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:

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 Records (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
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 Forrest as downstream classifiers. 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

1We 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, so as 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.

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 $M = \{ (m_1, t_1), \ldots, (m_n, t_n) \}$ 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.

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. 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.

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

\footnote{To simplify model assumption, we only use gender and age as demographic information.}

\footnote{We follow the same setting in [38].}
where 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.

### 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 \( T = \{ n_i | i = 1, \ldots, |V| \} \), the topological sequence \( S \) is defined as

\[
S = \{ n_i.label + level | i = 1, \ldots, |V|, \text{ and } n_i \in T \}
\]

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

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 \( t\_p_1 = \{v_{11}, \ldots, v_{1m}\} \) where

\[
v_{1j} = \begin{cases} d_j, & \text{if } S[j] \in S_1 \\ -1, & \text{otherwise} \end{cases}
\]

and define the temporal signature for \( g_2 \) as \( t\_p_2 = \{v_{21}, \ldots, v_{2m}\} \) where

\[
v_{2j} = \begin{cases} d_j, & \text{if } S[j] \in S_2 \\ -1, & \text{otherwise} \end{cases}
\]

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 \( t\_p_1, t\_p_2 \). 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 temporal signature \( t\_p_1, t\_p_2 \) as:

\[
K_{tp}(g_1, g_2) = e^{-||t\_p_1 - t\_p_2||}
\]

where \( ||t\_p_1 - t\_p_2|| \) is the Euclidean distance between \( t\_p_1, t\_p_2 \).

#### Shortest path kernel

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:

\[
K_{node}(g_1, g_2) = \sum_{n_1 \in g_1, n_2 \in g_2} K_{label}(n_1, n_2)
\]

where \( K_{label} \) is defined as:

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

---

*We modify original all-pair shortest path algorithm used in [7] to directed acyclic graph version to reduce time complexity.*
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 \( K_i \in R^{n \times n} \) where \( K_{t,g,g} = K_{1}(g_i, g_j) \) and \( t \in \{tp, sp, node\} \), we use a Multi-layer perceptron (MLP) to generate the corresponding kernel representation \( g_{emb} \in R^{m \times m} \) where \( m \ll n \). In this case, each row in \( K_i \) 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 \( g_{emb} \) 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. \( g_{emb} \in R^{m} \) is converted to \( g_{emb} \in R^{m} \) under kernel type \( t \) as follows:

\[
g_{emb} = ReLU(W_{g}g_{i} + b_{t})
\]

by using the kernel embedding weight matrix \( W_{g} \in R^{m \times n} \) 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 \( W_{l} \) and \( b_{l} \) within layer by using the same way that we compute the embedding for input kernel gram matrix such as:

\[
g_{emb_{t_{l}}} = ReLU(W_{t_{l}}g_{emb_{t_{l-1}}} + b_{t_{l}})
\]

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 \( g_{emb_{f}} \in R^{f} \):

\[
g_{emb_{sum}} = \frac{1}{3} \sum_{t \in \{tp, sp, node\}} g_{emb_{last}}
\]

\[
g_{emb_{avg}} = ReLU(W_{f}g_{emb_{avg}} + b_{f})
\]

in which \( W_{f} \in R^{l \times q} \) 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:

\[
\hat{y} = \text{Sigmoid}(W_{p}g_{emb_{f}} + b_{p})
\]

where \( W_{p} \in R^{l \times f} \) 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 \( g_{emb} \) 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:

\[
\text{argsort}(g_{emb_{i}})[1 : k]
\]
Figure 7: Interpretation steps. Given an embedded patient vector \( \mathbf{g}_{\text{emb}} \), we sort it in a descending order and select top 3 value dimension and find corresponding training examples in \( \mathbf{g}_{tp} \), which contribute most, through weight matrix \( \mathbf{W}_{tp} \).

\[
\text{argsort}(\mathbf{W}_t[i,:][1:k'])
\] (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 (ICD-9-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.

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

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

---

\(^5\)https://nhird.nhri.org.tw/en/

\(^6\)All word embedding is performed by Word2Vec [24].
Table 2: Performance comparison

<table>
<thead>
<tr>
<th>Model</th>
<th>Pneumonia</th>
<th>Acute otitis media</th>
<th>Acute cystitis</th>
<th>Urinary tract infection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>ACC</td>
<td>F1</td>
<td>AUC</td>
</tr>
<tr>
<td>MGKF</td>
<td>0.7056</td>
<td>0.6744</td>
<td>0.6583</td>
<td>0.6912</td>
</tr>
<tr>
<td>Deep Patient</td>
<td>0.5184</td>
<td>0.6258</td>
<td>0.3829</td>
<td>0.6076</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.5650</td>
<td>0.6074</td>
<td>0.4249</td>
<td>0.5929</td>
</tr>
<tr>
<td>Doctor AI</td>
<td>0.4689</td>
<td>0.5250</td>
<td>0.2963</td>
<td>0.6054</td>
</tr>
<tr>
<td>CNN</td>
<td>0.5161</td>
<td>0.6171</td>
<td>0.3932</td>
<td>0.6036</td>
</tr>
<tr>
<td>Fusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN-S</td>
<td>0.5655</td>
<td>0.6308</td>
<td>0.3804</td>
<td>0.6053</td>
</tr>
<tr>
<td>Fusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN-EF</td>
<td>0.5596</td>
<td>0.6142</td>
<td>0.3932</td>
<td>0.6042</td>
</tr>
<tr>
<td>Fusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN-SF</td>
<td>0.5718</td>
<td>0.6279</td>
<td>0.4045</td>
<td>0.6045</td>
</tr>
<tr>
<td>Fusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN-LF</td>
<td>0.5667</td>
<td>0.6476</td>
<td>0.3329</td>
<td>0.6165</td>
</tr>
<tr>
<td>Med2Vec</td>
<td>0.5506</td>
<td>0.6308</td>
<td>0.3134</td>
<td>0.6174</td>
</tr>
<tr>
<td>Retain</td>
<td>0.5559</td>
<td>0.5800</td>
<td>0.4531</td>
<td>0.6176</td>
</tr>
<tr>
<td>Deepr</td>
<td>0.5400</td>
<td>0.6180</td>
<td>0.2204</td>
<td>0.6081</td>
</tr>
<tr>
<td>Dipole-g</td>
<td>0.5802</td>
<td>0.5860</td>
<td>0.4759</td>
<td>0.6031</td>
</tr>
<tr>
<td>Dipole-c</td>
<td>0.5668</td>
<td>0.5900</td>
<td>0.4533</td>
<td>0.5943</td>
</tr>
<tr>
<td>Dipole-l</td>
<td>0.5357</td>
<td>0.5520</td>
<td>0.4455</td>
<td>0.6025</td>
</tr>
<tr>
<td>GRNN-HA</td>
<td>0.5553</td>
<td>0.5709</td>
<td>0.4624</td>
<td>0.5767</td>
</tr>
<tr>
<td>Timeline</td>
<td>0.5458</td>
<td>0.6300</td>
<td>0.2629</td>
<td>0.6200</td>
</tr>
<tr>
<td>Patient2Vec</td>
<td>0.5497</td>
<td>0.6053</td>
<td>0.3785</td>
<td>0.6010</td>
</tr>
<tr>
<td>MCA-RNN</td>
<td>0.6121</td>
<td>0.6532</td>
<td>0.4762</td>
<td>0.6440</td>
</tr>
<tr>
<td>ClinicalBERT</td>
<td>0.5000</td>
<td>0.5089</td>
<td>0.3373</td>
<td>0.5946</td>
</tr>
<tr>
<td>SVM</td>
<td>0.6463</td>
<td>0.6120</td>
<td>0.5369</td>
<td>0.6209</td>
</tr>
<tr>
<td>LR</td>
<td>0.6468</td>
<td>0.6023</td>
<td>0.5328</td>
<td>0.6152</td>
</tr>
<tr>
<td>RF</td>
<td>0.6603</td>
<td>0.6134</td>
<td>0.5874</td>
<td>0.6190</td>
</tr>
</tbody>
</table>

Traditional approaches

• 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 cross-validation.

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

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 to avoid severe overfitting on noisy data via kernel embedding. The ranking for training examples is ordered by weight in their belonging coordinate. The top weighted training examples in each coordinate provide insights on the contribution weight 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. Also, 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 representation learning model on electronic medical record which learns the effect 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 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. 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 \( K_{sp} \) 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.

Due to space limitations we only select maximum value coordinate.

The ranking for training examples is ordered by weight in their belonging coordinate.
Table 3: Coordination information of two selected patients

<table>
<thead>
<tr>
<th>Patient/Kernel type</th>
<th>$k_{tp}$</th>
<th>$k_{sp}$</th>
<th>$k_{node}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinate number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patient 10 Failure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>257 Success</td>
<td>10.946</td>
<td>17.291</td>
<td>28.322</td>
</tr>
<tr>
<td>193 Failure</td>
<td>15.362</td>
<td>7.957</td>
<td>2.173</td>
</tr>
<tr>
<td>332 Success</td>
<td>4.955</td>
<td>4.252</td>
<td>3.462</td>
</tr>
<tr>
<td>589 Success</td>
<td>2.901</td>
<td>0.626</td>
<td>1.765</td>
</tr>
<tr>
<td>102 Success</td>
<td>4.579</td>
<td>17.460</td>
<td>2.221</td>
</tr>
<tr>
<td>Coordinate number</td>
<td>304</td>
<td>218</td>
<td>499</td>
</tr>
<tr>
<td>Patient 19 Success</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>207 Success</td>
<td>3.604</td>
<td>6.691</td>
<td>20.930</td>
</tr>
<tr>
<td>110 Success</td>
<td>22.351</td>
<td>17.284</td>
<td>3.808</td>
</tr>
<tr>
<td>343 Success</td>
<td>3.304</td>
<td>13.011</td>
<td>8.773</td>
</tr>
<tr>
<td>222 Success</td>
<td>10.021</td>
<td>0.179</td>
<td>5.347</td>
</tr>
<tr>
<td>532 Failure</td>
<td>10.081</td>
<td>30.395</td>
<td>3.404</td>
</tr>
</tbody>
</table>

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.


