AUTOMED: Automated Medical Risk Predictive Modeling on Electronic Health Records

Suhan Cui, Jiaqi Wang, Xinning Gui, Ting Wang, Fenglong Ma
Pennsylvania State University, United States
{suhan, jqwang, xinning, ting, fenglong}@psu.edu

2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-6819-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/BIBM55620.2022.9995209

***Abstract*—Electronic health records (EHR) have been widely applied to various tasks in the medical domain such as risk predictive modeling, which aims to predict further health conditions by analyzing patients' historical EHR. Existing work mainly focuses on modeling the sequential and temporal characteristics of EHR data with advanced deep learning techniques. However, the network architectures of these models are all manually designed based on experts' prior knowledge, which largely impedes non-experts from exploring this task. To address this issue, in this paper, we propose a novel automated risk prediction model named AUTOMED to automatically search the optimal model architecture for modeling the complex EHR data and improving the performance of the risk prediction task. In particular, we follow the idea of neural architecture search to design a search space that contains three separate searchable modules. Two of them are used for analyzing sequential and temporal features of EHR data, respectively. The third is to automatically fuse both features together. Besides these three modules, AUTOMED contains an embedding module and a prediction module. All the three searchable modules are jointly optimized in the search stage to derive the optimal model architecture. In such a way, the model design can be automatically achieved with few human interventions. Experimental results on three real-world datasets show that AUTOMED outperforms state-of-the-art baselines in terms of PR-AUC, F1, and Cohen's Kappa. Moreover, the ablation study shows that AUTOMED can obtain reasonable model architectures and offer useful insights to the future risk prediction model design.**

I. INTRODUCTION

Medical risk prediction is a representative task in healthcare, which aims at building actionable predictive models to forecast the future health conditions or outcomes of patients based on their historical electronic health records (EHR) [1], [2]. EHR data consist of a time-ordered sequence of visits, and each visit contains several clinical codes such as International Classification of Diseases (ICD) codes. To model such sequential characteristics of EHR data, most of existing approaches usually apply recurrent neural networks (RNN) [3], [4] and Transformer [5] as the backbone and equip advanced techniques such as attention mechanisms with them to improve the prediction performance [6]–[11].

Besides, EHR data have temporal characteristics since each visit is associated with a timestamp, which is the key factor in modeling disease progression. Existing work that models the time information in the risk prediction task can be categorized into two groups. One follows the information decay assumption and uses monotonically non-increasing functions to model irregular time intervals between two consecutive visits, such as T-LSTM [12]. The other such as HiTANet [13] treats visits as words in a sentence and time stamps as words' positions and employs Transformer to model the EHR data. These approaches are powerful and effective to enhance the prediction performance, but *designing such time-aware models requires substantial efforts of human experts*. Although some automated machine learning-based frameworks are proposed in the medical domain recently such as AutoPrognosis [14] and Clairvoyance [15], they mainly focus on configuring machine learning **pipelines**, instead of automatically designing network architectures. Therefore, it is an urgent need to develop new models to automatically model sequential yet temporal EHR data simultaneously with minimal human interventions.

To tackle the aforementioned challenges, in this paper, we propose a new automated medical risk prediction model, named AUTOMED, which can automatically search an optimal network architecture on time-ordered EHR data as shown in Figure 1. AUTOMED consists of five modules: (1) The *embedding module* that maps discrete medical codes with each visit and the associated timestamp to dense embeddings $\mathbf{D}$ and $\mathbf{T}$, respectively. (2) The *time encoding module* contains a directed acyclic graph (DAG), i.e., a cell, and a searchable feature selector. The cell can automatically search for the optimal operation between a pair of computation nodes of DAG, and the feature selector can output the representative representation $\hat{\mathbf{x}}_T$, which is taken as the input of the fusion module. (3) The *diagnosis encoding module* has the same structure as the time encoding module, and its output $\hat{\mathbf{x}}_D$ is also the input of the fusion module. (4) The *fusion module* also contains a cell to search the optimal architecture for fusing two types of features simultaneously and learning the final EHR representation $\mathbf{H}$ as the input of the risk prediction module. (5) The *prediction module* is designed to make the search stage learning more stable, which consists of an RNN layer with attention mechanisms. We use the bi-level optimization technique as DARTS [16] to jointly optimize three cells and two feature selectors and further obtain the optimal model architecture.

Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to design an NAS-based model to solve the health risk prediction problem in the medical domain, which largely reduces the human interventions of model design.
- The proposed AUTOMED tailors a novel search space to model sequential yet temporal EHR data. Correspondingly,

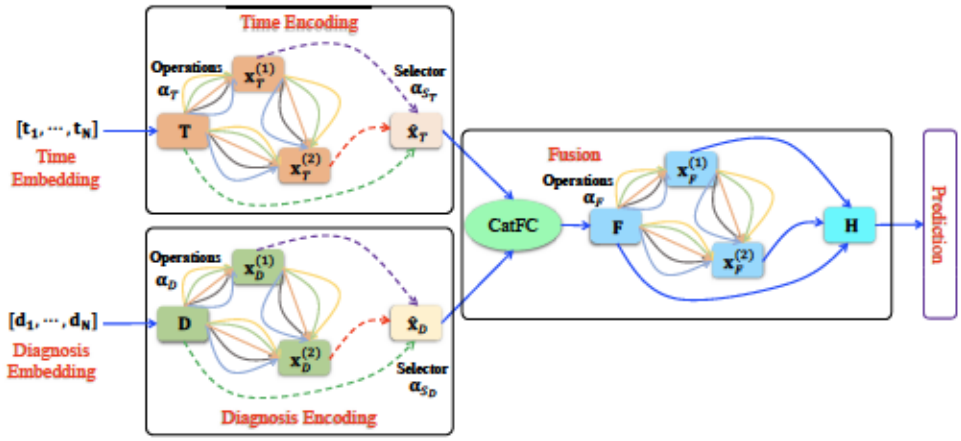

Fig. 1: Overview of the proposed AUTOMED in the searching stage, i.e., the supernet.

two separate modules are used to search the optimal architectures for discrete medical codes within each visit and the associate time information. Moreover, a fusion cell is designed to search the optimal fusion strategy. These designs make AUTOMED learn better representation and further improve the prediction performance.

- Experimental results on three real-world claims data show that AUTOMED achieves significant improvement over all state-of-the-art baselines, and ablation study further shows the effectiveness of all the designed modules.

II. Related Work

Existing health risk prediction models are mainly to model the sequential characteristics of EHR data using RNN [3], [4] and Transformer [5] as the backbones. Then they are equipped with different types of attention mechanisms [6], [7], [10] or incorporating external knowledge such as ICD hierarchy [17]–[19], medical text [20], and medical knowledge graphs [21]–[24], to further improve the prediction performance. There are several approaches are proposed to model the **time information**, such as T-LSTM [12], RetainEX [25], Timeline [26]. They mainly design the model architecture based on human prior assumptions about the effect of time information, which limits the models' learning ability. Thus, there is an urgent need of the automatic model design for health risk prediction.

Neural architecture search (NAS) [27] is a general approach for automatically discovering the optimal model architecture for deep neural networks, which is a bi-level optimization problem in essence that aims to optimize both the network parameters and the model architecture simultaneously. Some work aims to directly solve the searching problem with huge computing cost, such as using reinforcement learning [28] or evolutionary search [29]. To improve the searching efficiency of NAS methods from different perspectives, weight sharing [30], sequential model-based optimization [31], and Bayesian optimization [32] are used. More recently, differentiable architecture search (DARTS) [16] is proposed and achieves remarkable improvement in terms of searching efficiency, which introduces a continuous relaxation to the discrete model architecture and designs a unified gradient optimization framework for both the network weights and architecture. In this paper, we utilize the differentiable methods as the search algorithm and design a unified searching space for learning heterogeneous EHR features and the fusion strategy at the same time.

III. Methodology

A. Data & Task

The **EHR data** of each patient consists of multiple time-ordered visits $V = [(\mathbf{v}_1, t_1), (\mathbf{v}_2, t_2), \cdots, (\mathbf{v}_N, t_N)]$, where N is the total number of visits. At each visit, a set of diagnosis codes is recorded, which is represented as a binary vector $\mathbf{v}_n \in \{0,1\}^M$, where M represents the total number of unique codes in the dataset. $\mathbf{v}_n^m = 1$ denotes that the m-th code appears in the i-th visit; otherwise, $\mathbf{v}_n^m = 0$. In addition, a timestamp in terms of date t_n is recorded at each visit. The **task** of health risk prediction is to predict whether the patient will suffer from the target disease or condition in the future according to the historical EHR data V.

B. Overview of AUTOMED

To investigate the optimal way of integrating the heterogeneous features of EHR data, we propose AUTOMED as shown in Figure 1, which contains five modules: the embedding module, the time encoding module, the diagnosis encoding module, the fusion module, and the prediction module. The **embedding module** aims to map the input diagnosis $\mathbf{v}_n$ and time t_n features into dense vector representations $\mathbf{d}_n$ and $\mathbf{t}_n$, respectively. Then we use three modules to automatically fuse $\mathbf{d}_n$ and $\mathbf{t}_n$ following the idea of the neural architecture search (NAS) [27] to learn the optimal architectures of these three modules in a unified way. Specifically, in each module, we design a searchable cell, which shares the same search space but uses different network weights. The **time** and **diagnosis encoding modules** take $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_N]$ and $\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_N]$ as the inputs and automatically learn a representation for the computational node in each cell, respectively. Then a searchable feature selector is developed to select optimal representations outputted by computational nodes. The selected features from the time and diagnosis encoding modules are then considered as the inputs of the **fusion module**

to generate the final visit-level EHR representations, which contains a searchable cell followed by a linear combination layer (CatFC in Figure 1). Finally, the EHR representations are used as the inputs of the **prediction module** to make risk predictions. Next, we introduce the details of each module.

C. Embedding Diagnosis and Time Features

1) Diagnosis Embedding: Given the binary input visit vector $\mathbf{v}_n$, we apply a linear function to transform it into a latent representation $\mathbf{d}_n \in \mathbb{R}^d$, i.e., $\mathbf{d}_n = \mathbf{W}_d\mathbf{v}_n + \mathbf{b}_d$, where $\mathbf{W}_d \in \mathbb{R}^{d\times M}$ and $\mathbf{b}_d \in \mathbb{R}^d$ is the weight matrix and bias vector, respectively. Since there are N visits in each patient's EHR data, the diagnosis features of a patient will become a sequence of representations $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_N]$.

2) Time Embedding: Following [13], 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:

$$\mathbf{t}_n = \mathbf{W}_t\mathbf{r}_n + \mathbf{b}_t,\ \mathbf{r}_n = \mathbf{1} - \tanh((\mathbf{W}_r\frac{\Delta t_n}{180} + \mathbf{b}_r)^2), \quad (1)$$

where $\mathbf{W}_r \in \mathbb{R}^a$, $\mathbf{W}_t \in \mathbb{R}^{d\times a}$, $\mathbf{b}_r \in \mathbb{R}^a$, and $\mathbf{b}_t \in \mathbb{R}^d$ are all network parameters. Similarly, the time features of the patient will be represented by a sequence of representations $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_N]$. The *network parameters* in the embedding module are $\mathbf{W}_E = [\mathbf{W}_d, \mathbf{b}_d, \mathbf{W}_r, \mathbf{W}_t, \mathbf{b}_r, \mathbf{b}_t]$.

D. Encoding Diagnosis Representations

1) Cell Design: Given the input diagnosis features $\mathbf{D}$, we aim to find an optimal neural architecture to encode them. In particular, following DARTS [16], we adopt the general DAG (directed acyclic graph) setting that a cell contains an ordered sequence of C computation nodes[1], where each node $\mathbf{x}_D^{(i)}$ is a latent representation, and each directed edge (i, j) is associated with some operation $o_D^{(i,j)}$ that draw from an operation set $\mathcal{O}$ to transform $\mathbf{x}_D^{(i)}$. During the search stage, each node is computed based on all of its predecessors, i.e.,

$$\mathbf{x}_D^{(j)} = \sum_{i<j} o_D^{(i,j)}(\mathbf{x}_D^{(i)}) = \sum_{i<j}\sum_{o\in\mathcal{O}} \frac{\exp(\alpha_{Do}^{(i,j)})}{\sum_{o'\in\mathcal{O}}\exp(\alpha_{Do'}^{(i,j)})} o(\mathbf{x}_D^{(i)}), \quad (2)$$

where $\mathbf{x}_D^{(0)} = \mathbf{D}$ and all of $\mathbf{x}$'s have the same shape as $\mathbf{D}$. The operations $\mathcal{O}$ include *1-D convolution*, *multi-head self attention*, *recurrent layer*, *feed-forward layer*, *identity*, and *zero*. The details of these operations are introduced in Section IV-A3. $\alpha_{Do}^{(i,j)}$ denotes the weight of the operation o on edge (i, j) in the diagnosis encoding module.

2) Searchable Feature Selector: . Existing methods generate the output of the diagnosis encoding module by averaging or concatenating $[\mathbf{x}_D^{(0)}, \mathbf{x}_D^{(1)}, \cdots, \mathbf{x}_D^{(C)}]$ learned by Eq. (2) [16], [30]. Such mandatory operations require to use all the node outputs, and the averaged or concatenated output may not be the most representative one. To avoid this issue and increase the capability of AUTOMED, we design a searchable feature selector. Let $\alpha_{S_D}^{(k)}$ denotes the architecture weight of the k-th computation node in the cell. In the search stage, we define the mixed selection on C nodes as follows:

$$\hat{\mathbf{x}}_D = \sum_{k=0}^{C} \frac{\exp(\alpha_{S_D}^{(k)})}{\sum_{k'=0}^{C}\exp(\alpha_{S_D}^{(k')})}\mathbf{x}_D^{(k)}, \quad (3)$$

where $\hat{\mathbf{x}}_D$ is the output of the diagnosis encoding module. In this module, we need to optimize the *model architecture parameters*, including the operation weights $\boldsymbol{\alpha}_D$ on all edges and the selection weights $\boldsymbol{\alpha}_{S_D}$ on all computation nodes. We need to optimize the *operation parameter set* $\mathbf{W}_{O_D}$.

E. Encoding Time Representations

We apply the same cell and feature selector design introduced in the previous subsection to encode time representations. Taking $\mathbf{T}$ as the input of the time encoding module, we first obtain a list of computation node features $\{\mathbf{x}_T^{(k)}\}, k \in \{0, \cdots, C\}$ using Eq. (2) with operation weight parameters $\boldsymbol{\alpha}_T$. Then we generate the module output $\hat{\mathbf{x}}_T$ via Eq. (3) with node selection parameters $\boldsymbol{\alpha}_{S_T}$. The *operation parameters* to be optimized in this module are denoted as $\mathbf{W}_{O_T}$.

F. Fusing Diagnosis and Time Representations

After obtaining the selected features $\hat{\mathbf{x}}_D = [\hat{\mathbf{x}}_1^D, \cdots, \hat{\mathbf{x}}_N^D]$ and $\hat{\mathbf{x}}_T = [\hat{\mathbf{x}}_1^T, \cdots, \hat{\mathbf{x}}_N^T]$, we first concatenate them together and then apply a linear transformation to map the concatenation into a single feature, i.e.,

$$\mathbf{f}_n = \mathbf{W}_c\text{concat}(\hat{\mathbf{x}}_n^D, \hat{\mathbf{x}}_n^T) + \mathbf{b}_c, \quad (4)$$

where $\mathbf{W}_c \in \mathbb{R}^{d\times 2d}$, $\mathbf{b}_c \in \mathbb{R}^d$ are network parameters of the linear transformation layer. Then the obtained features $\mathbf{F} = [\mathbf{f}_1, \cdots, \mathbf{f}_N]$ are taken as the input of the fusion cell, which has the same design as the cells in the diagnosis and time encoding modules.

Similarly, we can obtain a list of node features as well, i.e., $\{\mathbf{x}_F^{(k)}\}, k \in \{0, \cdots, C\}$ using Eq. (2), when taking $\mathbf{F}$ as the input and using $\boldsymbol{\alpha}_F$ as the operation weights. However, different from the previously two encoding modules, we do not apply a feature selector here since we need to get the comprehensive representation for the whole EHR data. Thus, we combine all the node features into a single representation:

$$\mathbf{h}_n = \sum_{k=0}^{C} w_k\mathbf{x}_{Fn}^{(k)}, \quad (5)$$

where $w_k \in \mathbb{R}$ is the network weight parameter of the k-th computation node and $\mathbf{w}_f = [w_0, \cdots, w_C]^\top$. The output of the fusion module is $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_N]$. In this module, we need to optimize the *model architecture parameters* $\boldsymbol{\alpha}_F$ and the *network parameters* $\mathbf{W}_F = [\mathbf{W}_c, \mathbf{b}_c, \mathbf{w}_f, \mathbf{W}_{O_F}]$, where $\mathbf{W}_{O_F}$ is the operation parameter set used in the fusion cell.

G. Predicting Health Risks

To make the search stages more stable, we add a fixed RNN layer (GRU [33]) to transform the features and aggregate them over the sequence dimension through the attention mechanism

[1]To reduce the computational complexity, we set $C = 2$ in this paper, i.e., three computation nodes with IDs 0, 1, and 2 in the DAG.

and then pass the aggregated EHR representation through the final classifier as follows:

$$\begin{aligned}
&[\mathbf{h}'_1, \cdots, \mathbf{h}'_N] = \text{RNN}([\mathbf{h}_1, \cdots, \mathbf{h}_N]), \\
&[\beta_1, \cdots, \beta_N] = \text{softmax}(\mathbf{w}_h^\top \mathbf{h}'_1 + b_h, \cdots, \mathbf{w}_h^\top \mathbf{h}'_N + b_h), \\
&\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_y \mathbf{u} + \mathbf{b}_y) = \text{softmax}(\mathbf{W}_y \sum_{n=1}^{N} \beta_n \mathbf{h}'_n + \mathbf{b}_y),
\end{aligned} \tag{6}$$

where $\mathbf{w}_h \in \mathbb{R}^d$, $b_h \in \mathbb{R}$, $\mathbf{W}_y \in \mathbb{R}^{2\times d}$, and $\mathbf{b}_y \in \mathbb{R}^2$ are all network parameters, β's are the aggregation weights for all N time steps, and $\hat{\mathbf{y}} \in \mathbb{R}^2$ is the final output distribution. The network parameters of the prediction module are $\mathbf{W}_P = [\mathbf{w}_h, b_h, \mathbf{W}_y, \mathbf{b}_y, \mathbf{W}_{rnn}]$, where $\mathbf{W}_{rnn}$ is the parameter set of the RNN layer.

H. Optimization

Let $\boldsymbol{\alpha}$ denote the collection of architecture weights, which includes $\boldsymbol{\alpha}_T$ for the time cell, $\boldsymbol{\alpha}_D$ for the diagnosis cell, $\boldsymbol{\alpha}_F$ for the fusion cell, $\boldsymbol{\alpha}_{S_T}$ for the time feature selector, and $\boldsymbol{\alpha}_{S_D}$ for the diagnosis feature selector. Let $\mathbf{W}$ denote the network weights, which contains $\mathbf{W}_E$ for the embedding module, $\mathbf{W}_F$ for the fusion module, $\mathbf{W}_P$ for the prediction module, and $\mathbf{W}_O = [\mathbf{W}_{O_D}, \mathbf{W}_{O_T}, \mathbf{W}_{O_F}]$ for the operation parameters used in the three cells. We use the bi-level optimization technique as DARTS [16] to optimize the model architecture $\boldsymbol{\alpha}$ and the network weights $\mathbf{W}$ simultaneously:

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} \mathcal{L}_{val}(\mathbf{W}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\
&\text{s.t. } \mathbf{W}^*(\boldsymbol{\alpha}) = \text{argmin}_{\mathbf{W}} \mathcal{L}_{train}(\mathbf{W}, \boldsymbol{\alpha})
\end{aligned} \tag{7}$$

where $\mathcal{L}_{val}$ and $\mathcal{L}_{train}$ mean the validation loss and training loss, respectively.

I. Deriving Discrete Architectures

Using the learned architecture parameters $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_D, \boldsymbol{\alpha}_T, \boldsymbol{\alpha}_F, \boldsymbol{\alpha}_{S_D}, \boldsymbol{\alpha}_{S_T}]$, we are able to derive the discrete model architectures based on the optimal $\boldsymbol{\alpha}$ for each module. For each searched cell, based on the obtained $\alpha^{(i,j)}$ for each edge in the DAG, we can choose the optimal operation, which is $o'^{(i,j)} = \text{argmax}_{o\in\mathcal{O}} \{\alpha_o^{(i,j)}\}$. Then we compute the node feature via $\mathbf{x}^{(j)} = \sum_{i<j} o'^{(i,j)}(\mathbf{x}^{(i)})$. Also, for the feature selection in the diagnosis and time cells, we choose the node features by $\mathbf{x}'_D = \text{argmax}_{k\in\{0,\cdots,C\}} \{\alpha_{S_D}^{(k)}\}$ and $\mathbf{x}'_T = \text{argmax}_{k\in\{0,\cdots,C\}} \{\alpha_{S_T}^{(k)}\}$. Eventually, we can derive the final model architecture and train the model from scratch for evaluation.

IV. Experiments

A. Experimental Setup

1) Datasets: In our experiments, we conduct retrospective analysis on three common chronic and progressive health conditions, which are Chronic Obstructive Pulmonary Disease (COPD), Amnesia and Kidney Disease. With the guidance of clinicians, we extract the corresponding EHR data, which includes positive cases and negative/control cases, from a real-world claims database. We randomly partition the datasets into the training set, validation set, and testing set with the ratio of $0.75 : 0.10 : 0.15$. The best model is selected based on the performance on the validation set. The statistics of the datasets are shown in Table I.

TABLE I: Statistics of the three EHR datasets.

Dataset	COPD	Amnesia	Kidney
Positive Cases	7,314	2,982	2,810
Negative Cases	21,942	8,946	8,4300
Avg. Visits/Patient	30.39	39.00	39.09
Avg. Codes/Visit	3.50	4.70	4.40
Unique Codes	10,053	9,032	8,802

2) Baselines: We select traditional and state-of-the-art risk prediction models as our baselines, which are divided into two categories: (1) Without using time information: LSTM [3], Dipole [7], Retain [6], SAnD [10], AdaCare [8], LSAN [11]. (2) Using time information: RetainEx [25], Timeline [26], T-LSTM [12], HiTANet [13].

3) Operations: In this paper, we use the following six operations when searching the network architectures: 1-D Convolution (*conv*), Multi-head Self Attention (*attention*), Recurrent Layer (*rnn*), Feed-Forward Layer (*ffn*), Identity and Zero.

4) Implementation Details: During the searching stage, we set different optimization configurations for the architecture weights $\boldsymbol{\alpha}$ and network weights $\mathbf{W}$. For both of them, we apply Adam optimizer [34]. For $\mathbf{W}$, we use the learning rate of 10^{-4} and weight decay of 10^{-4}, while for $\boldsymbol{\alpha}$, we use the learning rate of 10^{-5} and weight decay of 10^{-4}. We tune the learning rate and weight decay from a candidate set of $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. Through grid search method, we obtain the most suitable values and use them in the experiments. After searching, we train the model from scratch with the derived architecture, and we also apply the Adam optimizer with learning rate of 10^{-4} and weight decay of 10^{-4}, which is tuned in the same way as aforementioned. Besides, the hidden dimension size d of all the node features within our framework is set to 256, and the dimension of the intermediate time encoding a is set to 64. The setting of these dimensions maintains the same during both searching and training stages. We implement the **baselines** on the same platform with the proposed model and apply the same optimization settings as training the searched architecture. We use the standard cross-entropy loss for all baselines. The numbers of hidden dimensions of baselines are all 256 no matter for RNN or Transformer based models. We use *PR-AUC* (area under the precision-recall curve), *F1 score*, and *Cohen's Kappa* as the evaluation metrics considering the imbalanced data property in our datasets shown in Table I.

B. Performance Evaluation

Table II shows the overall performance of the proposed AutoMed and baselines on three datasets. We report the **average values** of five runs and the corresponding standard deviations (***std.***). We also conduct significance testing (**t-test**) to justify whether the proposed AutoMed is significantly better than the best baseline model.

TABLE II: Performance comparison in terms of PR-AUC, F1 score, and Cohen's Kappa (*mean*±*std.*). The results produced by the best baseline and the best model in each column are marked by underlined and **boldfaced**, respectively. $*$ denotes that the p-value is smaller than 0.01.

Dataset	COPD			Amnesia			Kidney		
Metrics	PR-AUC (%)	F1 (%)	Kappa (%)	PR-AUC (%)	F1 (%)	Kappa (%)	PR-AUC (%)	F1 (%)	Kappa (%)
LSTM [3]	55.34 ± 3.05	55.96 ± 0.97	41.78 ± 1.13	55.36 ± 3.35	61.14 ± 0.50	48.38 ± 1.39	61.96 ± 2.49	63.69 ± 1.18	50.36 ± 1.78
Dipole [7]	58.70 ± 1.19	56.18 ± 1.29	42.18 ± 1.44	58.04 ± 1.77	60.16 ± 2.84	46.46 ± 3.39	64.88 ± 3.35	64.65 ± 1.74	51.60 ± 2.22
Retain [6]	53.56 ± 0.69	50.96 ± 0.65	37.46 ± 0.80	56.04 ± 3.20	55.06 ± 1.52	43.48 ± 1.88	61.72 ± 2.76	57.15 ± 2.40	44.61 ± 2.83
SAnD [10]	51.70 ± 2.27	52.12 ± 2.36	37.66 ± 2.36	52.50 ± 4.98	56.38 ± 2.81	41.68 ± 2.70	57.69 ± 3.21	60.36 ± 1.06	45.75 ± 1.10
Adacare [8]	60.50 ± 1.61	55.08 ± 0.36	42.34 ± 0.85	59.68 ± 2.10	60.68 ± 1.21	47.84 ± 2.87	71.29 ± 2.46	65.01 ± 1.49	52.89 ± 2.29
LSAN [11]	63.84 ± 1.75	54.98 ± 0.98	43.52 ± 0.88	68.16 ± 1.52	64.12 ± 1.64	52.88 ± 1.87	72.31 ± 0.63	64.44 ± 1.43	52.48 ± 1.87
RetainEx [25]	60.52 ± 0.61	54.04 ± 2.69	43.44 ± 2.55	63.44 ± 1.92	58.92 ± 4.00	49.06 ± 4.27	69.15 ± 1.48	61.61 ± 1.48	50.61 ± 1.65
Timeline [26]	54.86 ± 1.85	49.02 ± 0.85	36.40 ± 1.10	56.46 ± 2.52	58.24 ± 2.04	45.52 ± 2.48	63.89 ± 3.12	59.87 ± 1.18	46.71 ± 1.12
T-LSTM [12]	68.62 ± 0.80	62.92 ± 0.61	51.55 ± 1.06	63.19 ± 2.14	62.91 ± 0.83	51.08 ± 1.62	68.90 ± 3.29	66.16 ± 0.61	54.26 ± 0.73
HiTANet [13]	68.46 ± 0.44	63.70 ± 0.80	51.78 ± 0.57	70.80 ± 0.96	65.40 ± 1.87	53.28 ± 2.18	75.65 ± 0.44	70.20 ± 0.74	56.72 ± 0.81
AUTOMED	**71.57**$*$± 2.48	**65.08**$*$± 2.13	**54.34**$*$± 1.86	**73.13**$*$± 2.58	**68.91**$*$± 1.73	**58.42**$*$± 2.60	**76.63**$*$± 1.83	**70.41**$*$± 1.26	**59.13**$*$± 2.23

From Table II, we can observe that the baselines incorporating time information usually perform better than those without considering the importance of time information. Especially, time-aware LSTM (T-LSTM) [12] that uses an information decay function to model the time information in the LSTM cell achieves the best PR-AUC score on the COPD dataset among all the baselines. HiTANet [13] takes the time information as word positions in Transformer and achieves the best performance on all three datasets. These two kinds of approaches are representative in the health risk prediction task when modeling time information.

Although existing approaches can improve the prediction performance by modeling time information via human prior knowledge, they all entangle the time features with diagnosis features during the model architecture design. Since two different features have inconsistent patterns and scales, it is extremely difficult for human-designed architecture to fuse them together appropriately. Thus, our proposed AUTOMED uses disentangled cells to process each type of features independently and designs a fusion cell to automatically search the feature fusion strategy, which can solve the feature inconsistency problem better. In such a way, the proposed AUTOMED significantly outperforms all the baselines in terms of PR-AUC, F1, and Cohen's Kappa.

C. Ablation Study

The benefit of the proposed AUTOMED is to automatically discover optimal network architectures via the three designed cells. Next, an ablation study is conducted to investigate the performance change when we add the cells one by one. Besides, for both the diagnosis and time encoding modules, we use a searchable feature selector to automatically learn the representative module outputs. To validate the efficiency of the proposed feature selector, we also conduct an ablation study. Specifically, we design the following four settings:

- **Fusion Only**: In this setting, we do not use the diagnosis and time modules and only use the fusion module. We achieve this by replacing $\hat{\mathbf{x}}_T$ and $\hat{\mathbf{x}}_D$ in Eq. (4) with $\mathbf{T}$ and $\mathbf{D}$ obtained in the embedding module.
- **Fusion+Time**: We use two searchable cells in this setting, i.e., the fusion and time cells. Towards this end, we replace $\hat{\mathbf{x}}_D$ with $\mathbf{D}$ in Eq. (4). In the time encoding module, we use the searchable feature selector.
- **Fusion+Diagnosis**: Similar to the above ablation setting, we replace $\hat{\mathbf{x}}_T$ with $\mathbf{T}$ in Eq. (4). In the diagnosis encoding module, we also use the searchable feature selector.
- **W.O. Selectors**: This setting means that AUTOMED removes the feature selector for the diagnosis and time encoding modules (i.e., without using Eq. (3)) and uses the average computation node representations as the outputs of these modules, i.e., $\hat{\mathbf{x}}_D = \frac{1}{C}\sum_{k=0}^{C}\mathbf{x}_D^{(k)}$ and $\hat{\mathbf{x}}_T = \frac{1}{C}\sum_{k=0}^{C}\mathbf{x}_T^{(k)}$.

TABLE III: Ablation study results in terms of F1 score (%).

Dataset	COPD	Amnesia	Kidney
AUTOMED	65.08	**68.91**	**70.41**
Fusion Only	62.32	64.75	69.42
Fusion+Time	62.81	68.79	69.02
Fusion+Diagnosis	61.90	63.95	69.31
W.O. Selectors	**65.90**	66.00	69.35

We present the ablation study results in Table III in terms of F1 score (%). Note that the results of the other two metrics have similar patterns as those of F1 scores. We can observe that removing any of the cells will lead to performance drop to some degree, which can validate that it is necessary to design three cells to jointly learn the optimal model architecture for risk prediction. Additionally, the contribution of each cell varies on different datasets.

Compared to Fusion Only, AUTOMED designs separate cells for each type of feature, which enables the search algorithm to find the best model architecture for each one of them. Thus, AUTOMED can largely improve the model learning ability on heterogeneous EHR data. Another noteworthy thing is that it would lead to performance drop compared to single-cell search when adding the diagnosis cell. This indicates that simply searching for one type of feature might lead to the inconsistency of time and diagnosis features, which affects the learning of the fusion cell. Therefore, it is optimal to design both time and diagnosis cells and combine them with the fusion cell to learn the overall model architecture simultaneously.

When we use the average representations of nodes in each cell as the output of the encoding modules (i.e., W.O. Selectors),

we can find that on the COPD dataset, the F1 score sightly increases. For other two datasets, the performance of W.O. Selectors is worse than that of AUTOMED. These results demonstrate that using the designed searchable feature selector does not harm the model performance, and in turn, it can boost the performance in most cases.

V. CONCLUSIONS

In this paper, we propose a novel automated risk predictive modeling approach, named AUTOMED, which is able to automatically search the optimal model architecture for dealing with the sequential and temporal EHR data with minimal human interventions. The designed model consists of five modules, and they tightly work together to optimize not only model architecture parameters and generate the optimal network architecture. Experiments on three real-world medical datasets show that the proposed AUTOMED achieves state-of-the-art performance compared with baselines. Moreover, the ablation study demonstrates the effectiveness of the designed modules, and the case study of presenting searched architectures offers some important insights, which are helpful for the future model design. Our future work will explore how to incorporate medical knowledge graphs into automated risk predictive modeling design.

ACKNOWLEDGEMENT

This work is partially supported by the National Science Foundation (NSF) under Grant No. 1953893 (T. Wang), 1951729 (T. Wang), 2119331 (T. Wang) and 2212323 (T. Wang, F. Ma, and X. Gui), and the National Institutes of Health (NIH) under Grant No. 1R01AG077016-01 (F. Ma). 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 NSF and NIH.

REFERENCES

[1] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

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

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

[4] 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.

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

[6] 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.

[7] 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 *KDD*, 2017, pp. 1903–1911.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] A. Alaa and M. Schaar, "Autoprognosis: Automated clinical prognostic modeling via bayesian optimization with structured kernel learning," in *ICML*. PMLR, 2018, pp. 139–148.

[15] D. Jarrett, J. Yoon, I. Bica, Z. Qian, A. Ercole, and M. van der Schaar, "Clairvoyance: A pipeline toolkit for medical time series," in *International Conference on Learning Representations*, 2020.

[16] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

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

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

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

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

[25] 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.

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

[27] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.

[28] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[29] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *AAAI*, vol. 33, no. 01, 2019, pp. 4780–4789.

[30] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *ICML*. PMLR, 2018, pp. 4095–4104.

[31] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy, "Progressive neural architecture search," in *ECCV*, 2018, pp. 19–34.

[32] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing, "Neural architecture search with bayesian optimisation and optimal transport," *NeurIPS*, vol. 31, 2018.

[33] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

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