Ph.D. DISSERTATION

Learning Medical Concepts and Patient Representations with Deep Neural Networks for Medical Applications

딥 뉴럴 네트워크를 활용한 의학 개념 및 환자 표현 학습과 의료 문제에의 응용

BY

Kwak Heeyoung
AUGUST 2022

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Learning Medical Concepts and Patient Representations with Deep Neural Networks for Medical Applications

딥 뉴럴 네트워크를 활용한 의학 개념 및 환자 표현 학습과 의료 문제에의 응용

BY

Kwak Heeyoung

AUGUST 2022

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Learning Medical Concepts and Patient Representations with Deep Neural Networks for Medical Applications

딥 뉴럴 네트워크를 활용한 의학 개념 및 환자 표현 학습과 의료 문제에의 응용

지도교수 정 교 민

이 논문을 공학박사 학위논문으로 제출함

2022년 8월

서울대학교 대학원

전기 컴퓨터 공학부

곽 희 영

곽희영의 공학박사 학위 논문을 인준함

2022년 8월

위 원 장:	최 진 영
부위원장:	정 교 민
위 원:	심 규 석
위 원:	문 태 섭
위 원:	최 윤 재

Abstract

This dissertation proposes a deep neural network-based medical concept and patient representation learning methods using medical claims data to solve two healthcare tasks, i.e., clinical outcome prediction and post-marketing adverse drug reaction (ADR) signal detection. First, we propose SAF-RNN, a Recurrent Neural Network (RNN)-based model that learns a deep patient representation based on the clinical sequences and patient characteristics. Our proposed model fuses different types of patient records using feature-based gating and self-attention. We demonstrate that high-level associations between two heterogeneous records are effectively extracted by our model, thus achieving state-of-the-art performances for predicting the risk probability of cardiovascular disease. Secondly, based on the observation that the distributed medical code embeddings represent temporal proximity between the medical codes, we introduce a graph structure to enhance the code embeddings with such temporal information. We construct a graph using the distributed code embeddings and the statistical information from the claims data. We then propose the Graph Neural Network(GNN)-based representation learning for post-marketing ADR detection. Our model shows competitive performances and provides valid ADR candidates. Finally, rather than using patient records alone, we utilize a knowledge graph to augment the patient representation with prior medical knowledge. Using SAF-RNN and GNN, the deep patient representation is learned from the clinical sequences and the personalized medical knowledge. It is then used to predict clinical outcomes, i.e., next diagnosis prediction and CVD risk prediction, resulting in state-of-the-art performances.

keywords: Healthcare AI, Disease Prediction Model, Deep Neural Network

student number: 2014-22543

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recently with the accumulation of a vast amount of clinical data, clinicians and researchers have been able to apply many advanced machine learning techniques to improve the quality of healthcare services. Among the various types of biomedical data, we focus on sequential clinical data such as Electronic health records (EHRs) or healthcare claims data. These data include longitudinal patient records accumulated over a considerable period of time along with the various demographic information. One of the essential healthcare tasks that can best leverage such data is to predict future clinical outcomes. Clinical outcome prediction refers to various health risk prediction such as morbidity (i.e., the risk of disease onset), mortality, hospitalization, and treatment outcomes. A well-developed clinical outcome prediction model can assist healthcare practitioners in making more accurate decisions, hence improving the quality of healthcare.

Another application where these data can be utilized well is an adverse drug reaction (ADR) signal detection in post-marketing drug surveillance. Most ADR detection research has been aimed to predict ADRs in pre-marketing phases, using biomedical information sources such as chemical structures, protein targets, and therapeutic indications. Nevertheless, capturing potential ADRs from the entire population in post-marketing phases is also essential to fully establish the ADR profiles [1]. In post-

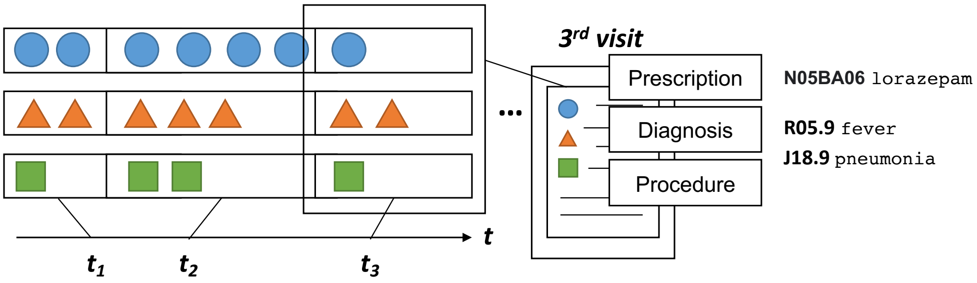


Figure 1.1: Data structure of a NHIS-NSC sample.

marketing drug surveillance, one needs to monitor of drugs once they reach the market after clinical trials. The potential causal relationship between an adverse event and a drug is called an ‘ADR signal’ when the relation is previously unknown or incompletely documented. To capture ADR signals, it is important to evaluate drugs taken by individuals over an extended period of time. Therefore, clinical databases with longitudinal patient records can be an important data source for post-marketing ADR signal detection.

In this dissertation, we use National Health Insurance Service-National Sample Cohort (NHIS-NSC), the 12-year healthcare claims data [2] to solve two healthcare tasks, i.e., clinical outcome prediction and post-marketing ADR signal detection.

The basic element for applying deep learning techniques to sequential clinical data is to represent medical concepts and patients as computable vectors. Unlike text in which tokens are just sequentially arranged, clinical sequence data have multiple medical codes for each visit and this visit occupies a single time step of the sequence as depicted in Figure 1.1. Considering these characteristics, it is important how to represent each medical code and how to model a single visit. Ultimately, how to model individual patients with multiple visits and demographic information is paramount.

We propose various methods to learn deep patient representations for clinical outcome prediction, and to enhance medical concept embeddings using graph structure for post-marketing ADR detection. First, we propose a Recurrent Neural Network (RNN)

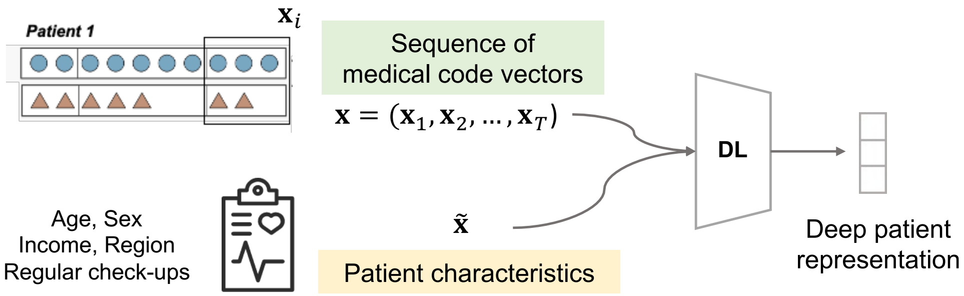

Patient 1
$\mathbf{x}_i$
Sequence of medical code vectors
$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$
DL
Age, Sex
Income, Region
Regular check-ups
$\tilde{\mathbf{x}}$
Patient characteristics
Deep patient representation

Figure 1.2: Deep patient represenation learned upon two different types of patient information.

model that learns patient representations based on the clinical sequences along with the fixed patient characteristics, and predicts the risk probability of a cardiovascular disease onset. The data sample of NHIS-NSC consisted of two different patient information, i.e., time-varying sequential information and the fixed patient characteristics as shown in Figure 1.2. We provide the information about the patient characteristics in Table 1.1

To fully exploit both temporal records and the patient characteristics together, we proposed a self-attentive fusion encoder (SAF) through the following research:

- (SCI) **H Kwak**, J Chang, B Choi, S Park, and K Jung, Interpretable Disease Prediction Using Heterogeneous Patient Records with Self-Attentive Fusion Encoder, *Journal of the American Medical Informatics Association(JAMIA)*, July 2021

In this work, we proposed a RNN-based disease prediction model that efficiently fuses different types of information using self-attention. Self-attention is an attention mechanism that enables different positions of an input sequence to interact with each other[17, 18, 19]. It computes the attention scores for each interaction and outputs the representation of each position of the sequence. In our proposed SAF, self-attention is applied after the RNN encodes of the temporal sequence, and the patient characteristics

Table 1.1: Patient characteristics extracted from NHIS-NSC

DB Category	# of variables	Description
Qualification DB	4	Sex, age group, income-level and residential area
Health Check-up DB	4	Body measurement (body mass index, waist circumference and blood pressure)
	11	Blood test (fasting blood glucose level, total cholesterol level, gamma-GTP and etc.)
	11	Patient and family history of major diseases (hypertension, cancer, and etc.)
	5	Surveys on smoking, drinking and physical activity

are combined with feature-based gating. We demonstrate that high-level associations between two heterogeneous patient records are effectively extracted during the process of feature-based gating and the computation of self-attention.

In a comparison with other fusion mechanisms, we show that our SAF-RNN successfully combines two pieces of heterogeneous information and therefore significantly increases the predictability. We further explain the obtained results by showing the relative importance of each time step in the temporal sequence for affecting the risk probability. Hence, our model provides interpretability for the predictions so that they can be understood by a human.

In the previous study, additional performance improvements were made by representing the medical codes in a distributed code representation. This confirms that the distributed representation of medical code contains the information necessary to process temporal information in patient records. We therefore, introduced a graph structure in the subsequent study to enhance such temporal information. We constructed a graph using the similarity between the distributed vectors of medical code and the statistical information between medical codes. We then obtained the medical code representations with the enhanced temporal information using the Graph Neural Networks(GNN)-based representation learning.

Based on the obtained medical code representations, we proposed a model for detecting potential ADR signals of post-marketing drugs. As GNN models have been demonstrated [3, 4] their power to solve many tasks with graph-structured data, we used GNN-based approach for ADR detection through the following research:

- **H Kwak**, M Lee, S Yoon, J Chang, S Park, K Jung, Drug-disease Graph: Predicting Adverse Drug Reaction Signals via Graph Neural Network with Clinical Data, *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, May 2020, Singapore

This study is the first to propose a method of simultaneously detecting ADR signals for every possible drugs with the graph-structured clinical records. The proposed model empirically showed competitive ADR prediction performance on the side effect resource database (SIDER). It especially predicted ADR candidates that do not exist in the existing ADR database, showing its capability to supplement the ADR database. The constructed graph is a heterogeneous graph with drug and disease nodes, as it is depicted in Figure 1.3. The corresponding graph construction only requires simple data processing and well-established medical terminologies. Therefore, our work does not demand case-by-case feature engineering that requires expertise, and thereby the detection for the whole drug candidates can be fully automated.

Finally, rather than just learning patient representations using patient records alone, we utilized Semantic Medline Knowledge Graph (SemMed KG) that specifies relationships between medical entities to augment the deep patient representation with prior medical knowledge. Here, a personalized knowledge graph is made by extracting only the subgraph of the SemMed KG consisting of the medical code in the patient's record.

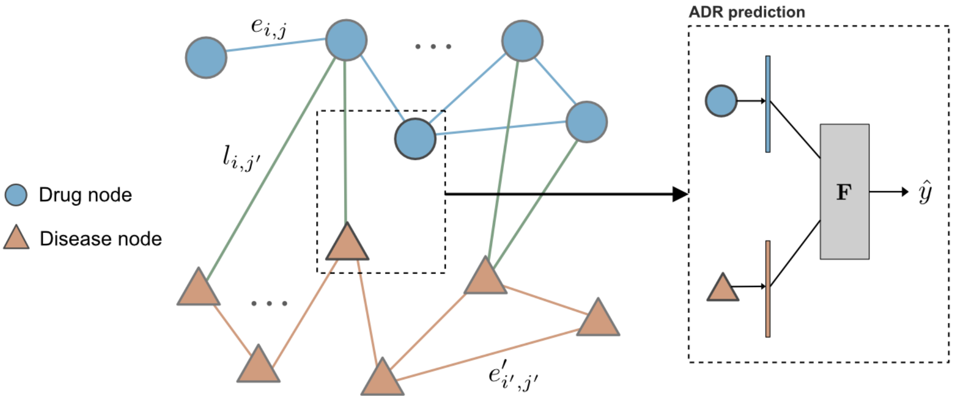


Figure 1.3: Heterogeneous graphs consisting of drug and disease nodes

Based on the personalized KG, we build the deep patient representations upon the personalized medical knowledge using GNNs. Along with the temporal information, the deep patient representation is used to predict some clinical outcomes. Specifically, we evaluated the performances of our model on two tasks, i.e., the next diagnosis prediction and the CVD prediction as in the first study.

Additionally, we seek to harness pre-training for GNNs to fully exploit the logical rules inherent in the KGs. We perform the KG completion (KGC) task, which is to predict the plausibility of a given triplet, as a pre-training task to further enhance the subgraph representation. The SemMed KG expresses various relations between medical entities, and these relations are strongly associated with other adjacent relations according to certain logical rules. To encode these logical rules, we utilize the subgraph representations for the KGC. Just as Masked Language Modeling (MLM) provides self-supervision for learning the contexts, KGC, which predicts the masked edges based on the surrounding subgraphs, can provide self-supervision for learning the structure of the SemMed KG.

To improve the performance of KGC task, we utilize the method, proposed in our other research:

- **H Kwak**, H Bae, K Jung, Subgraph Representation Learning with Hard Negative Samples for Inductive Link Prediction, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, Singapore

In this work, we suggest a novel inductive link prediction model, called **Subgraph Infomax (SGI)**, where the relation embedding is trained to contain more meaningful information about subgraphs via the mutual information (MI) maximization objective. Specifically, SGI consists of a GNN-based scoring network for computing the score of a given triplet and a module for MI maximization. We trained SGI to maximize the MI between the relation embedding and the subgraph representation. Performing the ranking evaluation of previously presented SOTA models, GraIL [5], TACT [6], our model showed superior performances in both inductive versions of Nell-995 and FB15k-237 KG datasets.

After pre-training SGI with the KGC training objective on SemMed KGs, we used subgraph representations encoded with the SGI model for the clinical outcome prediction. As a result, it showed better CVD prediction performances in the case of using personalized KG representation together with SAF-RNN than in the case of using SAF-RNN alone. This behavior was also observed in the next diagnosis prediction.

The remainder of this dissertation is organized as follows. Chapter 2 provides a background on medical concept embeddings and deep patient representation in clinical outcome prediction. In Chapter 3, we explain the method to fuse two heterogeneous patient records . Chapter 4 explains a model for the post-marketing ADR signal detection that uses the medical concept embeddings, enhanced with the graph structure. Further investigation on the knowledge-enhanced deep patient representation for clinical event prediction is discussed in Chapter 5. Finally, the dissertation is concluded in Chapter 6.

Chapter 2

Background

This dissertation presents our research on deep neural network-based models for various healthcare tasks. To build a deep neural network model that deals with discrete information in the clinical visits, i.e., the sequence of medical codes, we first need to understand how to represent the medical concepts, each clinical visit, and the patient. In this chapter, we introduce widely-used techniques for medical representations, which are the building blocks in developing a healthcare AI model.

2.1 Medical Concept Embedding

To represent medical concepts such as diagnosis, medication, and procedure, medical codes are used for encoding Electronic Health Records (EHR) and the claims data. For example, ICD-10 (International Classification of Diseases, 10th revision), published by the United States for classifying diagnoses and reasons for visits in all health care settings, is the most commonly-used diagnosis code. We use National Health Insurance Service-National Sample Cohort (NHIS-NSC) data, and the concepts of diagnosis and medication in NHIS-NSC are represented in the form of KCD (Korean Standard Classification of Diseases) codes, which is the Korean translation of ICD-10, and ATC (Anatomical Therapeutic Chemical Classification System) codes.

Just as words and phrases in text data are embedded into computable vectors for NLP applications, medical concepts are mapped to vector representations to apply deep learning techniques to clinical data; we call them medical concept/code embeddings. The medical concept embedding greatly influences the performance of clinical prediction models as it is the primary input feature for the model. We introduce essential techniques for medical code-level representations, i.e., Multi-hot code representation, Grouped code representation, and distributed code representation.

One-hot/ Grouped Code Representation

The naive approach for encoding medical codes is one-hot encoding, which is to represent categorical variables as binary vectors; all the components are marked with zero except the index of the variable, marked with 1. Since clinical sequence data have multiple medical codes for a single clinical visit, each visit is often represented by multi-hot vectors. However, most medical codes are composed of categorical variables with hierarchical structures. There are many well-constructed ontologies in the medicine area, such as International Classification of Diseases (ICD), Clinical Classifications Software (CCS) [7] , Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) [8]. To leverage the medical knowledge that can be induced from these structures, medical concepts are often represented as the grouped one-hot vectors as in 2.1. The one-hot vector format is adopted to represent each categorical information of medical codes. We utilize the hierarchical structure of categorical codes (i.e. ATC and ICD-10 codes) by adopting the one-hot vector format. Since there are multiple categories for each code, it is shown as a concatenation of one-hot vectors, thus, a multi-hot vector.

Distributed Code Representation

Alternatively, medical concept vectors based on distributed code representations were proposed and helped clinical prediction models obtain performance gain. Most large-

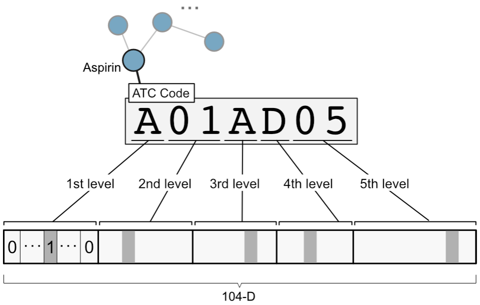


Figure 2.1: Example of grouped code representations.

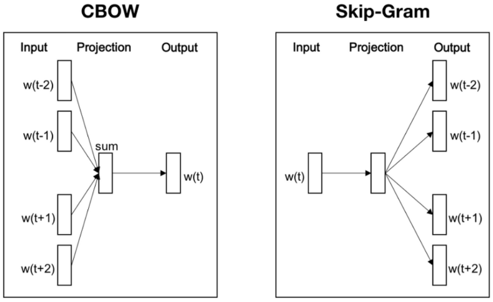


Figure 2.2: Two approaches for distributed code representations

scale clinical databases including NHIS-NSC, are collected in the form of longitudinal visit records of the patients. Therefore, unsupervised methods for word representations using the huge text corpus can be adopted. Choi et al. first used word2vec model, a widely-used word embedding technique in natural language processing, to learn the latent representation of medical codes in EHR data, in a way that captures the temporal proximity between them [9].

Word2vec suggested by Mikolov et al. (2013) [10], introduces two model architectures called continuous bag-of-words (CBOW) and skip-gram for computing continuous vector representations of a word based on the surrounding context. The difference between two approaches is depicted in Figure 2.2. Word2vec showed improved performances in evaluating syntactic and semantic similarities between words. Similarly, distributed code representations embed contextually similar medical codes close to each other in the embedding space. To build a DNN-based clinical prediction model, we can either use distributed code vectors as the initial weights of the embedding layer or use the pre-trained code vectors as the building blocks to represent a single clinical visit.

Now, we explain how to process the patient's longitudinal records for applying skip-gram model. In the patient's longitudinal records, each patient can be treated as a sequence of hospital visits $\{v_1^{(n)}, v_2^{(n)}, ..., v_{T_n}^{(n)}\}$ where n represents each patient in the data, and T_n is the total number of visits of the patient. The i^{th} visit can be denoted as $v_i^{(n)} = \{\mathcal{P}_i{}^{(n)}, \mathcal{D}_i{}^{(n)}\}$ where $\mathcal{P}_i{}^{(n)}$ is the set of prescribed codes and $\mathcal{D}_i{}^{(n)}$ is the set of diagnosed codes in the i^{th} visit. Within a set of codes, codes are listed in arbitrary order. The size of each set is variable since the number of prescribed/diagnosed codes varies from visit to visit. With these sets of codes, we form a drug sequence $\mathbf{Seq}_{drug}{}^{(n)}$ and a disease sequence $\mathbf{Seq}_{disease}{}^{(n)}$ of n^{th} patient by listing each of the codes in a temporal order, as it is described below (Here, we leave out the symbol n):

$$\begin{aligned} \mathbf{Seq}_{drug} &= \{p_1, p_2\,, \dots, p_{T_p}\}\,,\; p_x \in \mathcal{P}_i, \\ \mathbf{Seq}_{disease} &= \{d_1, d_2\,, \dots, d_{T_d}\}\,,\; d_y \in \mathcal{D}_i, \end{aligned} \tag{2.1}$$

where $p_x \in \mathbb{R}^{V_p}$ and $d_y \in \mathbb{R}^{V_d}$ are the one-hot vectors representing each of the medical codes in the sequences. V_p and V_d are the vocabulary size of the whole prescription and diagnosis codes within the data, respectively. T_p and T_d represent the total number of prescription/diagnosis codes of the patient's record. In this way, we can build a corpus consisting of $\mathbf{Seq}_{drug}$ or $\mathbf{Seq}_{disease}$. With $\mathbf{Seq}_{drug}$ or $\mathbf{Seq}_{disease}$, we can apply the Skip-gram model with negative sampling scheme.

2.2 Encoding Sequential Information in Clinical Records

Predicting future clinical events such as morbidity (i.e., the risk of disease onset), mortality, hospitalization, and treatment outcomes is an essential healthcare task. With the help of a vast amount of clinical data, many advanced machine learning techniques have been used to develop effective prediction models. A well-developed prediction model using various deep learning approach can assist healthcare practitioners in making more accurate decisions, hence improving the quality of healthcare AI.

One prominent method for obtaining a patient representation is first expressing an entire longitudinal patient record as a sequence of medical concept vectors and then applying deep architectures [9, 11, 12, 13, 14, 9, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. During this process, medical concept representations are either pre-trained or jointly learned during the process of end-to-end learning.

For learning a deep patient representation, Convolutional Neural Networks(CNNs) have been employed in several works [26, 21, 22, 23]. They transform a medical record into a temporal matrix or a sequence of discrete clinical event codes and perform a convolutional operation over them. However, the most popular architectures for learning a patient representation are an RNN and its variants since they are developed to model the sequential data. Much research on clinical event prediction has yielded a recurrent neural network (RNN)-based approach to capture the temporal patterns within longitudinal patient records [9, 11, 12, 13, 14, 27, 15, 16, 17, 18, 19, 20].

Choi et al. trained a GRU-RNN on sequences of pretrained medical concept vectors to predict future diagnoses or the onset of heart failure [9, 11, 12]. Pham et al. used an LSTM-RNN for predicting the next diagnosis and intervention for specific groups of patients [13]. More recent work on clinical event prediction has incorporated an attention mechanism with RNNs to interpret the prediction results [27, 15, 16, 17, 18, 19, 20]. An attention mechanism allows a model to place more attention weights on the parts of the model that are more relevant to the given prediction [28, 29, 30]. Choi et al. were the first to utilize an attentional RNN model for identifying significant visits and features for heart failure prediction task [27]. Other studies [15, 16, 17, 18] also used attentional RNN models to measure the importance of features of various levels and of various types (i.e., the medical code-level, hospital visit-level, within/between subsequences-level, and multichannel attention). Self-attention has also been employed to capture the relations between different visiting events [19] and medical codes [20].

Chapter 3

Deep Patient Representation with Heterogeneous Information

Predicting future clinical events such as morbidity (i.e., the risk of disease onset), mortality, hospitalization, and treatment outcomes is an essential healthcare task. With the help of a vast amount of clinical data, many advanced machine learning techniques have been used to develop effective prediction models. A well-developed prediction model can then assist healthcare practitioners in making more accurate decisions, hence improving the quality of healthcare.

Electronic health records (EHRs) or healthcare claims data are commonly used since they include various patient information, such as longitudinal patient records accumulated over a considerable period of time. Much research on clinical event prediction has yielded a recurrent neural network (RNN)-based approach to capture the temporal patterns within longitudinal patient records [9, 11, 12, 13, 14, 27, 15, 16, 17, 18, 19, 20]. In addition to temporal patient records, many studies also often utilize patient characteristics (i.e., demographic profiles or health examination results) for prediction purposes. However, these studies incorporate patient characteristics into the model simply by concatenating them to the inputs or by hidden representation [14, 21, 22, 23, 24].

To fully exploit both temporal records and the patient characteristics together, we propose a self-attentive fusion encoder (SAF) for an RNN-based disease prediction model that efficiently fuses different types of information using self-attention. Specifically, we propose SAF-RNN, which applies an SAF module to the GRU-RNN model to predict cardiovascular disease (CVD) events using the medical histories of general patients from healthcare claims data. Self-attention is an attention mechanism that enables different positions of an input sequence to interact with each other [31, 32, 33]. It computes the attention scores for each interaction and outputs the representation of each position of the sequence. In our proposed SAF, self-attention is applied after the RNN encodes of the temporal sequence, and the patient characteristics are combined with feature-based gating. We demonstrate that high-level associations between two heterogeneous patient records are effectively extracted during the process of feature-based gating and the computation of self-attention.

The experimental results on a general patient dataset show that the proposed method achieves superior AUROC and AUPRC performances on CVD prediction compared to all other methods. In a comparison with other fusion mechanisms, we show that our SAF-RNN successfully combines two pieces of heterogeneous information and therefore significantly increases the predictability. We further explain the obtained results by showing the relative importance of each time step in the temporal sequence for affecting the risk probability. Hence, our model provides interpretability for the predictions so that they can be understood by a human. Additionally, we performed a sensitivity analysis to examine the model's sensitivity to the most obvious factors (e.g, outpatient CVD diagnosis before CVD admission) by masking them. We show that our model consistently outperforms the other methods, even in this challenging setting.

3.1 Related Work

Patient Representation Learning and Clinical Outcome Prediction

Recently, there have been many efforts to apply DL methods to understand medical data such as EHRs. Many of these studies learn deep patient representations from medical data so that the learned representations are projected into a vector space. The qualities of the derived patient representations are then evaluated on clinical outcome prediction tasks [34]. Such research includes predicting the risks of disease onset, mortality, and any future events that can be encountered by the patient, such as readmission, multilabel diagnoses in the next encounter, transfer to the ICU, etc [9, 11, 12, 13, 14, 9, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26].

One prominent method for obtaining a patient representation is first expressing an entire longitudinal patient record as a sequence of medical concept vectors and then applying deep architectures such as Convolutional Neural Networks (CNNs) [9, 11, 12, 13, 14, 9, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. The most popular architectures for learning a patient representation are an RNN and its variants since they were developed to model sequential data. Choi et al. trained a GRU-RNN on sequences of pretrained medical concept vectors to predict future diagnoses or the onset of heart failure [9, 11, 12]. Pham et al. used an LSTM-RNN for predicting the next diagnosis and intervention for specific groups of patients [13].

More recent work on clinical event prediction has incorporated an attention mechanism with RNNs to interpret the prediction results [27, 15, 16, 17, 18, 19, 20]. An attention mechanism allows a model to place more attention weights on the parts of the model that are more relevant to the given prediction [28, 29, 30]. Choi et al. were the first to utilize an attentional RNN model for identifying significant visits and features for heart failure prediction task [27]. Other studies [15, 16, 17, 18] also used attentional RNN models to measure the importance of features of various levels and of various types (i.e., the medical code-level, hospital visit-level, within/between subsequences-

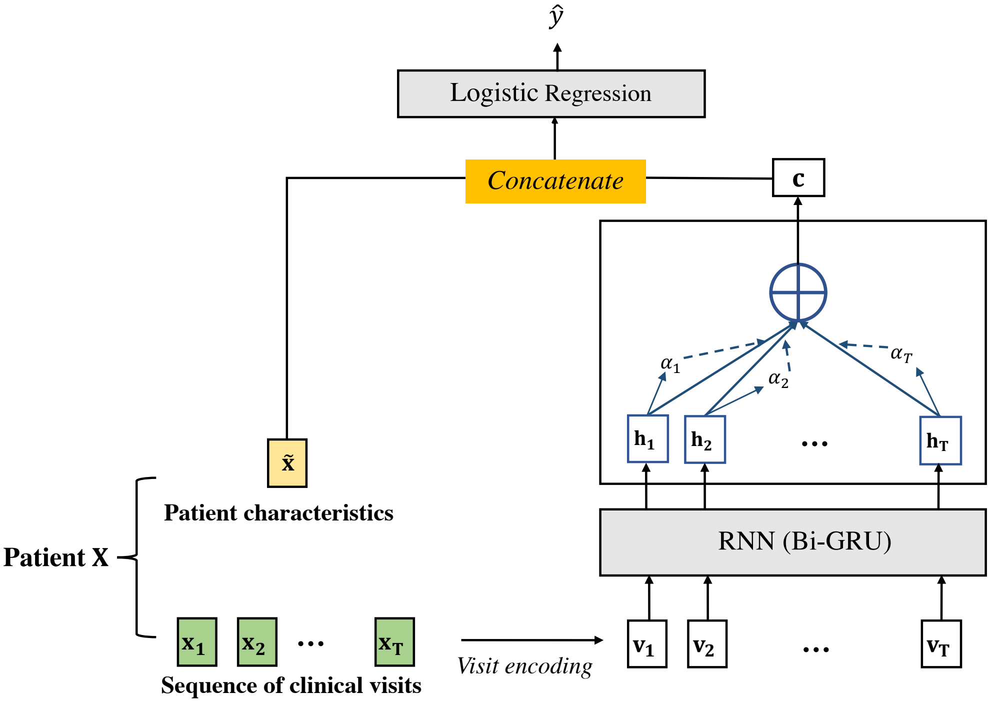


Figure 3.1: Standard approach to incorporate the patient characteristics.

level, and multichannel attention). Self-attention has also been employed to capture the relations between different visiting events [19] and medical codes [20]. Our work also utilizes self-attention to facilitate the interpretation of the obtained results. However, the main purpose of using self-attention in our model is to fuse heterogeneous patient records adeptly.

Using Heterogeneous Patient Records in Clinical Event Prediction

There have been several attempts to use patient characteristics such as demographic profiles and health examination results to predict clinical events. Studies such as [14, 21, 22, 23, 24] used patient characteristics, together with other clinical information. Esteban et al. classified patient data into static and dynamic features and combined these two types of features into an input for an RNN model to predict the complications related to kidney transplantation [14]. Lin et al. proposed a neural network model that predicts hypertension by combining the demographic information with initial signatures and laboratory results, such as heart rates and sodium and creatine levels [22]. Heo et al. additionally used health examination information in an X-ray based deep learning diagnostic model [23]. The model proposed by Finneas et al. encodes the clinical records during the most recent several hours with CNNs and combines these records with demographic information to make predictions about critical risks [24]. However, far too little attention has been paid to the fusion of heterogeneous information, and all of these previous studies have simply concatenated different feature vectors. The most standard approach to incorporate patient characteristics is depicted in 3.1. On the other hand, our research effectively combines temporal patient records with patient characteristics using a self-attentive fusion mechanism.

Attention-based Fusion Mechanism in Multimodal Deep Learning

The methodologies used to fuse different information channels can also be found in the field of multimodal deep learning. In multimodal deep learning, multiple modalities are fused for a single prediction task, such as speech emotion recognition [30], which uses audio, visual and textual data, and visual question answering(VQA) [35]. Recent approaches in these areas have introduced attention mechanism to capture the high-level associations between multiple heterogeneous data [36, 37, 38, 39]. In the VQA domain, Yu et al. used a co-attention learning module to jointly learn the attention for both images and questions [36, 37]. While [36] used self-attention only for question embedding [37], modeled self-attention for both questions and images. For speech emotion recognition, [38] employed a GRU-RNN for each modality (i.e., acoustic, textual and visual) and fused them using attention. [39] suggested a self-attentive feature-level fusion method that applies self-attention after fusing the audio and textual features. Similar to these works on multimodal deep learning, we fused heterogeneous patient records using self-attention with feature-level gating.

3.2 Problem Statement

We aimed to predict the patient-specific risks of CVD events in the next visit given a 2-year clinical visit history and patient characteristics. We defined the problem as follows: Given one patient's record denoted as $\mathbf{X} = (\mathbf{x}, \tilde{\mathbf{x}})$, where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ is a sequence of clinical visits and $\tilde{\mathbf{x}}$ denotes the patient characteristics, the goal was to estimate the risk probability $\hat{y}$ of the patient (here, we leave out the notation for each patient). The labels were given as values of 0 and 1, where $y = 1$ indicates that the patient had the disease. $\mathbf{x}_i$ is a set of prescriptions and diagnosis codes for the ith visit, and the sequence $\mathbf{X}$ was pre-trained to obtain a computable input vector v, which is described in the following subsubsection. To express the patient characteristics $\tilde{\mathbf{x}}$, we used the patient's demographic profile (e.g., age, sex, residential area and income level) and their most recent health examination results. We encoded the patient characteristics into a one-hot vector form. More information about the patient characteristics is in Table 1.1.

Pre-trained Representations of the Medical Codes

In a patient's longitudinal visit sequence, each visit can be represented as a set of diagnosed disease codes and prescribed medication codes. These multiple medical codes can be represented in the form of multi-hot encoded binary vectors, for which the dimensionality is the total number of unique medical codes. However, this naïve representation cannot capture the temporal proximity between the medical codes in sequential records. Hence, to capture the temporal proximity between the medical codes and facilitate vector computation, we encoded each diagnosis and prescription code into a low-dimensional real-valued vector space. Motivated by the successful applications of Skip-gram in constructing medical concept vectors [9, 11, 12], we used Skip-gram, a widely-used word embedding technique [10], to learn representations for medical codes. The details of the learning process of Skip-gram embeddings are described in

Chapter 2. Then, we represented each clinical visit as a sum of the learned Skip-gram embeddings of each medical code within the visit, as follows:

$$\mathbf{v}_i = [\sum_{p_x \in \mathcal{P}_i} \mathbf{v}(p_x), \sum_{d_y \in \mathcal{D}_i} \mathbf{v}(d_y)], \tag{3.1}$$

where $[\cdot, \cdot]$ represents the vector concatenation; $\mathcal{P}_i$ is the set of prescription codes, and $\mathcal{D}_i$ is the set of diagnosis codes in the ith visit. $\mathbf{v}(c)$ is the Skip-gram embedding of a medical code c.

The Operational Definition for a Diagnosis of Cardiovascular Disease (CVD)

In this study, we operationally defined a CVD diagnosis as CVD events resulting in hospitalization or death, following the previous works that use the same data source [9, 11, 12]. A CVD event was defined as 2 or more days of hospitalization or death due to the International Classification of Diseases, Tenth Revision (ICD-10) codes pertaining to CVD. Upon admission, the Korean National Health Insurance Service (NHIS) requires physicians to designate ICD-10 codes for which the patient was hospitalized. Causes of death were also determined by ICD-10 codes. The qualifying ICD-10 codes corresponding to CVD were divided into coronary heart disease(CHD) and stroke in accordance with the AHA guidelines [11]. The qualifying IDC-10 codes are shown in Table 3.1. The frequency of each ICD-10 code within total cases is provided in the last column. In Table 3.2, we additionally reported the frequency of few codes which may present similar symptoms as a stroke.

3.3 Method

3.3.1 RNN-based Disease Prediction Model

In our model, the patient records were processed in three steps: (1) First, we encoded the time-dependent visit history into a sequence of hidden representations. (2) Second, to obtain the global representation of the entire set of patient records, we used an SAF

Table 3.1: A list of qualifying ICD-10 codes.

Disease Category	**ICD-10 Code**	**Description**	**Frequency (%)**
Coronary heart disease	I20	Angina pectoris	35.1
	I21	Acute myocardial infarction	7.5
	I22	Subsequent myocardial infarction	0.1
	I23	Certain current complications following acute myocardial	0.1
	I24	Other acute ischemic heart diseases	0.5
	I25	Chronic ischemic heart diseases	6.5
Stroke	I60	Subarachnoid haemorrhage	3.5
	I61	Intracerebral haemorrhage	4.2
	I62	Other nontraumatic intracranial haemorrhage	1.3
	I63	Cerebral infarction	26.7
	I64	Stroke, not specified as haemorrhage or infarction	1.2
	I65	Occlusion and stenosis of precerebral arteries, not resulting in cerebral infarction	1.7
	I66	Occlusion and stenosis of cerebral arteries, not resulting in cerebral infarction	1.5
	I67	Other cerebrovascular diseases	6.6
	I68	Cerebrovascular disorders in diseases classified elsewhere	0.1
	I69	Sequelae of cerebrovascular disease	3.5

Table 3.2: Frequencies of unusual conditions.

ICD-10 Code	**Description**	**Frequency (%)**
167.1	Cerebral aneurysm, nonruptured	2.6
167.3	Progressive vascular leukoencephalopathy	0.0
167.4	Hypertensive encephalopathy	0.3
167.7	Cerebral arteritis, NEC	0.0
168.0	Cerebral amyloid angiopathy	0.0
168.1	Cerebral arteritis in other diseases classified elsewhere	0.0
168.2	Cerebral arteritis in infectious and parasitic diseases classified elsewhere	0.0

module that fuses the hidden representations of the visits and the patient characteristics. (3) Finally, we used the obtained global representation for binary classification. The entire architecture of our model is shown in 3.2. To capture the temporal relations between the clinical events in each of the visits, we used an RNN model to process the visit history given as the sequence of the visit embedding vectors, which is $v = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T)$. The RNN model updates the visit representations with respect to the informative events that occurred in the past. The high-level representation of a hidden state is computed as follows:

$$\mathbf{h}_i = \mathbf{RNN}(\mathbf{v}_i, \mathbf{h}_{(i-1)}). \tag{3.2}$$

We specifically implemented the Bi-directional GRU(Gated Recurrent Units)-RNN model to address the problem of long-term dependencies.

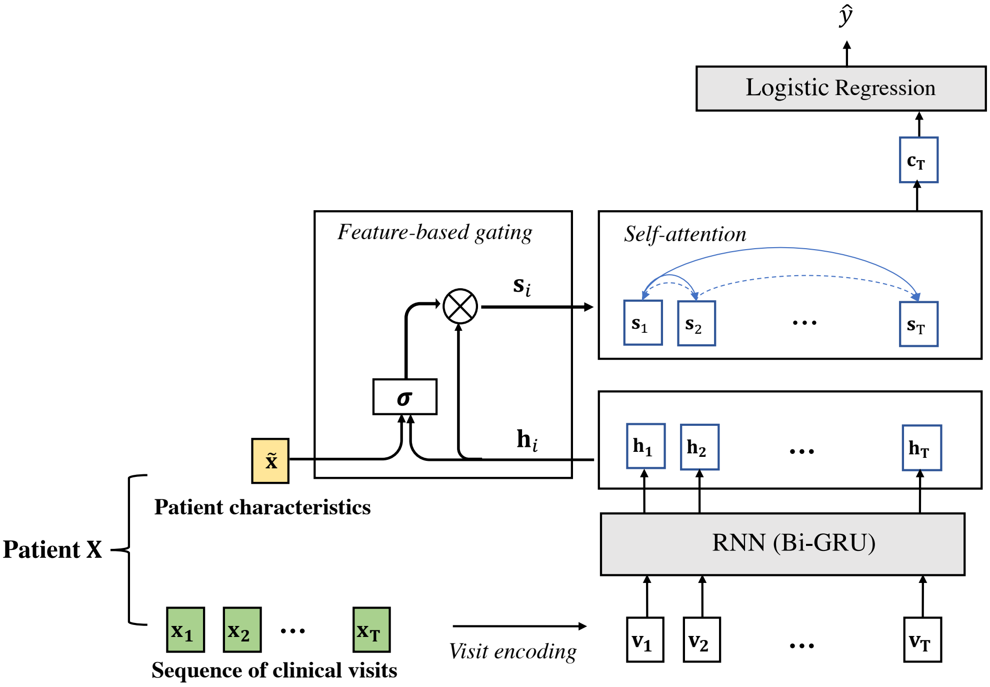


Figure 3.2: The architecture of the SAF-RNN model.

3.3.2 Self-Attentive Fusion (SAF) Encoder

Next, to obtain the global representation of the patient's history, considering the patient characteristics, we applied the SAF encoder. As depicted in Figure 3.1, a previously dominant method to incorporate patient characteristics was a simple concatenation of the RNN features with the vector encoding the patient characteristics. However, this approach does not consider the complex relations between two heterogeneous patient records. On the other hand, our proposed SAF encoder captures the relations between patient characteristics and the RNN hidden states from different time steps by using the self-attention after the feature-based gating. First, the patient characteristics $\tilde{\mathbf{x}}$ is fused with each of the visit representations $\mathbf{h}_i$ during the feature-based gating. Here, the hypernetwork is fed with the concatenation of $\tilde{\mathbf{x}}$ and each $\mathbf{h}_i$, yielding an element-wise gating that is applied to $\mathbf{h}_i$. A gate function f_g with a sigmoid activation function

σ generates a mask vector for $\mathbf{h}_i$, conditioned on $\tilde{\mathbf{x}}$. Formally:

$$\mathbf{s}_i = f_g(\mathbf{h}_i, \tilde{\mathbf{x}}) = \sigma(W_g^{\intercal}[\mathbf{h}_i, \tilde{\mathbf{x}}] + \mathbf{b}_g) \odot \mathbf{h}_i, \tag{3.3}$$

where W_g and $\mathbf{b}_g$ are learnable parameters. After the salient features of $\mathbf{h}_i$ are selected with respect to the patient characteristics, the self-attention mechanism is applied over the updated visit representations $\mathbf{s}_i$. Self-attention, also known as intra-sequence attention, computes the compositional relationships between visits within a sequence. Here, we use a bilinear function f_a to measure the alignment between the query input $\mathbf{s}_i$ and the key input $\mathbf{s}_t$. The alignment $e_{(i,t)}$ is computed with a learnable weight matrix W_a as shown below:

$$e_(i, t) = f_a(\mathbf{s}_i, \mathbf{s}_t) = \mathbf{s}_i^{\intercal} W_a \mathbf{s}_t \tag{3.4}$$

Then we compute the normalized attention score $\alpha_{(i,t)}^{(1)}$ across the inputs and obtain each visit representation $\mathbf{c}_i$ as a weighted sum:

$$\alpha_{(i,t)}^{(1)} = \frac{exp(e_{i,t})}{\sum_{j=1}^{T} exp(e_{i,j})} \tag{3.5}$$

$$\mathbf{c}_i = \sum_{t=1}^{T} \alpha_{(i,t)}^{(1)} \mathbf{s}_t \tag{3.6}$$

Lastly, we apply logistic regression to the final visit representation $\mathbf{c}_T$. It produces the scalar value $\hat{y}$, which estimates the patient-specific risk score for a disease diagnosis in the next visit.

$$\hat{y} = \sigma(W^{\intercal}\mathbf{c}_T + b) \tag{3.7}$$

3.4 Dataset and Experimental Setup

3.4.1 Dataset

NHIS-NSC as the Primary Data Source

We obtained data from the Sample Cohort Database (NHIS-NSC), a nationwide population-based cohort established by the National Health Insurance Service (NHIS) of South

Korea [40]. The NHIS-NSC provides a wide variety of information about the demographic profiles, medical insurance claims, and health examinations of one million patients sampled from 2002 to 2013. It is considered representative of the entire Korean population because 97% of the population is obliged to enroll in national health insurance, which covers all forms of health care services. Moreover, the NHIS-NSC uses systematic stratified random sampling to create a highly representative sample. The groups from which the samples are taken divide the entire population based on the shared characteristics such as age, sex, region, and income level. Notably, medical insurance claims in the NHIS-NSC provide a sequence of clinical records for each patient, consisting of the diagnoses, medication prescriptions, and procedures given during each clinical visit.

Data Processing

To train and test our model on the general patient population, we extracted samples from the NHIS-NSC by adopting a case/control design with incidence density sampling. In the incidence density sampling process, the selection of controls is decided by the diagnosis dates of cases. A diagnosis date is the day of the visit during which a CVD diagnosis was made. We operationally defined a CVD diagnosis as a CVD event resulting in hospitalization or death by following the previous works that used the same data source [41, 42, 43]. The results of our analysis should be interpreted with the awareness of the broad definition of CVD used for case sampling. The definition includes conditions such as ”Cerebral aneurysm, nonruptured” and ”Hypertensive encephalopathy,” which may present similar symptoms as a stroke. However, these diseases are uncommon and represent only 2.9% of cases used in the analysis.

Among the cohort participants, patients who were diagnosed with CVD before 2007 were excluded from the analysis. Cases were sampled between 2007 and 2013. For each case, approximately nine controls were sampled from a pool of participants who had not been diagnosed with CVD prior to the case’s event date. Age, sex and number of visits within two years were matched between the cases and the controls using nearest neighbor matching. The same diagnosis date was assigned to all controls, and all the clinical records of the selected cases and controls during the time window of two years before the diagnosis date were collected. We named this time window as an observation period because the model makes decisions based on the observations during this period. The participants were 40–90 years of age on the diagnosis date. We also avoided selection bias by death when extracting the controls, which could occur if ill people had already died and so were not selected as cases. Thus, we excluded the patients who died within one month of the diagnosis date. [12, 11, 9]

3.4.2 Experimental Design

In this research, we extracted the visit data of 75,604 patients from the NHIS-NSC data, following the strategy described in the subsubsection 'Data Preprocessing.' Consequently, 7,981 cases and 67,623 controls were extracted with diagnosis and prescription codes. The average visit length for each patient was approximately 57, and the total numbers of unique codes were 1,628 and 1,502 for diagnoses and prescriptions, respectively. Then, we designed more tailored experimental settings as follows. [15, 27, 14, 13] An immediate outpatient CVD diagnosis before CVD admission is not a cause for CVD admission; rather, it should be considered as a point of the first contact in the natural course of CVD detection. However, because our operational definition of CVD was CVD with inpatient admission, cases very often had CVD outpatient visits immediately prior to admission. With such highly-correlated cases, the model was incentivized to predict based on CVD outpatient diagnosis rather than looking at other non-obvious factors. [22, 21, 20, 19, 18, 17, 16] Thus, we cleaned our data by masking all medical data, including CVD outpatient diagnosis codes, within the 7 days (and 14 days) prior to CVD admission on the diagnosis date. We defined this data as the **MASKED_7** and **MASKED_14** dataset, in contrast to the original **RAW** dataset. For each dataset, we used 80% of the data for training, 10% for validation, and the remaining 10% for testing. [30, 29, 28, 26, 25, 34, 33, 32, 31, 24, 23]

Baselines

We trained six classification models as the baselines – a regularized logistic regression (LR), a multi-layer perceptron (MLP), a vanilla-GRU model (RNN), and three variants of the GRU models, including Patient2Vec [23]. Instead of the time-varying sequence vectors, the aggregated counts of medical codes were used as inputs for the LR and MLP models. Also, a sum of the embedding vectors of the documented medical codes was concatenated to the input. [10, 43, 42, 41, 40, 39, 38, 37, 36, 35] We denoted the GRU variants that learned the attention weights for each RNN hidden state using location-based attention(LA) as attentional RNNs (ARNN). The GRU variant that used the bilinear self-attention was denoted as RNN-SA. The models that concatenated the patient characteristics before the last prediction are indicated with a suffix '(+concat).' Patient2Vec [23] is an ARNN-based state-of-the-art model. We trained the MLP model with two hidden layers, and all the GRU-based models had two layers with residual connections between layers. We trained Patient2vec using the default implementation in the original work. Patient2Vec used the same training scheme as that of our model, which used the pretrained Skip-gram embedding vectors. Hyperparameters such as the L2 regularization coefficient and drop-out rates were optimized, but the time interval required for constructing subsequences was the same as that in the original work. The hidden dimension size was set to 100 for all the models, and we trained them until early stopping criteria were met.

3.4.3 Implementation Details

We implemented and trained the models using python Tensorflow 1.14.0. We used Adam optimizer trained with the mini-batch of 64 patients.

3.5 Experimental Results

3.5.1 Evaluation of CVD Prediction

We reported the model performances on the test set in terms of the area under the ROC curve (AUROC) and the area under the precision-recall curve (AUPRC) results. The average performances obtained on the RAW and MASKED datasets are shown in Table 3.3 and Table 3.4. The GRU-based models clearly outperformed the other conventional machine learning models. These results represent the ability of RNN models to discover complex relationships within the patient history. The attention-based models generally performed better than the vanilla GRU model. Patient2Vec from [23] also achieved fairly high performances. The performance of SAF-RNN was significantly higher than that of the other attention-based models, showing that it can leverage patient characteristics for prediction purposes. Furthermore, the other models did not benefit from concatenating the patient characteristics.

3.5.2 Sensitivity Analysis

Almost all the models' performances were decreased on the MASKED sets as the models cannot exploit the strong CVD signals immediately prior to the diagnosis date. LR and MLP-based models did not change much since they make predictions upon the aggregated counts of medical codes, which are relatively consistent across two datasets. Therefore, we verify that the models make prediction based on CVD outpatient diagnosis immediately before admission when provided with the highly-correlated cases. However, the SAF-RNN still showed its ability to leverage the patient characteristics, significantly outperforming the other models. Figure 3.3 also shows the performance degradation of the models on the MASKED sets. Here, SAF-RNN clearly displayed its robustness against eliminating the highly-correlated cases, demonstrating its ability to focus on more diverse factors.

Table 3.3: AUROC performances for predicting CVD on dataset RAW and MASKED (7 days and 14 days)

Model		RAW	MASK_7d	MASK_14d
Without Patient Characteristics	LR	0.741±0.001	0.679±0.003	0.668±0.007
	MLP	0.782±0.003	0.733±0.004	0.702±0.006
	RNN	0.823±0.001	0.779±0.004	0.749±0.004
	ARNN	0.826±0.002	0.775±0.003	0.750±0.002
	RNN-SA	0.830±0.000	0.778±0.003	0.778±0.003
With Patient Characteristics	LR(+concat)	0.756±0.001	0.695±0.003	0.692±0.003
	MLP(+concat)	0.781±0.003	0.744±0.003	0.725±0.005
	RNN(+concat)	0.826±0.003	0.770±0.002	0.743±0.003
	ARNN(+concat)	0.827±0.003	0.773±0.003	0.747±0.005
	RNN-SA(+concat)	0.830±0.001	0.774±0.004	0.745±0.004
	Patient2Vec[15]	0.819±0.003	0.771±0.005	0.744±0.008
	SAF-RNN	**0.839±0.000**	**0.784± 0.001**	**0.760± 0.001**

Table 3.4: AUPRC performances for predicting CVD on dataset RAW and MASKED (7 days and 14 days).

Model		RAW	MASK_7d	MASK_14d
Without Patient Characteristics	LR	0.477±0.002	0.379±0.005	0.343±0.005
	MLP	0.490±0.005	0.393±0.005	0.479±0.006
	RNN	0.655±0.004	0.529±0.004	0.509±0.004
	ARNN	0.653±0.003	0.529±0.003	0.490±0.003
	RNN-SA	0.654±0.003	0.530±0.002	0.490±0.002
With Patient Characteristics	LR(+concat)	0.493±0.001	0.395±0.003	0.382±0.004
	MLP(+concat)	0.502±0.005	0.411±0.004	0.382±0.004
	RNN(+concat)	0.647±0.004	0.528±0.003	0.491±0.005
	ARNN(+concat)	0.649±0.005	0.528±0.003	0.492±0.006
	RNN-SA(+concat)	0.650±0.001	0.531±0.004	0.494±0.004
	Patient2Vec[15]	0.643±0.006	0.528±0.006	0.488±0.005
	SAF-RNN	**0.661± 0.001**	**0.540± 0.001**	**0.501± 0.002**

3.5.3 Ablation Studies

Table 3.5: AUROC performances of different fusion methods

Models		RAW	MASK_7d
SAF-RNN (RNN+gating+SA)		**0.839±0.000**	**0.784±0.001**
\gating	RNN+concat+SA	0.828±0.001	0.773±0.001
	RNN-SA(+concat)	0.830±0.001	0.774±0.004
\SA	RNN+gating+LA	0.830±0.002	0.779±0.004
	ARNN(+gating)	0.826±0.001	0.777±0.001

Table 3.6: AUPRC performances of different fusion methods

Models		RAW	MASK_7d
SAF-RNN (RNN+gating+SA)		**0.661±0.001**	**0.540±0.001**
\gating	RNN+concat+SA	0.649±0.001	0.538±0.003
	RNN-SA(+concat)	0.650±0.001	0.531±0.004
\SA	RNN+gating+LA	0.649±0.001	0.540±0.007
	ARNN(+gating)	0.648±0.002	0.539±0.004

We also conducted ablation studies to demonstrate the effect of each part of the SAF module. We eliminated the gating mechanism and self-attention, individually. We first replaced the gating mechanism with a simple concatenation. In RNN+concat+SA, the patient characteristics were concatenated to each of the RNN hidden states, and then, self-attention was applied. In RNN-SA(+concat), self-attention was employed before the information fusion, and then, the patient characteristics were combined using concatenation. Secondly, RNN+gating+LA and ARNN(+gating) used the gating mechanism to incorporate the patient characteristics but did not use self-attention. Although the high performances of these models demonstrate the strong abilities of the self-attention and gating mechanisms, the results imply that SAF-RNN is the most effective method for information fusion.

3.6 Further Investigation

3.6.1 Case Study: Patient-Centered Analysis

We showed the interpretability of our model by assessing the importance of each clinical visit for a selected CVD case. Given all the attention weights, we considered the visits with higher attention weights to be more critical to CVD diagnoses since they had a greater impact on the final prediction results. We illustrate the visit-level attention weights provided by SAF-RNN and ARNN(+concat) in 3.4. Consequently, the compared models showed a difference in the attention weight distributions. Both models produced the highest attention weight for the 3rd visit since the diagnosis code indicating hypertension, one of the strongest CVD risk factors, appeared during the 3rd visit. However, SAF-RNN paid comparably high attention to the 4th visit, whereas the ARNN(+concat) put most of its attention on the 3rd visit. The prescription of Olmesartan (which occurred during the 4th visit) is highly associated with CVD since it is used to treat hypertension. Provided with the same patient characteristics showing high blood pressure, our SAF-RNN model focused on the 4th visit more than the ARNN(+concat) model did. Another distinct feature in the 4th visit was the code indicating hyperglyceridemia, a well-documented CVD risk factor. Considering the extremely high cholesterol and LDL levels of the patient, which is related to hyperglyceridemia, this result shows that SAF-RNN revealed the informative parts of the patients' history by efficiently fusing heterogeneous information.

3.6.2 Data-Driven CVD Risk Factors

To further examine the interpretability of our model, we extracted CVD risk factors using the calculated attention weights. We applied a code-level attention mechanism along with the visit-level attention to measure the extent to which medical codes affected the model's prediction. The code-level attention mechanism was implemented as in previous works [27, 17], although it resulted in a slight performance degradation(-2.28%) compared to the original SAF-RNN model. Using both code-level and visit-level attention weights, we computed the average attention given by the model to each code. Using both code-level attention weight and visit-level attention weight, we defined the average model's attention w_j paid on each code j as follows:

$$w_j = \frac{1}{|\mathcal{P}_j|} \sum_{x \in \mathcal{P}_j} \frac{\sum_i exp(\alpha_i^{(x)} \beta_{ij}^{(x)})}{\sum_{i'} \sum_{j'} exp(\alpha_{i'}^{(x)} \beta_{i'j'}^{(x)})} \cdot \hat{p}^{(x)}, \tag{3.8}$$

where $\mathcal{P}_j$ is a group of patients with code j in the records. $\alpha_i^{(x)}$ denotes the visit-level attention weight assigned to the ith visit and $\beta_{ij}^{(x)}$ denotes the code-level attention weight assigned to the code j in the ith visits of patient x. The normalized product of two attention weights is averaged across all the patients in $\mathcal{P}_j$, weighted by the predicted risk probability $\hat{p}^{(x)}$ for each patient.

We considered the medical codes with the greatest attention values as the CVD risk factors that the model learned. As a result, the top-10 diagnosis and prescription codes are listed in Table 3.7 and Table 3.8, respectively. The diagnosis codes directly indicating CVD were excluded from these tables. The relevance of each code to CVD was judged by a physician, who was given categories of 'relevant,' 'possibly relevant,' and 'irrelevant.' All of the extracted diagnosis codes were considered 'relevant' to CVD except for one code indicating the umbrella term. Additionally, the extracted medication codes were considered 'relevant' or 'possibly relevant' to CVD, confirming the interpretability of SAF-RNN. These observations show a potential application of SAF-RNN in identifying CVD risk factors.

3.7 Conclusion

In this work, we proposed an interpretable disease prediction model that efficiently fuses heterogeneous patient records using self-attentive fusion encoder. We demonstrated the model's ability to learn representations for heterogeneous patient records in various experimental settings, and the constructed model consistently achieved superior performances. An analysis on attention weights also indicated the degree to which medical codes can affect the model prediction, hence providing interpretability.

Table 3.7: Top-10 diagnosis-related risk factors judgement

Top-10 ICD-10 codes	**Model's attention (# of occurrences)**	**Relevancy to CVD**	**Reason**
Chronic kidney disease	0.080 (309)	relevant	risk factor for CVD
Other cardiac arrhythmias	0.049 (133)	relevant	risk factor for CVD
Atrial fibrillation and flutter	0.039 (140)	relevant	risk factor for CVD
Pain in throat and chest	0.027 (782)	relevant	symptom of CVD (myocardial infarction)
Dementia in Alzheimer's disease	0.027 (169)	relevant	has similar risk factors
Heart failure	0.026 (276)	relevant	risk factor for CVD
Medical observation and evaluation for suspected diseases and conditions	0.023 (194)	irrelevant	umbrella term for diagnostic process
Recurrent depressive disorder	0.022 (116)	relevant	risk factor for CVD
Secondary hypertension	0.021 (109)	relevant	risk factor for CVD
Hypertensive heart disease	0.019 (523)	relevant	risk factor for CVD
Fracture of lower leg, including ankle	0.019 (171)	relevant	immobility from this may increase risk of CVD
Paroxysmal tachycardia	0.018 (117)	relevant	risk factor for CVD
Intracranial injury	0.016 (129)	relevant	immobility from this may increase risk of CVD

Essential (primary) hypertension	0.015 (5961)	relevant	risk factor for CVD
Headache	0.014 (941)	relevant	symptom of CVD

Table 3.8: Top-10 medication-related risk factors judgement

Top-10 generic medication codes	**Model's attention (# of occurrences)**	**Relevancy to CVD**	**Reason**
diltiazemHCl	0.120 (197)	relevant	used for treating angina (chest pain)
nitroglycerindiluted	0.078 (272)	relevant	used for treating angina (chest pain)
nicorandil(e)	0.074 (221)	relevant	used for treating angina (chest pain)
isosorbidedinitrate	0.072 (199)	relevant	used for treating angina (chest pain)
clopidogrel	0.056 (299)	relevant	used for treatment of ischemic stroke or myocardial infarctions, may also cause bleeding which may result in hemorrhagic stroke

buflomedilpyridoxalphosphate	0.046 (165)	relevant	a vasoactive drug which was suspended in 2011 for increased cardiac toxicity
candesartancilexetil	0.041 (209)	relevant	antihypertensive drug which may be indicative of hypertension patients with increased risk of CVD
venlafaxinHCl	0.040 (159)	possibly relevant	SNRI antidepressant drug which may increase sympathetic pathways leading to increased heart rate and blood pressure
trimetazidine(2)HCl	0.033 (573)	relevant	used for treating angina (chest pain)
isosorbidemononitrate	0.031 (197)	relevant	used for treating angina (chest pain)
ramipril	0.022 (178)	relevant	antihypertensive drug which may be indicative of hypertension patients with increased risk of CVD

benidipineHCl	0.019 (156)	relevant	antihypertensive drug which may be indicative of hypertension patients with increased risk of CVD
fluvastatin	0.019 (160)	relevant	used to treat dyslipidemia, which may be indicative of dyslipidemic patients who are at higher risk of CVD
cholinealfoscerate	0.019 (232)	possibly relevant	used to treat cognitive impairment which may be a signal for preclinical symptoms of stroke
cilostazol	0.018 (459)	relevant	used for treatment of intermittent claudication which is indicative of vascular disease with higher risk of CVD

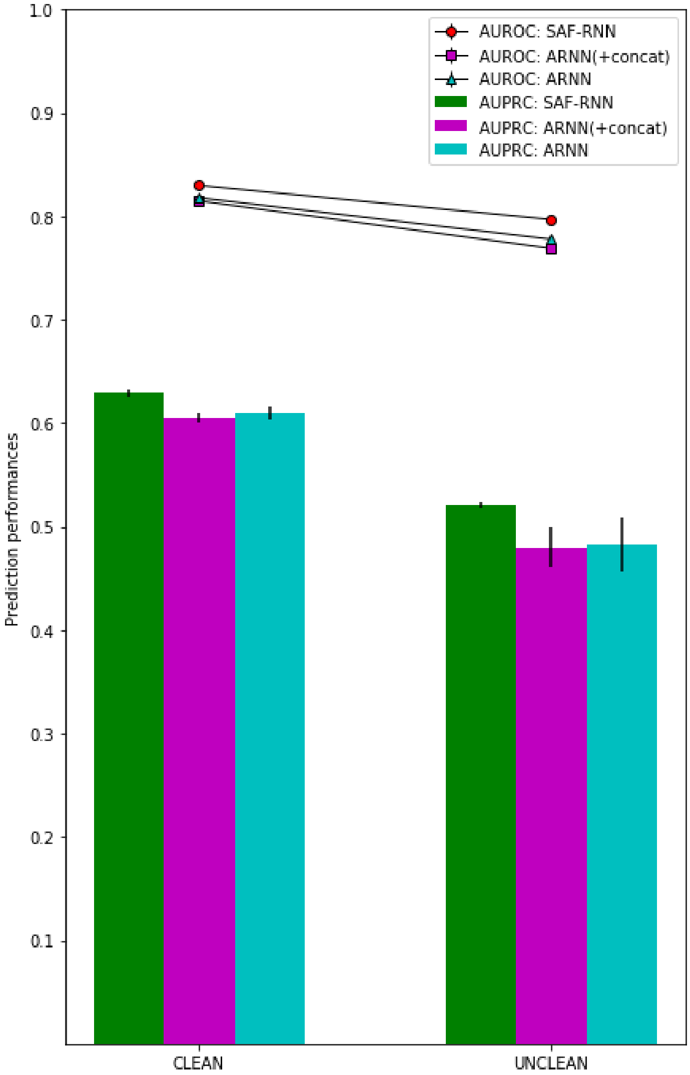
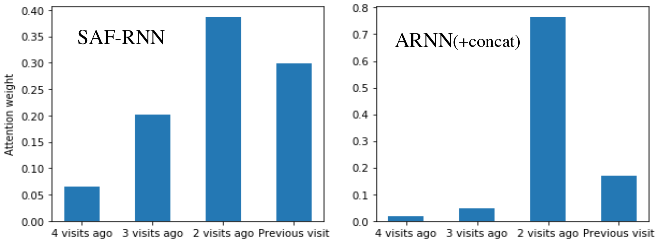


Figure 3.3: CVD prediction performances for different datasets.



Age, Sex	50, male
Systolic BP level	≥ 135 (mmHg)
Diastolic BP level	≥ 85 (mmHg)
Cholesterol level	> 239 (mg/dL)
Triglycerides level	≥ 400 (mg/dL)
HDL level	35-59 (mg/dL)
LDL level	≥ 189 (mg/dL)
Smoke	Yes

Visit order	Diagnosis	Prescription
4 visits ago (1st visit)	Functional dyspepsia Fracture of rib(s), sternum and thoracic spine	mosapride citrate hydrate streptokinase propionic acid derivative
3 visits ago (2nd visit)	The same codes as in the 1st visit	
2 visits ago (3rd visit)	Fracture of rib(s), sternum and thoracic spine Essential hypertension	-
Previous visit (4th visit, the last visit before CVD diagnosis)	Functional dyspepsia Fracture of rib(s), sternum and thoracic spine Disorders of lipoprotein metabolism and other lipidaemias Essential hypertension	mosapride citrate hydrate streptokinase propionic acid derivative orphenadrine olmesartan

Figure 3.4: Case study of a selected case using the visit-level attention weights.

Chapter 4

Graph-Enhanced Medical Concept Embedding

An adverse drug reaction (ADR) is considered to be one of the significant causes of morbidity and mortality, estimated to be the fourth to sixth highest cause of death in the United States [44]. Most ADR detection research has been aimed to predict ADRs in pre-marketing phases, using biomedical information sources such as chemical structures, protein targets, and therapeutic indications. Especially, studies using graph-structured data have demonstrated the superiority of modeling biomedical interactions as graphs. Nevertheless, capturing potential ADRs from the entire population in post-marketing phases is also essential to fully establish the ADR profiles [1]. The potential causal relationship between an adverse event and a drug is called a 'signal' when the relation is previously unknown or incompletely documented. Traditional ADR signal detection research in post-marketing phases mainly counts on a spontaneous and voluntary reporting system that collects spontaneous reports of suspected drug-related events, such as the WHO Uppsala Monitoring Center [45, 46, 47]. However, the spontaneous reporting system has inherent limitations such as underreporting [48, 49], selective reporting [47], and the lack of drug usage data. Therefore, recent studies have attempted algorithmic approaches to detect ADR signals on large clinical databases such as electronic health records (EHR) and healthcare claims data [1, 50]. Many of these studies apply basic machine learning techniques such as random forest,

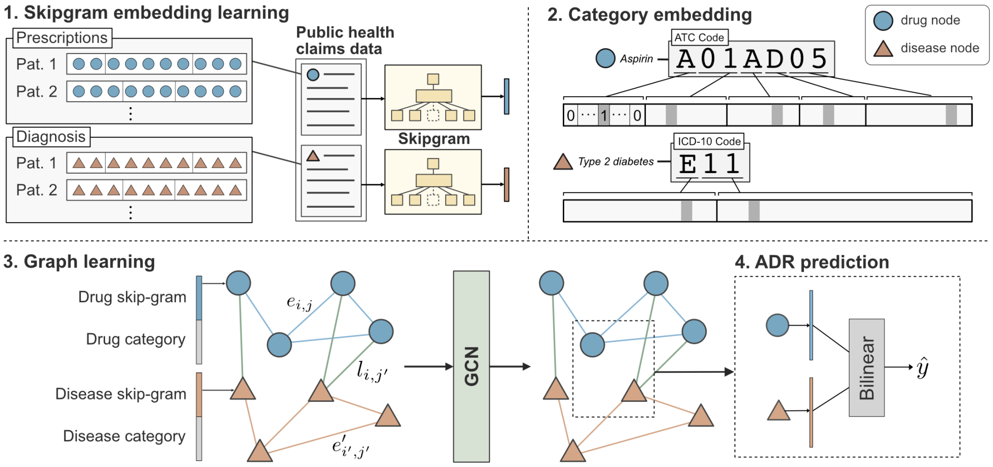


Figure 4.1: Overview of the proposed pipeline for the ADR detection task.

support vector machines, and neural networks. However, fewer studies are using graph-based approaches on the clinical databases in the field of post-marketing ADR signal detection. Due to the complex polypharmacy and multiple relations among drugs and diseases, we expect that graph structure can provide insights to potential ADRs, which may not otherwise be apparent using disconnected structures.

In this study, we develop a novel graph-based framework for ADR signal detection using healthcare claims data to construct a Drug-disease graph. Specifically, we use National Health Insurance Service-National Sample Cohort (NHIS-NSC), the 12-year healthcare claims data that covers medical histories for one million population [2].

The constructed graph is a heterogeneous graph with drug and disease nodes, as it is depicted in Figure 4.1. The nodes represent the medicine prescription codes and disease diagnosis codes derived from the healthcare claims data. To represent the relations among these codes, we define edge weights using information from the data. For example, *l*2-distance between two node embeddings, which are learned from the data, is used to define the drug-drug and disease-disease edge weights. Also, the conditional probability computed on the data is used for the drug-disease relationship. As Graph

Neural Network (GNN) models have been demonstrated [3, 4] their power to solve many tasks with graph-structured data, showing state-of-the-art performances, we use GNN-based approach for ADR detection. We verify that GNNs can learn node representations that are indicative of various relations between drugs and diseases. Then our model makes a prediction on whether a drug node and a disease node will have an ADR relation based on the learned node representations.

To evaluate the performance of the proposed approaches, we conduct experiments with the newly generated dataset using the side effect resource database (SIDER). The empirical results demonstrate the superiority of our proposed model, which outperforms other alternative machine learning algorithms with a significant margin in terms of the area under the receiver operating characteristic (AUROC) score and the area under the precision-recall curve (AUPRC) score. Furthermore, our method unveils ADR candidates that are examined to be very useful information to the medical community. Our model uses only simple data processing and well-established medical terminologies. Therefore, our work does not demand case-by-case feature engineering that requires expertise.

4.1 Related Work

There have been numerous studies on ADR prediction in pre-marketing phases, attempting graph-based approaches on biomedical information sources [51, 52, 53, 54]. These studies predicted potential side-effects of drug candidate molecules based on their chemical structures [52] and additional biological properties [51]. Although such studies may play important roles in preventing ADRs in pre-marketing phases, capturing potential ADRs in real-world use cases has been considered very important.

A spontaneous and voluntary reporting system has been an important data source of the real world drug usages. Most of the traditional ADR signal detection research used voluntary reporting systems with disproportionality analysis (DA), which measures disproportionality of observed drug-adverse event pairs existing in data and the null expectations [45, 46, 47]. Recently, large-scale clinical databases such as EHR (Electronic Health Records) or healthcare claims data have gained popularity as an alternative or additive data source in ADR signal detection research. Much of the studies applied machine learning techniques such as support vector machine (SVM), random forest (RF), logistic regression (LR) and other statistical machine learning methods to model the decision boundary to detect ADR in post-marketing phases [1, 55, 56, 50, 57].

More recently, researchers explored neural network-based models over clinical databases. Shang et al. [58] combined graph structure with the memory network to recommend a personalized medication. The longitudinal electronic health records and drug-drug interaction information were embedded as a separate graph to be jointly considered for the recommendation. There also exists research for the recommendation, but the architectures are limited to the single use of instance symptoms [59, 60], or patient history [61]. However, none of these research explored graph neural network model for predicting the ADR reactions in the post-marketing phase.

4.2 Problem Statement

The task is to predict the potential causal relationship between a given drug and a disease pair, which represents the prescription code and the diagnosis code in clinical data. To consider the various relationships between drugs and diseases, we convert our clinical data into a novel graph structure that consists of drug and disease nodes. The node representations and the edge weights are given according to the information retrieved from the clinical data NHIS-NSC in this study. We first learn a node embedding that reflects the temporal proximity between homogeneous nodes, i.e., drug-drug and disease-disease node pairs. In order to model the proximity between two codes, we form drug/disease sequences from patients' records.

After the drug-disease graph is constructed, we build a GNN model that predicts the signal of side effects between any pairs of drug and disease. The side effect labels, which are taken from the SIDER database, are given to a subset of drug-disease pairs in graph G. We define the label function $l\colon V_{\text{drug}}^{\text{SIDER}} \times V_{\text{dis}}^{\text{SIDER}} \to \{0\,,1\}$ as follows:

$$l(v,w) = \begin{cases} 1 & \text{if } (v,w) \in E^{\text{SIDER}}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.1}$$

where $V_{\text{drug}}^{\text{SIDER}}$ and $V_{\text{dis}}^{\text{SIDER}}$ are the subsets of V_{drug} and V_{dis} registered in the SIDER database respectively, and E^{SIDER} is the set of drug-disease pairs that are known to have side effect relation according to the SIDER database.

4.3 Method

4.3.1 Code Embedding Learning with Skip-gram Model

Most large-scale clinical databases including NHIS-NSC, are collected in the form of longitudinal visit records of the patients. In this section, we explain how we process the patient's longitudinal records as sequential data and apply skip-gram model to learn the code embeddings.

Definition 1 (Drug/Disease Sequence) In the patient's longitudinal records, each patient can be treated as a sequence of hospital visits $\{v_1^{(n)}, v_2^{(n)}, ..., v_{T_n}^{(n)}\}$ where n represents each patient in the data, and T_n is the total number of visits of the patient. The i^{th} visit can be denoted as $v_i^{(n)} = \{\mathcal{P}_i{}^{(n)}, \mathcal{D}_i{}^{(n)}\}$ where $\mathcal{P}_i{}^{(n)}$ is the set of prescribed codes and $\mathcal{D}_i{}^{(n)}$ is the set of diagnosed codes in the i^{th} visit. Within a set of codes, codes are listed in arbitrary order. The size of each set is variable since the number of prescribed/diagnosed codes varies from visit to visit. With these sets of codes, we form a drug sequence $\mathbf{Seq}_{drug}{}^{(n)}$ and a disease sequence $\mathbf{Seq}_{disease}{}^{(n)}$ of n^{th} patient by listing each of the codes in a temporal order, as it is described below (Here, we leave out the symbol n):

$$\begin{aligned} \mathbf{Seq}_{drug} &= \{p_1, p_2\,, \ldots, p_{T_p}\}\,,\ p_x \in \mathcal{P}_i, \\ \mathbf{Seq}_{disease} &= \{d_1, d_2\,, \ldots, d_{T_d}\}\,,\ d_y \in \mathcal{D}_i, \end{aligned} \tag{4.2}$$

where $p_x \in \mathbb{R}^{V_p}$ and $d_y \in \mathbb{R}^{V_d}$ are the one-hot vectors representing each of the medical codes in the sequences. V_p and V_d are the vocabulary size of the whole prescription and diagnosis codes within the data, respectively. T_p and T_d represent the total number of prescription/diagnosis codes of the patient's record. In this way, we can build a corpus consisting of $\mathbf{Seq}_{drug}$ or $\mathbf{Seq}_{disease}$, in which the proximity-based code embedding learning can be implemented.

We use Skip-gram [10] model to learn the latent representation of medical codes in our data, in a way that captures the temporal proximity between them. With $\mathbf{Seq}_{drug}$ or $\mathbf{Seq}_{disease}$, we use the context window size of 16, meaning 16 codes behind and 16 codes ahead, and apply the Skip-gram learning with negative sampling scheme. As a result, we project both diagnosis codes and prescription codes into the separate lower-dimensional spaces, where codes are embedded close to one another that are in close proximity to them. The trained Skip-gram vectors are then used as the proximity-based code embeddings.

4.3.2 Drug-disease Graph Construction

Here, we describe how we construct our unique Drug-disease graph from NHIS-NSC. In Definition 2, we explain the concept of the Drug-disease graph. Then, we explain the node representations and edge connections.

Definition 2 (Drug-disease Graph) We construct a single heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of drug and disease nodes, where $\mathcal{V} = \mathcal{V}_{\text{drug}} \cup \mathcal{V}_{\text{dis}}$ is the union of drug and disease nodes, and $\mathcal{E} = \mathcal{E}_{\text{drug}} \cup \mathcal{E}_{\text{dis}} \cup \mathcal{E}_{\text{inter}}$ is the union of homogeneous edges $\mathcal{E}_{\text{drug}}$ and $\mathcal{E}_{\text{dis}}$ (i.e. consisting of same type of nodes) and heterogeneous edges $\mathcal{E}_{\text{inter}}$ (i.e. consisting of different types of nodes).

To represent $\mathbf{v}_{\text{drug}} \in \mathcal{V}_{\text{drug}}$ and $\mathbf{v}_{\text{dis}} \in \mathcal{V}_{\text{dis}}$, we jointly use proximity-based node representation along with category-based node representation. Proximity-based node representation is obtained by initial Skip-gram code embedding as in section 4.3.1. We denote a proximity-based drug node as $\mathbf{v}'_{\text{drug}}$ and a disease node as $\mathbf{v}'_{\text{dis}}$. Category-based node representation is designed to represent the categorical information of medical codes. We utilize the hierarchical structure of categorical codes (i.e. ATC and ICD-10 codes) by adopting the one-hot vector format. Since there are multiple categories for each code, the category-based node representation is shown as a concatenation of one-hot vectors, thus, a multi-hot vector. Finally, the initial node representation of the Drug-disease graph are represented as the concatenation of the proximity-based node embeddings and the category-based node embeddings. Following are the definitions for the drug and disease node representations.

Definition 3 (Node Representations)

$$\begin{aligned} \mathbf{v}''_{drug} &= \{\mathbf{v}^1_{drug}||\mathbf{v}^2_{drug}||\mathbf{v}^3_{drug}||\mathbf{v}^4_{drug}||\mathbf{v}^5_{drug}\}, \\ \mathbf{v}''_{dis} &= \{\mathbf{v}^1_{dis}||\mathbf{v}^2_{dis}\}, \\ \mathbf{v}_{drug} &= \{\mathbf{v}'_{drug} \,||\, \mathbf{v}''_{drug}\}, \\ \mathbf{v}_{dis} &= \{\mathbf{v}'_{dis} \,||\, \mathbf{v}''_{dis}\}, \end{aligned} \tag{4.3}$$

where $\mathbf{v}''_{drug}$ is a category-based drug node, $\mathbf{v}''_{dis}$ is a category-based disease node, $\mathbf{v}_{drug}$ is an initial drug node, $\mathbf{v}_{dis}$ is an initial disease node, and $||$ is a vector concatenation function. Each $\mathbf{v}^i_{drug}$ represents the each level in the ATC code structure and $\mathbf{v}''_{drug} \in \mathbb{R}^{104}$. Because the ATC code structure is represented in 5 levels, a drug node vector is also represented as the concatenation of 5 one-hot vectors. Similarly, each $\mathbf{v}^i_{dis}$ represents each of the first two levels in the ICD-10 code structure and $\mathbf{v}''_{dis} \in \mathbb{R}^{126}$. We only use two classification levels of the ICD-10 code structure, therefore, the disease node vector is represented as the concatenation of 2 one-hot vectors.

For homogeneous edges like $\mathcal{E}_{\text{drug}}$ and $\mathcal{E}_{\text{dis}}$, we view the relationships between homogeneous nodes as the temporal proximity of two entities, meaning that two nodes are likely to be close together in the records. Therefore, using the proximity-based node embeddings, we compute *l2*-distance between two node embeddings to estimate the temporal proximity. For heterogeneous edges, which are the edges connecting drug nodes and disease nodes, are given as the conditional probability of drug prescription given the diagnosed disease. The definitions of the two types of edges are given as follows:

Definition 4 (Homogeneous Edges) For any node $i, j \in \mathcal{V}_{\text{drug}}$ (or $\mathcal{V}_{\text{dis}}$), the edge weight w_{ij} between two nodes are defined using Gaussian weighting function as follows:

$$w_{ij} = \begin{cases} exp(-\frac{\|\mathbf{v}'_i - \mathbf{v}'_j\|^2}{2\theta^2}) & \text{if } \|\mathbf{v}'_i - \mathbf{v}'_j\| \leq threshold, \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

for some parameters *threshold* and θ. $\mathbf{v}'_i$ and $\mathbf{v}'_j$ are the proximity-based node embeddings of two nodes i and j. Later, we additionally use edge-forming thresholds to control the sparsity of the graph.

Definition 5 (Heterogeneous Edges) For any drug node $i \in \mathcal{V}_{\text{drug}}$ and any disease node $j \in \mathcal{V}_{\text{dis}}$, the edge weight w_{ij} between two nodes are given as:

$$w_{ij} = \frac{n_{ij}}{n_j}, \tag{4.5}$$

where n_{ij} is number of patients' histories in the NHIS-NSC database that is recorded with a diagnosis j and a prescription i in tandem. n_j is the number of patients' histories with the diagnosis j.

4.3.3 A GNN-based Method for Learning Graph Structure

We aggregate neighborhood information of each drug/disease node from the constructed graph using the Graph Neural Network (GNN) framework. In each layer of

GNN, the weighted sum of neighboring node features in the previous layer is computed to serve as the node features (after applying a RELU nonlinearity σ) as follows:

$$\mathbf{z_i}^{(l+1)} = \sigma(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} W \mathbf{z_j}^{(l)}), \tag{4.6}$$

where $\mathcal{N}(i)$ denotes the set of neighbors of i^{th} node, $\mathbf{z_i}^{(l)}$ denotes feature vector of i^{th} node at l^{th} layer, W denotes a learnable weight matrix and $\alpha_{ij}^{(l)}$ denotes the normalized edge weight between i^{th} and j^{th} nodes at the l^{th} layer. In the first layer, the initial drug/disease node representations are each passed through a nonlinear projection function to match their dimensions.

We use two weighting schemes for $\alpha_{ij}^{(l)}$. The first variant follows the definition in [3], and the weight is defined as follows:

$$\alpha_{ij} = \frac{w_{ij}}{\sqrt{d_i d_j}}, \tag{4.7}$$

where d_i and d_j are the degree of nodes i and j respectively, and w_{ij} are the edge weights defined in section 4.3.2. The weights are fixed for all layers. The second weighting scheme instead learns the weighting scheme using attention mechanism [4] as follows:

$$\alpha_{ij}^{(l)} = \frac{\exp(g(\mathbf{z_i}^{(l)}, \mathbf{z_j}^{(l)}))}{\sum_{k \in \mathcal{N}(i)} \exp(g(\mathbf{z_i}^{(l)}, \mathbf{z_k}^{(l)}))}, \tag{4.8}$$

where g is a single fully-connected layer with LeakyReLU nonlinearity that takes a pair of node features as input. In the rest of this paper, we call the network with the first weighting scheme as **GCN** and the network with the second scheme as **GAT**.

We predict the ADR signal of a drug-disease pair using the learned embeddings from the GNN model with a single bilinear layer as follows:

$$\hat{y}_{ij} = \sigma(\mathbf{z_i}^{(L)} W_p \mathbf{z_j}^{(L)} + b), \tag{4.9}$$

where W_p, b are the learnable weights, and $v_i^{(L)}$, $v_j^{(L)}$ are the node features of drug node i and disease node j at the last GNN layer. The whole model is trained by minimizing the cross-entropy loss.

4.4 Dataset and Experimental Setup

4.4.1 Dataset

Healthcare claims dataset

For this study, we obtain data from the Sample Cohort Database (NHIS-NSC), a healthcare claims data established by national health insurance (NHI) of South Korea. The NHIS-NSC is a retrospective cohort data from a population of one million patients sampled from 2002 to 2013, providing longitudinal observations of patient's diagnosis, medication prescription, and procedures. With this data, we extract target drugs and diseases and compute the statistics between any pairs of the drug-disease combinations. These statistics are used in determining edges in the drug-disease graph.

We represent drugs and diseases in NHIS-NSC data in a form of ATC codes (medication codes) and ICD-10 codes (International Classification of Diseases, 10th revision). The number of converted ATC and ICD-10 codes are 1,201 and 1,872, respectively.

Adverse drug reaction dataset

As a labeled dataset, we use Side Effect Resource (SIDER) database which contains 139,756 drug-side effect pairs over 1,430 drugs and 5,868 side effects. These were extracted from public information on recorded adverse drug reactions using natural language processing techniques. To leverage medical and pharmacological knowledge, we extract drug and side effect information in a form of categorical codes (i.e. ATC and ICD-10 codes) with hierarchical structures. Since drug and side effect information in SIDER are represented as STITCH (Search Tool for Interactions of CHemicals) compound identifiers and UMLS (Unified Medical Language System) concept identifiers, we convert them to ATC and ICD-10 codes. The number of converted ATC and ICD-10 codes are 562 and 931, respectively.

Table 4.1: Summary statistics of the constructed graph and datasets

Edge-forming threshold	Low	High
# Drug nodes	1,201	
# Labeled drug-dis pairs in train	37,016	
# Labeled drug-dis pairs in test	6,092	
# Disease nodes	10,117	
# Drug2drug-Edges	19,918	7,199
# Drug2dis-Edges	1,306	
# Dis2dis-Edge	401,801	

Data preprocessing

As we get the labels from the SIDER database and the edge weight from the NHIS-NSC database, we retrieve the drug and disease nodes over the joint set of two databases. The resulting dataset is composed of 607 drugs and 556 diseases, and the number of positive samples, indicating the drug-side effect relationships, are 28,746 pairs. A negative sample is defined as a combination of drugs and diseases over the dataset, where the known 28,746 positive samples are excluded. We randomly select negative samples, setting the size of negative samples same as the size of positive samples.

4.4.2 Experimental Design

Since we extract those combinations from the SIDER database, it is plausible to believe that they have not been reported as ADRs. Although the labels are only given to the drug-disease pairs over the joint set of two databases, we make use of all the drugs and diseases in NHIS-NSC as graph nodes to utilize the relations among the drugs and diseases.

To predict the link between the drugs and diseases, we split drug-disease pairs from the ADR dataset into training, validation, and test sets, ensuring that the classes of diseases included in each set do not overlap. The reason we split the data without overlapping disease classes is to increase the usability of the ADR signal detection model. It is also because only a few classes of diseases exist in our dataset, and therefore there could be a data leakage if the same disease class exists in both training and validation. The class of disease means the classification up to the third digit of ICD-10 codes. Note that we make the inference very difficult by not letting the model know which classes of diseases are linked with drugs as ADRs. We use 80% of data for training, 10% for validation, and the remaining 10% for testing.

To control the sparsity of a graph, we build two types of graphs where the edge-forming threshold is either low or high. When the edge-forming threshold is low, the graph has more edges, having more information as a result. We examine whether it is beneficial or detrimental to have more edge information. We distinguish two graphs by setting the thresholds for $\mathcal{E}_{\text{drug}}$ differently. The summary statistics of the constructed graphs and datasets are provided in Table 4.1.

Baselines

To verify the performance of the GNN approach, we compare GNN models with DeepWalk approach and non-graph-based ML techniques. DeepWalk [62] is an unsupervised graph embedding method that uses random walks on graphs. In DeepWalk approach, we use the DeepWalk embeddings, pre-trained by the DeepWalk model. We apply vanilla GCN and its variants to examine the effect of considering the edge types. The followings are the models used for the graph embedding learnings. All the neural-network-based models use two layers with a hidden dimension of 300.

- **LR** is a logistic regression (LR) approach with information of the graph topology. The vector composed of initial node representations of the node itself and its neighbor nodes are input to the LR model. The number of neighbors is limited to 10.

- **NN** is a 2-layer fully-connected neural network which is solely based on the initial node representations.

- **DW** directly feeds the input, which is the concatenation of the DeepWalk embedding and the initial node representation, to the prediction layer.

- **DW + NN** is a 2-layer fully-connected neural network that uses the concatenation of the DeepWalk embedding and the initial node representation as its input.

- **DW + GCN** is a 2-layer GCN model that uses the concatenation of DeepWalk embedding and the initial node representation as its input.

- **GCN$_{low}$** is a graph convolution network, a representative GNN model that uses graph convolutions [3].

- **GAT$_{low}$** is a GNN that applies the attention mechanism on the node embeddings. Here we use GAT with two layers, where the number of heads are (4,4) for each layer.

- **adrGCN$_{low}$** is an adapted version of GCN, that uses separate GCN layers according to the edge types and then aggregate them.

- **GCN$_{high}$**,**GAT$_{high}$**, **adrGCN$_{high}$** are the GCN models applied to the sparser graph, i.e. the edge-forming threshold is high.

4.4.3 Implementation Details

We implement all the baselines and our proposed models with PyTorch 0.4.14. For training models, we use Adam optimizer with a mini-batch of 32 (drug, disease) tuples. We train on 1 GPU (1080Ti) for 20 epochs, with an early stopping and a learning rate of 0.0001. We use the area under the receiver operating characteristic curve (AUROC) in the test sets as a measure for comparing the performance of all the methods. We use Adam Optimizer with the learning rate of 0.0001.

Table 4.2: Model AUROC and AUPRC performances (including 95% CI)

Model	**AUROC**	**AUPRC**
LR	0.631 ± 0.006	0.585 ± 0.007
NN	0.739 ± 0.005	0.701 ± 0.006
DW	0.728 ± 0.004	0.709 ± 0.004
DW + NN	0.772 ± 0.005	0.758 ± 0.005
DW + GCN_{low}	0.794 ± 0.003	0.768 ± 0.003
GCN_{low}	**0.795** ± 0.006	**0.775** ± 0.006
GAT_{low}	0.732 ± 0.005	0.686 ± 0.009
adrGCN_{low}	0.755 ± 0.008	0.726 ± 0.009
GCN_{high}	0.784 ± 0.006	0.761 ± 0.008
GAT_{high}	0.733 ± 0.008	0.692 ± 0.009
adrGCN_{high}	0.756 ± 0.004	0.732 ± 0.006

4.5 Experimental Results

4.5.1 Evaluation of ADR Detection

As shown in Table 4.2, the proposed graph-based approaches surpass all the non-graph-based approaches. The best AUROC performance is achieved when **GCN** is applied with the low edge-forming threshold. The results show that the **GCN** model efficiently leverages the information from sufficiently selected edges. To compare the performances of graph-based methods, we plot the learning curves as in Figure 4.2. The performances of DeepWalk-based approaches increases as the more advanced neural architecture is applied. This result shows that the graph structure learned by **DW** does not provide sufficient information to predict ADR signals. Also, slow convergence of **DW** + **GCN**, compared to **DW** and **DW** + **NN**, implies that GNN uses more information in this graph.

To see the robustness of the proposed method, we also examine whether our model

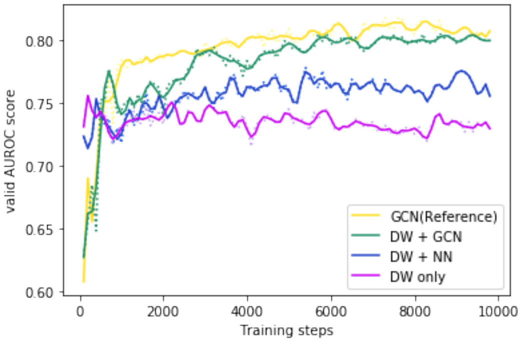


Figure 4.2: Learning curves of the models using DeepWalk embeddings.

works well for the infrequent drug-ADR pairs. We evaluate model performance for the infrequent drug-ADR pairs, which are labeled as 'rare' or 'post-marketing' in SIDER. As a result, the best average test accuracy in infrequent drug-ADR pairs is achieved with **adrGCN*high*** (0.746), demonstrating that using multiple GCNs according to the edge types is useful to detect rare symptoms. According to the SIDER database, the ADRs with 'rare' or 'post-marketing' labels are reported with frequencies under 0.01.

4.5.2 Newly-Described ADR Candidates

To verify the power of the graph-based approach to discover ADR candidates which are unseen in the dataset, we extract the drug-disease pairs which are predicted to be positive with high probability — over 0.97 but labeled as negative (false positive). To demonstrate the genuine power of graph-based methods, we exclude the candidates that are also positively predicted by the baseline neural network, which does not use relational information. As a result, clinical experts (M.D.) confirm that there exist pairs that are clearly considered to be real ADRs. The pairs are listed in Table 4.3.

Many of the discovered pairs, including umbrella terms like edema, are rather symptoms and signs than diseases. This can be explained by the fact that the SIDER database is less comprehensive to cover all the specific symptoms, that can be induced by taking medicine. Especially, cardiac murmur and abnormal reflex are fre-

Table 4.3: Newly-described drug-ADR pairs predicted by the proposed method

Drug name	**ADR symptom**	**Probability**
Dasatinib	Cardiac murmur	0.985
Hydroxycarbamide	Abnormal reflex	0.981
Alendronic acid	Tetany	0.978
Ibandronic acid	Unspecified edema	0.976
Etidronic acid	Abnormal reflex	0.972

quent symptoms, but it is reasonable to say that the suggested pairs are ADRs. For example, Dasatinib is used to treat leukemia and can have significant cardiotoxicity, which can lead to cardiac murmurs. Hydroxycarbamide is a cytotoxic drug used for certain types of cancer, and it is known that cytotoxic medications can cause electrolyte imbalance leading to abnormal reflex.

There are also significant pairs such as alendronic acid and tetany in the third row. Severe and transient hypocalcemia is a well-known side-effect of bisphosphonates, which can lead to symptoms of tetany. Alendronic acid is classified as bisphosphonates, and therefore, tetany can be described as ADR of alendronic acid. Ibandronic acid and etidronic acid in the last two rows are also bisphosphonates, and the paired symptoms are relevant to the usage of bisphosphonates. Unspecified edema may signify bone marrow edema caused by bisphosphonate use, and electrolyte imbalance, which can lead to abnormal reflex, can be caused by etidronic acid use. All these explanations show that the ADR pairs we extract are based on various relations among drugs and diseases.

4.6 Conclusion

In this study, we propose a novel graph-based approach for ADR detection by constructing a graph from the large-scale healthcare claims data. Our model can capture various relations among drugs and diseases, showing improved performance in predicting drug-ADR pairs. Furthermore, our model even predicts drug-ADR pairs that do not exist in the established ADR database, showing that it is capable of supplementing the ADR database. The explanation by clinical experts verifies that the graph-based method is valid for ADR detection. In this study, we only make inferences within the labeled dataset, yet we plan to make inferences on unlabeled data to discover unknown ADR pairs, which will be a huge breakthrough in ADR detection.

Chapter 5

Knowledge-Augmented Deep Patient Representation

In this work, we aim to predict clinical outcomes using National Health Insurance Service-National Sample Cohort (NHIS-NSC), augmented with expertise knowledge. Although many deep learning techniques have demonstrated state-of-the-art performances in modeling patient representation for clinical outcome prediction, only a limited part of the data is used for training. This is because there are many overlapping parts in EHR or the claims data, as doctors tend to label existing symptoms on the visits of the same patient. For example, when we sample 87,384 Cardiovascular Disease (CVD) cases and controls, about 87% of the total diagnosis codes of each patient are repeated on average.

To address this issue, several works have utilized medical knowledge for EHR representation learning. There are many well-constructed ontologies in the medicine area, such as International Classification of Diseases (ICD), Clinical Classifications Software (CCS) [7] , Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) [8], and Human Phenotype Ontology (HPO) [63], and knowledge graphs, such as Semantic Medline Knowledge Graph (SemMed KG) [64] and KnowLife [65]. The proposed approaches injecting such medical knowledge for EHR representation learning showed improved performances on various clinical outcome prediction tasks [66, 67, 68, 69, 70, 71]. The types of medical knowledge and studies using them for clin-

Table 5.1: Types of medical knowledge and studies using them for clinical prediction

Knowledge Types		Model
Ontologies	ICD, CCS	GRAM [66]
		KAME [67]
	SNOMED-CT	Snomed2Vec [71]
	HPO	KGDAL [69]
General Knowledge Graph	KnowLife	DG-RNN [68]
	SemMed	MedPath [70]
		KA-SAF (Ours)

ical prediction are summarized in Table 5.1. They also demonstrated enhanced interpretability by showing explicit reasoning path or using knowledge graph attention mechanism.

As an extension of these studies, we propose a method to leverage prior medical knowledge for clinical outcome prediction. We construct a personalized Knowledge Graph(KG) for each patient to incorporate patient-specific knowledge from the entire KG. Here, the personalized KG is created by extracting a subgraph consisting only of medical codes, i.e., diagnosis and prescription codes, of patient records. Based on the personalized KG, we build the KG representation using Graph Neural Network (GNN). The KG representation is then used with the deep patient representation modeled on the patient's clinical records to predict clinical outcomes. To encode the deep patient representation, we use Self-Attentive Fusion Encoder (SAF)-based RNN model (SAF-RNN), which achieved the state-of-the-art performance in our previous work, described in Chapter 3.

Additionally, we seek to harness pre-training for GNNs to further enhance the personalized KG representation. We perform KG completion (KGC) task as a pre-training task, which is to predict the plausibility of a given triplet. Here, we compute the plausibility score by applying GNN model on the enclosing subgraphs. We expect

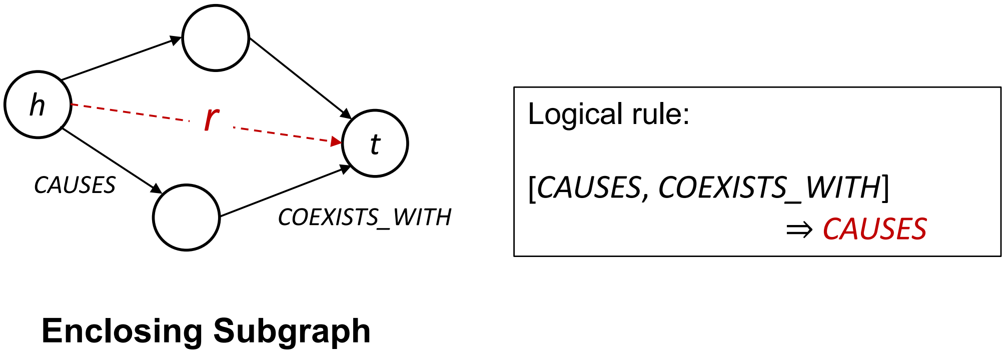


Figure 5.1: Logical rules contained in the enlosing subgraph.

the GNN model to encode the logical rules inherent in the subgraph. The SemMed KG expresses various relations between medical entities, and these relations are strongly associated with other adjacent relations according to specific logical rules as depicted in 5.1.

To improve the performance of KGC task, we utilize the method, proposed in our other research. In this work, we suggest a novel inductive link prediction model, called **Subgraph Infomax (SGI)**, where the relation embedding is trained to contain more meaningful information about subgraphs via the mutual information (MI) maximization objective. SGI consists of a GNN-based scoring network for computing the score of a given triplet and a module for MI maximization. We trained SGI to maximize the MI between the relation embedding and the subgraph representation. After pre-training the GNN encoder with the KGC training objective on SemMed KGs, the GNN encoder is further fine-tuned via the supervision coming from clinical outcome prediction.

We evaluate the performances of our model on two tasks, i.e., the next diagnosis prediction and the CVD prediction. As a result, it shows improved prediction performances than in the case of using SAF-RNN alone. The performances are further improved when we pre-train the GNN encoder with KGC training objective using SGI, showing the enhanced KG representation.

5.1 Related Work

5.1.1 Incorporating Prior Medical Knowledge for Clinical Outcome Prediction

There are several works that suggest incorporating prior medical knowledge to facilitate EHR and patinet representation learning. GRAM [66] , for example, learns the embedding of a medical code by attending over each hierarchical information extracted from medical ontologies for sequential diagnoses prediction and heart failure prediction tasks. KAME [67] also proposes an attention mechanism to further exploit high-level knowledge to improve the diagnosis prediction task. Like these studies that mainly use the hierarchical information from medical ontologies, some studies that use knowledge graphs that explain the direct relationship between entities have been suggested. DG-RNN and KGDAL leverage the medical knowledge graphs called KnowLife and HPO, respectively [68, 69]. DG-RNN uses the knowledge graph attention mechanism to learn the medical code embedding for heart failure prediction. KGDAL also suggests knowledge-based attention mechanism for mortality prediction of critically ill patients with acute kidney injury requiring dialysis. However, these two works do not leverage the personalized KG.

Similar to our work, MedPath [70] extracts a personalized medical knowledge graph for each patient and enhances the patient representation learning with prior medical knowledge. However, there are two main differences between two works. First, MedPath only uses the medical concept corresponding to the diagnosis code to interpret the disease progression path. On the other hand, our work uses both diagnosis and prescription codes to fully utilize the potential of SemMed KG. Secondly, MedPath uses different initial concept vectors for each of the EHR encoder and the graph encoder, using TransE [72] embedding as the initial node embedding for the graph encoder. All the previous works leveraging KGs apply knowledge graph embedding methods such as TransE on the entire KGs to obtain medical concept embeddingsTo the best of our knowledge, our work first uses the same medical concept embedding as the initial node embedding of KG in this literature.

5.1.2 Inductive KGC based on Subgraph Learning

Knowledge Graphs (KGs) have been very useful for many information retrieval (IR)-related tasks such as query answering, entity linking, and knowledge-augmented question answering. However, due to the incompleteness problem of the KGs, there has been an increasing interest in Knowledge Graph Completion (KGC) task. KGC is a link prediction task to predict missing edges in KG, and can be seen as a question of whether a given triplet is valid or not. Especially, the evolving nature of KGs has led to active research on inductive link prediction, where one needs to make inferences on triplets with entities that are not seen during training.

Recently, many inductive KGC studies have tried inducing the logical rules contained in a local subgraph to learn entity-independent semantics [73, 5, 6, 74]. Since Zhang & Chen [75] have theoretically proven that the enclosing subgraphs surrounding the target triplets are informative for link prediction, several subsequent studies have suggested graph representation learning to encode the logical rules within the local subgraphs [5, 6, 74].

5.2 Method

5.2.1 Extracting Personalized KG

Selecting Medical Knowledge Graph

To augment the patient's records with medical knowledge, it is essential to select the appropriate KG. We select the KG based on two criteria., (1) compatibility between medical concept identifiers and (2) ability to express rich relationships between medical concepts.

We first need to search for KGs in which the concept identifiers that compose the KG can be mapped with the medical codes in NHIS-NSC data. The diagnosis and prescription codes in NHIS-NSC are represented in ICD-10 codes (International Classification of Diseases, 10th revision) and ATC codes (Anatomical Therapeutic Chemical Classification System). The SNOMED-CT codes in SNOMED-CT and the UMLS Concept Unique Identifiers (CUIs) in SemMed KG have similar properties to ICD-10 and ATC codes in that they can express clinical information in EHRs, and, therefore, can be mapped to each other.

Secondly, in order to incorporate knowledge related to the clinical outcome prediction, a KG representing rich relationships between medical concepts is needed. Specifically, we explore a KG with the relation types that represent causal or temporal relations. We select SemMed KG as it satisfies all the above conditions.

Mapping concept identifiers in SemMed KG with the medical codes

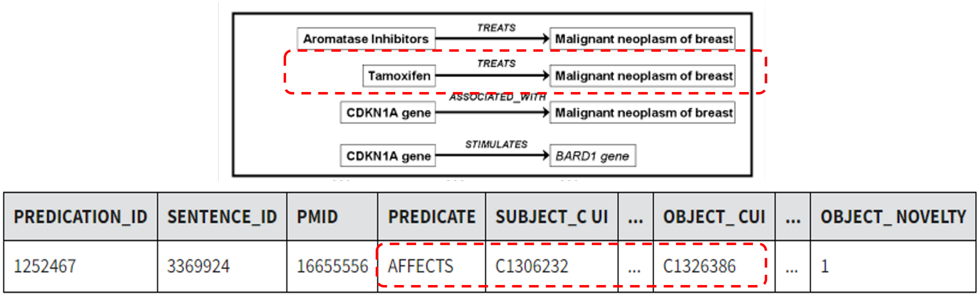
To extract personalized KGs from the original SemMed KG using the medical codes in each patient's records, we map concept identifiers in SemMed KG with the medical codes in NHIS-NSC. For diagnosis code, we map the ICD-10 codes with UMLS CUIs using the mapping table provided by the UMLS metathesaurus. We also map the ATC codes in NHIS-NSC by converting them into SNOMED-CT concepts and then convering SNOMED-CT concepts into UMLS CUIs. Here, we use only the overlapping parts between two sources.

Subgraph Extraction

SemMed KG contains information about approximately 96.3 million predications (relations) among medical entities from all PubMed citations (about 29.1 million citations) [63, 64]. We use only 19 relation types that connect entities contained in the domains of 'Pharmacologic Substance' and 'Disease or Syndrome'. The predicates (relation types) we use are given in a Table 5.2. Also, the example of the extracted

Table 5.2: Relation types in extracted personalized KGs

Predicate Types
'COMPLICATES', 'COEXISTS_WITH', 'MANIFESTATION_OF', 'ISA', 'LOCATION_OF', 'TREATS', 'INHIBITS', 'PRECEDES', 'AFFECTS', 'PART_OF', 'AUGMENTS', 'PREVENTS', 'ASSOCIATED_WITH', 'PRODUCES', 'CAUSES', 'DISRUPTS', 'OCCURS_IN', 'PREDISPOSES', 'DIAGNOSES'



PREDICATION_ID	SENTENCE_ID	PMID	PREDICATE	SUBJECT_C UI	...	OBJECT_ CUI	...	OBJECT_ NOVELTY
1252467	3369924	16655556	AFFECTS	C1306232	...	C1326386	...	1

Figure 5.2: Exmaples of an extracted knowledge triplet from SemMed KG.

knowledge triplet is given in Figure 5.2. We extract each enclosing subgraph by following process.

First, we select a set of nodes N that indicates the diagnosis and prescription codes of each patient. Then, we extract the direct triplets between the two nodes $u, v \in N$ in this set. We denote this set of edges as $E(N)$. To also consider the 2-hop path between two nodes u, v, we extract all the 1-hop neighbors of each $u, v \in N$. We denote them as $\mathcal{N}(u)$ and $\mathcal{N}(v)$. After retrieving triplets from $E(\{\mathcal{N}(u) \cup \mathcal{N}(v), \forall u, v \in N\})$, we exclude the triplets in $E(\{\mathcal{N}(u) \cap \mathcal{N}(v), \forall u, v \in N\})$. Finally, we use the resulting subgraph together with $E(N)$.

5.2.2 KA-SAF: Knowledge-Augmented Self-Attentive Fusion Encoder

Based on the personalized KG, we build the KG representation using GNN-encoder. The KG representation is then augmented to the deep patient representation built upon the patient's clinical records. To encode the deep patient representation, we use SAF-RNN, which achieved the state-of-the-art performance in our previous work, described in Chapter 3.

RNN-based Disease Prediction Model

In our model, the patient records were processed in three steps: (1) First, we encoded the time-dependent visit history into a sequence of hidden representations. (2) Second, to obtain the global representation of the entire set of patient records, we used an SAF module that fuses the hidden representations of the visits and the patient characteristics. (3) Finally, we used the obtained global representation for binary classification. To capture the temporal relations between the clinical events in each of the visits, we used an RNN model to process the visit history given as the sequence of the visit embedding vectors, which is $v = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T)$. The RNN model updates the visit representations with respect to the informative events that occurred in the past. The high-level representation of a hidden state is computed as follows:

$$\mathbf{h}_i = \mathbf{RNN}(\mathbf{v}_i, \mathbf{h}_{(i-1)}). \tag{5.1}$$

We specifically implemented the Bi-directional GRU(Gated Recurrent Units)-RNN model to address the problem of long-term dependencies.

Self-Attentive Fusion (SAF) Encoder

Next, to obtain the global representation of the patient's history, considering the patient characteristics, we applied the SAF encoder. As depicted in Figure 2, a previously dominant method to incorporate patient characteristics was a simple concatenation of the RNN features with the vector encoding the patient characteristics. However, this approach does not consider the complex relations between two heterogeneous patient records. On the other hand, our proposed SAF encoder captures the relations between patient characteristics and the RNN hidden states from different time steps by using the self-attention after the feature-based gating. First, the patient characteristics $\tilde{\mathbf{x}}$ is fused with each of the visit representations $\mathbf{h}_i$ during the feature-based gating. Here, the hypernetwork is fed with the concatenation of $\tilde{\mathbf{x}}$ and each $\mathbf{h}_i$, yielding an element-wise gating that is applied to $\mathbf{h}_i$. A gate function f_g with a sigmoid activation function σ generates a mask vector for $\mathbf{h}_i$, conditioned on $\tilde{\mathbf{x}}$. Formally:

$$\mathbf{s}_i = f_g(\mathbf{h}_i, \tilde{\mathbf{x}}) = \sigma(W_g^{\intercal}[\mathbf{h}_i, \tilde{\mathbf{x}}] + \mathbf{b}_g) \odot \mathbf{h}_i, \tag{5.2}$$

where W_g and $\mathbf{b}_g$ are learnable parameters. After the salient features of $\mathbf{h}_i$ are selected with respect to the patient characteristics, the self-attention mechanism is applied over the updated visit representations $\mathbf{s}_i$. Self-attention, also known as intra-sequence attention, computes the compositional relationships between visits within a sequence. Here, we use a bilinear function f_a to measure the alignment between the query input $\mathbf{s}_i$ and the key input $\mathbf{s}_t$. The alignment $e_{(i,t)}$ is computed with a learnable weight matrix W_a as shown below:

$$e_(i, t) = f_a(\mathbf{s}_i, \mathbf{s}_t) = \mathbf{s}_i^{\intercal} W_a \mathbf{s}_t \tag{5.3}$$

Then we compute the normalized attention score $\alpha_{(i,t)}^{(1)}$ across the inputs and obtain each visit representation $\mathbf{c}_i$ as a weighted sum:

$$\alpha_{(i,t)}^{(1)} = \frac{exp(e_{i,t})}{\sum_{j=1}^{T} exp(e_{i,j})} \tag{5.4}$$

$$\mathbf{c}_i = \sum_{t=1}^{T} \alpha_{(i,t)}^{(1)} \mathbf{s}_t \tag{5.5}$$

Lastly, we apply logistic regression to the final visit representation $\mathbf{c}_T$. It produces the scalar value $\hat{y}$, which estimates the patient-specific risk score for a disease diagnosis in the next visit.

$$\hat{y} = \sigma(W^{\intercal} \mathbf{c}_T + b) \tag{5.6}$$

GNN-based Knowledge Graph Encoder

We use multi-relational R-GCN [76], a GNN-based method designed for modeling multi-relational data, to obtain a subgraph-level representation. The embedding of a node i in the kth layer is given by:

$$\mathbf{h}_i^k = \text{ReLU}(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}_r^k \mathbf{h}_j^{k-1} + \mathbf{W}_{self}^k \mathbf{h}_i^{k-1}), \tag{5.7}$$

where the first term is the aggregated message from the neighbors $\mathcal{N}_i$ of node i. The initial node representation $\mathbf{h}_i^0$ is a skip-gram based medical code embeddings. Note that we use the same code representations as the input of the SAF-RNN. $\alpha_{i,j}$ denotes an edge attention weight of the edge (i, j) and is given as a function of the source node i, neighbor node j, relation type r_{ij} of the edge (i, j).

After L layers of message passing, we obtain a subgraph-level representation $\mathbf{h}_{\mathcal{G}_{(h,r,t)}}$ by concatenating the node representation of the source, target, and the average-pooled representation of all the node:

$$\mathbf{h}_{\mathcal{G}} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{h}_i^L, \tag{5.8}$$

where $\mathcal{V}$ denotes the set of vertices in $\mathcal{G}$. To make the model's performances robust to the number of GNN layers, we adopt JK-connections [77].

Combining with KG representation

We combine the KG representation and the deep patient representation using two approaches: `Concat` and `Attend`. `Concat` simply concatenates two representation vectors and input the concatenation to the prediction layer. Meanwhile, `Attend` considers the deep patient representation in the process of neighborhood aggregation of GNN by computing the attention weight conditioned on the deep patient representation.

Two different approach to combine the KG representation with the deep patient

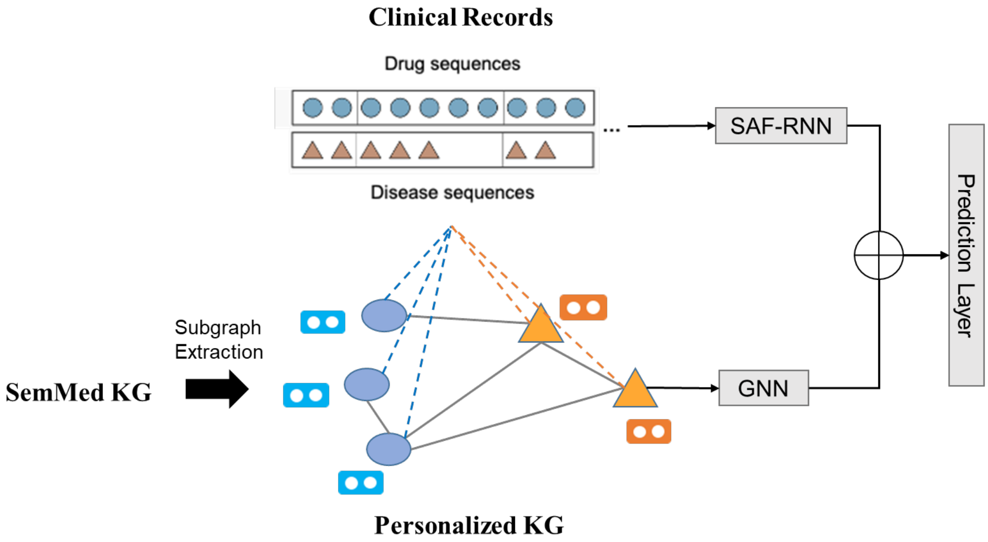


Figure 5.3: Concatentation-based KA-SAF.

representation are depicted in **??** and 5.4. If we use `Concat` approach, then the final representation to be fed into the regression/classification layer is given as:

$$\mathbf{c} = \mathbf{c}_T \oplus \mathbf{h}_\mathcal{G} \tag{5.9}$$

However. if we use `Attend` approach for combining GNN and SAF-RNN representations, we use the SAF-RNN's context vector $\mathbf{c}_T$ for computing the attention weight α_{ij} .

$$\begin{aligned} \mathbf{s} &= \text{ReLU}(\mathbf{A_1}^k[\mathbf{h}_i^{k-1} \oplus \mathbf{h}_j^{k-1} \oplus \mathbf{c}_T \oplus \mathbf{e}_{r_{ij}}] + \mathbf{b_1}^k), \\ \alpha_{ij} &= \sigma(\mathbf{A_2}^k\mathbf{s} + \mathbf{b_2}^k), \end{aligned} \tag{5.10}$$

Finally, the average-pooled node representation $\mathbf{h}_\mathcal{G}$ is given as the input of the regression/classification layer.

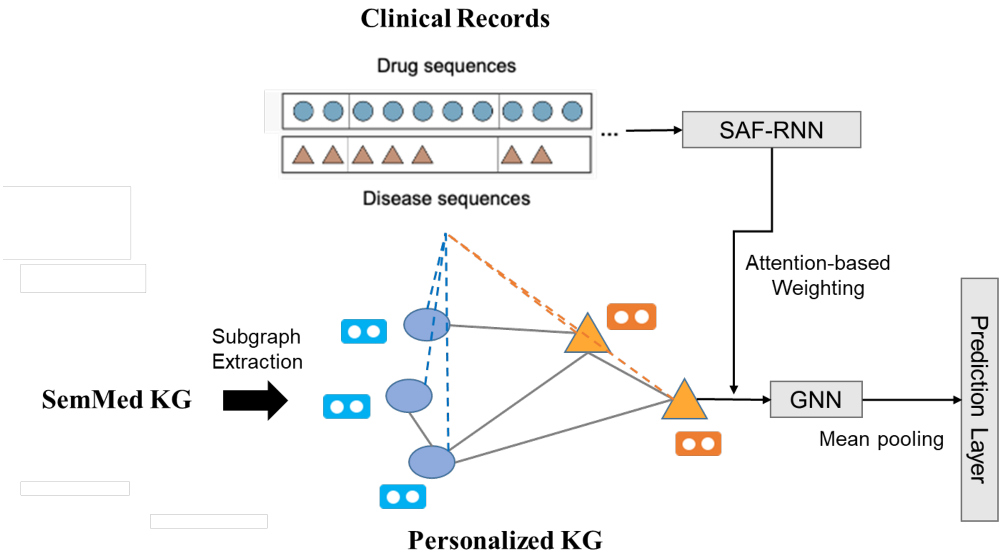


Figure 5.4: Attention-based KA-SAF.

5.2.3 KGC as a Pre-training Task

Additionally, we seek to harness pre-training for GNNs to further enhance the personalized KG representation. We perform KG completion (KGC) task as a pre-training task, which is to predict the plausibility of a given triplet solely on the logical rules inherent in the KGs. The SemMed KG expresses various relations between medical entities, and these relations are strongly associated with other adjacent relations according to specific logical rules as depicted in 5.1. To encode these logical rules, we use GNN representation learning based on the enclosing subgraph.

To improve the performance of KGC task, we utilize the method, proposed in our other research. In this work, we suggest a novel inductive link prediction model, called **Subgraph Infomax (SGI)**, where the relation embedding is trained to contain more meaningful information about subgraphs via the mutual information (MI) maximization objective. SGI consists of a GNN-based scoring network for computing the score of a given triplet and a module for MI maximization. We trained SGI to maximize the

MI between the relation embedding and the subgraph representation. After pre-training the GNN encoder with the KGC training objective on SemMed KGs, the GNN encoder is further fine-tuned via the supervision coming from clinical outcome prediction.

5.2.4 Subgraph Infomax: SGI

In this section, we provide an overview of our SGI model. The KGC task is to score a triplet (h, r, t) to estimate the probability of a relation r between a head entity h and a tail entity t. Similar to GraIL, we extract an enclosing subgraph $\mathcal{G}_{(h,r,t)}$ around the target nodes, h and t, and use the subgraph structure to score a triplet independently of the node embeddings. In particular, the enclosing subgraph is extracted by intersecting two subsets of k-hop neighboring nodes of the head or tail. The shortest distance of the nodes to the head or tail is used as node features, known as double-radius vertex labeling.

SGI first summarizes the subgraph through a GNN encoder and computes the triplet's score using the encoded-subgraph representation and relation embedding. At the same time, the MI estimator estimates the MI between the relation embedding and the subgraph representation.

GNN-based Scoring Network for Inductive KGC

We use multi-relational R-GCN [76], a GNN-based method designed for modeling multi-relational data, to obtain a subgraph-level representation. The embedding of a node i in the kth layer is given by:

$$\mathbf{h}_i^k = \text{ReLU}(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}_r^k \mathbf{h}_j^{k-1} + \mathbf{W}_{self}^k \mathbf{h}_i^{k-1}), \tag{5.11}$$

where the first term is the aggregated message from the neighbors $\mathcal{N}_i$ of node i. The initial node representation $\mathbf{h}_i^0$ is a node feature labeled by aforementioned double-radius vertex labeling. $\alpha_{i,j}$ denotes an edge attention weight of the edge (i, j) and is given as a function of the source node i, neighbor node j, relation type r_{ij} of the edge (i,j), and the target relation r.

$$\begin{aligned} &\mathbf{s} = \text{ReLU}(\mathbf{A_1}^k[\mathbf{h}_i^{k-1} \oplus \mathbf{h}_j^{k-1} \oplus \mathbf{e}_r \oplus \mathbf{e}_{r_{ij}}] + \mathbf{b_1}^k), \\ &\alpha_{ij} = \sigma(\mathbf{A_2}^k \mathbf{s} + \mathbf{b_2}^k), \end{aligned} \tag{5.12}$$

where $\mathbf{e}_r$ and $\mathbf{e}_{r_{ij}}$ are relation embeddings for relation r and r_{ij}. Different from GraIL that uses attention relation embeddings, separate from the relation embeddings, we use the same relation embeddings during computing attention weights. This allows the GNN encoder to learn a subgraph representation with enhanced connectivity between the subgraph representation and the input relation embedding through the MI estimator that will be described later.

After L layers of message passing, we obtain a subgraph-level representation $\mathbf{h}_{\mathcal{G}_{(h,r,t)}}$ by concatenating the node representation of the source, target, and the average-pooled representation of all the node:

$$\mathbf{h}_{\mathcal{G}_{(h,r,t)}} = \mathbf{h}_h^L \oplus \mathbf{h}_t^L \oplus \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{h}_i^L, \tag{5.13}$$

where $\mathcal{V}$ denotes the set of vertices in $\mathcal{G}_{(h,r,t)}$. To make the model's performances robust to the number of GNN layers, we adopt JK-connections [77] as in GraIL. Finally, we compute a score for (h,r,t) using the subgraph representation and the target relation embedding as follows:

$$\text{score}(h,r,t) = \mathbf{W}[\mathbf{h}_{\mathcal{G}_{(h,r,t)}} \oplus \mathbf{e}_r] \tag{5.14}$$

Subgraph-Relation MI Maximization

The key idea behind our approach is to strengthen the connectivity between the enclosing subgraph around the target triplet and the target relation. Based on the assumption that the enclosing subgraphs contain the logical rules related to the target triplet, we implement the objective to maximize the MI between the subgraph representation encoded with an R-GCN model and the relation embedding.

Motivated by previous works [78, 79] on using MI estimator for graph representation learning, we employ a discriminator $\mathcal{D}_{\psi}(\mathbf{h}_{\mathcal{G}_{(h,r,t)}}, \mathbf{e}_r)$ that represents the feasibility of the subgraph-relation pair. ψ refers to the parameters of $\mathcal{D}$, which is a bilinear function in our study. Negative samples for subgraph-relation pair are given as $(\mathbf{h}_{\mathcal{G}_{(h,r,t)}}, \mathbf{e}_{r'})$, where $\mathbf{h}_{\mathcal{G}_{(h,r,t)}}$ is the graph representation from the positive triplet and $\mathbf{e}_{r'}$ is the relation embedding for a random negative relation. We use a noise-contrastive type objective with a binary-cross entropy loss suggested in [80] for MI maximization so the estimated MI on subgraph-relation pairs over the training set $\mathcal{G}_{train}$ is given as follows:

$$\begin{aligned} \mathcal{I}_{\phi,\psi} := \sum_{(h,r,t)\in\mathcal{G}_{train}} \log[\mathcal{D}_{\psi}(\mathbf{e}_r, \mathbf{h}_{\phi,\mathcal{G}_{(h,r,t)}})] + \\ \sum_{(h,r,t)\in\mathcal{G}_{train}, r'\in\mathcal{R}, r'\neq r} \log[(1 - \mathcal{D}_{\psi}(\mathbf{e}_{r'}, \mathbf{h}_{\phi,\mathcal{G}_{(h,r,t)}}))], \end{aligned} \tag{5.15}$$

where $\mathcal{I}_{\phi,\psi}$ is the MI estimator modeled by discriminator $\mathcal{D}_{\psi}$ and ϕ denotes the set of parameters of a R-GCN encoder. By maximizing the above MI objective, we train the whole networks to learn the subgraph representation that is strongly connected to the target relation embedding.

We use binary-cross entropy (BCE) loss to discriminate the positive and negative triplets as follows:

$$\mathcal{L} = \sum_{p_i\in\mathcal{G}_{train}} -\log(\text{score}(p_i)) - \log(1 - \text{score}(n_i)), \tag{5.16}$$

where p_i is the positive triplet and n_i is the negative triplet extracted by using SANS strategy. Combined with the MI objective in (5.15), the total loss function for pre-training is defined by:

$$\mathcal{L}_{p.t.} = \mathcal{L} - \mathcal{I}_{\phi,\psi}. \tag{5.17}$$

5.3 Dataset and Experimental Setup

5.3.1 Clinical Outcome Prediction

In this research, we extracted the visit data of 75,604 patients from the NHIS-NSC data. Consequently, 7,981 cases and 67,623 controls were extracted with diagnosis and prescription codes. The average visit length for each patient was approximately 57, and the total numbers of unique codes were 1,628 and 1,502 for diagnoses and prescriptions, respectively. Then, we designed more tailored experimental settings as follows. An immediate outpatient CVD diagnosis before CVD admission is not a cause for CVD admission; rather, it should be considered as a point of the first contact in the natural course of CVD detection. However, because our operational definition of CVD was CVD with inpatient admission, cases very often had CVD outpatient visits immediately prior to admission. With such highly-correlated cases, the model was incentivized to predict based on CVD outpatient diagnosis rather than looking at other non-obvious factors. Thus, we cleaned our data by masking all medical data, including CVD outpatient diagnosis codes, within the 7 days (and 14 days) prior to CVD admission on the diagnosis date. We defined this data as the **MASKED_7** and **MASKED_14** dataset, in contrast to the original **RAW** dataset. For each dataset, we used 80% of the data for training, 10% for validation, and the remaining 10% for testing. For next diagnosis prediction task, we randomly sampled 82,311 patients from NHIS-NSC data. We added the classification layer to predict the softmax probabilities to all 1,623 diagnoses.

5.3.2 Next Diagnosis Prediction

For next diagnosis prediction task, we randomly sample 87,384 patients from the entire NHIS-NSC. Our task is to predict the diagnosis codes given the clinical history and the patient characteristics of a patient. We use 80% of the data for training, 10% for validation, and the remaining 10% for testing. Top-k recall is used to evaluate the performance.

5.4 Experimental Results

5.4.1 Cardiovascular Disease Prediction

The experimental results are shown in Table 5.4. Compared to the previous state-of-the-art method, which is SAF-RNN, the knowledge-aumented SAF-RNN shows improved performances in both AUROC and AUPRC. We observe that KA-SAF using `Attend` is more effective than KA-SAF with `Concat`. The performance of KA-SAF with `Concat` is even more improved after pre-training GNN encoder with KGC task. However, KA-SAF with `Attend` shows worsen results. We assume that different schemes to compute the attention weights between pre-training and fine-tuning states interferes with the prediction performances.

5.4.2 Next Diagnosis Prediction

The experimental results are shown in Figure 5.7. Top-k recall is evaluated for $k = 10$ and $k = 30$. For both different k values, SAF-RNN augmented with SemMed KG shows the best performances. Here, we use the KA-SAF with `Concat`. As in CVD prediction task, the performances are even more improved after pre-training GNN encoder with KGC task.

5.4.3 KGC on SemMed KG

In KGC on SemMed KG, we aim to predict the other entity given another entity and the relation type, i.e., predicting the tail given $(h, r, ?)$ or predicting the head given $(?, r, t)$. We evaluate the models on Mean Reciprocal Rank (MRR), Hits at 1 (H@1), and H@10, by ranking each validation triplet among 50 other negative candidates as reported in Table 5.3. The evaluated models are GraIL, SGI, and SGI_f.t., which are the models for inductive KGC. SGI_f.t. is a SGI model where further fine-tuning technique

Table 5.3: KGC results evaluated on SemMed KG with uniform negative triplets

Datasets	SemMed KG		
Models	MRR	Hits@1	Hits@10
GraIL	49.64	41.12	56.88
SGI	55.35	48.28	61.33
SGI_f.t.	57.02	51.75	62.89

Table 5.4: CVD prediction results evaluated on three datasets

Datasets	RAW		MASKED_7		MASKED_14	
Models	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
SAF-RNN	0.839± 0.000	0.661± 0.001	0.784±0.001	0.540± 0.001	0.760± 0.001	0.501± 0.002
KA-SAF (Concat)	0.842± 0.001	0.683± 0.003	0.809± 0.001	0.565± 0.003	0.784± 0.004	0.532± 0.005
KA-SAF (Attend)	0.849± 0.001	0.695± 0.002	0.816± 0.002	0.572± 0.002	0.791± 0.003	0.543± 0.005
KA-SAF (Concat) + p.t.	**0.855± 0.002**	**0.697± 0.003**	**0.818± 0.001**	**0.581± 0.002**	**0.799± 0.003**	**0.550± 0.003**
KA-SAF (Attend) + p.t.	0.848 ±0.003	0.692± 0.003	0.810 ±0.002	0.570± 0.002	0.784± 0.003	0.537± 0.004

is applied. As a result, SGI_f.t. showed the best inductive link prediction performances as in Table 5.3.

5.5 Conclusion

We utilize Semantic Medline Knowledge Graph (SemMed KG) to augment the deep patient representation with prior medical knowledge. A personalized knowledge graph is made by extracting the subgraph of the SemMed KG consisting of the medical codes in each patient's record. Based on the personalized KG, we build the deep patient representations upon the personalized medical knowledge using GNNs. Along with the deep patient representation learned by SAF-RNN, the KG representation is used to predict some clinical outcomes. Specifically, we evaluated the performances of our model on two tasks, i.e., the next diagnosis prediction and the CVD prediction as in the first study. The knowledge-augmented SAF (KA-SAF) showed improved performances in both two tasks, compared to the previous state-of-the-art method. The performances are further improved when we pre-train the GNN encoder with KGC training objective using SGI, showing the enhanced KG representation.

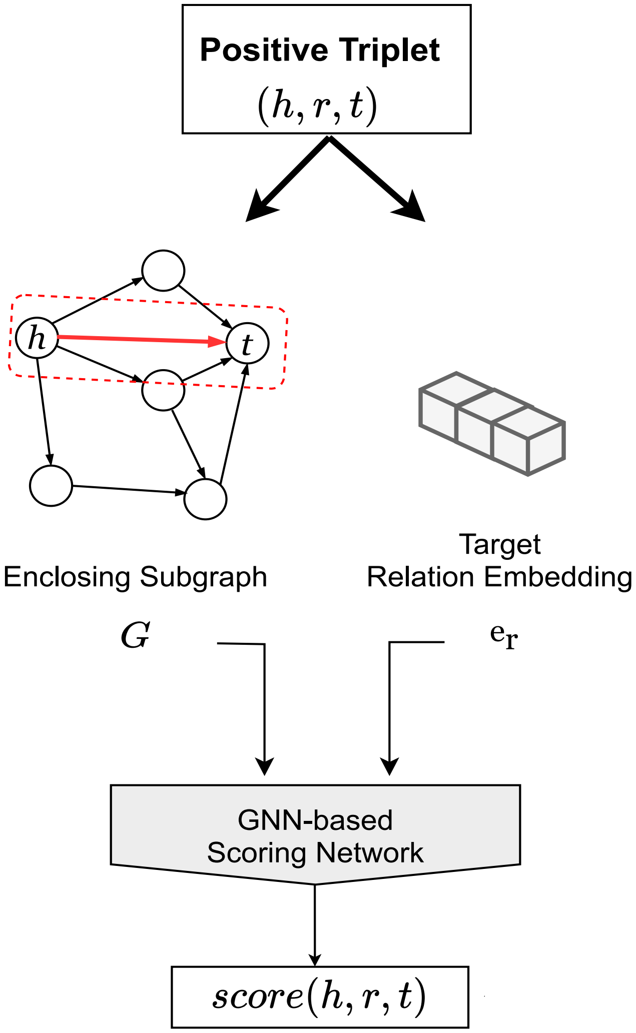


Figure 5.5: GNN-based Scoring Network.

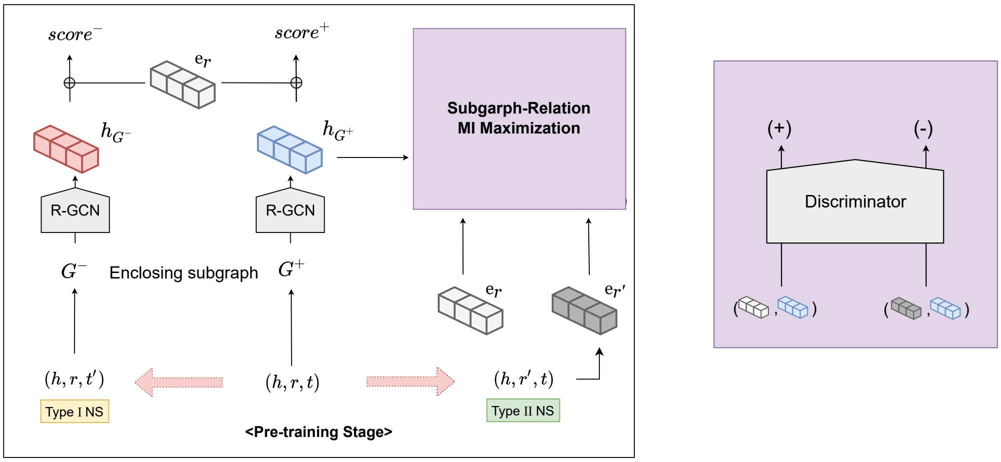


Figure 5.6: Architecture of Subgraph Infomax(SGI).

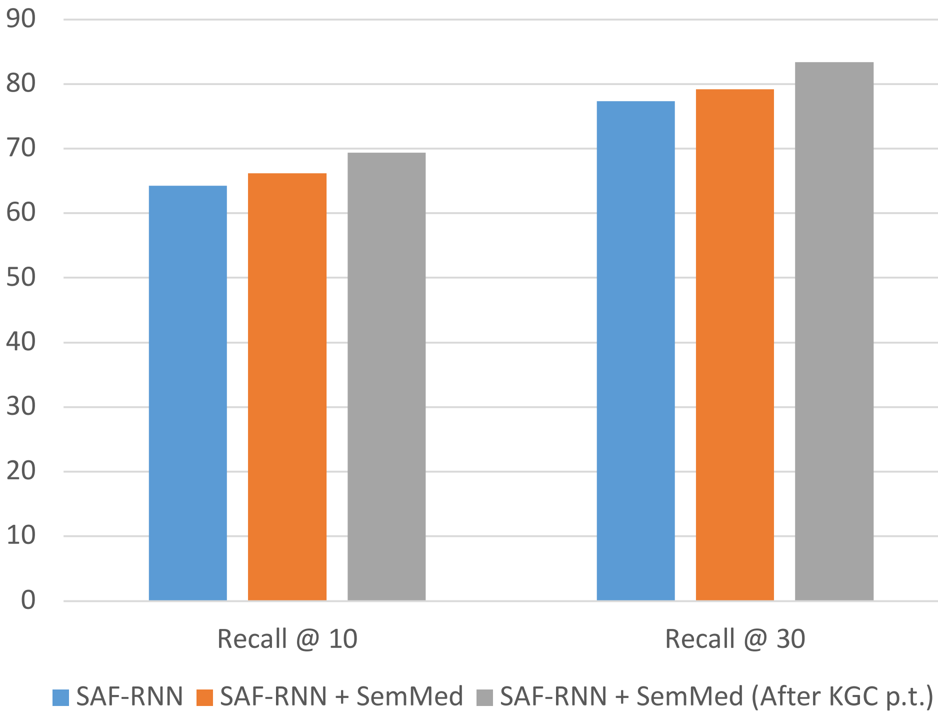


Figure 5.7: Top-k recall results for next diagnosis prediction task.

Chapter 6

Conclusion

This dissertation proposes a deep neural network-based medical concept and patient representation learning methods using National Health Insurance Service-National Sample Cohort (NHIS-NSC) to solve two healthcare tasks, i.e., clinical outcome prediction and post-marketing ADR signal detection.

First, we proposed a Recurrent Neural Network (RNN) model that learns patient representations based on the clinical sequences along with the fixed patient characteristics, and predicts the risk probability of a cardiovascular disease onset. Our proposed model efficiently fuses different types of information using feature-based gating and self-attention. We demonstrated that high-level associations between two heterogeneous patient records are effectively extracted during the process of feature-based gating and the computation of self-attention.

Secondly, based on the observation that the distributed representation of medical code contains temporal information, we introduced a graph structure in to enhance the code embedding with such temporal information. We constructed a graph using the similarity between the distributed vectors of medical code and the statistical information between medical codes. We then proposed the Graph Neural Networks(GNN)-based representation learning approach for post-marketing ADR detection. Our model showed competitive performance in predicting drug-ADR pairs. It especially predicted

ADR candidates that do not exist in the existing ADR database, showing its capability to supplement the ADR database.

The suggest graph construction method only requires simple data processing and well-established medical terminologies. Therefore, our work does not demand case-by-case feature engineering that requires expertise, and thereby the detection for the whole drug candidates can be fully automated.

Finally, rather than just learning patient representations using patient records alone, we utilized Semantic Medline Knowledge Graph (SemMed KG) that specifies relationships between medical entities to augment the deep patient representation with prior medical knowledge. Here, the personalized KG was created by extracting the subgraph consisting of only the medical codes of each patient. Based on the personalized KG, we built the KG representation using GNN-encoder. We combined the KG representation and the deep patient representation using two approaches: `Concat` and `Attend`.

We additionally seek to harness pre-training for GNNs to further enhance the personalized KG representation. We perform KG completion (KGC) task as a pre-training task using the method, proposed in our other research. In this work, the relation embedding is trained to contain more meaningful information about subgraphs via the mutual information (MI) maximization objective. After pre-training the GNN encoder with the KGC training objective on SemMed KGs, the GNN encoder is further fine-tuned via the supervision coming from clinical outcome prediction. We evaluated the performances of our model on two tasks, i.e., the next diagnosis prediction and the CVD prediction. As a result, it showed improved prediction performances compared to the case of using SAF-RNN alone. The performances are further improved when we pre-train the GNN encoder with KGC training objective using SGI, showing the enhanced KG representation.

Bibliography

[1] E. Jeong, N. Park, Y. Choi, R. W. Park, and D. Yoon, "Machine learning model combining features from algorithms with different analytical methodologies to detect laboratory-event-related adverse drug reaction signals," *PloS one*, vol. 13, no. 11, 2018.

[2] J. Lee, J. S. Lee, S.-H. Park, S. A. Shin, and K. Kim, "Cohort profile: the national health insurance service–national sample cohort (nhis-nsc), south korea," *International journal of epidemiology*, vol. 46, no. 2, pp. e15–e15, 2016.

[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[5] K. K. Teru, E. Denis, and W. L. Hamilton, "Inductive relation prediction by subgraph reasoning.," *arXiv: Learning*, 2020.

[6] J. Chen, H. He, F. Wu, and J. Wang, "Topology-Aware Correlations Between Relations for Inductive Link Prediction in Knowledge Graphs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, 2021.

[7] J. P. Bynum, P. V. Rabins, W. Weller, M. Niefeld, G. F. Anderson, and A. W. Wu, "The relationship between a dementia diagnosis, chronic illness, medicare expen-

ditures, and hospital use," *Journal of the American Geriatrics Society*, vol. 52, no. 2, pp. 187–194, 2004.

[8] M. Q. Stearns, C. Price, K. A. Spackman, and A. Y. Wang, "Snomed clinical terms: overview of the development process and project status.," in *Proceedings of the AMIA Symposium*, p. 662, American Medical Informatics Association, 2001.

[9] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor ai: Predicting clinical events via recurrent neural networks," in *Machine learning for healthcare conference*, pp. 301–318, PMLR, 2016.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[11] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Using recurrent neural network models for early detection of heart failure onset," *Journal of the American Medical Informatics Association*, vol. 24, no. 2, pp. 361–370, 2017.

[12] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Medical concept representation learning from electronic health records and its application on heart failure prediction," *arXiv preprint arXiv:1602.03686*, 2016.

[13] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Deepcare: A deep dynamic memory model for predictive medicine," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 30–41, Springer, 2016.

[14] C. Esteban, O. Staeck, S. Baier, Y. Yang, and V. Tresp, "Predicting clinical events by combining static and dynamic information using recurrent neural networks," in *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 93–101, IEEE, 2016.

[15] Y. Xu, S. Biswal, S. R. Deshpande, K. O. Maher, and J. Sun, "Raim: Recurrent attentive and intensive model of multimodal patient monitoring data," in *Proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining*, pp. 2565–2573, 2018.

[16] J. Zhang, K. Kowsari, J. H. Harrison, J. M. Lobo, and L. E. Barnes, "Patient2vec: A personalized interpretable deep representation of the longitudinal electronic health record," *IEEE Access*, vol. 6, pp. 65333–65346, 2018.

[17] 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 *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1903–1911, 2017.

[18] Y. Sha and M. D. Wang, "Interpretable predictions of clinical outcomes with an attention-based recurrent neural network," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 233–240, 2017.

[19] L. Wang, Q. Wang, H. Bai, C. Liu, W. Liu, Y. Zhang, L. Jiang, H. Xu, K. Wang, and Y. Zhou, "Ehr2vec: representation learning of medical concepts from temporal patterns of clinical notes based on self-attention mechanism," *Frontiers in genetics*, p. 630, 2020.

[20] T. Bai, S. Zhang, B. L. Egleston, and S. Vucetic, "Interpretable representation learning for healthcare via capturing disease progression through time," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 43–51, 2018.

[21] F. López-Martínez, E. R. Núñez-Valdez, R. G. Crespo, and V. García-Díaz, "An artificial neural network approach for predicting hypertension using nhanes data," *Scientific Reports*, vol. 10, no. 1, pp. 1–14, 2020.

[22] E. Lin, J. L. Hefner, X. Zeng, S. Moosavinasab, T. Huber, J. Klima, C. Liu, and S. M. Lin, “A deep learning model for pediatric patient risk stratification,” *Am J Manag Care*, vol. 25, no. 10, pp. e310–5, 2019.

[23] S.-J. Heo, Y. Kim, S. Yun, S.-S. Lim, J. Kim, C.-M. Nam, E.-C. Park, I. Jung, and J.-H. Yoon, “Deep learning algorithms with demographic information help to detect tuberculosis in chest radiographs in annual workers’ health examination data,” *International journal of environmental research and public health*, vol. 16, no. 2, p. 250, 2019.

[24] F. J. Catling and A. H. Wolff, “Temporal convolutional networks allow early prediction of events in critical care,” *Journal of the American Medical Informatics Association*, vol. 27, no. 3, pp. 355–365, 2020.

[25] Y. Cheng, F. Wang, P. Zhang, and J. Hu, “Risk prediction with electronic health records: A deep learning approach,” in *Proceedings of the 2016 SIAM international conference on data mining*, pp. 432–440, SIAM, 2016.

[26] P. Nguyen, T. Tran, N. Wickramasinghe, and S. Venkatesh, “Deepr: a convolutional net for medical records (2016),” *ArXiv160707519 Cs Stat*.

[27] 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,” *Advances in neural information processing systems*, vol. 29, 2016.

[28] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” *arXiv preprint arXiv:1508.04025*, 2015.

[29] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[30] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition," in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 3687–3691, IEEE, 2013.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[32] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[33] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," *arXiv preprint arXiv:1601.06733*, 2016.

[34] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.

[35] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

[36] Z. Yu, J. Yu, C. Xiang, J. Fan, and D. Tao, "Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 5947–5959, 2018.

[37] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6281–6290, 2019.

[38] S. Yoon, S. Dey, H. Lee, and K. Jung, "Attentive modality hopping mechanism for speech emotion recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3362–3366, IEEE, 2020.

[39] D. Hazarika, S. Gorantla, S. Poria, and R. Zimmermann, "Self-attentive feature-level fusion for multimodal emotion detection," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 196–201, IEEE, 2018.

[40] J. Lee, J. S. Lee, S.-H. Park, S. A. Shin, and K. Kim, "Cohort profile: the national health insurance service–national sample cohort (nhis-nsc), south korea," *International journal of epidemiology*, vol. 46, no. 2, pp. e15–e15, 2017.

[41] J. S. Son, S. Choi, K. Kim, S. M. Kim, D. Choi, G. Lee, S.-M. Jeong, S. Y. Park, Y.-Y. Kim, J.-M. Yun, *et al.*, "Association of blood pressure classification in korean young adults according to the 2017 american college of cardiology/american heart association guidelines with subsequent cardiovascular disease events," *Jama*, vol. 320, no. 17, pp. 1783–1792, 2018.

[42] S. M. Kim, G. Lee, S. Choi, K. Kim, S.-M. Jeong, J. S. Son, J.-M. Yun, S. G. Kim, S.-s. Hwang, S. Y. Park, *et al.*, "Association of early-onset diabetes, prediabetes and early glycaemic recovery with the risk of all-cause and cardiovascular mortality," *Diabetologia*, vol. 63, no. 11, pp. 2305–2314, 2020.

[43] S. R. Kim, S. Choi, N. Keum, and S. M. Park, "Combined effects of physical activity and air pollution on cardiovascular disease: A population-based study," *Journal of the American Heart Association*, vol. 9, no. 11, p. e013611, 2020.

[44] J. Lazarou, B. H. Pomeranz, and P. N. Corey, "Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies," *JAMA*, vol. 279, no. 15, pp. 1200–1205, 1998.

[45] A. Bate, M. Lindquist, I. R. Edwards, and R. Orre, "A data mining approach for signal detection and analysis," *Drug Safety*, vol. 25, no. 6, pp. 393–397, 2002.

[46] M. Hauben and A. Bate, "Decision support methods for the detection of adverse events in post-marketing data," *Drug discovery today*, vol. 14, no. 7-8, pp. 343–357, 2009.

[47] A. Hochberg and M. Hauben, "Time-to-signal comparison for drug safety data-mining algorithms vs. traditional signaling criteria," *Clinical Pharmacology & Therapeutics*, vol. 85, no. 6, pp. 600–606, 2009.

[48] L. Hazell and S. A. Shakir, "Under-reporting of adverse drug reactions," *Drug safety*, vol. 29, no. 5, pp. 385–396, 2006.

[49] A. Sarker, R. Ginn, A. Nikfarjam, K. O'Connor, K. Smith, S. Jayaraman, T. Upadhaya, and G. Gonzalez, "Utilizing social media data for pharmacovigilance: a review," *Journal of biomedical informatics*, vol. 54, pp. 202–212, 2015.

[50] M. Y. Park, D. Yoon, K. Lee, S. Y. Kang, I. Park, S.-H. Lee, W. Kim, H. J. Kam, Y.-H. Lee, J. H. Kim, *et al.*, "A novel algorithm for detection of adverse drug reaction signals using a hospital electronic medical record database," *Pharmacoepidemiology and drug safety*, vol. 20, no. 6, pp. 598–607, 2011.

[51] M. Liu, Y. Wu, Y. Chen, J. Sun, Z. Zhao, X.-w. Chen, M. E. Matheny, and H. Xu, "Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs," *Journal of the American Medical Informatics Association*, vol. 19, no. e1, pp. e28–e35, 2012.

[52] E. Pauwels, V. Stoven, and Y. Yamanishi, "Predicting drug side-effect profiles: a chemical fragment-based approach," *BMC bioinformatics*, vol. 12, no. 1, p. 169, 2011.

[53] C. Su, J. Tong, Y. Zhu, P. Cui, and F. Wang, “Network embedding in biomedical data science,” *Brief. Bioinform*, pp. 1–16, 2018.

[54] X. Yue, Z. Wang, J. Huang, S. Parthasarathy, S. Moosavinasab, Y. Huang, S. M. Lin, W. Zhang, P. Zhang, and H. Sun, “Graph embedding on biomedical networks: Methods, applications, and evaluations,” *arXiv preprint arXiv:1906.05017*, 2019.

[55] I. Karlsson, J. Zhao, L. Asker, and H. Boström, “Predicting adverse drug events by analyzing electronic patient records,” in *Conference on Artificial Intelligence in Medicine in Europe*, pp. 125–129, Springer, 2013.

[56] M. Liu, E. R. McPeek Hinz, M. E. Matheny, J. C. Denny, J. S. Schildcrout, R. A. Miller, and H. Xu, “Comparative analysis of pharmacovigilance methods in the detection of adverse drug reactions using electronic medical records,” *JAMIA*, vol. 20, no. 3, pp. 420–426, 2012.

[57] D. Yoon, M. Park, N. Choi, B. J. Park, J. H. Kim, and R. Park, “Detection of adverse drug reaction signals using an electronic health records database: Comparison of the laboratory extreme abnormality ratio (clear) algorithm,” *Clinical Pharmacology & Therapeutics*, vol. 91, no. 3, pp. 467–474, 2012.

[58] J. Shang, C. Xiao, T. Ma, H. Li, and J. Sun, “Gamenet: Graph augmented memory networks for recommending medication combination,” in *Proceedings of the AAAI*, vol. 33, pp. 1126–1133, 2019.

[59] M. Wang, M. Liu, J. Liu, S. Wang, G. Long, and B. Qian, “Safe medicine recommendation via medical knowledge graph embedding,” *arXiv:1710.05980*, 2017.

[60] Y. Zhang, R. Chen, J. Tang, W. F. Stewart, and J. Sun, “Leap: learning to prescribe effective and safe treatment combinations for multimorbidity,” in *Proceedings of the 23rd ACM SIGKDD*, pp. 1315–1324, ACM, 2017.

[61] H. Le, T. Tran, and S. Venkatesh, "Dual memory neural computer for asynchronous two-view sequential learning," in *Proceedings of the 24th ACM SIGKDD*, pp. 1637–1645, ACM, 2018.

[62] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

[63] M. Fiszman, T. C. Rindflesch, and H. Kilicoglu, "Abstraction summarization for managing the biomedical research literature," in *Proceedings of the computational lexical semantics workshop at HLT-NAACL 2004*, pp. 76–83, 2004.

[64] H. Kilicoglu, G. Rosemblat, M. Fiszman, and T. C. Rindflesch, "Constructing a semantic predication gold standard from the biomedical literature," *BMC bioinformatics*, vol. 12, no. 1, pp. 1–17, 2011.

[65] P. Ernst, C. Meng, A. Siu, and G. Weikum, "Knowlife: a knowledge graph for health and life sciences," in *2014 IEEE 30th International Conference on Data Engineering*, pp. 1254–1257, IEEE, 2014.

[66] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: graph-based attention model for healthcare representation learning," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 787–795, 2017.

[67] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, and J. Gao, "Kame: Knowledge-based attention model for diagnosis prediction in healthcare," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 743–752, 2018.

[68] C. Yin, R. Zhao, B. Qian, X. Lv, and P. Zhang, "Domain knowledge guided deep learning with electronic health records," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 738–747, IEEE, 2019.

[69] L. J. Liu, V. Ortiz-Soriano, J. A. Neyra, and J. Chen, “Kgdal: knowledge graph guided double attention lstm for rolling mortality prediction for aki-d patients,” in *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 1–10, 2021.

[70] M. Ye, S. Cui, Y. Wang, J. Luo, C. Xiao, and F. Ma, “Medpath: Augmenting health risk prediction via medical knowledge paths,” in *Proceedings of the Web Conference 2021*, pp. 1397–1409, 2021.

[71] K. Agarwal, T. Eftimov, R. Addanki, S. Choudhury, S. Tamang, and R. Rallo, “Snomed2vec: Random walk and poincar\’e embeddings of a clinical knowledge base for healthcare analytics,” *arXiv preprint arXiv:1907.08650*, 2019.

[72] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” *Advances in neural information processing systems*, vol. 26, 2013.

[73] A. Sadeghian, M. Armandpour, P. Ding, and D. Z. Wang, “DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[74] S. Mai, S. Zheng, Y. Yang, and H. Hu, “Communicative Message Passing for Inductive Relation Reasoning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021.

[75] M. Zhang and Y. Chen, “Link prediction based on graph neural networks,” in *Advances in Neural Information Processing Systems*, pp. 5165–5175, 2018.

[76] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in *ESWC*, 2018.

[77] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462, PMLR, 10–15 Jul 2018.

[78] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep Graph Infomax," in *International Conference on Learning Representations*, 2019.

[79] F.-Y. Sun, J. Hoffman, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *International Conference on Learning Representations*, 2019.

[80] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *International Conference on Learning Representations*, 2019.

초 록

본 학위 논문은 전국민 의료 보험데이터인 표본코호트DB를 활용하여 딥 뉴럴 네트워크 기반의 의학 개념 및 환자 표현 학습 방법과 의료 문제 해결 방법을 제안한다. 먼저 순차적인 환자 의료 기록과 개인 프로파일 정보를 기반으로 환자 표현을 학습하고 향후 질병 진단 가능성을 예측하는 재귀신경망 모델을 제안하였다. 우리는 다양한 성격의 환자 정보를 효율적으로 혼합하는 구조를 도입하여 큰 성능 향상을 얻었다. 또한 환자의 의료 기록을 이루는 의료 코드들을 분산 표현으로 나타내 추가 성능 개선을 이루었다. 이를 통해 의료 코드의 분산 표현이 중요한 시간적 정보를 담고 있음을 확인하였고, 이어지는 연구에서는 이러한 시간적 정보가 강화될 수 있도록 그래프 구조를 도입하였다. 우리는 의료 코드의 분산 표현 간의 유사도와 통계적 정보를 가지고 그래프를 구축하였고 그래프 뉴럴 네트워크를 활용, 시간·통계적 정보가 강화된 의료 코드의 표현 벡터를 얻었다. 획득한 의료 코드 벡터를 통해 시판 약물의 잠재적인 부작용 신호를 탐지하는 모델을 제안한 결과, 기존의 부작용 데이터베이스에 존재하지 않는 사례까지도 예측할 수 있음을 보였다. 마지막으로 분량에 비해 주요 정보가 희소하다는 의료 기록의 한계를 극복하기 위해 지식그래프를 활용하여 사전 의학 지식을 보강하였다. 이때 환자의 의료 기록을 구성하는 지식그래프의 부분만을 추출하여 개인화된 지식그래프를 만들고 그래프 뉴럴 네트워크를 통해 그래프의 표현 벡터를 획득하였다. 최종적으로 순차적인 의료 기록을 함축한 환자 표현과 더불어 개인화된 의학 지식을 함축한 표현을 함께 사용하여 향후 질병 및 진단 예측 문제에 활용하였다.

주요어: 의료데이터, 질병 예측 시스템, 딥 뉴럴 네트워크

학번: 2014-22543

ACKNOWLEGEMENT

I would like to thank my family and the MILAB members who gave me abundant insights and expertise. Especially, I would like to express my gratitude to professor K. Jung for the advisory and discussion throughout the academic years.