Multiple Graph Kernel Fusion Prediction of Drug Prescription

Hao-Ren Yao
hy301@georgetown.edu
Georgetown University
Washington, D.C., USA

Der-Chen Chang
chang@georgetown.edu
Georgetown University
Washington, D.C., USA

Ophir Frieder
ophir@ir.cs.georgetown.edu
Georgetown University
Washington, D.C., USA

Wendy Huang
al0357186@hotmail.com
Meng Cheng Family Medicine Clinic
Kaohsiung City, Taiwan

Tian-Shyug Lee
036665@mail.fju.edu.tw
Fu Jen Catholic University
New Taipei City, Taiwan

ABSTRACT

We present an end-to-end interpretable deep architecture that predicts the success of drug prescription based on multiple graph kernel fusion using a graphical representation of electronic health records. We formulate the predictive model as a binary graph classification problem with a set of graph kernels proposed to capture different aspects of graph structures through deep neural networks. Results using the Taiwanese National Health Insurance Research Database demonstrate that our approach outperforms current start-of-the-art models on accuracy and interpretability. The approach is in preliminary deployment.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Applied computing** → **Health informatics**.

KEYWORDS

Health informatics; Predictive model; Deep learning; Multiple kernel fusion; Graph kernel

ACM Reference Format:
Hao-Ren Yao, Der-Chen Chang, Ophir Frieder, Wendy Huang, and Tian-Shyug Lee. 2019. Multiple Graph Kernel Fusion Prediction of Drug Prescription. In *10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (ACM-BCB '19), September 7–10, 2019, Niagara Falls, NY, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3307339.3342134

1 INTRODUCTION

Predictive modeling of drug prescription is paramount yet not always achieved. For any clinical visit, medical doctors are obligated to prescribe the most suitable and least harmful drug that combats the ailment while minimizing adverse side-effects [3, 39]. Many machine learning techniques exist to solve such predictive modeling problems; their development was fueled by the rapid growth of Electronic Health Recoreds (EHRs), providing opportunities to mine medical data [1, 29]. EHRs provide historical medical road-maps for patients enabling the design of intelligent predictive systems.

The complex nature of EHR, such as high dimensional information and temporal event relationships, complicates their use in developing predictive models. Traditional approaches transform EHRs into vector representations via various feature extraction techniques (e.g. electronic phenotyping) [37]. The extracted feature vectors, where each dimension corresponds to a certain medical concept, are fed into a linear classifier. This flattening formulation of EHRs ignores temporal relationships between medical events in a patient's history, reducing effectiveness. On the other hand, many extraction tasks require domain medical knowledge to generate hand-crafted features which is not efficient and cost prohibitive at large scale. [37].

We address this accuracy loss via a graphical EHR formulation. Such formulation compactly encompasses all the medical information, without the need of electronic phenotyping. A set of graph kernels is proposed to compute the similarity between graphical EHR with multi-view captured by different types of kernels. To achieve the best kernel combination, deep learning formulation for an end-to-end multiple kernel learning is applied, resulting in meaningful and noise-resistant refined kernel embedding, while maintaining the interpretability. Dimension reduction by virtue of kernel embedding boosts the efficiency for large scale learning.

A common strategy, derived from recent development of deep learning, is to apply representation learning to embed EHR into low dimensional space to represent the medical concept combining with downstream classifier [33]. The end-to-end learning performs various prediction tasks such as diagnosis code prediction [6, 13], mortality prediction [32], and risk prediction [9, 12] and achieves impressive accuracy.

On the other hand, interpretability is not only critical for doctors and patients but also essential for knowledge discovery in the medical domain. At the same time, it is unreliable for people to trust such an elusive system resulting in low utilization. More and more studies introduce interpretable models to resolve such issues. However, producing a good interpretation is still a challenge for current deep learning models.

We continue our previous work [38] and introduce a kernel based deep architecture to predict the success or failure for drug prescription given to a patient. The success and failure of medication on patients are identified for targeted disease treatment to generate

ACM-BCB '19, September 7–10, 2019, Niagara Falls, NY, USA

ACM ISBN 978-1-4503-6666-3/19/09…$15.00
https://doi.org/10.1145/3307339.3342134

the training data. Observation windows with user specified periods are used to define the success and failure cases. An EHR prior to the disease diagnosis is included for each patient, and their graphical representation (e.g., patient graph), where nodes denote all medical events with day differences as edge weights, are built. The binary graph classification task is performed directly on the patient graph via a deep architecture. Interpretability is readily available and easily accepted by users without further post-processing due to the nature of the graph structure.

We also propose a novel graph kernel: Temporal proximity kernel, which efficiently calculates temporal similarity between two patient graphs. The kernel function is proven to be positive definite, increasing the model availability by using a kernelized classifier such as Support Vector Machine (SVM). To obtain the multi-view aspect, we combine the temporal proximity kernel with the node kernel and the shortest path kernel as a single kernel through multiple kernel learning.

To perform large scale and noise-resistant learning objectives, we transfer the original task to similarity-based classification [11], where each row in the kernel gram matrix is considered as a feature vector with each dimension expressing the similarity measurement with specific training examples. A multiple graph kernel fusion approach is proposed to learn kernel representation in an end-to-end manner for the best kernel combination. We argue that representation learning is a typical kernel approximation which preserves the similarity while reducing the dimension for the original kernel matrix. The embedding weight for each kernel supports the interpretation to the prediction via most similar cases by selecting top relevant embedding dimension.

We evaluate our proposed method by using the National Health Insurance Research Database (NHIRD); a real world population claim-based database from Taiwan. The task is to predict the outcome of success or failure for a prescribed drug to patients given their disease diagnosis at the time of treatment and their medical history prior to the diagnosis. The experimental results show that our proposed multiple graph kernel fusion approach outperforms the current state-of-the-art deep learning models as well as the traditional feature extraction approaches. In addition, we demonstrate the interpretability by analyzing the kernel embedding to infer relevant features corresponding to the original feature space, providing insight on prediction. Finally, we discuss the observation on possible model biases for major state-of-the-art deep learning models on NHIRD. To our best knowledge, we are the first to propose the predictive model by combining deep architecture and graph kernel method, and compare the majority state-of-the-art deep learning baselines on large scale, real-world, different population-based dataset. The described approach is now under limited preliminary use. Our contributions using the proposed model are:

- We propose and discuss how multiple kernel fusion can be utilized to develop an interpretable predictive model for drug prescriptions.
- We show the improvement of interpretability of a deep architecture by combining it with a kernel method.
- We compare the majority of the state-of-the-art deep learning baselines to demonstrate our effectiveness as well as interpretability on a large scale real-world dataset.
- We discuss the possible model biases when performing the prediction task on different population-based datasets.

2 RELATED WORK

2.1 Predictive models for Drug prescription

An erroneous medication treatment process results in an unsuccessful treatment or harmful outcome to patients [3]. One interest is predicting Adverse Drug Reactions (ADRs) or possible medication errors for given prescriptions. A data mining technique to derive treatment algorithms from EHR to improve theoretical empirical therapy for outpatient urinary tract infections is developed in [1]. Predicting ADRs by a hierarchical Bayesian model formulation is proposed in [36]. Another paper [28] describes a method that uses a probability model with association rule mining to predict a possible unsafe drug prescription, and a machine learning approach for predicting a failure in drug prescription on anti-diabetic drugs is introduced in [19].

2.2 Models for EHR analytics

Traditionally, major approaches rely on extracting features or phenotypes from diverse EHR data representation with linear models and ensemble methods like Random Forset as downstream classifiers [37]. Due to the recent prosperity of deep learning approaches, representation learning, embedding EHR from high dimensional input space into low dimensional space, plays the major role in EHR analytics. Models like autoencoder [25] and multilayer perceptron (MLP) [14] are two examples. The majority of related work focuses on using Recurrent Neural Networks (RNNs) to model the temporal event sequences of clinical visits in EHR, where an EHR is treated as a sequence of feature vectors [6, 13]. Convolution Neural Networks (CNNs) are also applied to learn local patterns [9, 12] or clinical visit progression motifs [27]. Once the representation is learned, an end-to-end classification task is performed.

2.3 Interpretable Deep learning models

For classification or prediction, Med2Vec [14] uses MLP with Rectified linear unit (ReLU) activation function to embed medical codes and clinical visits into interpretable low dimensional spaces. The success of attention mechanism in nerual machine translation [4, 21] provides opportunities for interpretability for RNN-based models. Retain [16] used a two-level neural attention model to assign an attention weight for each visit and capture relevant medical information. Dipole [22] introduces a three attention mechanism with bi-directional long short term memory network (LSTM) to predict patient future medical diagnosis. A hierarchical attention networks is introduced in [32] with Gated Recurrent Units (GRUs) to detect code level and visit level attention. Timeline [5] proposes an attention based RNN to learn time progression patterns of disease by learning time decay factors for every medical code. GRAM [15] combines medical ontologies to learn medical concept representation by graph-based attention model to address data insufficiency and align the interpretation with medical knowledge [1]. In [20], attention based RNN along with conditional variational

[1]We will not compare GRAM since we do not use and incorporate external medical ontologies.

autoencoder (CVAE) is developed to learn both temporal medical events and patient demographic information.

2.4 Problems on current interpretable models

Although many studies successfully improve the interpretability of deep learning models, few problems are yet unsolved. First, interpretation should lead to medical knowledge discovery. Posing attention on relevant medical code or visit enables inference on cause of prediction. However, this inference only happens in local perspective, which the interpretation is only used for individual patient, and lowers the ability to perform global knowledge inference such as discovering inter-patient disease progression. Second, to deal with complex representation learning on EHR, models are developed in a very complex structure resulting in implementation difficulty and overfitting possibility. Additional attention layer to design interpretable model expands such complexity. Third, the offered interpretation should be familiar with medical doctor, where case-based study is a common clinical practice performed by medical clinician [18, 23], and improve model utilization. We design a case learning based interpretation, which mimics the real medical learning method, instead of returning attention spots in the specific part of information.

2.5 Graph kernel and Multiple Kernel Fusion

Graph kernels compute the similarity between pairs of graphs. A positive definite kernel performs the inner product by mapping from an input space to the Hilbert space implicitly. Recent graph kernel approaches compare two input graphs based on their common substructures [17]. Many of the applications introduced by graph kernels are within the domains of bioinformatics [17]. An application using EHRs is proposed in [38], however it is still relatively limited. We design a kernel approach to predict medication success using EHRs.

Multiple kernel learning is a framework. The optimization problem for minimizing objective loss function is performed simultaneously with the convex combination on a set of kernels, soas to manipulate the interdependent behavior of different features collectively by different kernel functions [34]. Recently, a multiple kernel learning framework, working on feature fusion, with a deep learning approach was proposed [34]. Instead of traditional multiple kernel learning framework, the learned representation can be considered as a replacement of convex combination approach by using representation learning on kernel merging. In our work, we proposed a multi-layer embedding framework with associated ReLU activation function, where an embedding is learned for each kernel, and a fusion embedding is learned for their joint representation followed by a fully-connected layer with sigmoid activation function commuting the binary cross-entropy loss. Given a set of kernel gram matrix, instead of minimizing the classification loss as well as finding the best convex combination, we aim to learn their individual kernel representation jointly which derives the best combination in an end-to-end fashion.

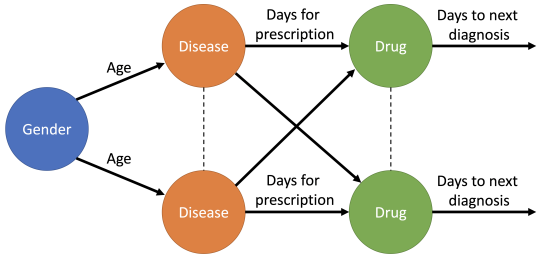


Figure 1: An example of patient graph

3 METHODOLOGY

3.1 EHR patient graph

Following our previous work [38], a patient's EHR is represented by a directed acyclic graph where each node represents a medical event, and an edge between two nodes represents ordering and time difference (e.g., days) as a edge weight. All patient demographic information, e.g., gender, connect to the first medical event with age as an edge weight. Figure 1 describes an example patient graph.

Given n medical events, set $M = \{(m_1, t_1), \ldots, (m_n, t_1)\}$ represents a patient's EHR with m_i denoting a medical event such as diagnosis or drug prescription, and t_i denoting the time for m_i. For each patient, their demographic information is represented as a set of string (e.g., {Male, Student, Postcode....}) with length k $D = \{d_1, \ldots, d_k\}$. We define the patient graph as follows:

Definition 3.1 (Patient Graph). The patient graph $P_g = (V, E)$ of events M and demographic information D is a weighted directed acyclic graph with its vertices V containing all events $m_i \in M$ and $d_j \in D$. Edges E contains all pairs of consecutive events (m_i, m_j) and all pairs of demographic information connected to all the first medical events. The edge weight from node i to node j is defined as $W_{ij} = t_j - t_i$ which defines the time interval between m_i, m_j if both node i, j are medical events, and $W_{ij} = age$ if node i is a demographic information and node j is a medical event [2].

3.2 Success and Failure cases

Given a disease diagnosis of a patient, a drug prescription for the diagnosis is considered a failure if the patient has a second same diagnosis within an observation window. Otherwise, the prescription is considered a success. Figure 2 explains this criterion [3]. The failure is labelled as positive, and the success is labelled as negative. To capture historical factors, each case contains previous medical events prior to the diagnosis date in a user-defined period. We treat each case as a subset of patient EHRs as Figure 3, which contains a multiple-event single-patient EHR. In short, each case contains the medical events before and after the disease diagnosis for a user-defined period.

3.3 Problem Definition

We aim to perform a binary graph classification on graph EHR. Given success and failure cases with their associated label (g_i, y_i),

[2] To simplify model assumption, we only use gender and age as demographic information.

[3] We follow the same setting in [38].

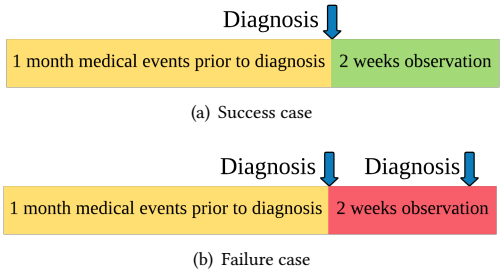


Figure 2: Criteria for success and failure cases

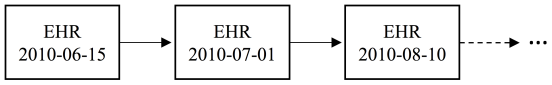


Figure 3: A sample subset of EHRs

we want to learn a classifier such that $f(g_i) = y_i$ where $y_i \in \{0, 1\}$ to predict the success or failure outcome y_i of the given prescription in g_i. This problem can be easily tackled via a kernelized support vector machine (Kernel-SVM) with a proper graph kernel which is demonstrated in our previous work [38]. The predictive system is illustrated in Figure 4

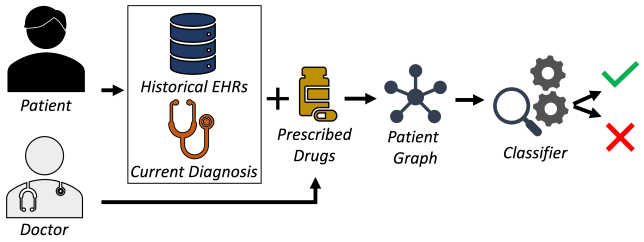


Figure 4: Predictive framework

3.4 Graph Kernel

For a given pair of graph input g_1, g_2, we want to calculate their kernel value via a kernel function. Before introducing our proposed ***Temporal proximity kernel***, we need to define ***Topological sequence*** and ***Temporal signature***.

Topological sequence

Let T be a topological ordering of graph $G = (V, E)$ such that $T = \{n_i \mid i = 1, \ldots, |V|\}$, the topological sequence S is defined as

$$S = \{n_i.label + level \mid i = 1, \ldots, |V|, \text{ and } n_i \in T\} \quad (1)$$

where + represents the string concatenation and *level* denotes the order of occurrence of *label* associated to node n_i in T. Namely, every node in the topological sequence has an attached number to indicate the level. The level indicates the order of occurrence of the same node label in the topological ordering.

Topological signature

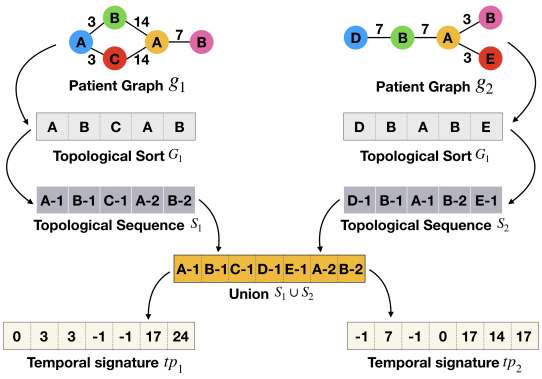


Figure 5: Input graphs to temporal signatures.

Let S_1, S_2 be topological sequences of two input graphs g_1, g_2, and $S = S_1 \cup S_2$ with the union set length $m = |S|$. We define the temporal signature for g_1 as $tp_1 = \{v_{11}, \cdots, v_{1m}\}$ where

$$v_{1j} = \begin{cases} d_j, & \text{if } S[j] \in S_1 \\ -1, & \text{otherwise} \end{cases}, \text{ for } j = 1, \ldots, m \quad (2)$$

and define the temporal signature for g_2 as $tp_2 = \{v_{21}, \cdots, v_{2m}\}$ where

$$v_{2j} = \begin{cases} d_j, & \text{if } S[j] \in S_2 \\ -1, & \text{otherwise} \end{cases}, \text{ for } j = 1, \ldots, m \quad (3)$$

for d_j denotes the total passage day from the root node to node n_j in its belonging patient graph. We now transfer g_1, g_2 into their vector representation tp_1, tp_2. Figure 5 illustrates the process of transferring the input from graphs to temporal signature.

Temporal proximity kernel

Temporal proximity kernel K_{tp} calculates the kernel value between g_1, g_2 via ***temporal signature*** tp_1, tp_2 as:

$$K_{tp}(g_1, g_2) = e^{-\|tp_1 - tp_2\|} \quad (4)$$

where $\|tp_1 - tp_2\|$ is the Euclidean distance between tp_1, tp_2.

Shortest path kernel[4]

Shortest path kernel K_{sp} calculates the edge walk similarity on the shortest path graphs for two input graphs. We use the same kernel definition in [7].

Node kernel

Node kernel K_{node} compares the node labels of two input graphs. The kernel value is the total number of same node labels:

$$K_{node}(g_1, g_2) = \sum_{n_1 \in g_1.V, n_2 \in g_2.V} K_{label}(n_1, n_2) \quad (5)$$

where K_{label} is defined as:

$$K_{label}(n_1, n_2) = \begin{cases} 1, & \text{if } label(n_1) = label(n_2) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

[4]We modify original all-pair shortest path algorithm used in [7] to directed acyclic graph version to reduce time complexity.

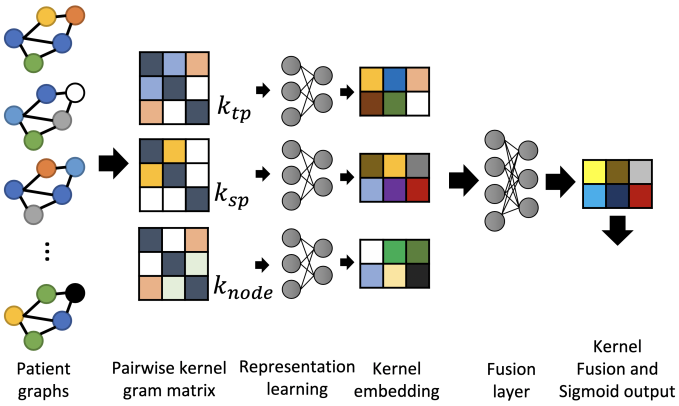


Figure 6: Multiple graph kernel fusion

3.5 Positive Definiteness

Here, we prove all the above kernels are positive definite. K_{tp} is positive definite since the transformation of exponential function of euclidean distance is still positive definite [8]. K_{sp} is already proven to be positive definite in [7]. Finally, K_{node} is positive definite since the K_{label} is a dirac delta function which is proven to be positive definite in [31], and it is known that positive definiteness is closed under addition on positive definite kernels.

3.6 Multiple kernel fusion architecture

To capture multi-view characteristics on patient graphs, we use two additional kernels; shortest path kernel and node kernel, in conjunction with our proposed temporal proximity kernel and find the best combination of them in an end-to-end manner. Specifically, temporal proximity kernel K_{tp} focuses on temporal similarity between substructure such as node ordering and their time difference, shortest path kernel K_{sp} aims to capture similarity in overall connection, and node kernel K_{node} offers a balance between local and global similarity by comparing all node labels between two patient graphs to achieve best accuracy as well as prevent overfitting from noise collaboratively by kernels. The architecture is described in Figure 6.

Given gram matrices on all pair of n graphs for each kernel type $K_t \in R^{n \times n}$ where $K_{t\,g_i,g_j} = k_t(g_i, g_j)$ and $t \in \{tp, sp, node\}$, we use a Multi-layer perceptron (MLP) to generate the corresponding kernel representation $g_{emb_t} \in R^{n \times m}$ where $m \ll n$. In this case, each row i in K_t represents a high-dimensional feature vector with each dimension being a kernel value (e.g., similarity score) between its associated graph g_i and all other graphs, and its kernel embedding g_{emb_t} can be treated as a dimension reduction by using traditional kernel approximation technique [30, 35, 41] to generate low dimensional features for g_i such that efficient linear classifier can be used directly. $g_{i_t} \in R^n$ is converted to $g_{emb_t} \in R^m$ under kernel type t as follows:

$$g_{emb_t} = ReLU(W_t g_{i_t} + b_t) \tag{7}$$

by using the kernel embedding weight matrix $W_t \in R^{m \times n}$ and the bias vector $b_t \in R^m$ where n is the number of input graphs, and m is the dimension for the embedding space. The rectified linear unit (ReLU) activation is defined as $ReLU(val) = max(val, 0)$. For deep architecture, we can compute the layer l with its previous layer $l-1$ with related parameters W_{t_l} and b_{t_l} within layer by using the same way that we compute the embedding for input kernel gram matrix such as:

$$g_{emb_{t_l}} = ReLU(W_{t_l} g_{emb_{t_{l-1}}} + b_{t_l}) \tag{8}$$

For combining three kernels, we first average their embedding from last layer and use another dense layer with ReLU activation that learns the kernel fusion $g_{emb_F} \in R^f$:

$$\begin{aligned} g_{emb_{sum}} &= \sum_{t \in \{tp, sp, node\}} g_{emb_{t_{last}}} \\ g_{emb_{avg}} &= \frac{g_{emb_{sum}}}{3} \\ g_{emb_F} &= ReLU(W_F g_{emb_{avg}} + b_F) \end{aligned} \tag{9}$$

in which $W_F \in R^{f \times q}$ is the fusion weight matrix with fusion embedding dimension f and the bias vector $b_F \in R^f$ assuming the last embedding layer dimension is q.

Finally, the label of success or failure for g_{emb_F} is produced by using Sigmoid layer defined as:

$$\hat{y} = Sigmoid(W_p g_{emb_F} + b_p) \tag{10}$$

where $W_p \in R^{1 \times f}$ and $b_p \in R$ are trainable weight used to generate class label $\hat{y} \in \{0, 1\}$. We also use binary cross-entropy loss function to optimize the best embedding under the fusion setting to learn all kernel embedding weight matrices.

3.7 Interpretation

As we stated in Section 3.6, each row (e.g., each patient) depicts a high dimensional feature vector with each dimension corresponding a kernel value to a specific training example. Since the kernel value can be treated as a similarity measurement, we can use the concept in similarity-based classification, in which class labels are inferred by a set of most similar training examples [10], and consult the top k most similar patients to get prediction insights based on the nature such that features with higher weight contribute more to the result in a linear classifier [26]. Kernel embedding, reducing input dimension, for each kernel type facilitates similarity measurement refinement, reducing the number of training examples used to infer. Similar patients with allied graph similarity are grouped into one coordinate (e.g., dimension) in the embedding space.

Since kernel embedding space is trained in an end-to-end manner through ReLU operation in Equation 8, which achieves the interpretability [14], we can select a set of candidates that contribute most to the prediction, via top k value coordinates in the embedding space. The selected ones under different kernel type can be interpreted as multi-view representative cases (e.g., time propagation or disease connection) in case-based learning [23]. In practice, we sort patient g_{emb_t} in kernel embedding space and pick up top k coordinates. We then select top k' training examples for the i-th coordinate in top k coordinates. We illustrate the interpretation process under k_{tp} in Figure 7. All sorts are in a reverse order:

$$argsort(g_{emb_t})[1:k] \tag{11}$$

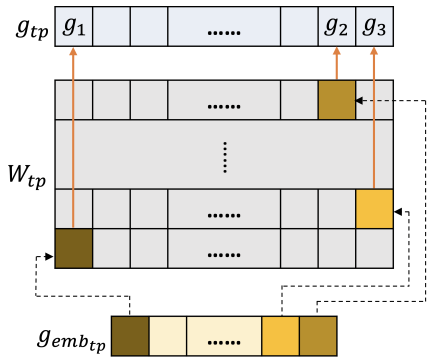


Figure 7: Interpretation steps. Given an embedded patient vector $g_{emb_{tp}}$, we sort it in a descending order and select top 3 value dimension and find corresponding training examples in g_{tp}, which contribute most, through weight matrix W_{tp}.

$$argsort(W_t[i,:])[1:k'] \tag{12}$$

4 EXPERIMENTS

4.1 Dataset

Data Source

We use a subset of the Taiwanese National Health Insurance Research Database (NHIRD) [5] as our data source. Our sample contains over a 20-year, complete, medical history for one-million randomly sampled patients. The database is provided by the National Health Insurance Administration and the Ministry of Health and Welfare. NHIRD is composed of registration files and original claim data for the hospitals that participate in the National Health Insurance (NHI) program for reimbursement. The International Classification of Diseases, 9th Revision, Clinical Modification (ICD9-CM) code indicates the diagnosed disease. The unique identifier is used for drug prescription and can be further linked to the Anatomical Therapeutic Chemical (ATC) code. All personal information are de-identified. Institutional Review Board (IRB) approvals for our research were granted by all associated institutions.

Diseases

Four diseases, namely, pneumonia, acute otitis media, acute cystitis, and urinary tract infection, are studied. Their treatments primarily rely on drug prescriptions (e.g., antibiotics), and the effectiveness for treatment depends on what stage of the disease the patient is at, with early treatment effecting recovery. The goal is to predict the success or failure of the given drug prescription for disease diagnosis. For creating patient cases, a 1 month observation window is established after the drug is prescribed. For each patient, 2 months of medical history is included prior to their diagnosis. Table 1 summarizes the dataset statistic.

[5]https://nhird.nhri.org.tw/en/

Table 1: Disease Data Statistic

Disease	# of patient	# of failure	# of success
Pneumonia	37,677	12,439	25,238
Acute otitis media	40,008	14,999	25,009
Acute cystitis	113,513	35,728	77,785
Urinary tract infection	279,645	94,105	185,540

4.2 Experimental Setup

Baselines

We selected 14 deep learning and 3 traditional approaches as our baselines[6].

Deep learning approaches:

- Deep Patient [25]. Deep Patient learns an EHR unsupervised representation through three-layer stack denoising autoencoder with Random Forest used as classifier to predict future diagnosis.
- LSTM [6]. This model uses LSTM to classify medical code diagnosis given the time series clinical measurements. Word embedding is used to embed medical code before feeding to LSTM.
- Med2Vec [14]. Med2Vec tries to learn interpretable code and visit representations from EHR by using multi-layer perceptron and uses the current visit information to predict medical codes in the following visit.
- Doctor AI [13]. Doctor AI uses Gate Recurrent Unit to learn representation for patient status at each timestamp to make multilabel predictions. We connect a timestamp to the most recent diagnosis time and change the softmax layer to sigmoid layer.
- Retain [16]. This is an interpretable model to predict the future diagnosis of heart failure via two-level RNN attention model incorporated with reverse time attention mechanism. By using attention, influential past visits which contribute to the final prediction can be selected.
- CNN [9]. This model uses word embedding to learn medical code embedding from raw EHRs and transfer each visit into a fixed dimension vector. The multi-layer CNN is introduced to capture local and short temporal dependency in EHRs for risk prediction.
- Temporal Fusion CNN [12]. Four types of CNNs, namely Single-frame (S), Early Fusion (EF), Late Fusion (LF), and Slow Fusion (SF) are proposed to extract phenotypes from patient EHR represented as a temporal event matrix.
- Deepr [27]. Deepr uses CNN to learn and detect meaningful clinical motifs from EHR to predict unplanned readmission. In their work, EHRs are transformed into a sentence where each medical event is represented as a phrase and connected with each other by special keywords as a time gap.
- Dipole [22]. In their work, bidirectional RNN with three different attention mechanism is proposed. The three attention

[6]All word embedding is performed by Word2Vec [24].

Table 2: Performance comparison

	Pneumonia			Acute otitis media			Acute cystitis			Urinary tract infection		
Model	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1
MGKF	**0.7056**	**0.6744**	**0.6583**	**0.6912**	**0.6920**	**0.6351**	**0.7201**	**0.7200**	**0.6883**	**0.7249**	**0.7224**	**0.7202**
Deep Patient	0.5184	0.6258	0.3829	0.6076	0.6051	0.5747	0.5874	0.6400	0.4994	0.5714	0.6017	0.5178
LSTM	0.5650	0.6074	0.4249	0.5929	0.5912	0.6031	0.5986	0.6184	0.5106	0.5841	0.5877	0.5457
Doctor AI	0.4689	0.5250	0.2963	0.6054	0.6046	0.5968	0.6008	0.6233	0.5066	0.6286	0.6350	0.4511
CNN	0.5616	0.6171	0.3932	0.6036	0.6048	0.5837	0.6085	0.6372	0.5014	0.5975	0.6055	0.5407
Fusion CNN-S	0.5655	0.6308	0.3804	0.6053	0.6070	0.5783	0.6127	0.6437	0.5019	0.6000	0.6087	0.5408
Fusion CNN-EF	0.5596	0.6142	0.3932	0.6042	0.6064	0.5660	0.6096	0.6355	0.5070	0.5992	0.6088	0.5356
Fusion CNN-SF	0.5718	0.6279	0.4045	0.6045	0.6067	0.5667	0.6112	0.6429	0.4934	0.6088	0.6222	0.5306
Fusion CNN-LF	0.5667	0.6476	0.3329	0.6165	0.6195	0.5738	0.6130	0.6550	0.4597	0.6133	0.6286	0.5169
Med2Vec	0.5506	0.6308	0.3134	0.6174	0.6204	0.5537	0.6040	0.6493	0.4382	0.6028	0.6224	0.4777
Retain	0.5559	0.5800	0.4531	0.6176	0.6183	0.5970	0.6073	0.6576	0.4403	0.6027	0.6170	0.5030
Deepr	0.5400	0.6180	0.2204	0.6081	0.6113	0.5639	0.5996	0.6509	0.4242	0.6073	0.6266	0.4922
Dipole-g	0.5802	0.5860	0.4759	0.6031	0.6040	0.5775	0.5999	0.6436	0.4434	0.5982	0.6131	0.5060
Dipole-c	0.5668	0.5900	0.4533	0.5943	0.5959	0.5716	0.5994	0.6351	0.4689	0.5936	0.6024	0.5280
Dipole-l	0.5357	0.5520	0.4455	0.6025	0.6049	0.5709	0.5973	0.6367	0.4604	0.5926	0.6051	0.5096
GRNN-HA	0.5553	0.5709	0.4624	0.5767	0.5763	0.5778	0.5763	0.5824	0.5184	0.5730	0.5730	0.5502
Timeline	0.5458	0.6300	0.2629	0.6200	0.6280	0.5613	0.6470	0.6400	0.5982	0.6022	0.6000	0.6226
Patient2Vec	0.5497	0.6053	0.3785	0.6010	0.6029	0.5672	0.5995	0.6351	0.4729	0.5851	0.5975	0.5059
MCA-RNN	0.6121	0.6532	0.4762	0.6440	0.6464	0.6081	0.6065	0.6457	0.4680	0.6123	0.6263	0.5296
ClinicalBERT	0.5000	0.5089	0.3373	0.5946	0.6018	0.5170	0.5000	0.5855	0.3693	0.5000	0.5089	0.3373
SVM	0.6463	0.6120	0.5369	0.6209	0.5955	0.5035	0.5950	0.6241	0.2809	0.6463	0.6120	0.5369
LR	0.6486	0.6023	0.5328	0.6152	0.5839	0.5791	0.5939	0.5720	0.4991	0.6486	0.6023	0.5328
RF	0.6603	0.6134	0.5874	0.6190	0.5772	0.5344	0.5887	0.6069	0.4372	0.6603	0.6134	0.5874

mechanisms, namely, general (g), concatenation-based (c), and location-based (l) are used to calculate attention weight for each patient visit.

- GRNN-HA [32]. This model introduces a hierarchical attention network to learn attention weight from medical code level to patient visit level.
- MCA-RNN [20]. An attention-based contextual RNN is used, and patient information (e.g., demographic) is derived from conditional variational autoencoders. They combine the contextual features with RNN by using medical context attention to generate final representation to make prediction.
- Timeline [5]. Timeline is an interpretable model with attention mechanism to learn time decay factors for every medical code and improves visit embedding. By analyzing attention and disease propagation functions, Timeline provides interpretation for prediction and insights on how future risks are changed over time.
- Patient2Vec [40]. Patient2Vec proposes a hierarchical representation learning framework to capture complex relationships between medical events in EHR with the attention mechanism used to learn personalized representation for patient.
- ClinicalBERT [7] [2]. The pre-trained BERT model is used to learn embedding for clinical text and performed on clinical natural language processing tasks.

[7]In our task, patient is represented as a document containing all medical codes.

Traditional approaches [8]:

- Linear Support Vector Machine (SVM).
- Logistic Regression (LR).
- Random Forest (RF).

Implementation Detail

We use Keras with Tensorflow backend to build our model. We set 1000 dimensions for dense embedding layer of each kernel and 500 dimensions for kernel fusion layer. Between the dense embedding layer and kernel fusion layer, we setup 3-layer neural network with size 800, 600, and 500 to learn the deep representation for each kernel. We also use dropout for each layer with the fine tuning rate except for the final fusion layer which is set to 0.9. All these parameters were empirically determined. For the training stage, we use the adam optimizer with 128 batch size to optimize the binary cross-entropy loss and train for 10 epochs. All the experiments are executed on an Intel Core i7, with 64GB memory and one Nvidia 1080 Ti GPU.

Evaluation Metrics

We use accuracy (ACC), F1-score (F1), and the area under the receiver operating characteristic curve (AUROC) as our evaluation metrics. For each disease, we divide our datasets into training, validation, and testing in an 80:10:10 ratio. All parameters for all models and dropout rate in our proposed model are fine tuned via 10-fold

[8]All traditional approaches use word embedding to embed medical code into 256 dimension vector.

cross validation on the validation set. We repeat all experiments 100 times and report their best performance scores. The pairwise t-test is used with p-value set to 0.05 to reject the null-hypothesis to test the statistical significance of our proposed method statistically. We find that our solution statistically significantly differs from previous efforts.

4.3 Experimental Results

Results shown in Table 2 illustrate that our proposed method (MGKF) outperforms all baseline approaches by a large margin. We are surprised that most of the deep learning approaches failed to yield better results than traditional methods. We surmise that the possible reason might be the characteristic difference between development datasets. For most deep learning models, their datasets are primarily collected from regional sources such as local hospitals or private healthcare data partnerships where uniform patient population type is expected. Data cleaning and normalization is often conducted. For models developed on a widely used public open dataset, namely MIMIC3, the primary difficulty is the relatively few history records for each patient, diminishing the detection on long term medical information. On the contrary, NHIRD is a national-wide, in production, and clinical usage dataset with complete medical history, mirroring true medical practice from daily clinical activities. High variance and noise is inevitable.

In NHIRD, some clinical visit records are for reimbursement or request for refill of prescription purposes. Due to the system limitation for maximum number of drugs allowed to store in one record, physicians split prescription orders into multiple records. The diagnosis or drug codes in such cases are pointless, which prevents deep learning models from learning event sequence patterns from those pointless events. We can see all deep learning models especially Doctor AI, Deepr, Med2Vec, and Timeline, where code level representation learning acts as the major part, performing poorly on F1 for pneumonia since it is one of the most frequent diseases that uses record splitting for reimbursement purposes. The attention mechanism on visit level (e.g., Retain, Dipole, GRNN-HA, and MCA-RNN) eases the effect by memorizing relevant visits toward classification result. Although Timeline introduces time decay factors with attention, the capricious medical records listed in those pointless events may lead to overfitting in Timeline. Another concern, the patient hospital-shopping habit in Taiwan generates lots of sequences for the same disease diagnosis. Distinguishing the event sequence originated from a hospital-shopping habit from a true medical condition is difficult, causing RNN-based models to overfit.

On the other hand, traditional approaches are shallow architectures that usually fit the data in a simple manner without much representation learning process (e.g., LR and SVM). They are easy to interpret and avoid severe overfitting in high noise environments as compared to deep architectures. Thus, our approach does not rely on data representation learning, reducing the likelihood of falling into NHIRD-driven potential pitfalls. Also, with the help of representation learning on multi-view kernel value (e.g., similarity measurement), the reduced dimension keeps the most relevant consulting cases and filters noise such as hospital-shopping events. We explain this further in Section 4.4.

4.4 Interpreting Results

We select two patients who had a successful and failure treatment for pneumonia respectively. All patients from top coordinate under all kernel types are selected [9] following interpretation steps in Section 3.7. We show top 5 weighted patients with their kernel value between 2 selected patients from each coordinate in Table 3, and two selected patient graphs in Figure 8.

For patient level interpretation, it is simple to see patient's disease progression using an EHR graphical representation. Patient 10 failed to treat Pneumonia at visit 7 while patient 19 is successful at visit 4. The medical event ordering and their connection is straightforward for the treating physician to understand. We also see that similar patients, whose treatment is successful, under all kernel types do not affect the prediction for patient 10 since the weight contribution for success training examples are reduced by K_{sp} and K_{node}; top failure patients from coordinate 365 in K_{sp} and coordinate 499 in K_{node} balance the final weight [10]. Also, for patient 19 whose treatment is a success, the node similar patients from coordinate 499 in K_{node} do not affect the prediction because of the contribution weight reduced by K_{tp} and K_{sp}. Each coordinate in the kernel embedding space reveals similarity patient group, where top weighted training examples in each coordinate provide contribution to prediction, diminishing the effects from pointless events described in Section 4.3. The representative cases, which are denoted by top weighted training examples, provide insights on how overall treatment and disease progression look like for success and failure outcomes. These results demonstrate how multiple graph kernel fusion with multiple layer embedding prevents the model from overfitting due to noise and offers interpretation.

5 CONCLUSION

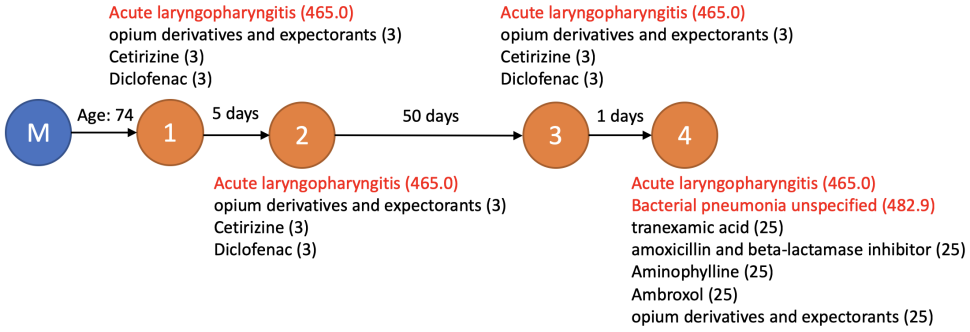
Sparsity, temporal relationships, and heterogeneity challenge the development of a predictive model on patient EHRs. The bias among different population groups also provides a gap between model implementation and utilization. Those are issues that must be addressed in model development. It is a trade-off to develop a model that achieves high accuracy versus high interpretability, for example, recurrent neural network versus logistic regression solutions. To balance the issue, current approaches try to add interpretability via different ways of attention mechanism to existing deep neural networks. This, however, increases the complexity of the models themselves and may result in overfitting issues when performing those models on different population based datasets (e.g., NHIRD).

Consequently, we proposed a model, Multiple Graph Kernel Fusion, that achieves both high accuracy and interpretability to predict the success or failure of drug prescription. We presented a deep learning approach where the prediction task uses a graphical represented EHR without the need of electronic phenotyping or representation learning. Three of our proposed kernel functions capture different aspects from patient graph structures that provide meaningful insights for clinical practice. The multiple graph kernel fusion with the help of deep neural networks helps the prediction task to refine and concentrate on the most relevant and similar patients to prevent overfitting on noisy data via kernel embedding.

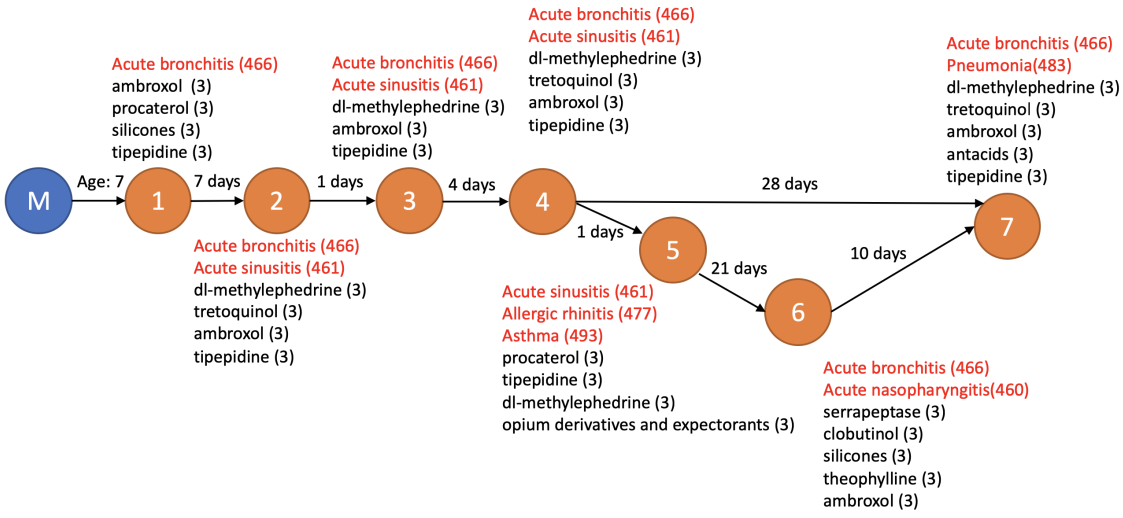
[9] Due to space limitations we only select maximum value coordinate.
[10] The ranking for training examples is ordered by weight in their belonging coordinate.

Table 3: Coordination information of two selected patients

Patient/Kernel type	k_{tp}		k_{sp}		k_{node}	
Coordinate number	11		365		499	
	Patient Number	Kernel Value	Patient Number	Kernel Value	Patient Number	Kernel Value
Patient 10 Failure	257 Success	10.946	96 Failure	17.291	546 Success	28.322
	193 Failure	15.362	747 Success	7.957	523 Failure	2.173
	332 Success	4.955	415 Success	4.252	733 Failure	3.462
	589 Success	2.901	558 Failure	0.626	654 Success	1.765
	102 Success	4.579	112 Failure	17.460	35 Success	2.221
Coordinate number	304		218		499	
	Patient Number	Kernel Value	Patient Number	Kernel Value	Patient Number	Kernel Value
Patient 19 Success	207 Success	3.604	470 Success	6.691	546 Success	20.930
	110 Success	22.351	494 Success	17.284	523 Failure	3.808
	343 Success	3.304	4 Success	13.011	733 Failure	8.773
	222 Success	10.021	297 Success	0.179	654 Success	5.347
	532 Failure	10.081	337 Failure	30.395	35 Success	3.404



(a) Patient 19



(b) Patient 10

Figure 8: Two sample patient graphs. Due to space limitation, instead of displaying connection between disease nodes and drug nodes, we show drug information by text. For each medical event, red color denotes disease diagnosis following a ICD9-CM code and black color denotes drug prescription with prescribed days within parentheses.

The classification performance of drug prescription success/failure shown by experimental results surpasses all evaluated approaches. The interpretation is simple, and the medical clinician can put more attention on the most relevant cases for the given patient.

Overall, we have shown that our described approach has the ability to predict the outcome of drug prescription with the high scores in all evaluation metrics with high interpretability. It was also reviewed by a medical clinician to confirm that our proposed approach is able to predict the failure of a drug used for a specific diagnosis and identify which drug prescription path to pursue. The approach is now in limited clinical use.

REFERENCES

[1] Hannah Alphs-Jackson, John Cashy, Ophir Frieder, and Anthony J Schaeffer. 2011. Data mining derived treatment algorithms from the electronic medical record improve theoretical empirical therapy for outpatient urinary tract infections. *The Journal of urology* 186, 6 (2011), 2257–2262.

[2] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. 72–78.

[3] JK Aronson. 2009. Medication errors: what they are, how they happen, and how to avoid them. *QJM : monthly journal of the Association of Physicians* 102, 8 (2009), 513–521.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[5] Tian Bai, Shanshan Zhang, Brian L Egleston, and Slobodan Vucetic. 2018. Interpretable representation learning for healthcare via capturing disease progression through time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 43–51.

[6] Jacek M Bajor and Thomas A Lasko. 2016. Predicting medications from diagnostic codes with recurrent neural networks. (2016).

[7] K. M. Borgwardt and H. P. Kriegel. 2005. Shortest-path kernels on graphs, In Fifth IEEE International Conference on Data Mining (ICDM'05). *Fifth IEEE International Conference on Data Mining (ICDM'05)*, 8 pp.–. https://doi.org/10.1109/ICDM.2005.132

[8] Der-Chen Chang, Ophir Frieder, and Hao-Ren Yao. 2018. On Bochner's Theorem and Its Application to Graph Kernels. *Journal of Nonlinear and Convex Analysis* 19, 12 (2018).

[9] Zhengping Che, Yu Cheng, Zhaonan Sun, and Yan Liu. 2017. Exploiting convolutional neural network for risk prediction with medical feature embedding. *arXiv preprint arXiv:1701.07474* (2017).

[10] Yihua Chen, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti. 2009. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research* 10, Mar (2009), 747–776.

[11] Yihua Chen, Maya R Gupta, and Benjamin Recht. 2009. Learning kernels from indefinite similarities. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 145–152.

[12] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. 2016. Risk prediction with electronic health records: A deep learning approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 432–440.

[13] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*. 301–318.

[14] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. 2016. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1495–1504.

[15] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. 2017. GRAM: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 787–795.

[16] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*. 3504–3512.

[17] Swarnendu Ghosh, Nibaran Das, Teresa Gonçalves, Paulo Quaresma, and Mahantapas Kundu. 2018. The journey of graph kernels through two decades. *Computer Science Review* 27 (2018), 88–111.

[18] Alejandra Hurtado-de Mendoza, Adriana Serrano, Qi Zhu, Kristi Graves, Nicole Fernández, Aileen Fernández, Paola Rodriguez-de Liebana, Valeria Massarelli, Claudia Campos, Florencia González, et al. 2018. Engaging Latina breast cancer survivors in research: building a social network research registry. *Translational behavioral medicine* 8, 4 (2018), 565–574.

[19] Seokho Kang, Pilsung Kang, Taehoon Ko, Sungzoon Cho, Su-jin Rhee, and Kyung-Sang Yu. 2015. An efficient and effective ensemble of support vector machines for anti-diabetic drug failure prediction. *Expert Systems with Applications* 42, 9 (2015), 4265–4273.

[20] Wonsung Lee, Sungrae Park, Weonyoung Joo, and Il-Chul Moon. 2018. Diagnosis Prediction via Medical Context Attention Networks Using Deep Generative Modeling. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1104–1109.

[21] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[22] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1903–1911.

[23] Susan F McLean. 2016. Case-based learning and its application in medical and health-care fields: a review of worldwide literature. *Journal of Medical Education and Curricular Development* 3 (2016), JMECD–S20377.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[25] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. 2016. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports* 6 (2016), 26094.

[26] Dunja Mladenić, Janez Brank, Marko Grobelnik, and Natasa Milic-Frayling. 2004. Feature selection using linear classifier weights: interaction with classification models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 234–241.

[27] P. Nguyen, T. Tran, N. Wickramasinghe, and S. Venkatesh. 2017. `Deepr`: A Convolutional Net for Medical Records. *IEEE Journal of Biomedical and Health Informatics* 21, 1 (Jan 2017), 22–30. https://doi.org/10.1109/JBHI.2016.2633963

[28] Phung Anh Nguyen, Shabbir Syed-Abdul, Usman Iqbal, Min-Huei Hsu, Chen-Ling Huang, Hsien-Chang Li, Daniel Livius Clinciu, Wen-Shan Jian, and Yu-Chuan Jack Li. 2013. A probabilistic model for reducing medication errors. *PloS one* 8, 12 (2013), e82401.

[29] Jyotishman Pathak, Abel N Kho, and Joshua C Denny. 2013. Electronic health records-driven phenotyping: challenges, recent advances, and perspectives.

[30] Ali Rahimi and Benjamin Recht. 2008. Random features for large-scale kernel machines. In *Advances in neural information processing systems*. 1177–1184.

[31] Bernhard Scholkopf and Alexander J Smola. 2001. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[32] Ying Sha and May D Wang. 2017. 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*. ACM, 233–240.

[33] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. 2018. Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE journal of biomedical and health informatics* 22, 5 (2018), 1589–1604.

[34] Huan Song, Jayaraman J Thiagarajan, Prasanna Sattigeri, Karthikeyan Natesan Ramamurthy, and Andreas Spanias. 2017. A deep learning approach to multiple kernel fusion. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2292–2296.

[35] Christopher KI Williams and Matthias Seeger. 2001. Using the Nyström method to speed up kernel machines. In *Advances in neural information processing systems*. 682–688.

[36] Cao Xiao, Ping Zhang, W Art Chaovalitwongse, Jianying Hu, and Fei Wang. 2017. Adverse drug reaction prediction with symbolic latent Dirichlet allocation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[37] Pranjul Yadav, Michael Steinbach, Vipin Kumar, and Gyorgy Simon. 2017. Mining Electronic Health Records: A Survey (https://dl.acm.org/citation.cfm?id=3127881). *Comput. Surveys* 50 (02 2017). https://doi.org/10.1145/3127881

[38] Hao-Ren Yao, Der-Chen Chang, Ophir Frieder, Wendy Huang, and Tian-Shyug Lee. 2019. Graph Kernel Prediction of Drug Prescription. In *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI) (IEEE BHI 2019)*. Chicago, USA.

[39] Andrew Yates, Nazli Goharian, and Ophir Frieder. 2015. Extracting Adverse Drug Reactions from Social Media. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

[40] Jinghe Zhang, Kamran Kowsari, James H Harrison, Jennifer M Lobo, and Laura E Barnes. 2018. Patient2Vec: A Personalized Interpretable Deep Representation of the Longitudinal Electronic Health Record. *IEEE Access* 6 (2018), 65333–65346.

[41] Kai Zhang and James T Kwok. 2009. Density-weighted Nyström method for computing large kernel eigensystems. *Neural Computation* 21, 1 (2009), 121–146.