DISEASE PREDICTION VIA GRAPH-BASED MODEL

NGUYEN ANH MINH
VIETTEL DIGITAL TALENT PROGRAM 2023

ABSTRACT

Disease prediction is a well-known classification problem in medical applications. Graph Convolutional Networks (GCNs) provide a powerful tool for analyzing the patients' features relative to each other. This can be achieved by modeling the problem as a graph node classification task, where each node is a patient or disease. Graph-based models overcome the limitations of traditional machine learning models by not considering individual patient data records as discrete entities but rather establishing connections between symptoms, similar diseases, and generating different disease groups. In this report, the author proposes a Graph Neural Network model with various information aggregation mechanisms such as Gated Recurrent Unit (GRU), ... the mechanism updates information from neighboring nodes through Attention mechanism, building node features for the disease graph based on symptoms using the TF-IDF formula and achieves a 100% F1-Macro evaluation result for disease prediction based on symptoms dataset from India.

KEYWORDS

Graph-based model, Graph Attention, Create node feature by TF-IDF

1 INTRODUCTION

EMRs, short for electronic medical records, are a commonly utilized data management scheme employed in hospitals to store comprehensive clinical information obtained from patients' visits. In recent times, advancements in information technology and machine learning have led to a more manageable volume of EMRs. Consequently, the analysis of EMRs using machine learning and data mining techniques has emerged as a prominent research direction, aiming to enhance healthcare services [5]. One notable application of machine learning in the healthcare domain involves disease prediction, which focuses on determining whether a patient is afflicted with a specific ailment. This task typically involves training a classifier to make predictions based on the information derived from EMRs [17], [11]. For instance, Palaniappan and Awang employed various data mining techniques, such as Decision Trees [14], and Neural Networks [8], to construct a predictive system for heart diseases [12]. Leveraging the capabilities of Convolutional Neural Networks (CNNs), Suo et al. [17] initially identified similarities among patients based on their EMRs and subsequently conducted personalized disease predictions.

Existing deep neural networks lack the incorporation of patient interactions and associations within their architecture. However, considering these relationships can prove advantageous as it facilitates the analysis and study of similar patient cohorts. Graphs offer a natural means of representing the interactions among a population by treating patients as nodes and their associations as edges. Constructing a graph between patients based on a subset of their features allows for the summarization of these features within the graph edges, reducing feature dimensionality and mitigating overfitting caused by a large number of features [2], [18].

Related work

In recent years, GCNs have been adopted in different applications, especially in medical domains e.g., brain analysis [9], mammogram analysis [4],... The disease prediction problem has been also widely explored by GCN-based methods [1], [10]. Following the promising results of GCNs in the medical domain, [13] exploit GCNs in the disease prediction problem with multi-modal datasets. They chose ChebyNet with constant filter size as the classifier and investigated the affinity graph construction by computing the distance between the subjects for brain analysis. InceptionGCN [7] extends ChebyNet and designs filters with different kernel sizes, and chooses the optimal one for the final disease prediction. A branch of methods focuses on improving the graph structure. [6] and [16] start with a pre-constructed graph and update it during the training, but [3] learn the whole graph from features directly in an end-to-end manner.

Main contribution

In this report, my main contributions are as follows:

- Combining GCN layers, Multi Layer Perceptron (MLP), and mechanisms for information exchange and update, such as using Gated Recurrent Neural Network (GRU) or Attention mechanism, to build a disease prediction model with a dataset from India.
- Proposing methods for constructing different graphs: disease graph and patient graph.
- Introducing a training method that combines Multi Head Attention layer and reduces the size of the input graph while still achieving good prediction results with various metrics like Recall, Precision, and F1.
- Proposing a method for building a feature set for the disease graph based on the TF-IDF formula.

2 DATA PREPROCESSING

In this report, I utilize two datasets for disease prediction based on relevant symptoms to construct multi-class classification and binary classification problems.

2.1 Disease Symptom Prediction

This dataset [1] consists of 4920 records with 18 different fields representing disease symptoms. There are 41 distinct diseases, with each disease having 120 records. After conducting the Exploratory Data Analysis (EDA), I observed that each disease exhibits 2-3 distinctive symptoms. There are also common symptoms shared among different diseases, such as headaches and fatigue. Additionally, patients infected with the same disease tend to have 1-2 similar

Supervised by LE HOANG NGAN.

Mini Project in VDT, gen 3, (2023)
(C).

[1] https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset

symptoms. These findings inspire the idea of constructing graphs for different disease types and patient graphs based on shared symptoms. These graphs serve to generalize the information of the patients, and by utilizing Graph Neural Networks (GNN), we can extract latent features from the data.

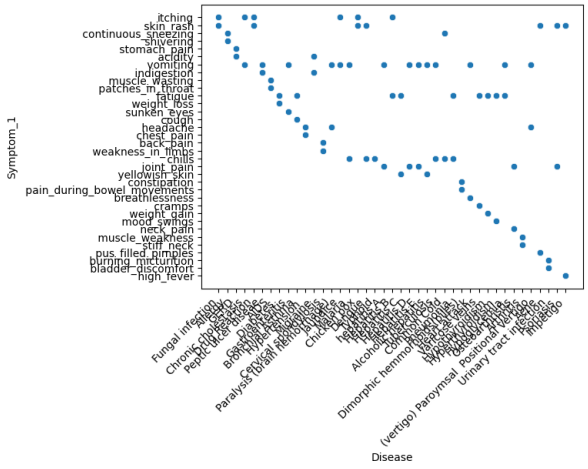

Figure 1: Relation between Symptom and Disease

2.2 Pima Indian Diabetes (Diabetes) [15]

The dataset is produced by the "National Institute of Diabetes and Digestive and Kidney Diseases". The goal of this dataset is to recognize the diabetic status of patients (binary classification). Every patient has 7 numeric features which show the diagnostic measurements of diabetes including the number of pregnancies, plasma glucose, blood pressure, skin thickness, Body Mass Index (BMI), insulin level, diabetes pedigree function, and age.

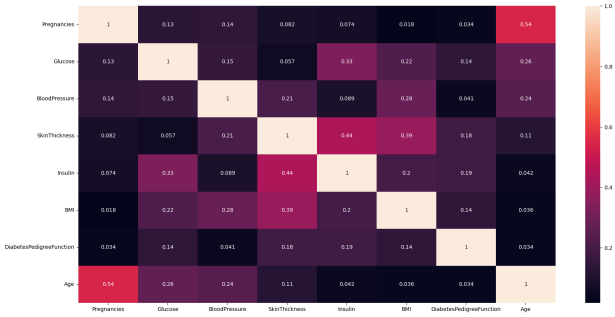

Figure 2: Correlation betweem feature columns

Based on the data, we can observe a relatively high correlation between the age field and other fields in the dataset. Therefore, I perform binning on the age field to create different age groups and utilize them to construct graph connections between these age groups.

3 THE PROPOSED METHOD

In this report, I will build various graph-based models, including both single-graph and multi-graph inputs. Additionally, I will utilize different mechanisms to update and aggregate messages between nodes.

3.1 Definitions

Assume that the graph G is given with N nodes, represented by $G(V, E, X)$, where V is the set of nodes ($|V| = N$), E is the set of edges, and $X \in \mathbb{R}^{N\times F}$ indicates the node feature matrix. $A \in \mathbb{R}^{N\times N}$ is the unweighted and undirected adjacency matrix of the graph. Every node v_i has a corresponding feature vector x_i, a one-hot label vector y_i, and the true class label $c_i \in C$ where C is the set of classes. The label information is available for a subset of nodes and the task is to learn the parametric function $f_\theta(X, A)$ which takes the adjacency matrix and node features as input, and its goal is to predict the true label of the unlabeled nodes. It should be noted that a probabilistic classifier predicts a probability distribution over the $|C|$ classes and the class with maximum probability is selected as the label. In our proposed method q_i is the output probability distribution of classifier defined over $|C|$ classes for the sample x_i where c-th element represents the confidence of the classifier about assigning label c to x_i. Thus, the problem will be formulated as follow:

$$Q = f_\theta(X, A),$$

where $Q \in \mathbb{R}^{N\times|C|}$ is the prediction matrix of the classifier for all nodes including unlabeled ones.

3.2 Neural Graph Encoder

To begin with, for a graph G, we uniformly represent a disease, symptom, or patient node as $v \in G$ to be succinct. Then, at the l-th information propagation layer, the embedding $\mathbf{h}_v^l$ of node v is calculated as:

$$\mathbf{h}_{\mathcal{N}(v)}^l = AGGREGATE\left(\left\{\mathbf{h}_{v'}^{l-1}, \forall v' \in \mathcal{N}(v)\right\}\right)$$
$$\mathbf{h}_v^l = \sigma\left(\mathbf{W}^l \cdot \left[\mathbf{h}_v^{l-1}; \mathbf{h}_{\mathcal{N}(v)}^l\right]\right)$$

where $\mathbf{W}^l$ is the weight matrix to be learned at the l-th layer, $\mathbf{h}_v^{l-1}$ is node v 's embedding at the previous layer, and we denote the total layer size as L. We use $[\cdot;\cdot]$ to represent the concatenation of two vectors, and use $\mathcal{N}(v)$ to denote the set of evenly sampled neighbor nodes of v. Note that for $l = 0$, the node embedding $\mathbf{h}_v^0 \in \mathbb{R}^d$ is initialized via either randmonized values or side information from the data (subject to availability). For instance, given a patient node, with the available patient demographics and medical profiles in the EMR data, then $\mathbf{h}_v^0$ will be initialized as a real-valued dense feature vector, and each digit in $\mathbf{h}_v^0$ represents the observed value of a feature dimension (e.g., age). $\mathbf{h}_{\mathcal{N}(v)}^l$ is the synergic representation resulted from the aggregation function,which is designed to aggregate the embeddings of node v 's neighbors at the $(l-1)$-th layer. σ is a non-linear activation function (e.g., tanh), and the aggregator can be chosen as mean, max pooling, RNNs, etc. By default, we deploy mean $(\cdot)$ in our model for information aggregation.

Then, we take a normalization step before reaching the final embedding for all nodes at the last layer L :

$$\mathbf{h}_v = \frac{\mathbf{h}_v^L}{\left\|\mathbf{h}_v^L\right\|_2}, \forall v \in G$$

After passing through the GCN layers, we will feed $\mathbf{h}_v$ through an MLP layer to obtain the final embedding representation.

$$\mathbf{z}_v = \sigma\left(\mathbf{W}\mathbf{h}_v\right), \forall v \in G$$

where $\mathbf{W}$ is the learnable weight, and $\mathbf{z}_v$ is the final embedding for node v.

Attention mechanism

We can apply attention mechanisms to selectively encode the information from neighbors according to their importance to the target node v. This is achieved by taking a weighted sum of the representations of all v 's neighbor nodes:

$$\mathbf{h}_v^l = \sum_{v' \in \mathcal{N}(v)} \alpha_{v'v} \mathbf{M} \mathbf{h}_{v'}^{l-1}$$

where $\mathbf{M}$ is the transformation weight matrix, and $\alpha_{v'v}$ is the attentive weights indicating the importance of neighbor node $v' \in \mathcal{N}(v)$ when calculating $\mathbf{h}_v^l$. Each $a_{v'v}$ is computed via the following attention network:

$$\alpha_{v'v} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{N}\mathbf{h}_v^{l-1} \| \mathbf{N}\mathbf{h}_{v'}^{l-1}\right]\right)\right)}{\sum_{k \in \mathcal{N}(v)} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{N}\mathbf{h}_v^{l-1} \| \mathbf{N}\mathbf{h}_k^{l-1}\right]\right)\right)}$$

with a projection vector a and the weight matrix $\mathbf{N}$. Essentially, the learned attentive weights allows the aggregator to lay more emphasis on neighbor nodes having more contributions to the message passing process, thus being able to generate highly expressive node embeddings.

Disease Prediction for Patients

The purpose of the graph decoder in our model is to translate the information contained in the symptom, disease, and patient node embeddings into predictions of possible diseases associated to a given patient. In summary, given the embedding $\mathbf{z}_p$ of a patient p, the graph decoder maps it to a vectorized output $\hat{\mathbf{c}}_p \in \{0,1\}^{|C|}$ which approximates this patient's multi-hot disease label $\mathbf{c}_p \in \{0,1\}^{|C|}$. Specifically, in this decoding process, every element in $\hat{\mathbf{c}}_p$ is computed via:

$$\hat{\mathbf{c}}_{p,n} = \text{sigmoid}\left(\mathbf{z}_p^T \mathbf{Q}_n\right), \forall m_n \in \mathcal{V}_M$$

where $\hat{\mathbf{c}}_{p,n}$ is the n-th element in $\hat{\mathbf{c}}_p$, while $n \leq |\mathcal{V}_M|$ is used for indexing all the diseases m. The closer $\hat{\mathbf{c}}_{p,n} \in \hat{\mathbf{c}}_p$ is to 1 , the more likely patient p is diagnosed with disease m_n.$\mathbf{Q} \in \mathbb{R}^{|\mathcal{V}_M| \times d}$ carries the corresponding regression weights for all diseases, and $\mathbf{Q}_n$ is the n-th column of it. To train our model, we quantify the prediction error via the following negative log likelihood loss function:

$$\mathcal{L} = -\sum_{n=1}^{|\mathcal{V}_M|} \mathbf{c}_{p,n} \log\left(\hat{\mathbf{c}}_{p,n}\right)$$

3.3 Proposed Model

I will build two different types of Graph-based models, using either a single graph or multiple graphs as inputs. To avoid the over-smoothing problem, I will only use two layers of Graph Convolution Network (GCN) and incorporate skip connections to preserve information. To enhance the model's representational capacity, I will combine MLP layers, including Fully Connected and Batch Normalization layers, to extract information effectively. Moreover, during the training process, I will employ various mechanisms to aggregate information, such as SUM, MAX, MEAN, and Attention, to effectively combine the information from neighboring nodes. For updating node representation, I will utilize MLP layers to learn isomorphic or non-isomorphic sub-graphs and employ a GRU network to learn sequential representations of neighboring nodes.

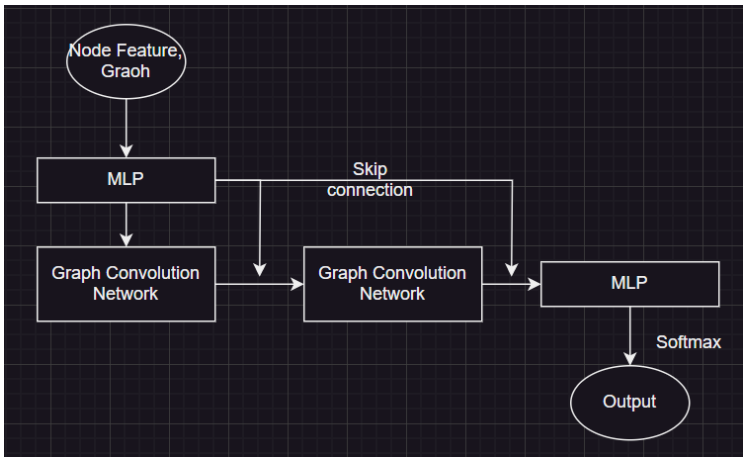

Figure 3: Proposed model with 1 input graph

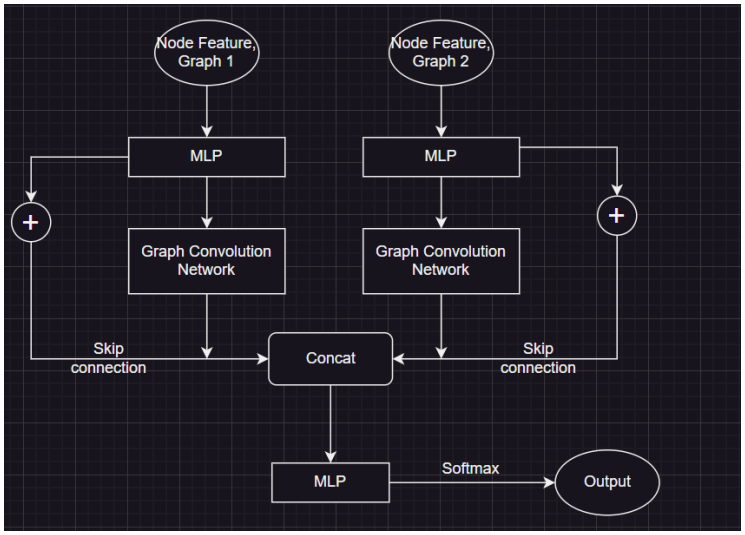

Figure 4: Proposed model with 2 input graph

Training method

When training model, I have this Hypothesis Hypothesis: Reducing the size of the input graph by sampling a small percentage of connections (5%, 10%, 20%, 30%, etc.) does not decrease the model's performance if information is combined using the Attention mechanism. Because, based on the data, each individual is only grouped with 120 other individuals (1/41) with the same disease, so only a small representative group is needed for each disease group. The Attention mechanism helps the model learn important connections within the graph. A node learns which connections are significant with its neighboring nodes. Therefore, removing edges with low probability does not affect the performance. The experimental results in the following section demonstrate the effectiveness of this training approach.

Create feature for Disease Graph

I propose to construct features for the disease graph based on the TF-IDF formula by these steps:

(1) Calculate the top 3 most common symptoms for each disease.
(2) TF (Term Frequency): Count the number of individuals with each disease who exhibit the corresponding common symptom.
(3) IDF (Inverse Document Frequency): Count the occurrence of each common symptom across the entire dataset.
(4) Calculate the weight by multiplying TF with 1/IDF.

So we have a feature matrix for graph disease.

4 RESULTS & CONCLUSION

4.1 Result

In this report, to evaluate the classification performance of the model, I optimize the parameters based on the F1-macro, Recall-macro, and Precision-macro metrics. This approach is chosen because in reality, there are numerous rare diseases with very few data records. Optimizing parameters using the macro strategy helps to address the issue of data imbalance to some extent.

Graph Construction

Firstly, we need to define how to construct graph. I proposed some stretagy to construct 2 type of graph: Disease Graph, Patient Graph.

For establishing connections between two patients, we can:

- Strategy 1: Create an edge between two patients if they have the same type of disease (To avoid data leakage, only use the training set to construct the graph).
- Strategy 2: Create an edge between two patients if they share two similar symptoms. (This approach can be combined with Strategy 1 to create a new graph).
- Strategy 3: Create an edge between two patients if their cosine similarity, measured by a threshold γ, exceeds a certain threshold.

$$a_{ij} = \begin{cases} 1, & \text{if cosine-simalarity } (x_i, x_j) \geq \gamma \\ 0, & \text{otherwise} \end{cases}$$

where a_{ij} is indicator for representing the existence of an edge connecting patients i-th and j-th and x_i is sample of patient i-th.

For establishing connections between two diseases, we can create a connection between two diseases if they share the top two prominent symptoms.

We divided the dataset into train/validation/test sets with a ratio of 70/10/20, respectively. The results on 2 data sets will be represented below.

4.1.1 Multi-class Classification. After conducting experiments on the Kaggle dataset, we obtained the following results for each graph construction strategy and methods of aggregating information from neighboring nodes and updating node representations:

Model	Accuracy	F1-macro	Recall-macro	Precision-macro
Baseline Random Forest	0.91 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.95 ± 0.01
Strategy 1 (Mean + GRU)	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
Strategy 2 (Sum + Concat)	0.98 ± 0.01	0.98 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
Strategy 3 (Attention + GRU + sample 20%)	0.95 ± 0.01	0.94 ± 0.01	0.95 ± 0.01	0.95 ± 0.01
Strategy 3 (MultiHeadAttention + concat + sample 5%)	0.99 ± 0.01	0.99 ± 0.01	1 ± 0.01	0.99 ± 0.01
Multi Graph Input (Mean + GRU)	1 ± 0.01	1 ± 0.01	1 ± 0.01	1 ± 0.01

We can observe that the Graph-based models all outperform the baseline model, Random Forest, demonstrating their effectiveness on this dataset. Furthermore, the training method that reduces the size of the input graph and utilizes the Multi Head Attention layer helps the model focus on important connections within the graph, resulting in excellent classification results.

Model	Accuracy	F1-macro	Recall-macro	Precision-macro
Baseline KNN	0.73 ± 0.01	0.73 ± 0.01	0.73 ± 0.01	0.74 ± 0.01
Strategy 1 (Mean + GRU)	0.77 ± 0.01	0.76 ± 0.01	0.76 ± 0.01	0.79 ± 0.01
Strategy 2 (Sum + Concat)	0.71 ± 0.01	0.65 ± 0.01	0.64 ± 0.01	0.79 ± 0.01
Strategy 3 (Attention + GRU + sample 20%)	0.77 ± 0.01	0.77 ± 0.01	0.77 ± 0.01	0.79 ± 0.01

4.1.2 Binary Classification. We proceeded to build a binary classification model using the Diabetes dataset. To construct the graph according to Strategy 2, I performed age binning to form age groups and established connections between patient age groups. We can see that the Graph-based models are still very effective; however, constructing the graph according to Strategy 2 resulted in less effective learning for the model.

4.2 Conclusion

Through this project, I have achieved the following results:

- Understanding and constructing Graph-based models such as GraphSage, Graph Attention, etc., with one or multiple graphs as inputs.

- Proposing different approaches to construct various graphs (e.g., disease graph, symptom graph) and building a feature set for diseases based on applying TF-IDF measure to prominent symptoms.
- Introducing the Graph-based MultiHeadAttention model with a mechanism to sample edges, which helps reduce the size of the graph while ensuring classification performance.

In the future, I will supplement the data by crawling additional data to evaluate the effectiveness of the model with a larger dataset. Also, Constructing various weighted graphs with different types of nodes (including graphs consisting of patient nodes, disease nodes, and interconnected symptom nodes).

REFERENCES

[1] Rushil Anirudh and Jayaraman J Thiagarajan. Bootstrapping graph convolutional neural networks for autism spectrum disorder classification. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3197–3201. IEEE, 2019.

[2] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[3] Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Latent-graph learning for disease prediction. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*, pages 643–653. Springer, 2020.

[4] Hao Du, Jiashi Feng, and Mengling Feng. Zoom in to where it matters: a hierarchical graph based model for mammogram analysis. *arXiv preprint arXiv:1912.07517*, 2019.

[5] Richard Hillestad, James Bigelow, Anthony Bower, Federico Girosi, Robin Meili, Richard Scoville, and Roger Taylor. Can electronic medical record systems transform health care? potential health benefits, savings, and costs. *Health affairs*, 24(5):1103–1117, 2005.

[6] Yongxiang Huang and Albert CS Chung. Edge-variational graph convolutional networks for uncertainty-aware disease prediction. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23*, pages 562–572. Springer, 2020.

[7] Anees Kazi, Shayan Shekarforoush, S Arvind Krishna, Hendrik Burwinkel, Gerome Vivar, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, and Nassir Navab. Inceptiongcn: receptive field aware graph convolutional network for disease prediction. In *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26*, pages 73–85. Springer, 2019.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[9] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 2021.

[10] Yang Li, Buyue Qian, Xianli Zhang, and Hui Liu. Graph neural network-based diagnosis prediction. *Big Data*, 8(5):379–390, 2020.

[11] Fenglong Ma, Jing Gao, Qiuling Suo, Quanzeng You, Jing Zhou, and Aidong Zhang. Risk prediction on electronic health records with prior medical knowledge. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1910–1919, 2018.

[12] Sellappan Palaniappan and Rafiah Awang. Intelligent heart disease prediction system using data mining techniques. In *2008 IEEE/ACS international conference on computer systems and applications*, pages 108–115. IEEE, 2008.

[13] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrerro Moreno, Ben Glocker, and Daniel Rueckert. Spectral graph convolutions for population-based disease prediction. In *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20*, pages 177–185. Springer, 2017.

[14] J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.

[15] Jack W Smith, James E Everhart, WC Dickson, William C Knowler, and Robert Scott Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the annual symposium on computer application in medical care*, page 261. American Medical Informatics Association, 1988.

[16] Xuegang Song, Feng Zhou, Alejandro F Frangi, Jiuwen Cao, Xiaohua Xiao, Yi Lei, Tianfu Wang, and Baiying Lei. Graph convolution network with similarity awareness and adaptive calibration for disease-induced deterioration prediction. *Medical Image Analysis*, 69:101947, 2021.

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

[18] Honglei Zhang and Mancef Gabbouj. Feature dimensionality reduction with graph embedding and generalized hamming distance. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1083–1087. IEEE, 2018.