MedSkim: Denoised Health Risk Prediction via Skimming Medical Claims Data

Suhan Cui, Junyu Luo, Muchao Ye, Jiaqi Wang, Ting Wang, Fenglong Ma
Pennsylvania State University
{suhan, junyu, muchao, jqwang, ting, fenglong}@psu.edu

2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00018

***Abstract*—Health risk prediction is a challenge task that aims to predict whether patients would suffer from a certain disease/condition in the near future based on their historical EHR data. Although existing approaches can achieve better performance, none of them can deal with the noise existing in the EHR data explicitly. In this paper, we hypothesize that automatically removing noise from EHR data should help the models further improve the performance. Correspondingly, we propose a novel model named `MedSkim`, which is able to automatically rule out irrelevant visits and codes by effectively skimming through the EHR data. In particular, the proposed model has a code selection module that can directly make a skipping decision to each individual diagnosis codes and then remove the target-irrelevant ones. A backward probing RNN (BPRNN) is designed to reversely process the EHR data and provide a coarse grained representation learning for visits. Besides, a forward skipping RNN (FSRNN) is proposed to read the EHR in a preceding way and dynamically select important visits and codes based on the results of previous two modules. Finally, the risk prediction module uses the output hidden states from FSRNN for generating the final representation to make predictions. Additionally, we also design an extra regularization term based on the skip rate of the model and combine it with standard cross entropy loss to train the model in an end-to-end setting. Experimental results show that `MedSkim` achieves the best performance on three real-world datasets compared with the state-of-the-art baselines in terms of PR-AUC, F1 and Cohen's Kappa. Moreover, the ablation study and case study confirm that the proposed `MedSkim` is reasonable and effective for removing noise from EHR data [1].**

***Index Terms*—Health Risk Prediction, Electronic Health Records, Denoising Algorithm**

I. Introduction

Deep learning techniques have been widely used in the medical domain to analyze comprehensive electronic health records (EHRs) in recent years. **Health risk prediction** is a representative task in the medical domain, which aims to predict patients' future health conditions based on analyzing their historical EHR data [1]. The commonly-used EHR data consist of a sequence of administrative *claims* encoded by medical code systems such as International Classification of Diseases (ICD) codes[2]. Such temporal, high-dimensional, discrete, and sparse EHR data make the design of health risk prediction models challenging.

Exiting work mainly focuses on applying recurrent neural networks (RNN) [2]–[7] and Transformer [8]–[10] to modeling the unique characteristics of EHR data. Besides, several works have been proposed to incorporate extra information such as time information associated with each visit [4], [5], [9], medical knowledge graph [11]–[13], medical rules [14], and medical text data [15], [16] to improve the health prediction performance of RNN or Transformer-based models. Although these models can further improve models' predictive power and interpretability, an open yet fundamental challenge in the health risk prediction task is still unsolved thoroughly, which is how to better handle the noise in the EHR data. To clearly demonstrate this challenge, we take a patient who will suffer from heart failure in the future as an example, and the patient's EHRs are shown in Figure 1.

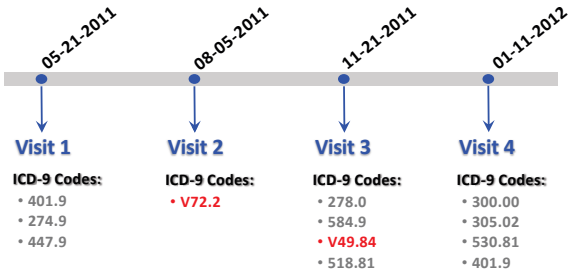


Fig. 1: An example of claims data of a patient who will suffer from heart failure in the future.

Visit-level noise. In this example, most visits are highly related to the target disease since they contain key risk factors of heart failure according to Mayo Clinic,[3] such as "401.9" (*unspecified essential hypertension*) in visit 1, "278.0" (*overweight and obesity*) in visit 3, and "305.02" (*alcohol abuse, episodic*) in visit 4. However, visit 2 only contains one ICD code "V72.2" representing *dental examination*,[4] which is largely irrelevant to our target disease. However, existing models still take all the visits as inputs to make a prediction. Even though they apply attention mechanisms [2], [3], [9], [10] to lower the weights for some visits, the noise will still be accumulated continuously during the model learning. To avoid this issue, an effective way is to directly skip such visits when models analyze EHR data.

Code-level noise. Besides the irrelevant visits among EHR data, even for each relevant visit, it still contains quite a

[1]The source code of the proposed `MedSkim` is available at https://github.com/SH-Src/MedSkim

[2]https://www.cdc.gov/nchs/icd/icd9.htm

[3]https://www.mayoclinic.org/diseases-conditions/heart-failure/symptoms-causes/syc-20373142

[4]http://www.icd9data.com/2015/Volume1/V01-V91/V70-V82/V72/V72.2.htm

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00018

few noisy diagnosis codes. For example, the third visit in Figure 1 contains a code "V49.84" (*bed confinement status*).[5] According to the Medicare Benefit Policy Manual from the Centers for Medicare & Medicaid Services (CMS), bed-confinement applies to those Medicare patients who are unable to tolerate any activity out of bed and may or may not, by itself, meet the requirement of an Paramedic or emergency medical technicians (EMT) monitoring him/her on their way to the hospital.[6] Obviously, this code does not explicitly or not even implicitly associate with the target heart failure. Therefore, recognizing and discarding these noisy codes from EHR data are of importance for further enhancing models' performance.

Our approach. To address the aforementioned challenges brought by EHR noise, in this paper, we propose a novel denoised risk prediction model named `MedSkim`, which can automatically identify both noise visits and codes within each visit and then directly discard them when the model **skim**s the **med**ical claims data. `MedSkim` consists of four key modules, including target-driven code selection, backward probing RNN, forward skipping RNN, and risk prediction.

Specifically, the **code selection** modules aims to remove target-irrelevant codes using Gumbel-Softmax [17] to learn a skipping indicator a_m for each individual code m. If the code is skipped, then it will be "removed" from all the visits. The backward probing RNN (BPRNN) and forward skipping RNN (FSRNN) will take the filtered visits as inputs. The **backward probing RNN** (BPRNN) tries to learn a coarse-grained feature representation $\mathbf{h}_n$ from the filtered visits in a reverse way, i.e., $\mathbf{h}_n = \text{BPRNN}([v_N, \cdots, v_{n+1}], \{a_m\}_{m=1}^{M})$, where N is the number of visits and M denotes the number of unique codes in the code set $\mathcal{C}$. The **forward skipping RNN** (FSRNN) will read the EHR data in a preceding way and learn a visit-level skipping indicator b_n for each visit v_n according to the embeddings of the remaining target-related codes, the target embedding, the time information, hidden state $\mathbf{h}_n$ learned from BPRNN, and the hidden state $\mathbf{s}_{n-1}$ outputted by FSRNN. If the visit is kept, `MedSkim` will update a new hidden state $\mathbf{s}_n$; otherwise, it will directly use the previous hidden state, i.e., $\mathbf{s}_n = \mathbf{s}_{n-1}$. Based on the learned hidden states $[\mathbf{s}_1, \cdots, \mathbf{s}_N]$ from FSRNN, the target embedding, and the time information, an attention mechanism is used to generate the final representation, which is used for making prediction in the **risk prediction** module.

In addition to automatically learning code-level and visit-level skipping indicators, we also take the skip rate into consideration, i.e., the percentage of skipped codes associated with visits. Finally, we design a **skip rate-based regularization** term working with the cross entropy loss generated from the risk prediction module. To sum up, our contributions are listed as follows:

- To the best of our knowledge, we are the first to deal with noise information in EHR data via learning skipping mechanisms, i.e., automatically removing target-irrelevant codes and visits simultaneously.
- Correspondingly, we propose a novel denoised health risk prediction model named `MedSkim` to skim EHR data using a bidirectional RNN structure, i.e., the backward probing RNN and the forward skipping RNN, over the denoised visits via the code selection module. Such a design not only enhances the prediction performance but also increases the efficiency.
- Experimental results on three real-world claims datasets show the proposed `MedSkim` model achieves a large performance improvement compared to state-of-art baselines and has excellent interpretability via a case study.

II. Related Work

A. Health Risk Prediction with Deep Learning

Many studies focus on modeling the temporal features of the EHR data and using the sequential models like RNN [18], [19] and Transformer [20] as the backbone model. Built upon the backbone models, some approaches consider to use other information such as visit time information and extra knowledge as well as advanced attention mechanisms [21] to further improve the prediction performance. Next, we briefly survey those state-of-the-art risk prediction models.

Basic Attention. Based on the naive recurrent structure, attention based enhancements are first proposed to improve the medical risk prediction tasks. Retain [2] is the first interpretable model for risk prediction. It learns visit-level weights and code level weights together with two independent RNNs. Compared to Retain, Dipole [3] tries to model longitudinal EHR data using bidirectional RNNs which have a stronger feature extraction ability. In addition, it applies more attention mechanisms and achieves a better result. SAnD [8] uses the self-attention [20] to calculate the importance of different visits. In `MedSkim`, we also utilize the attention mechanism and take advantages from the previous methods, learning the attention scores on multiple levels including diagnosis codes, visits and even the task.

Using Time Information. One important feature of the EHR records is that, the time gap between the visits are not identical. Thus, different adjacent visits actually have different gap time, and the differences can be huge. This may directly influence the final prediction task. As a result, many existing approaches try to take the time information into consideration. T-LSTM [22] is the first work proposed in this direction, which assumes that the patient information may decay as the time gap increases. RetainEX [4] and TimeLine [5] also adopt the similar ideas. LSAN [10] and HiTANet [9] apply a more open hypothesis on the time information, that is different diseases have different inner time models, and thus, we should let the model learn the time attention by itself. AdaCare [6] does not directly model the time information, but tries to model it using a multi-scale convolutions cores, which are set

[5]http://www.icd9data.com/2015/Volume1/V01-V91/V40-V49/V49/V49.84.htm

[6]https://www.cms.gov/Regulations-and-Guidance/Guidance/Manuals/Downloads/bp102c10.pdf

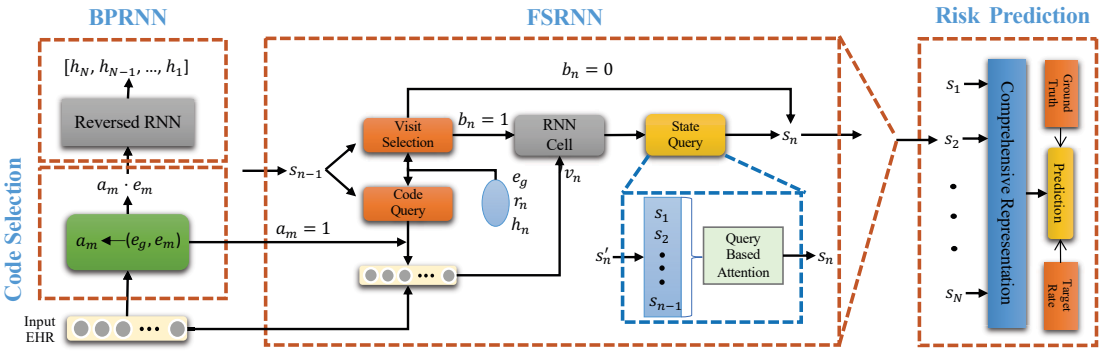


Fig. 2: Overview of the proposed MedSkim.

for different time-scales. Hence, different cores can represent different time scale information. In our work, we adopt the similar assumptions like LSAN and HiTANet, which is more flexible for task-specific optimization.

Using External Knowledge. On the other hand, some studies focus on the incorporation of the external knowledge on the prediction to improve the interpretation of the model. For example, the work [11], [15], [23] adopts the ICD hierarchy structure as external knowledge to enhance the code representations. The work of Ye et al. [16] applies the unstructured medical text to augment health risk prediction tasks. For the rest work, the majority solutions of using external knowledge is through the medical knowledge graph [12]–[14], [24] and use multi-sourced information as extra knowledge [25].

B. Selective Recurrent Neural Network

To address the long sequences data on the Natural Language Processing (NLP) domain, a special kind of RNN networks are developed. Compared to the normal RNN models, Selective Recurrent Neural Network has the ability to dynamically select parts of the data sequence as the input. In such a way, the total length of the processed sequence can be reduced. To achieve this goal, two lines of solutions are proposed. The first solution is to froze the state of the RNN models for unimportant steps, e.g., Variable Computation RNN (VCRNN) [26] and Skim-RNN [27]. The other methods choose to directly filter out the unimportant input sequence, such as LSTM-Jump [28], Skip RNN [29] and Leap LSTM [30], which need to predict whether to skip the next or multiple next inputs.

III. Methodology

The **EHR data** of each patient consists of multiple time-ordered visits $V = [(v_1, t_1), (v_2, t_2), \cdots, (v_N, t_N)]$, where N is the total number of visits. At each visit, a set of diagnosis codes is recorded, i.e., $v_n = [c_1^n, c_2^n, \cdots, c_M^n]$, where M represents the total number of unique codes in the dataset. $c_m^n = 1$ means the m-th code appears in the n-th visit; otherwise, $c_m^n = 0$. The **task** of health risk prediction is to predict whether the patient will suffer from the target disease g in the future according to the historical EHR data V.

A. Model Overview

Figure 2 shows the overall architecture of the proposed MedSkim, which consists of four major modules, i.e., code selection, backward probing RNN (BPRNN), forward skipping RNN (FSRNN), and risk prediction. The **code selection** module aims to filter out noisy diagnosis codes by learning a code level action a_m using the target embedding $\mathbf{e}_g$ and the code embedding $\mathbf{e}_m$. $a_m = 1$ indicates that the code will be kept; otherwise, it will be removed from visits. Both BPRNN and FSRNN will take the denoised visits as inputs. The **backward probing RNN** (BPRNN) module aims at learning hidden states $[\mathbf{h}_N, \cdots, \mathbf{h}_1]$ for denoised visits using a reversed RNN. The **forward skipping RNN** (FSRNN) first learns a visit-level action b_n for each denoised visit v_n using the previous output $\mathbf{s}_{n-1}$ from FSRNN, the target embedding $\mathbf{e}_g$, the time embedding $\mathbf{r}_n$, and the learned hidden state $\mathbf{h}_n$ from BPRNN. If $b_n = 0$, then the cell of FSRNN will not be updated, and $\mathbf{s}_n = \mathbf{s}_{n-1}$. Otherwise, FSRNN will take denoised v_t as input to generate $\mathbf{s}_n$. Finally, the **risk prediction** module uses outputs from FSRNN to make predictions. Since MedSkim is designed to automatically skip target-irrelevant information, a *skip rate-based regularization term* is then attached to the classification loss as the final loss function.

B. Code Selection

The main aim of this paper is to predict the target disease g, and the design of the double RNN mechanism is to extract key features from two directions. To enhance the probing speed and efficiency, we only take target-related diagnosis codes as the inputs of two RNNs. The challenge here is how to identify target-related diagnosis codes. An easy way is to introduce extra knowledge such as a medical knowledge graph to select target-related diagnosis codes. However, such an approach may miss some codes that are implicitly related to the target. To address this problem, we propose a code selection mechanism to automatically identify important codes as follows.

• **Code and Target Embedding.** To identify target-related codes, we first map the m-th diagnosis code in the code set $\mathcal{C}$ into a high dimensional embedding $\mathbf{e}_m \in \mathbb{R}^{d_e}$. The target disease g is also mapped to the same space, which is denoted as $\mathbf{e}_g \in \mathbb{R}^{d_e}$. Note that the code embeddings and the target

embedding are shared across all the visits even for different patients.

• **Target-driven Code Selection.** Based on the learned code embeddings $[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_M]$ and the target embedding $\mathbf{e}_g$, `MedSkim` can automatically identify target-related codes using Gumbel-Softmax [17] over a one-layer feedforward network:

$$\begin{aligned} a_m &= \text{Binarize}\Big(\frac{\exp((\log(\mathbf{p}_m[0]) + g_0)/\tau)}{\sum_{j=0}^{1}\exp((\log(\mathbf{p}_m[j]) + g_j)/\tau)}\Big), \\ \mathbf{p}_m &= \text{Softmax}(\mathbf{W}_p[\mathbf{e}_m, \mathbf{e}_g] + \mathbf{b}_p), \end{aligned} \tag{1}$$

where $\mathbf{p}_m \in \mathbb{R}^2$ is a probability distribution indicating the relevance of the m-th code, τ is the softmax temperature, g_j is i.i.d samples drawn from Gumbel distribution $\text{Gumbel}(0, 1)$, and $[\cdot, \cdot]$ means the operation of concatenation. $\mathbf{W}_p \in \mathbb{R}^{2\times(2*d_e)}$ and $\mathbf{b}_p \in \mathbb{R}^2$ are parameters. $a_m = 1$ indicates that the m-th code is selected; otherwise, this code will be removed from the inputs. With the reparameterization trick of Gumbel Softmax sampling, the gradients can be approximated using the soft value of a_m.

C. Backward Probing RNN (BPRNN)

The goal of `MedSkim` is to dynamically skip irrelevant visits and codes to improve both the prediction performance and efficiency. However, only using the forward RNN may discard some key information, which may be highly related to the future visits. To avoid this issue, we propose a novel backward probing RNN (shorten for BPRNN) to extract key features from the following visits. Next, we introduce the design of BPRNN.

• **Backward Visit Embedding.** In Section III-B, we can automatically learn a global selection indicator a_m for each diagnosis code. When learning the embedding of each visit v_n, we only consider the selected codes as follows:

$$\mathbf{x}_n = \sum_{m=1}^{M} a_m * c_m^n * \mathbf{e}_m, \tag{2}$$

where $\mathbf{x}_n \in \mathbb{R}^{d_e}$ is the visit embedding.

• **Backward Hidden State Updating.** Based on the learned embeddings $[\mathbf{x}_1, \cdots, \mathbf{x}_N]$, we can generate the hidden state for each visit via a reversed long short-term memory (LSTM) network [18], i.e.,

$$[\mathbf{h}_N, \cdots, \mathbf{h}_1] = \text{BPRNN}[\mathbf{x}_N, \cdots, \mathbf{x}_1], \tag{3}$$

where $\mathbf{h}_n \in \mathbb{R}^{d_h}$ is the hidden state of the n-th visit. Note that to speed up the learning, we usually set d_h as a small number.

D. Forward Skipping RNN (FSRNN)

Based on the hidden states obtained by BPRNN, we further design a Forward Skipping RNN (FSRNN) to dynamically skip irrelevant visits and codes considering both historical and future visits information. Additionally, time information is also essential for identifying key information from EHR data [5], [9], [22]. Therefore, our design of FSRNN comprehensively analyzes the bidirectional and time information to make reasonable visit-level and code-level skipping.

In particular, FSRNN first analyzes each visit information and learns an action to decide whether this visit will be as the input of the RNN cell or directly skipped. If the visit is selected, then FSRNN learns an embedding for each visit by considering code importance, target embedding, and probing feature learned by Eq. (3). Finally, FSRNN updates the hidden states based on the learned action and visit embeddings. Next, we describe the details of FSRNN.

1) Time-aware Visit Selection: In EHR data, only a few of visits are of great importance for the final prediction. In other words, many visits can be considered as noise. Current approaches for dealing with noisy EHR data usually apply attention mechanism [2], [3], [9], [10], i.e., automatically assigning lower weights for those irrelevant visits. Although these approaches can improve the prediction performance, the existence of irrelevant visits still affects the learning of patient representations. Thus, to further enhance the prediction ability, we propose to select the important visits first and then only use the selected ones to learn patient representations.

• **Time-aware Visit Embedding.** Intuitively, the importance of a visit v_n is mainly determined by its embedding $\mathbf{x}_n$ learned by Eq. (2). However, as we mentioned before, time information t_n also plays a key role in the prediction. Thus, following [9], we embed the time information using the time interval Δt_n between the current time t_n and the last recorded time t_N, i.e., $\Delta t_n = t_N - t_n$, as follows:

$$\begin{aligned} \mathbf{f}_n &= \mathbf{1} - \tanh((\mathbf{W}_f\frac{\Delta t_n}{180} + \mathbf{b}_f)^2), \\ \mathbf{r}_n &= \mathbf{W}_r\mathbf{f}_n + \mathbf{b}_r, \end{aligned} \tag{4}$$

where $\mathbf{W}_f \in \mathbb{R}^{d_f}$, $\mathbf{W}_r \in \mathbb{R}^{d_e\times d_f}$, $\mathbf{b}_f \in \mathbb{R}^{d_f}$ and $\mathbf{b}_r \in \mathbb{R}^{d_e}$ are all parameters. In this way, all the time information can be embedded into the latent space, which is the same as that of the visit embedding $\mathbf{x}_n$. Thus, the time-aware visit embedding $\mathbf{z}_n \in \mathbb{R}^{d_e}$ can be represented by

$$\mathbf{z}_n = \mathbf{x}_n + \mathbf{r}_n. \tag{5}$$

• **Visit Selection.** In addition to the time-aware visit embedding $\mathbf{z}_n$, the target g, the following visits $[v_{n+1}, \cdots, v_N]$, as well as the previous visits $[v_1, \cdots, v_{n-1}]$ are also key factors to decide whether to skip the current visit v_n or not. Thus, we need to consider them all together. Similar to Eq. (1), we use the Gumbel-Softmax over another feedforward layer to learn the visit selection action b_n as follows:

$$\begin{aligned} \mathbf{d}_n &= \text{Softmax}(\mathbf{W}_d[\mathbf{z}_n, \mathbf{e}_g, \mathbf{h}_n, \mathbf{s}_{n-1}] + \mathbf{b}_d), \\ b_n &= \text{Binarize}\Big(\frac{\exp((\log(\mathbf{d}_n[0]) + g_0)/\tau)}{\sum_{j=0}^{1}\exp((\log(\mathbf{d}_n[j]) + g_j)/\tau)}\Big), \end{aligned} \tag{6}$$

where $\mathbf{W}_d \in \mathbb{R}^{2\times(2*d_e+d_h+d_s)}$ and $\mathbf{b}_d \in \mathbb{R}^2$ are parameters to be learned. $\mathbf{h}_t \in \mathbb{R}^{d_h}$ is the following feature extracted from BPRNN using Eq. (3), $\mathbf{e}_g$ is the target embedding, and $\mathbf{s}_{n-1} \in \mathbb{R}^{d_s}$ is the previous step hidden state of FSRNN that will be introduced in Section III-D3, respectively. A distribution $\mathbf{d}_n \in$

$\mathbb{R}^2$ is obtained for each visit, indicating whether to skip the current visit, from which we can also sample an action b_n. $b_n = 0$ indicates that the current visit v_n will be skipped, and $b_n = 1$ means FSRNN will take v_n as an input. In such a way, the gating module considers bidirectional context information and also the global target disease so that it can make the most reasonable decision.

2) Forward Visit Embedding: After making the visit-level skip decision, as long as the decision is to keep the current visit (i.e., $b_n = 1$), FSRNN will take v_n as the input to update the hidden state. An easy way is to directly use $\mathbf{z}_n$ learned by Eq. (5) as the visit embedding. However, it treats each selected diagnosis code equally. However, previous work [10] shows that assigning different weights to diagnosis codes can improve the prediction performance. Thus, we propose a query-based attention mechanism to assign weights to diagnosis codes within each visit. In particular, upon the global code selection introduced in Section III-B, we also conduct visit-level or local code selection again. Based on the learned attention weights and the selected codes, we can obtain the visit embedding.

• **Code Query Generation.** The query used for learning the importance of each diagnosis code should contain contextual information of the whole visits and the target information. Thus, we choose to concatenate the previous hidden state of the FSRNN cell $\mathbf{s}_{n-1}$, the following features extracted by BPRNN $\mathbf{h}_n$ along with the target embedding $\mathbf{e}_g$, and apply a linear projection to the concatenated features as follows:

$$\mathbf{q}_n = \mathrm{ReLU}(\mathbf{W}_q[\mathbf{s}_{n-1}, \mathbf{h}_n, \mathbf{e}_g] + \mathbf{b}_q), \tag{7}$$

where $\mathbf{q}_n \in \mathbb{R}^{d_e}$ is the contextualized query, and $\mathbf{W}_q \in \mathbb{R}^{d_e \times (d_s + d_h + d_e)}$, $\mathbf{b}_q \in \mathbb{R}^{d_e}$ are both parameters of the linear projection. We also use an ReLU activation function here to keep the positive values.

• **Query-based Code Attention.** Using the learned query $\mathbf{q}_n$, we can assign attention scores to diagnosis codes to get the importance of them as follows:

$$\boldsymbol{\alpha}^n = \mathrm{Softmax}(\frac{\mathbf{q}_n^\top [c_1^n * \mathbf{e}_1; c_2^n * \mathbf{e}_2; \cdots; c_M^n * \mathbf{e}_M]}{\sqrt{d_e}}), \tag{8}$$

where $\boldsymbol{\alpha}^n \in \mathbb{R}^M$ is a distribution over all the diagnosis codes. If $c_m^n = 0$, then the corresponding weight will be 0. $[\cdot;\cdot]$ denotes the column-wise stack operation.

• **Attention-based Visit Embedding.** Based on the learned attention weights $\boldsymbol{\alpha}^n$ using Eq. (8), we can directly learn the visit embedding by applying weighted summation on all the code embeddings. However, as we discussed in Section III-B, only a few diagnosis codes are highly related to the final prediction. The global code action a_m can help us to filter out a subset of target-irrelevant codes, which should be considered when learning the visit embedding. Besides, the time embedding $\mathbf{r}_n$ associated with each visit is an essential factor. Based on these embeddings, we can obtain the final visit embedding $\mathbf{v}_n \in \mathbb{R}^{d_e}$ as follows:

$$\mathbf{v}_n = \mathbf{r}_n + \sum_{m=1}^{M} \alpha_m^n * a_m * c_m^n * \mathbf{e}_m. \tag{9}$$

3) Forward Hidden State Updating: As mentioned in Section III-D1, the visit v_n may be skipped if $b_n = 0$ estimated by Eq. (6). In such a way, FSRNN does not need to update the hidden state for v_n and directly uses the previous $\mathbf{s}_{n-1}$. Otherwise, it will update the hidden state $\mathbf{s}_n$ based on the previous hidden states $[\mathbf{s}_1, \cdots, \mathbf{s}_{n-1}]$ and the visit embedding $\mathbf{v}_n$ learned by Eq. (9), i.e.,

$$\mathbf{s}_n = \begin{cases} \mathrm{FSRNN}([\mathbf{s}_1, \cdots, \mathbf{s}_{n-1}], \mathbf{v}_n), & \text{if } b_n = 1; \\ \mathbf{s}_{n-1}, & \text{if } b_n = 0. \end{cases} \tag{10}$$

Next, we introduce how to learn the hidden state $\mathbf{s}_n$ when $b_n = 1$. Intuitively, we can directly follow the standard RNN model to generate $\mathbf{s}_n$ using the FSRNN cell taking $\mathbf{s}_{n-1}$ and $\mathbf{v}_n$ as the input. However, existing work [3] points out that the key information may be forgotten especially when the number of visits is large. To address this issue, similar to Eq. (8), we propose to use query-based attention again to assign an attention score to each previous hidden state $\mathbf{s}_j$ ($j = 1, \cdots, n-1$). We then aggregate all the hidden states with their attention weights to generate the current hidden state $\mathbf{s}_n$.

• **State Query Generation.** To generate the hidden state query $\mathbf{k}_n \in \mathbb{R}^{d_s}$, we propose to only use the current visit embedding $\mathbf{v}_n$ and $\mathbf{s}_{n-1}$, i.e.,

$$\mathbf{k}_n = \mathrm{FSRNNCell}(\mathbf{s}_{n-1}, \mathbf{v}_n). \tag{11}$$

• **Query-based State Attention.** Similar to query-based code attention mechanism, we can generate the attention weight γ_j^n for each previous hidden state $\mathbf{s}_j$ as follows:

$$\boldsymbol{\gamma}^n = \mathrm{Softmax}(\frac{\mathbf{k}_n^\top [\mathbf{s}_1; \mathbf{s}_2; \cdots; \mathbf{s}_{n-1}]}{\sqrt{d_s}}). \tag{12}$$

• **Forward Hidden State Generation.** Based on the learned attention weights in Eq. (12) and the query vector $\mathbf{k}_n$ in Eq. (11), we can finally generate the hidden state as follows:

$$\begin{aligned} \mathbf{s}_n &= \mathbf{W}_\mathbf{s}(\mathrm{ReLU}(\mathbf{W}_k \mathbf{s}'_n + \mathbf{b}_k)) + \mathbf{b}_s, \\ \mathbf{s}'_n &= \mathbf{k}_n + \sum_{j=1}^{n-1} \gamma_j^n * \mathbf{s}_j, \end{aligned} \tag{13}$$

where $\mathbf{W}_k \in \mathbb{R}^{(4*d_s) \times d_s}$, $\mathbf{b}_k \in \mathbb{R}^{4*d_s}$, $\mathbf{W}_s \in \mathbb{R}^{d_s \times (4*d_s)}$, and $\mathbf{b}_s \in \mathbb{R}^{d_s}$ are all parameters.

From Eq. (13), we can observe that all the historical hidden states are stored into a dynamic memory. At each step, we generate a query $\mathbf{k}_n$ to perform attention operation over the history hidden states and retrieve important information from the stored memory. Then, we apply a feed forward network to further transform the features. In this way, we can get a comprehensive hidden state representation $\mathbf{s}_n$ that contains all the important historical information. Eventually, we attend $\mathbf{s}_n$ to the memory $[\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n]$, which can be dynamically maintained at every step and provide more comprehensive visit features for the prediction.

E. Risk Prediction

The outputs from the FSRNN are a sequence of hidden states from the stored memory $[\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N]$. They are used to generate a comprehensive representation for the patient, which is further used to make final prediction.

Towards this goal, we first use the target embedding $\mathbf{e}_g$ and time encoding to determine the importance of each visit feature and fuse them together effectively. We achieve this by using a softmax function over a feedforward layer to generate the weights for each visit as follows:

$$\begin{aligned}[\phi_1, \phi_2, \cdots, \phi_N] &= \mathrm{Softmax}(l_1, l_2, \cdots, l_N), \\ l_n &= \mathbf{W}_\phi^\top [\mathbf{s}_n, \mathbf{r}_n, \mathbf{e}_g] + b_\phi, \forall n \in [1, \cdots, N],\end{aligned} \tag{14}$$

where $\mathbf{W}_\phi \in \mathbb{R}^{d_s + 2*d_e}$ and $b_\phi \in \mathbb{R}$ are parameters. After obtaining the attention weight for each visit feature, we can fuse them together to get the comprehensive feature representation $\mathbf{u} \in \mathbb{R}^{d_s}$ for the whole EHR as follows:

$$\mathbf{u} = \sum_{n=1}^{N} \phi_n \mathbf{s}_n, \tag{15}$$

We further apply an output classifier to get the final prediction, i.e.,

$$\hat{\mathbf{y}} = \mathrm{Softmax}(\mathbf{W}_y \mathbf{u} + \mathbf{b}_y), \tag{16}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^2$ is the output classification distribution, $\mathbf{W}_y \in \mathbb{R}^{2 \times d_s}$ and $\mathbf{b}_y \in \mathbb{R}^2$ are parameters.

F. Loss Function

To train the proposed `MedSkim`, we can directly use traditional cross entropy function. However, this simple loss function cannot handle the *skip rate*, i.e., the ratio of how many diagnosis codes and visits that will be skipped. To explicitly control the skip rate, we follow the previous work [30] to add a regularization term with regard to the skip rate.

Let θ' be a hyperparameter that represents the desired skip rate, and θ be the actual skip rate, which is the percentage of skipped codes among all the visits, i.e.,

$$\begin{aligned}\theta &= \frac{\sum_{n=1}^{N} \sum_{m=1}^{M} [b_n * (c_m^n - a_m) + (1 - b_n) * c_m^n]}{\sum_{n=1}^{N} \sum_{m=1}^{M} c_m^n} \\ &= \frac{\sum_{n=1}^{N} \sum_{m=1}^{M} (c_m^n - b_n * a_m)}{\sum_{n=1}^{N} \sum_{m=1}^{M} c_m^n},\end{aligned} \tag{17}$$

where a_m is the global code selection via Eq. (1), and b_n is the visit selection action from Eq. (6). We can observe that if $b_n = 0$, all the codes within the n-th visit will be accounted. When $b_n = 1$, only the codes filtered by global code selection (i.e., $a_m = 0$) will be accounted. Finally, we define our loss function as follows:

$$\mathcal{L} = CrossEntropy(\mathbf{y}, \hat{\mathbf{y}}) + \lambda(\theta' - \theta)^2, \tag{18}$$

where $\mathbf{y}$ is the ground truth vector, and λ is a predefined hyperparameter to control the weight of the regularization term.

TABLE I: Statistics of the used claim datasets.

Dataset	Heart Failure	COPD	Amnesia
Positive Cases	3,080	7,314	2,982
Negative Cases	9,240	21,942	8,946
Average Visits per Patient	38.74	30.39	72.52
Average Codes per Visit	4.24	3.50	2.53
Unique ICD-9 Codes	8,692	10,053	9032

IV. Experiments

A. Experimental Setup

1) Datasets: In our experiments, we conduct retrospective analysis on three common chronic and progressive health conditions, which are Heart Failure, Chronic Obstructive Pulmonary Disease (COPD), and Amnesia. The corresponding EHR data including positive cases and negative/control cases are extracted from a real-world claims database, with the guidance of clinicians. When extracting the data for positive cases, we identify *the first disease diagnosis date* and then only keep the EHR data before six months of that date. For each positive case, we extract three control cases based on gender, age, race, and underlying diseases. We keep the whole EHR data for negative/control cases. The statistics of these datasets are shown in Table I.

2) Baselines: We consider both state-of-the-art health risk prediction models and selective RNNs as baselines. **Risk prediction models** include:

- LSTM [18] is the basic baseline, which first embeds visits and then feeds them to the LSTM cell to generate hidden states for making predictions.
- Dipole [3] is an RNN-based risk prediction model that apply attention mechanism to perform visit analysis on top of bidirectional GRU, which can use attention weights to determine the importance of each visit.
- Retain [2] applies reversed RNN model to conduct attentional analysis in both visit level and variable level by mimicking the physicians' diagnosis, which typically focuses more on recent visits.
- SAnD [8] is an early Transformer based risk prediction model that uses dense interpolation strategies to fuse visits information by incorporating temporal order.
- AdaCare [6] applies dilated convolutional neural network to extract the visit features from EHR data in multiple scales and further uses a GRU model to conduct risk prediction.
- RetainEx [4] is an extension work of Retain, which incorporates time information into the visit feature inputs of retain, which can enhance the original model to achieve better performance.
- Timeline [5] is also an RNN based model that designs a time-aware mechanism by modeling the time decaying factor of each diagnosis code to improve the representation learning of EHR data.
- T-LSTM [22] designs a time-aware mechanism based on LSTM that is able to handle irregular time intervals in longitudinal patient records.

- LSAN [10] is a recent Transformer based model that develops a hierarchical attention mechanism to jointly consider the long term and short term dependencies of EHR data.
- HiTANet [9] is the state-of-the-art Transformer-based risk prediction model that designs a time-aware attention mechanism to capture the dynamic disease progression patterns from EHR data.

We also choose two representative **selective RNNs** as baselines, we first transform the EHR data into a sequence of visit embedding and then feed it into these models:

- Leap-LSTM [30] is a step-wise selective RNN that uses a gating module to extract messages from preceding texts, following texts and the current word, and then determines whether to skip the current word.
- Skim-RNN [27] designs a partial updating mechanism for the RNN cell. It has an additional small RNN and could only update partial hidden states using the small RNN at unimportant steps.

Note that there are two commonly-used selective RNNs, which are LSTM-Jump [28] and Skip RNN [29]. However, they are not suitable as baselines for the risk prediction task because they design mechanisms to skip multiple consecutive steps at one time. For EHR data, skipping multiple visits would cause the training process to be unstable, since there are no strong associations among consecutive visits (i.e. a patient happened to go the hospital for an irrelevant disease among his multiple visits about an important disease) for those models to make reasonable jumping.

3) Implementation: We implement the proposed model by the PyTorch framework and run it on an NVIDIA RTX A6000 GPU. The parameters are trained by Adam optimizer [31] with the learning rate of 10^{-4} and weight decay of 10^{-3}, and the mini-batch size is set to 64. In `MedSkim`, the EHR embedding d_e and hidden state dimensions of the forward skipping RNN d_s are all set to 256. For the Back Probing RNN, the hidden state size d_h is set to 32. The temperature for Gumble Softmax sampling τ is set to 1.0. We set the weight of skip rate regularization term λ as 0.1, For the desired skip rate θ', we set different values on different datasets. $\theta' = 0.5$ on the heart failure dataset, $\theta' = 0.3$ on both COPD and Amnsia datasets. We will discuss the effect of λ and θ' on the performance in the following experiments. We implement the baselines on the same platform with the proposed model and apply the same optimization settings. We use standard cross-entropy loss for all baselines. The numbers of hidden state of baselines are all 256 no matter for RNN or Transformer based models.

Moreover, we randomly partition the datasets into training set, validation set, and test set in the ratio of 0.75:0.10:0.15. We select the best model based on the performance on the validation set, and we run the algorithms **five times** and report the mean results for performance evaluation.

4) Evaluation Metrics: To fairly compare the proposed model with baselines, we use *PR-AUC* (area under the precision-recall curve), *F1 score*, and *Cohen's Kappa* as the evaluation metrics. The reason of choosing these metrics is that our datasets have imbalanced class distributions as shown in Table I. All three metrics could take class imbalance into consideration and provide more reasonable performance measurements for this task.

B. Performance Evaluation

In Table II, we show the experimental results of all baselines and the proposed model on three datasets under three evaluation metrics. All the results are average results of five-round experiments. We can observe that the proposed `MedSkim` achieves the best performance on three datasets in terms of three metrics. Additionally, we conduct **significance testing (t-test)** to justify that our model can bring significant improvement over the best baseline (i.e., HiTANet), and all p-values are less than 0.01.

We first analyze the results of the selective RNNs. From the result we can find that compared to the naive LSTM, only the Skim-RNN achieves a positive improvement on the experiment results. Compared to the Leap-LSTM, the Skim-RNN does not skip visits, but it uses a partial mechanism to update the unimportant visits. It is a softer method compared to the directly skipping. As a result, it has a better generalization ability on the medical risk prediction task. However, both methods do not take into account the special features of the EHR data, and hence, they can only achieve limited improvements.

For attention-based risk prediction methods that do not include the time information, the performance is similar and with relative low scores, including Dipole, Retain, and SAnD. Adacare, RetainEx, Timeline, T-LSTM, LSAN and HiTANet are much better compared to non-time information models. Among them, LSAN and HiTANet are both designed with more advanced hierarchical attention mechanisms based on Transformer and could achieve better results than others. However, they only use soft attention weights to lower the unimportant visits and codes, which is not effective enough to rule out the noisy information.

As for `MedSkim`, it achieves the best performance on these three datasets measured by these three metrics. With the modeling of time information, we can find significant improvement on the final performance. `MedSkim` can dynamically perform visit skipping and code skipping, which could explicitly select important visits and codes from original EHR. Thus, our model could outperform the two strong baselines.

C. Ablation Study

Since the proposed `MedSkim` contains several key and effective components, to thoroughly validate the influence of each component for the final predictions, we conduct the following ablation studies:

- **Skip-rate Regularization**: we remove the second term in Eq. (18) and only use the cross entropy loss to train the model.

TABLE II: Performance comparison in terms of PR-AUC, F1 score, and Cohen's Kappa. Statistical significance of pairwise differences of MedSkim against the best baseline (*) is determined by the t-test ($p < 0.01$).

Dataset		Heart Failure			COPD			Amnesia		
Metrics		PR-AUC	F1	Kappa	PR-AUC	F1	Kappa	PR-AUC	F1	Kappa
Plain and Selective RNNs	LSTM	0.5498	0.5958	0.4392	0.5534	0.5596	0.4178	0.5536	0.6114	0.4838
	Leap-LSTM	0.5845	0.5580	0.4001	0.5227	0.5368	0.3916	0.4932	0.5596	0.4178
	Skim-RNN	0.6309	0.6100	0.4536	0.6265	0.5792	0.4492	0.6024	0.5978	0.4612
Attention based Models	Dipole	0.5680	0.5884	0.4302	0.5870	0.5618	0.4218	0.5804	0.6016	0.4646
	Retain	0.5390	0.4996	0.3488	0.5356	0.5096	0.3746	0.5604	0.5506	0.4348
	SAnD	0.5374	0.5538	0.3942	0.5170	0.5212	0.3766	0.5250	0.5638	0.4168
Time-aware Models	Adacare	0.6074	0.5746	0.4298	0.6050	0.5508	0.4234	0.5968	0.6068	0.4784
	LSAN	0.6824	0.6010	0.4638	0.6384	0.5498	0.4352	0.6816	0.6412	0.5288
	RetainEx	0.6246	0.5406	0.4108	0.6052	0.5404	0.4344	0.6344	0.5892	0.4906
	Timeline	0.6018	0.5708	0.4210	0.5486	0.4902	0.3640	0.5646	0.5824	0.4552
	T-LSTM	0.6466	0.6240	0.4889	0.6862	0.6292	0.5155	0.6319	0.6291	0.5108
	HiTANet	0.6756	0.6470	0.5219	0.6846	0.6370	0.5178	0.7080	0.6540	0.5328
	MedSkim	**0.7238***	**0.6717***	**0.5433***	**0.6932***	**0.6372***	**0.5201***	**0.7309***	**0.6685***	**0.5507***

TABLE III: Ablation study results in term of PR-AUC when removing each component from MedSkim.

Dataset	Heart Failure	COPD	Amnesia
MedSkim	**0.7238**	**0.6932**	**0.7309**
without Skip-rate Regularization	0.7136	0.6845	0.7115
without Backward Probing RNN	0.7082	0.6607	0.7101
without Code Selection	0.6847	0.6825	0.7034
without Visit Selection	0.6786	0.6573	0.7052
without State Query Attention	0.6821	0.6547	0.7065

- **Backward Probing RNN**: when we conduct visit-level selection and code query, we do not consider the features outputted by Eq. (3) in BPRNN anymore.
- **Target-Driven Code Selection**: we remove this module by setting $a_m = 1$ in Eq. (1) for all diagnosis codes.
- **Visit Selection**: at each visit, we set all actions as 1, i.e., $b_n = 1$ in Eq. (6). In such a way, we can use all visits to conduct risk prediction.
- **State Query Attention**: we remove the state query attention in Eq. (12), (13) and directly use the output hidden state of FSRNN cell $\mathbf{k}_n$ without aggregating from all historical visits.

Table III shows the results of ablation study. We can observe that removing any these components from the proposed MedSkim will lead to the drop of performance. These results clearly show the effectiveness and reasonableness of each designed component. In the meanwhile, we can find that the contribution of different modules to different tasks are variant.

Without using the skip-rate regularization term, the performance slightly drops, but it is still better than that of baselines as shown in Table II. The use of this regularization term is to control the skip rate. Even discarding this term, the proposed MedSkim can still automatically skip target-unrelated codes and visits. Under this scenario, the actual skip rates are 0.2625, 0.4395 and 0.2818 for heart failure, COPD and Amnesia, respectively. We also can observe that the designed backward probing RNN is helpful to enhance the performance since it can extract some useful information from the future visits. The designed state query attention also improve the performance for its ability to memorize important information from historical visits.

The main hypothesis of this work is that filtering out target-irrelevant information in EHR can improve the performance for the risk prediction task. The results of the two ablation studies, including without code selection and without visit selection, clearly validate our hypothesis. We can see that removing these key components from MedSkim makes the performance drops a lot. Thus, our design is reasonable, and each component has its own contributions for the performance improvement.

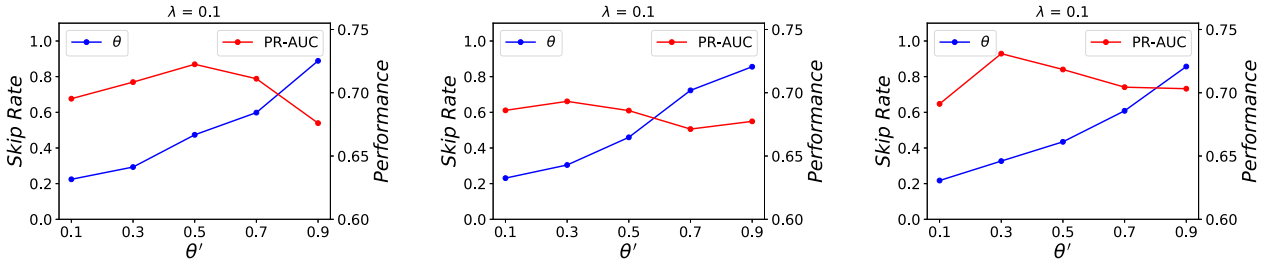
D. Case Study

In this section, we conduct further analysis of the proposed model. In order to show the interpretability of the proposed model and its internal mechanisms, we use a case study to analyse how the visit skipping and code skipping modules could infer about the possible causes of the target disease.

We draw a positive example from the Heart Failure dataset where the model successfully predicts the future condition of the patient and show the complete visit records in Table IV. At each visit, the model could obtain a visit selection action score b_n using Eq. (6) that indicates whether to skip the current visit, where $b_n = 1$ means to keep the current visit and 0 means to skip it. We can see that the model only keeps the first visit, since all of the codes from the remaining visits have weak associations with the target heart failure. Besides, for the selected visit, the model only keeps four ICD codes (i.e., Shortness of breath, Screening for diabetes mellitus, Morbid obesity, Edema) that have strong relations to heart failure, which have been reported by the Centers for Disease Control and Prevention (CDC)[7] and Mayo Clinic[8]. This case study also proves that our model is effective in filtering unrelated codes, and more importantly, this feature can improve the interpretabilty of the model on both visit and code level.

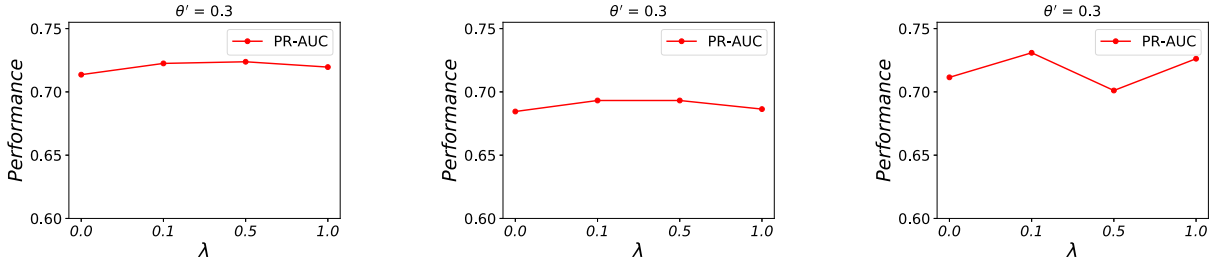
[7]https://www.cdc.gov/diabetes/library/features/diabetes-and-heart.html

[8]https://www.mayoclinic.org/diseases-conditions/edema/symptoms-causes/syc-20366493



(a) Heart Failure (b) COPD (c) Amnesia

Fig. 3: Performance on three datasets when trying different skip rates.



(a) Heart Failure (b) COPD (c) Amnesia

Fig. 4: Performance on three datasets when trying different penalty weight.

TABLE IV: Case study on the heart failure dataset.

Visit Skipping	Code Skipping
Visit 1: 1	[0.335] Shortness of breath (786.05) [0] Other abnormal blood chemistry (790.6) [0.113] Screening for diabetes mellitus (V77.1) [0] Other specified counseling (V65.49) [0] Other nonspecific abnormal findings (796.9) [0.106] Morbid obesity (278.01) [0.149] Edema (782.3) [0] Sprain of medial collateral ligament of knee (844.1)
Visit 2: 0	[0.278] Care involving other physical therapy (V57.1) [0.339] Old disruption of medial collateral ligament (717.82)
Visit 3: 0	[0.300] Care involving other physical therapy (V57.1) [0.371] Old disruption of medial collateral ligament (717.82)
Visit 4: 0	[0.592] Sprain of medial collateral ligament of knee (844.1)
Visit 5: 0	[0.536] Sprain of medial collateral ligament of knee (844.1)

E. Effect of Hyperparemeters

In this section, we conduct experiments to survey the effect of hyperparameters of `MedSkim`. There are two key hyperparameters in our model: The first is the skip rate θ' that controls how many visits and codes should be skipped; and the second is the weight of regularization term λ for punishing the model if the model does not follow the desired skip rate θ'. We first set the weight λ to be a fixed value and then try different values of skip rate θ'. After finding the optimal skip rate for each dataset, we fix the value of θ' and test different values of λ. We report the performance of the model in terms of PR-AUC and the actual skip rate θ under all the hyperparameter settings.

We show the results of `MedSkim` by testing for different skip rate on three datasets in Figure 3. The weight of regularization term λ is set to 0.1. We try five candidate target skip rates $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. On all three datasets, we run our algorithm under these five target skip rates and record the actual skip rates that the model achieve. In addition, we also report the model's performance under different settings of skip rates. Based on these results, we can obtain the optimal skip rate for each dataset. From these results, we can find that for the heart failure dataset, the optimal skip rate is 0.5, while for the other two datasets, the optimal skip rate is 0.3. Generally speaking, increasing the skip rate would force the model to discard more visits and codes, which will cause the decrease of model performance (e.g., the case when the skip rates are set to 0.7 and 0.9). However, we can see that small skip rates that are lower than the optimal skip rates would also cause the performance to decrease. In this case, too much noisy information is included in the model's input data, which has negative effects on the learning process of `MedSkim`. Hence, the choosing of the skip rate is a cautious process requiring careful validation using the development sets.

Moreover, we show the results of testing different regularization weights λ under the optimal skip rates discovered above on three datasets in Figure 4. We use four candidate values of λ: $\{0.0, 0.1, 0.5, 1.0\}$, and as above, we also run our algorithm on three datasets and report the model performance in terms of PR-AUC and actual skip rate under these settings. We first focus on the case when $\lambda = 0.0$, and the model is solely supervised by the standard cross-entropy loss. We can observe that the performance decreases on all three datasets since the actual skip rates of the model cannot be controlled, which indicates that the extra regularization term that we introduce is helpful for training the model and enables us to control the actual behaviors of the model. We can also pick an optimal value of λ on each dataset (0.5 for heart failure,

0.1 for COPD and Amnesia). We can also observe that the variation of the model performance are not very large under different settings of λ on Heart Failure and COPD datasets, which means that MedSkim is not very sensitive to the value of λ. However, the effect of λ becomes larger on the Amnesia dataset. This is because the number of visits per EHR within Amnesia dataset is much longer than that of Heart Failure and COPD datasets, which makes the model more sensitive to the extra skip rate regularization term. In this way, we should tune the values of λ based on the datasets that we use.

V. Conclusions

In this paper, we propose a novel risk prediction model named MedSkim, which dynamically selects important visits and codes during processing the EHR. MedSkim first conducts global code selection to remove the target-irrelevant codes from the input visits. Then MedSkim applies two RNNs (BPRNN and FSRNN) to process the EHR data in two directions, which enables the proposed model to fully consider the historical and future information when making skipping decisions. The FSRNN makes visit-level skipping operation at each visit based on time information and target information, and outputs hidden states eventually. Finally, the risk prediction module fuses the sequence of hidden states together to make predictions. We evaluate MedSkim on three real-world datasets and show that it outperforms existing state-of-the-art deep learning baselines steadily and robustly. Also, the experimental results can demonstrate that the explicit selection of visits and codes provides highly interpretable results.

Acknowledgement

T. Wang is partially supported by the National Science Foundation under Grant No. 1953893, 1951729, and 2119331. F. Ma is supported in part by the National Institutes of Health under Grant No. 1R01AG077016-01. 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 National Science Foundation and National Institutes of Health.

References

[1] F. Ma, M. Ye, J. Luo, C. Xiao, and J. Sun, "Advances in mining heterogeneous healthcare data," in *SIGKDD*, 2021, pp. 4050–4051.

[2] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *NeurIPS*, 2016, pp. 3504–3512.

[3] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao, "Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks," in *SIGKDD*, 2017, pp. 1903–1911.

[4] B. C. Kwon, M.-J. Choi, J. T. Kim, E. Choi, Y. B. Kim, S. Kwon, J. Sun, and J. Choo, "Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 299–309, 2018.

[5] T. Bai, S. Zhang, B. L. Egleston, and S. Vucetic, "Interpretable representation learning for healthcare via capturing disease progression through time," in *SIGKDD*, 2018, pp. 43–51.

[6] L. Ma, J. Gao, Y. Wang, C. Zhang, J. Wang, W. Ruan, W. Tang, X. Gao, and X. Ma, "Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration," in *AAAI*, 2020.

[7] F. Ma, Y. Wang, J. Gao, H. Xiao, and J. Zhou, "Rare disease prediction by generating quality-assured electronic health records," in *SDM*. SIAM, 2020, pp. 514–522.

[8] H. Song, D. Rajan, J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *AAAI*, vol. 32, no. 1, 2018.

[9] J. Luo, M. Ye, C. Xiao, and F. Ma, "Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records," in *SIGKDD*, 2020, pp. 647–656.

[10] M. Ye, J. Luo, C. Xiao, and F. Ma, "Lsan: Modeling long-term dependencies and short-term correlations with hierarchical attention for risk prediction," in *CIKM*, 2020.

[11] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, and J. Gao, "Kame: Knowledge-based attention model for diagnosis prediction in healthcare," in *CIKM*. ACM, 2018, pp. 743–752.

[12] C. Yin, R. Zhao, B. Qian, X. Lv, and P. Zhang, "Domain knowledge guided deep learning with electronic health records," in *ICDM*. IEEE, 2019, pp. 738–747.

[13] M. Ye, S. Cui, Y. Wang, J. Luo, C. Xiao, and F. Ma, "Medpath: Augmenting health risk prediction via medical knowledge paths," *The Web Conference*, 2021.

[14] F. Ma, J. Gao, Q. Suo, Q. You, J. Zhou, and A. Zhang, "Risk prediction on electronic health records with prior medical knowledge," in *SIGKDD*, 2018, pp. 1910–1919.

[15] F. Ma, Y. Wang, H. Xiao, Y. Yuan, R. Chitta, J. Zhou, and J. Gao, "A general framework for diagnosis prediction via incorporating medical code descriptions," in *BIBM*. IEEE, 2018, pp. 1070–1075.

[16] M. Ye, S. Cui, Y. Wang, J. Luo, C. Xiao, and F. Ma, "Medretriever: Target-driven interpretable health risk prediction via retrieving unstructured medical text," *CIKM*, 2021.

[17] E. Jang, S. S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *ArXiv*, vol. abs/1611.01144, 2017.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[21] Y. Zhou, H. Wang, J. He, and H. Wang, "From intrinsic to counterfactual: On the explainability of contextualized recommender systems," *CoRR*, vol. abs/2110.14844, 2021. [Online]. Available: https://arxiv.org/abs/2110.14844

[22] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware lstm networks," in *SIGKDD*, 2017, pp. 65–74.

[23] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: graph-based attention model for healthcare representation learning," in *SIGKDD*. ACM, 2017, pp. 787–795.

[24] E. Choi, C. Xiao, W. F. Stewart, and J. Sun, "Mime: multilevel medical embedding of electronic health records for predictive healthcare," in *Neural Information Processing Systems*, 2018, pp. 4552–4562.

[25] C. Chen, J. Liang, F. Ma, L. Glass, J. Sun, and C. Xiao, "Unite: Uncertainty-based health risk prediction leveraging multi-sourced data," in *The Web Conference*, 2021, pp. 217–226.

[26] Y. Jernite, E. Grave, A. Joulin, and T. Mikolov, "Variable computation in recurrent neural networks," *ArXiv*, vol. abs/1611.06188, 2017.

[27] M. Seo, S. Min, A. Farhadi, and H. Hajishirzi, "Neural speed reading via skim-rnn," in *ICLR*, 2018.

[28] A. W. Yu, H. Lee, and Q. Le, "Learning to skim text," in *ACL*, 2017, pp. 1880–1890.

[29] V. Campos, B. Jou, X. Giró-i Nieto, J. Torres, and S.-F. Chang, "Skip rnn: Learning to skip state updates in recurrent neural networks," in *International Conference on Learning Representations*, 2018.

[30] T. Huang, G. Shen, and Z.-H. Deng, "Leap-lstm: Enhancing long short-term memory for text categorization," *arXiv preprint arXiv:1905.11558*, 2019.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.