformer + HAM	✓	✓	✓	✗	80.6	84.1	78.1
LSAN	✓	✓	✓	✓	**84.6**	**86.7**	**83.1**

on the Heart Failure, Kidney Disease and Dementia datasets, respectively. In this comparison, the only difference between Vanilla and HAM is that HAM employs an attention mechanism in the hierarchy of diagnosis code and visit, which clearly confirms that the proposed hierarchical attention mechanism is effective.

By breaking down HAM into the hierarchy of diagnosis code and visit in the second and third row of Table 3, we can infer that both the attention computations in two hierarchies are necessary. We can see that the attention mechanism in the hierarchy of diagnosis code, *i.e.*, CodeAttn, can bring about 5% increment in AUC over Vanilla on the Heart Failure and Dementia datasets, and the one in the hierarchy of visit further improves the AUCs. Especially, note that Vanilla and CodeAttn both have an AUC of 50.0% on the Kidney Disease dataset, which shows that they cannot distinguish positive cases and negative ones on this dataset. However, with the help of the attention mechanism in the hierarchy of visit, HAM can achieve an AUC of 79.9% on the Kidney Disease dataset. Thus, the attention mechanism in both hierarchies are inseparable, and HAM is beneficial for digging out the potential of dense diagnosis code embedding for risk prediction.

Effect of TAM. We now analyze the utility of TAM, which is the other important module in our network design. TAM comprises two parts, namely Transformer and Conv. In the ablation study, we separate them to test their utility by first adding the Transformer module to the baseline HAM and adding the Conv to that later.

From Table 3, we can see that Transformer and Conv are compatible with each other. Compared to HAM, the introduction of Transformer for modeling the long-term dependencies between different visits enhances the AUC from 76.2%, 79.9% and 75.8% to 80.6%, 84.1% and 78.1% on the Heart Failure, Kidney Disease and Dementia datasets, respectively. Later, as Table 3 manifests, these AUC values on the three datasets can have further improvement by using Conv together with Transformer. The gradual enhancements in terms of AUC by adding Transformer and Conv show that these two components in TAM are compatible with each other.

A more interesting and important observation in Table 3 is that Transformer and Conv are complementary with each other. On the dataset of Kidney Disease, we can see that Transformer can bring an improvement of 4.2% in terms of AUC to HAM, while Conv can bring an improvement of 2.6% to the baseline Transformer + HAM. Conversely, on the dataset of Dementia, Conv can have an improvement of 5.0% while Transformer has a smaller improvement of 2.3% in terms of AUC. As we have discussed before, Transformer and Conv are designed for modeling different types of temporal interactions between visit embeddings. Now we can see that on different datasets, they bring improvement to different degrees. A possible explanation for this phenomenon is that sometimes long-term dependencies are more important than short-term correlations for risk prediction in some diseases, and vice versa for other ones. Therefore, it is vital to modeling both types of temporal interactions for risk prediction, and both Transformer and Conv are complementary to each other in learning these temporal interactions together.

In short, our ablation study results show that each component of the proposed LSAN enjoys good coexistence with one another, and integrating them together benefits the practice of risk prediction via deep learning models.

4.4 Model Interpretability and Robustness

We further examine the interpretability and robustness of our method with random examples selected from the datasets, which is

demonstrated in Figure 4. For one thing, LSAN enjoys good interpretability owing to the hierarchical attention mechanism in HAM. Through the attention weights in the hierarchy of diagnosis code, we can know which symptoms are paid more attention to during each visit for risk prediction. While through the attention weights in the hierarchy of visit, we learn about the importance of each visit for the decision making of LSAN. For another, the hierarchical attention mechanism also benefits the model robustness in the hierarchy of visit because of the comprehensive feature representation generated by HAM. Without loss of generality, we take the case of Heart Failure for detailed discussion.

From Figure 4(a), we can see that our model has good interpretability on the decision making of the heart failure disease. To be specific, in the first and third visits, HAM assigns the highest attention weights to the symptom of "unspecified voice and resonance disorder" (784.40) and "peritonsillar abscess" (475) respectively, which indicates that our model is able to extract the correlation between voice changes and heart failure [24]. Likewise, HAM pays greatest attention to the symptom of "pneumonia due to Methicillin susceptible Staphylococcus aureus" (482.41) in the second visit, which again shows that our model is able to attach greater attention to symptoms that are strongly related to heart failure like pneumonia [32]. As for robustness, when we remove the second visit from risk prediction, the probability of being diagnosed as positive will be down 4.43%, which shows that this visit contains important information for diagnosis. In addition, we can see that LSAN assigns small weights to irrelevant symptoms such as "erythema of upper limb" (943.10), "Sprain of lumbosacral" (846.0) and "aftercare for healing pathologic fracture of other bone" (V54.21). If we move these irrelevant diagnosis code, the probability of being diagnosed as positive remains the same as 0.662, so the prediction probability is barely affected by the removal of irrelevant diagnosis codes. We can infer from these results that the attention weight assignment of our model is based on the relatedness of symptoms to the disease rather than the date of visit, which contributes to the interpretability and robustness of the model.

Similarly, LSAN assigns the highest attention weights to the second visit due to the symptom of "gum recession" (523.20) in the case of kidney. It reflects the fact that poor oral health is common in patients of kidney disease [1]. As for the case in Dementia, the symptom of "mononeuritis of lower limb" (355.8) in the fourth visit are the most-weighted one, which demonstrates that Dementia affects the function of limbic and cortical structures [22]. We can also observe that the unrelated symptoms are assigned lower weights in these cases, so the case studies validate the interpretability and robustness of LSAN.

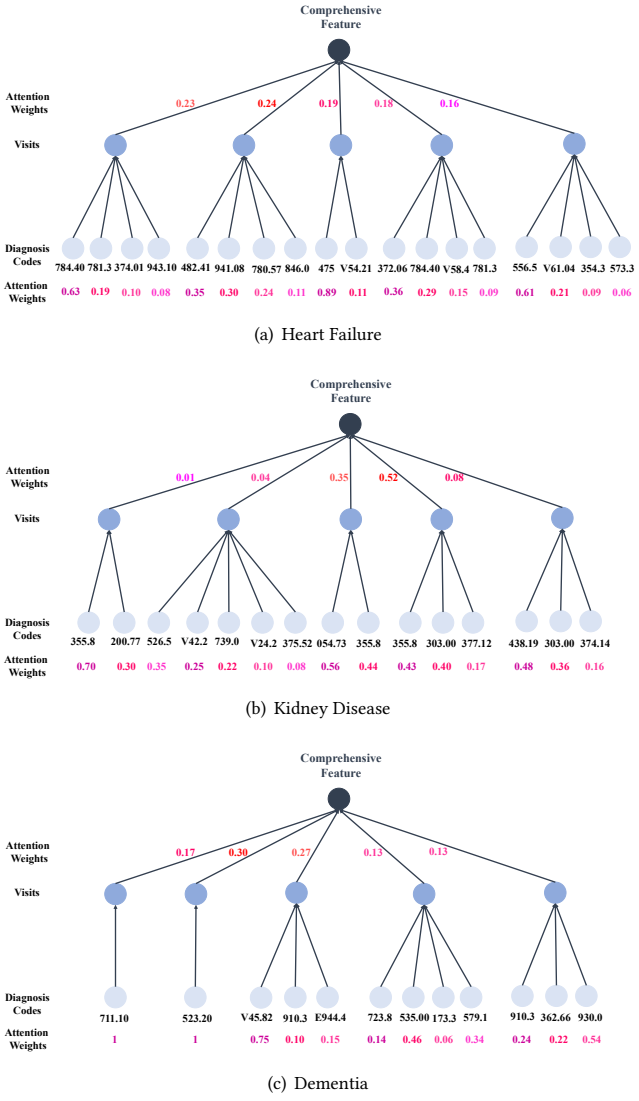


Figure 4: Illustration of model interpreterability and robustness for LSAN**. The patients have paid five visits to the doctors, and the diagnostic results in each visit are listed in the bottom hierarchy. Additionally, the attention weights learned by HAM are also shown in the illustration.**

5 CONCLUSIONS

In this paper, we introduce a novel deep neural network named LSAN for risk prediction task in EHR data mining. To tackle the issue of noise information in EHRs, LSAN decomposes the representation learning for EHR data into two hierarchies with HAM and TAM. In the hierarchy of diagnosis code, HAM learns a weighted sum of symptom embeddings according to their relevance to the disease in each visit. While in the hierarchy of visit, LSAN first uses TAM to aggregate each visit representation with the temporal information from long-term dependencies and short-term correlations by a two-pathway structure of Transformer encoder and convolutional layer. Then in the hierarchy of visit, HAM attends the temporally aggregated features to get a comprehensive feature for risk prediction by their significance to the diagnosis of the target disease. Experiments manifest that LSAN has higher AUCs, recalls and F1 scores against state-of-the-art models in the risk prediction task on three real-world EHR datasets, and it also has good interpretability and robustness thanks to the integration of HAM and TAM.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and suggestions. This work is supported in part by the seed grants from the College of Information Sciences and Technology (IST) and the Institute for Computational and Data Sciences (ICDS) at Pennsylvania State University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Pennsylvania State University.

REFERENCES

[1] Harun Akar, Gulcan Coskun Akar, Juan Jesús Carrero, Peter Stenvinkel, and Bengt Lindholm. 2011. Systemic consequences of poor oral health in chronic kidney disease patients. *Clinical Journal of the American Society of Nephrology* 6, 1 (2011), 218–226.

[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*. 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*. 65–74.

[4] 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.

[5] 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*. 787–795.

[6] 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.

[7] Edward Choi, Cao Xiao, Walter F. Stewart, and Jimeng Sun. 2018. MiME: Multilevel Medical Embedding of Electronic Health Records for Predictive Healthcare. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 4552–4562.

[8] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

[9] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. 2017. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 4 (April 2017), 677–691. https://doi.org/10.1109/TPAMI.2016.2599174

[10] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

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

[13] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[14] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[15] 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.

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

[17] 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*. 1903–1911.

[18] 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*. 1910–1919.

[19] 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.

[20] 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* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 743–752.

[21] 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.

[22] François Madore. 2009. Periodontal disease: a modifiable risk factor for cardiovascular disease in ESRD patients? *Kidney international* 75, 7 (2009), 672–674.

[23] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. 2018. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics* 19, 6 (2018), 1236–1246.

[24] Olivia M Murton, Robert E Hillman, Daryush D Mehta, Marc Semigran, Maureen Daher, Thomas Cunningham, Karla Verkouw, Sara Tabtabai, Johannes Steiner, G William Dec, et al. 2017. Acoustic speech analysis of patients with decompensated heart failure: A pilot study. *The Journal of the Acoustical Society of America* 142, 4 (2017), EL401–EL407.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[26] 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. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[27] 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.

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

[29] 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.

[30] Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters* 9, 3 (1999), 293–300.

[31] Qingxiong Tan, Andy Jinhua Ma, Mang Ye, Baoyao Yang, Huiqi Deng, Vincent Wai-Sun Wong, Yee-Kit Tse, Terry Cheuk-Fung Yip, Grace Lai-Hung Wong, Jessica Yuet-Ling Ching, et al. 2019. UA-CRNN: Uncertainty-Aware Convolutional Recurrent Neural Network for Mortality Risk Prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 109–118.

[32] Reimar W Thomsen, Nongyao Kasatpibal, Anders Riis, Mette Nørgaard, and Henrik T Sørensen. 2008. The impact of pre-existing heart failure on pneumonia prognosis: population-based cohort study. *Journal of general internal medicine* 23, 9 (2008), 1407.

[33] Cao Xiao, Edward Choi, and Jimeng Sun. 2018. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association* 25, 10 (2018), 1419–1428.

[34] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 2048–2057. http://proceedings.mlr.press/v37/xuc15.html

[35] 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)*. IEEE, 738–747.

[36] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. 2018. Auto-Conditioned Recurrent Networks for Extended Complex Human Motion Synthesis. In *International Conference on Learning Representations*.