Deep Learning Solutions for Classifying Patients on Opioid Use

Zhengping Che[1], Jennifer St. Sauver, PhD[2], Hongfang Liu, PhD[2], Yan Liu, PhD[1]
[1]Department of Computer Science, University of Southern California, Los Angeles, CA
[2]Department of Health Sciences Research, Mayo Clinic, Rochester, MN

Abstract

Opioid analgesics, as commonly prescribed medications used for relieving pain in patients, are especially prevalent in US these years. However, an increasing amount of opioid misuse and abuse have caused lots of consequences. Researchers and clinicians have attempted to discover the factors leading to opioid long-term use, dependence, and abuse, but only limited incidents are understood from previous works. Motivated by recent successes of deep learning and the abundant amount of electronic health records, we apply state-of-the-art deep and recurrent neural network models on a dataset of more than one hundred thousand opioid users. Our models are shown to achieve robust and superior results on classifying opioid users, and are able to extract key factors for different opioid user groups. This work is also a good demonstration on adopting novel deep learning methods for real-world health care problems.

1 Introduction

Opioid analgesics are effective, commonly prescribed medications used for management of both acute and chronic pain in patients with many different medical conditions and following many medical procedures[1,2]. Prescription of opioids in the United States is high, and between 2011 and 2012, nearly 7% of the adult population was estimated to have taken an opioid in the last thirty days[3,4]. However, as reported in several previous studies, these medications do not effectively control pain in all patients[5,6,7], and many patients are at high risk of adverse effects due to these medications[8,7]. A meta-analysis of randomized trials found that 80% of patients treated with opioids for chronic, non-cancer pain experienced at least one adverse event, with symptoms ranging from mild nausea to life-threatening respiratory depression[8]. In addition, the US is experiencing an opioid epidemic. Specifically, opioids are increasingly misused and diverted from their intended recipients, and abuse and overdoses have risen alarmingly in the last ten years[9]. The rate for drug overdose deaths, driven largely by opioid overdose, increased approximately 140% from 2000 to 2014[10]. In 2017, one of the largest pharmaceutical distributors in US was fined a record $150 million for failing to report suspicious orders linked to the opioid addiction epidemic[11]. Indeed, prompt and proper actions need to be taken to achieve a balanced opioid usage strategies, stem the tide of this public health epidemic, and prevent further devastating consequences.

The factors that contribute to opioid use – particularly the patient factors that contribute to long-term, chronic use of these medications and/or dependence or abuse of these drugs are poorly understood. Previous work found significant increases in incident opioid prescriptions for chronic, non-malignant pain between 1997 and 2005 in the Kaiser Permanente and Group Health populations[12]. Additionally, the proportion of the population receiving long-term therapy nearly doubled in the same time frame. The most common indications for long-term use in this study were chronic back pain, extremity pain, and osteoarthritis. Apart from these data, however, little is known about who receives opioid analgesic prescriptions in an average community. Additionally, with the exception of a few studies exploring the role of mental illness, depression, or previous patterns of substance abuse[13,14,15], patient characteristics that might contribute to these adverse outcomes have not been described.

The rapid growth in electronic health record (EHR) adoption provides a wealth of patient information that could help identify patients at high risk of long-term opioid use or dependence. If one predictive or classifying model can leverage such data for analysing opioid usage and/or dependence, that is, the model has the ability to identify patients likely to benefit from or get addicted to these medications and target therapy more appropriately to them, we can expect those models to be able to extract the knowledge of the clinical characteristics associated with the progression of a short-term to an episodic or long-term opioid prescribing pattern and aid in the identification of at-risk patients and provide the basis for developing targeted clinical interventions. In the era of data explosion, however, more powerful data-driven learning models are in urgent demand in order to fully utilize the large amount of EHR data, identify meaningful features for opioid dependence or abuse, provide precise information for clinicians to make early decisions, and ultimately contribute to better personalized health care quality.

In this paper, we utilized state-of-the-art deep learning models on a much larger data set for opioid usage prediction and factor investigation tasks. Deep learning models have brought lots of significant successes including but not limited to recognizing and distinguishing thousands of human faces at a time[16,17], understanding, translating, and generating human languages[18,19], and mastering games and beating top human professional players[20]. Deep learning is also revolutionizing the health care domain with the focuses on a variety of important and challenging tasks, such as computational phenotyping[21,22], predictive modeling[23,24] and medical imaging analyzing[25,26]. It is well known that deep learning solutions equipped with ample computational resources and large-scale datasets are able to go far beyond traditional statistical methods and shed light on intriguing real-world applications in health care. In this paper, we demonstrated our proposed deep learning solutions for identifying opioid user groups and showed that they provided superior classification results and outperformed other widely used learning baseline methods. We validated important factors and risk factors identified by deep learning models with previous clinical studies. Our work also provided a practical example on properly adopting novel deep learning methods for real-world health care problems leveraging large-scale EHR data.

2 Data and Task Descriptions

In this work, we took a cohort of $142\,377$ patients from the Rochester Epidemiology Project (REP)[27]. The total number of people identified by REP, as shown in previous work[28], cover about 98.7% of the population that reside in Olmsted County by the US Census. Thus, this large-scale dataset is well-representative for this population-based study and suitable for powerful and complex deep learning models.

Cohort Selection First, all outpatient drug prescriptions were obtained from Mayo Clinic and the Olmsted Medical Center from January 1, 2003 through March 31, 2016 for patients who authorized the use of their medical records for research purposes. The drug prescriptions were standardized using the 2016 version of RxNorm[29]. We kept the records for all patients who received at least one opioid analgesic prescription between July 1, 2013 and March 31, 2016 and did not have any opioid prescriptions 6 months prior to their first prescriptions within the study period. The analgesic prescriptions were determined by the RxNorm Code, with either National Drug File Reference Terminology (NDF-RT) code $C8834$ (Opioid Analgesics) or ingredient code 10689 (Tramadol) and 352362 (Acetaminophen/Tramadol). In order to remove incorrectly duplicated and modified prescription records, only the last prescription would be kept if same drug prescriptions were made for one patient within 30 minutes. A cohort of $N = 102\,166$ patients was created after these data cleaning and selection steps.

Group Identification All patients were classified into three groups, namely *short-term* users (*ST*), *long-term* users (*LT*), and *opioid-dependent* users (*OD*). ST and LT groups were defined by the CONSORT study[30] and the same as in our previous work[31]. Episodes of opioid prescription lasting longer than 90 days and with 120 or more total days supply or 10 or more prescriptions were classified as long-term ($N_{LT} = 21\,570$), while others were classified as short-term ($N_{ST} = 80\,596$). $N_{OD} = 749$ opioid-dependent patients were further identified by the diagnosis of "opioid dependence" from their problem lists. It is noting that the relatively low identification rate might be due to the fact that only part of dependent patients got explicit diagnosis in the problem lists by doctors. All identified dependent users were validated by clinicians. Table 1 shows detailed data characteristics of each patient group, which also match the finding in our previous related study on a smaller dataset[31]. Two classification tasks were considered in our experiments: 1) whether the patient will become a long-term opioid user or just a short-term opioid user (Task *ST-LT*), and 2) whether a long-term opioid user is an opioid-dependent patient or not (Task *LT-OD*).

3 Methodology

In this section, we first describe our feature extraction and temporal data processing steps. Next, two deep learning models deployed in our study are presented: A deep feed-forward neural network model with multiple hidden layers, and a recurrent neural network model with Long Short-Term Memory which can better model time series data. Several ways used to improve the model performance and obtain important features are discussed, followed by the descriptions of some machine learning baselines. In the following part, we use bold capital letter (e.g., $\boldsymbol{W}$) to refer to matrix variable, bold lowercase letter (e.g., $\boldsymbol{b}$) for vector variable, and unbold letter (e.g., l, D) for scalar, if not specified.

Table 1: Data characteristics of different patient groups.

		Short-Term/*ST*		Long-Term/*LT*[*]		Opioid-Dependent/*OD*		All	
		Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage
Total Number of Patients		80596	78.89%	21570	21.11%	749	–	102166	–
Sex	Men	37981	47.13%	8447	39.16%	345	46.06%	46428	45.44%
	Women	42453	52.67%	13075	60.62%	402	53.67%	55528	54.35%
	Other/Unknown	162	0.20%	48	0.22%	2	0.27%	210	0.21%
Age	≤ 18	5900	7.32%	447	2.07%	0	0.00%	6347	6.21%
	$19 - 29$	13701	17.00%	1311	6.08%	55	7.34%	15012	14.69%
	$30 - 49$	27696	34.36%	5416	25.11%	354	47.26%	33112	32.41%
	$50 - 64$	18027	22.37%	5570	25.82%	245	32.71%	23597	23.10%
	≥ 65	15272	18.95%	8826	40.92%	95	12.68%	24098	23.59%
Race	White	66184	82.12%	19297	89.46%	655	87.45%	85481	83.67%
	Hispanic	4151	5.15%	697	3.23%	34	4.54%	4848	4.75%
	African American	4131	5.13%	898	4.16%	49	6.54%	5029	4.92%
	Asian	3225	4.00%	361	1.67%	3	0.40%	3586	3.51%
	Other/Unknown	2905	3.60%	317	1.47%	8	1.07%	3222	3.15%
Mortality	Dead	4481	5.56%	17075	79.16%	628	83.85%	21556	21.10%
	Alive/Unknown	76115	94.44%	4495	20.84%	121	16.15%	80610	78.90%
Tobacco Use	Never/Unknown	46264	57.40%	7159	33.19%	76	10.15%	53423	52.29%
	Secondhand Only	746	0.93%	750	3.48%	25	3.34%	1496	1.46%
	Past/Current	33586	41.67%	13661	63.33%	648	86.52%	47247	46.25%
First Time of Anxiety or Depression	Never	58322	72.36%	11002	51.01%	115	15.35%	69324	67.85%
	Before FOT[†]	10431	12.94%	3230	14.97%	207	27.64%	13661	13.37%
	After FOT	11843	14.69%	7338	34.02%	427	57.01%	19181	18.77%
First Time of Substance Abuse	Never	70039	86.90%	15283	70.85%	45	6.01%	85322	83.51%
	Before FOT	4315	5.35%	1730	8.02%	221	29.51%	6045	5.92%
	After FOT	6242	7.74%	4557	21.13%	483	64.49%	10799	10.57%
First Time of Other Psychological Diagnosis	Never	35716	44.31%	3482	16.14%	9	1.20%	39198	38.37%
	Before FOT	27627	34.28%	9253	42.90%	412	55.01%	36880	36.10%
	After FOT	17253	21.41%	8835	40.96%	328	43.79%	26088	25.53%

Feature Extraction We retrieved structured EHR data of the chosen patients from REP historic sources, Olmsted Medical Clinic and Hospital, Mayo and Mayo Clinic Health System between January 1, 2003 and March 31, 2016. We extracted code records with time stamps and other information from three chart tables: diagnoses (*DX*), procedures (*PR*), and prescriptions (*RX*). The details are shown in Table 2. Instead of taking raw records in these tables, we mapped all the codes to a higher level code space, for the following two reasons: First, coding systems used in Mayo were different and often change from time to time[32]. For example, three different coding systems, ICD-9, ICD-10, and HICDA (Hospital International Classification of Diseases Adapted) were used for disease records in DX table. HICDA codes were used only before 2011, ICD-10 codes have not been in use until the year of 2015. This prevented us from taking one single raw code system and thus a consistent mapping of these conceptually-overlaid codes was required. Second, since there were tens of thousands of different raw codes in each table, the raw data tables were quite sparse and difficult to be examined in the feature level. Therefore, we mapped all DX and PR codes into categories in Clinical Classifications Software (CCS)[33] and all RX codes into NDF-RT class. In PR table, we also recorded the corresponding quantity together alone with the code.

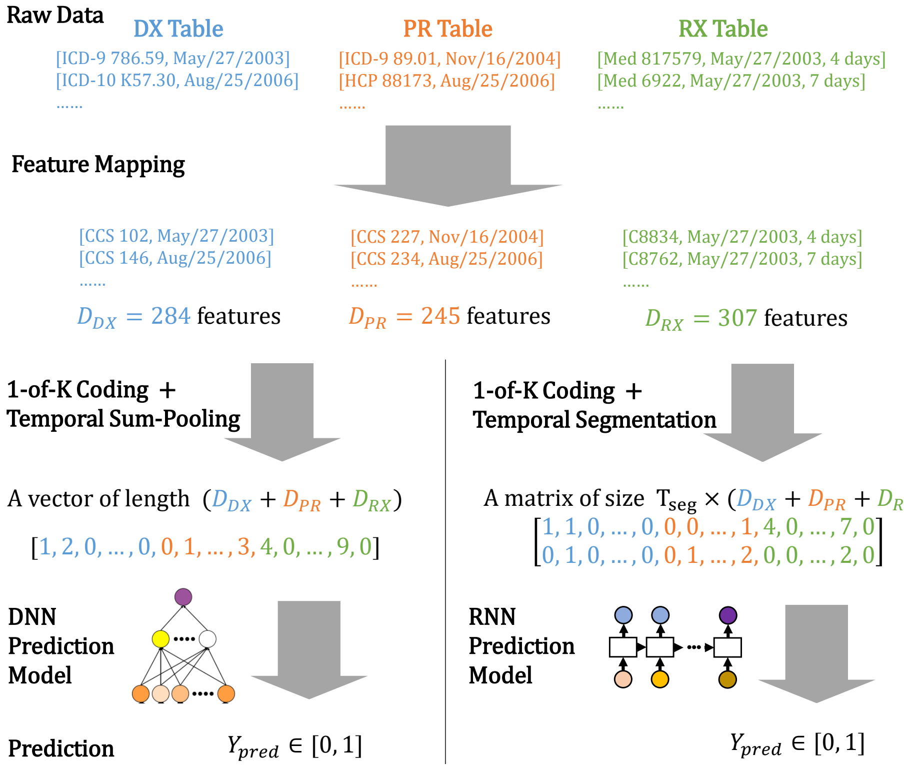
[*] Notice all patients included in *OD* group were also included in *LT* group.

[†] *FOT* refers to the time of the first opioid prescription for each patient.

Table 2: Record table descriptions and statistics.

Table Name	Descriptions	# of Records	Raw Code		Mapped Code	
			Coding Systems	Count	Coding System	Count
DX	Diagnosis records	56 229 157	ICD-9, ICD-10, HICDA	43 438	CCS	284
PR	Procedure, service, and surgical index records	46 386 740	ICD-9, ICD-10, CPT/HCPCS[34]	18 984	CCS	245
RX	Prescription records	8 102 477	Ingredient RxNorm Code	2 460	NDF-RT Class RxNorm Code	307

Temporal Data Processing We applied 1-of-K (one-hot) encoding[35] on the extracted features and used either the temporal sum-pooling or segmentations of the encoded features to get numerical features from sparse categorical features simply yet effectively. The 1-of-K encoding converts each record line to a single binary vector of the same length, and temporal sum-pooling and segmentation step (i.e., sum-pooling in each temporal segment) further aggregates features along the temporal direction. For RX table, we used the length of days instead of 1 when applying 1-of-K encoding on prescription records to utilize the important quantitative effective length information for each prescription. For example, a prescription record with length of 4 days was converted into a vector like $[0, \ldots, 4, \ldots, 0]$. Our recurrent neural network models were able to handle time series data directly and capture temporal information. We computed the sum-pooling vector[36] in each year and stacked them into a matrix, which we referred as yearly temporal segmentations. Since the medical records for patients might be of different length, the resulted matrices also had different length T_{seg}, thus these matrices could not be directly used in other models including non-recurrent deep networks and other machine learning baselines. For those model we chose the sum-pooling along all time steps and thus obtained a vector with fixed length. The length was the sum of the numbers of all mapped features from three tables ($D = D_{DX} + D_{PR} + D_{RX} = 836$) in our dataset. The prediction models took the obtained data as the input and provided predictive results. The entire pipeline is illustrated in Figure 1.



Raw Data
DX Table
[ICD-9 786.59, May/27/2003]
[ICD-10 K57.30, Aug/25/2006]
......
PR Table
[ICD-9 89.01, Nov/16/2004]
[HCP 88173, Aug/25/2006]
......
RX Table
[Med 817579, May/27/2003, 4 days]
[Med 6922, May/27/2003, 7 days]
......
Feature Mapping
[CCS 102, May/27/2003]
[CCS 146, Aug/25/2006]
......
[CCS 227, Nov/16/2004]
[CCS 234, Aug/25/2006]
......
[C8834, May/27/2003, 4 days]
[C8762, May/27/2003, 7 days]
......
$D_{DX} = 284$ features
$D_{PR} = 245$ features
$D_{RX} = 307$ features
1-of-K Coding +
Temporal Sum-Pooling
1-of-K Coding +
Temporal Segmentation
A vector of length $(D_{DX} + D_{PR} + D_{RX})$
$[1, 2, 0, \ldots, 0, 0, 1, \ldots, 3, 4, 0, \ldots, 9, 0]$
A matrix of size $T_{seg} \times (D_{DX} + D_{PR} + D_{RX})$
$\begin{bmatrix} 1, 1, 0, \ldots, 0, 0, 0, \ldots, 1, 4, 0, \ldots, 7, 0 \\ 0, 1, 0, \ldots, 0, 0, 1, \ldots, 2, 0, 0, \ldots, 2, 0 \end{bmatrix}$
DNN
Prediction
Model
RNN
Prediction
Model
Prediction
$Y_{pred} \in [0, 1]$
$Y_{pred} \in [0, 1]$

Figure 1: An illustration of the proposed pipeline from raw cohort data to final prediction. Left: DNN prediction model for data with temporal sum-pooling; Right: RNN prediction model for data with temporal segmentation.

Deep Feed-Forward Neural Network (DNN) Model A deep feed-forward neural network model (**DNN**)[37] is composed with multiple non-linear transformation layers. The output of each layer is fed to the next layer as input. For a DNN model with L layers (i.e., $L-1$ hidden layers and one final output layer), the input vector for the lth layer is denoted as $\boldsymbol{x}^{[l]} \in \mathbb{R}^{D^{[l]}}$, and rectified linear unit (ReLU)[38] function is used as the non-linear transformation function for each hidden layer. The output of layer l is

$$\boldsymbol{h}^{[l]} = ReLU(\boldsymbol{W}^{[l]}\boldsymbol{x}^{[l]} + \boldsymbol{b}^{[l]}) = \max\left(0, \boldsymbol{W}^{[l]}\boldsymbol{x}^{[l]} + \boldsymbol{b}^{[l]}\right) \in \mathbb{R}^{D^{[l+1]}},$$

which is also the input to the next layer $\boldsymbol{x}^{[l+1]}$. Here $\boldsymbol{W}^{[l]}$ and $\boldsymbol{b}^{[l]}$ are parameters which can be learned via back-propagation during training. Notice that in our case we had input dimension $D^{[1]} = D$. We chose ReLU function because 1) it is shown to not suffer from gradient vanishing problem during training compared with other non-linear transformations such as sigmoid and tanh functions, and 2) our model investigation step requires it. To conduct binary classification tasks, we applied sigmoid function $\sigma(x) = \frac{1}{1+\exp\{-x\}}$ in the last layer and set the output dimension to be 1. In other words, we had $\boldsymbol{W}^{[L]} \in \mathbb{R}^{D^{[L]}\times 1}$ and $h^{[L]} = \sigma\left(\boldsymbol{W}^{[L]}\boldsymbol{x}^{[L]} + \boldsymbol{b}^{[L]}\right) \in [0,1]$. With this proposed DNN model structure, we could learn the weights by optimizing binary cross-entropy loss function during training, which is

$$\ell_{loss} = -\sum_{n=1}^{N}\left(y_n \log h^{[L]} + (1-y_n)\log(1-h^{[L]})\right),$$

where y_n is the binary label for nth patient. In ST-LT prediction task, $y_n = 1/0$ indicates long-term/short-term opioid user. In LT-OD prediction task, $y_n = 1$ and $y_n = 0$ are for opioid-dependent and other patients, respectively.

Recurrent Neural Network (RNN) Model In order to handle sequential or temporal data of *arbitrary length* and capture temporal information from the data, recurrent neural network (**RNN**)[39] models are widely used. Unlike feed-forward neural network models, RNN models perform the same operation at each time step of the sequence input, and feed the output to the next time step as part of the input. Thus, RNN models are able to memorize what they have seen before and benefit from shared model weights (parameters) for all time steps. In order to capture complex long temporal dependency and avoid vanishing gradient problems, some modified RNN models such as Long Short-Term Memory (LSTM)[40] and Gated Recurrent Unit (GRU)[41] have been proposed with state-of-the-art performance. Assume the input is a matrix $\boldsymbol{X} \in R^{T\times D}$, where T is the number of temporal segmentations and varies for different patients, and D is the feature dimension. The tth row $\boldsymbol{x}_t \in R^D$ of the matrix represents the encoding vector at time step t. We used LSTM in our RNN prediction model. At each time step t, LSTM takes the input at that time step $\boldsymbol{x}_t$ and output at previous time step $\boldsymbol{h}_{t-1}$ to update its inner cell state $\boldsymbol{c}_t$ and produce the current output $\boldsymbol{h}_t$ as follows:

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f\boldsymbol{x}_t + \boldsymbol{U}_f\boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) \qquad \boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i\boldsymbol{x}_t + \boldsymbol{U}_i\boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right) \qquad \boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o\boldsymbol{x}_t + \boldsymbol{U}_o\boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right)$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot tanh\left(\boldsymbol{W}_c\boldsymbol{x}_t + \boldsymbol{U}_c\boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right) \qquad \boldsymbol{h}_t = \boldsymbol{o}_t \odot tanh\left(\boldsymbol{c}_t\right)$$

Here $\boldsymbol{W}_*, \boldsymbol{U}_*, \boldsymbol{b}_*$ are all model parameters, $\sigma()$ is the sigmoid function, $tanh()$ is the tangent function, and $\odot$ refers to element-wise multiplication between two vectors. We set the initial values of $\boldsymbol{h}_0$ and $\boldsymbol{c}_0$ to be $\boldsymbol{0}$. For binary classification tasks we applied another sigmoid layer of dimension 1 on top of the last output $\boldsymbol{h}_T$, i.e., $h^{top} = \sigma\left(\boldsymbol{W}\boldsymbol{h}_T + \boldsymbol{b}\right) \in [0,1]$, and trained the RNN model by optimizing the binary cross-entropy loss on the training dataset.

Implementation and Training Techniques We implemented our deep learning models with Python *Theano*[42] and *Keras*[43] libraries and all the models were reproducible. For both DNN and RNN models, we set the dimension of each hidden layer to be 256, which was chosen to have proper size and good performance. Several training techniques were designed or used to better handle our data. First, we applied an *L*-1 *regularizer* with coefficient $1 \cdot e^{-4}$ to make the model robust and able to select important features. Our preliminary experiments showed that L-1 provided more compact models with better or similar performance as L-2 or no regularizers. Second, *dropout* technique[44] with rate $p_{dr} = 0.5$ was used for all layers to reduce overfitting and avoid harmful weight co-adaptations. This was implemented by randomly dropping out units by probability of p_{dr} in the neural networks at training time and re-scaling all the weights by $\boldsymbol{W}_{test} = p_{dr}\boldsymbol{W}_{train}$ at test time. Third, we applied novel *batch normalization*[45] on all non-recurrent layers. The basic idea is to normalize the activations of the previous layer such that the outputs keep mean of 0 and standard deviation of 1 in each mini-batch during training. The running averages computed on training dataset are used to normalize the outputs at test time. This strategy speeded up training process and improved overall performance. Additionally in our experiments we found that applying batch normalization before the input layer had relatively the same impact as taking z-normalization on the input directly. They both improved model performance but the former one required less data pre-processing cost and more flexible. Fourth, we used *RMSprop*[46] as the gradient descent optimization algorithms to train these models. RMSprop utilizes an adaptive learning rate to normalize gradient values by their magnitudes. Finally, all our deep learning models could be efficiently trained within several hours on a single desktop with i5-4590S CPU and 16 GB memories.

Investigating Important Features Deep learning models are often argued to be difficult to interpret and investigate, especially because of their complex structures and thousands of or even millions of parameters. Furthermore, carelessly attempting to check and visualize individual units in neural networks might lead to misleading conclusions[47]. However, by checking the overall model weights and structures, it is still possible to identify important features extracted from the deep learning models and obtain rough quantitative evaluations. We designed the feature importance score $\mathcal{I}$ for such purpose. We first take the weight matrix in the first layer $\boldsymbol{W}^{[1]} \in \mathbb{R}^{D^{[2]} \times D}$ of a DNN model as an example, where each column $\boldsymbol{w}_d^{[1]}$ of $\boldsymbol{W}^{[1]}$ corresponds to the dth input feature. A simple way to quantify the feature importance is to take the summation of each column. The first importance score of dth input feature is formally defined as $\mathcal{I}_1(d) = \sum_{i=1}^{D^{[2]}} W^{[1]}[i, d]$, where $W[i, d]$ is the number in ith row and dth column of $\boldsymbol{W}$. However, we only consider the first layer in this score, which is definitely insufficient for a deep models. To overcome this issue, we need to take weights in higher layers into consideration. Since ReLU function was used as transformation function in all the hidden layers, we multiplied weights in all layers and took the value at the corresponding index as the importance score. We also need to take care of the impact of batch normalization since it introduces different scales on parameters, so we apply batch normalization operation before we multiply the weight matrix for each layer. Thus, the second importance score can be defined formally as

$$\mathcal{I}_2 = \boldsymbol{W}^{[L]} BN^{[L]} \left(\cdots \boldsymbol{W}^{[2]} BN^{[2]} \left(\boldsymbol{W}^{[1]} BN^{[1]}(\mathbf{1}) \right) \right) \in R^{1 \times D}$$

where $BN^{[l]}$ denotes the batch normalization operation for layer l. This process can also be viewed as a simplified version of the original deep neural networks without non-linear transformations or bias vectors. In our experiments, $\mathcal{I}_2$ was used for our DNN models. In order to validate the way of investigating important features and verify the selected features, we checked previous clinical studies and compared with features from our baseline models.

Other Machine Learning Baselines In order to evaluate the proposed deep models and validate the findings, we also compared some commonly used machine learning baselines in clinical research, including Logistic regression (**LR**), linear support vector machine with hinge loss (**SVM**), and random forest (**RF**). All the baselines are implemented in Python *Scikit-learn*[48] package. We kept most of the default settings and hyperparameters which are shown to be effective in practice, but made several specific changes to better fit our tasks. In order to distinguish important features an introduce sparsity into the model coefficients, we also used L-1 penalty in LR and SVM, tuned the regularization strength C by searching from $1 \cdot e^{-4}$ to 10 and finally chose $C = 0.1$ in our experiments since it usually provided best prediction results. In RF, using more trees usually leads to better results, but also possibly makes the model computationally inefficient and overfitted to training samples, and the model size also will grow linearly to the number of trees. Since using more trees brought negligible performance improvement but drastically increased the model size in our preliminary experiments, we took the default setting (10 trees) so that the RF model had moderate size as others. As shown in Table 3, all the tested models had comparable sizes and thus the performance comparison was fair.

4 Results and Discussions

Classification Result Comparison As mentioned before, we conducted two classification tasks (ST-LT and LT-OD). All $102\,166$ patients were included in the 5-fold cross validation for ST-LT task. Only 3.47% long-term users

Table 3: Model size comparison when saved into binary files in disk. All deep learning models are serialized and saved in HDF5 files, and other models are saved in cPickle files.

Model	DNN	RNN	LR	SVM	RF
File Size (KB)	$1,878$	$9,320$	21	23	$2,282$

Table 4: Long-term opioid patient prediction (ST-LT) results (mean ± 95% confidence interval). In ***Setting A***, we take all the medical records before the date when the patient is marked as long-term user or Mar, 31, 2016, whichever is earlier; ***Setting B*** is the same as *Setting A* except that we exclude all the opioid and non-opioid analgesics prescriptions; In ***Setting C*** we take records made before the patient's first opioid prescription. Best results shown in **bold**.

		Baseline Models			Deep Models			
		LR	SVM	RF	DNN-1hl[‡]	DNN-2hl	DNN-3hl	RNN
Setting A	Acc.	0.8946 ± 0.002	0.8938 ± 0.002	0.8666 ± 0.004	0.8960 ± 0.002	0.8954 ± 0.001	**0.8975** ± 0.002	0.8961 ± 0.002
	AUC	0.9074 ± 0.002	0.9038 ± 0.002	0.8747 ± 0.003	0.9086 ± 0.002	0.9082 ± 0.002	0.9091 ± 0.002	**0.9094** ± 0.002
	Prec.	0.8483 ± 0.007	0.8671 ± 0.006	0.8213 ± 0.009	0.8539 ± 0.013	0.8546 ± 0.009	0.8567 ± 0.009	**0.8719** ± 0.008
	Rec.	0.6099 ± 0.006	0.5868 ± 0.007	0.4702 ± 0.018	0.6122 ± 0.009	0.6082 ± 0.006	**0.6178** ± 0.006	0.5957 ± 0.007
	κ	0.6473 ± 0.007	0.6383 ± 0.007	0.5249 ± 0.018	0.6516 ± 0.007	0.6489 ± 0.004	**0.6571** ± 0.006	0.6472 ± 0.006
Setting B	Acc.	**0.8385** ± 0.002	0.8372 ± 0.002	0.8162 ± 0.002	0.8371 ± 0.002	0.8340 ± 0.002	0.8352 ± 0.002	0.8371 ± 0.002
	AUC	0.8369 ± 0.002	0.8366 ± 0.002	0.8044 ± 0.002	0.8412 ± 0.002	0.8362 ± 0.002	0.8362 ± 0.003	**0.8466** ± 0.002
	Prec.	0.7161 ± 0.010	0.7309 ± 0.011	0.6590 ± 0.011	**0.7319** ± 0.013	0.6999 ± 0.010	0.7121 ± 0.022	0.6889 ± 0.012
	Rec.	0.3892 ± 0.005	0.3623 ± 0.006	0.2683 ± 0.005	0.3612 ± 0.008	0.3749 ± 0.016	0.3712 ± 0.018	**0.4207** ± 0.020
	κ	0.4177 ± 0.007	0.4005 ± 0.009	0.2952 ± 0.006	0.3998 ± 0.008	0.3996 ± 0.013	0.4005 ± 0.009	**0.4297** ± 0.010
Setting C	Acc.	0.7917 ± 0.001	0.7908 ± 0.001	0.7890 ± 0.001	0.7919 ± 0.001	0.7920 ± 0.001	0.7915 ± 0.001	**0.7989** ± 0.001
	AUC	0.7323 ± 0.003	0.7327 ± 0.003	0.6936 ± 0.003	0.7220 ± 0.004	0.7340 ± 0.004	0.7218 ± 0.004	**0.7536** ± 0.003
	Prec.	0.5366 ± 0.019	0.5303 ± 0.021	0.5007 ± 0.010	0.5670 ± 0.031	**0.5943** ± 0.012	0.5774 ± 0.027	0.5692 ± 0.028
	Rec.	0.0996 ± 0.003	0.0800 ± 0.003	0.1279 ± 0.004	0.0646 ± 0.005	0.0672 ± 0.011	0.0490 ± 0.015	**0.1991** ± 0.002
	κ	0.1090 ± 0.005	0.0885 ± 0.005	0.1289 ± 0.006	0.0756 ± 0.004	0.0658 ± 0.013	0.0587 ± 0.016	**0.2076** ± 0.007

Table 5: Opioid-dependent patient prediction (LT-OD) results (mean ± 95% confidence interval). *Settings A, B, C* are defined the same as those in Table 4. Best results shown in **bold**.

		Baseline Models			Deep Models			
		LR	SVM	RF	DNN-1hl[‡]	DNN-2hl	DNN-3hl	RNN
Setting A	Acc.	0.6929 ± 0.010	0.6805 ± 0.007	0.7417 ± 0.007	0.7441 ± 0.010	0.7550 ± 0.008	0.7547 ± 0.009	**0.7607** ± 0.009
	AUC	0.7119 ± 0.010	0.6985 ± 0.010	0.7773 ± 0.011	0.7853 ± 0.012	0.7975 ± 0.010	0.8044 ± 0.011	**0.8060** ± 0.010
	Prec.	0.5385 ± 0.017	0.5212 ± 0.010	**0.7049** ± 0.022	0.6214 ± 0.021	0.6323 ± 0.012	0.6328 ± 0.018	0.6896 ± 0.020
	Rec.	0.5924 ± 0.022	0.5748 ± 0.019	0.3986 ± 0.016	0.6233 ± 0.028	0.6457 ± 0.031	**0.6471** ± 0.024	0.5205 ± 0.021
	κ	0.3262 ± 0.017	0.2966 ± 0.016	0.3555 ± 0.015	0.4273 ± 0.019	0.4520 ± 0.021	**0.4571** ± 0.017	0.4505 ± 0.019
Setting B	Acc.	0.6763 ± 0.007	0.6669 ± 0.009	0.7331 ± 0.010	0.7376 ± 0.011	0.7406 ± 0.009	**0.7427** ± 0.006	0.7417 ± 0.006
	AUC	0.6968 ± 0.008	0.6898 ± 0.010	0.7659 ± 0.013	0.7720 ± 0.009	0.7821 ± 0.012	0.7829 ± 0.008	**0.8010** ± 0.007
	Prec.	0.5156 ± 0.013	0.5029 ± 0.012	0.6784 ± 0.022	0.6214 ± 0.023	0.6146 ± 0.017	0.6289 ± 0.017	**0.7107** ± 0.019
	Rec.	0.5743 ± 0.022	0.5600 ± 0.018	0.3867 ± 0.024	0.5733 ± 0.026	**0.6162** ± 0.025	0.5810 ± 0.039	0.3976 ± 0.026
	κ	0.2951 ± 0.020	0.2734 ± 0.020	0.3301 ± 0.028	0.4046 ± 0.020	**0.4201** ± 0.019	0.4098 ± 0.018	0.3787 ± 0.021
Setting C	Acc.	0.6404 ± 0.007	0.6332 ± 0.012	0.6994 ± 0.007	0.6870 ± 0.009	0.6911 ± 0.009	**0.7065** ± 0.008	0.6956 ± 0.008
	AUC	0.6512 ± 0.009	0.6429 ± 0.010	0.6999 ± 0.011	0.7130 ± 0.014	0.7216 ± 0.014	**0.7279** ± 0.014	0.7144 ± 0.011
	Prec.	0.4639 ± 0.030	0.4554 ± 0.017	0.6019 ± 0.021	0.5491 ± 0.020	0.5485 ± 0.023	**0.6193** ± 0.024	0.5975 ± 0.018
	Rec.	0.4605 ± 0.020	0.4629 ± 0.017	0.3067 ± 0.018	0.4590 ± 0.075	**0.5338** ± 0.065	0.3305 ± 0.030	0.2895 ± 0.028
	κ	0.1906 ± 0.023	0.1821 ± 0.022	0.2342 ± 0.020	0.2702 ± 0.029	**0.3006** ± 0.026	0.2542 ± 0.024	0.2542 ± 0.032

are opioid dependent and thus the labels are quite imbalanced for LT-OD task. To get robust prediction and features, we randomly generated 14 datasets with class ratio of $\frac{1}{3}$ by downsampling the non-opioid-dependent patients. Each generated dataset had records from 2 237 patients. We further introduced three different settings (*A, B, C*) to test model performances in different simulated situations. The definitions of the settings are described in the caption of Table 4. *Setting A* was the ideal case and the best prediction results could be achieved in this setting since all possible information was taken into consideration. After we found the analgesics usage can be good indicators for our prediction tasks and might hide other indicators, we designed *Setting B* which might impair the prediction performance but help us find some hidden but useful features. *Setting C* was the most practical case among the three and we took it to demonstrate the early prediction capacity of our methods. For all the settings and tasks, classification accuracy

[‡] DNN-$k$hl refers to DNN models with k hidden layers and one output layer, $k \in \{1, 2, 3\}$.

(*Acc.*), area under the receiver operating characteristic curve score (*AUC*), precision (*Prec.*), recall (*Rec.*), and Cohen's kappa coefficient (κ) are reported. Results for ST-LT and LT-OD tasks are shown in Table 4 and Table 5, respectively. First, deep models provided the best performance in terms of most evaluation metrics. The improvements on LT-OD were larger than ST-LT. Second, the RNN models which captured temporal information usually but not always beat standard DNN models. It obtained the best AUC score in 5 out of 6 settings. This implies that even loosely segmented time series contains useful temporal information. However, the superiority of RNN was shown to be less on LT-OD than ST-LT, and one possible reason is the lack of training samples on LT-OD.

Feature Analysis It is useful to get to know which features are more related to opioid use, or played more important roles in the prediction models. We take the DNN-3hl models in *Setting A* and show the top ten most important features ordered by the absolute value of importance score $\mathcal{I}_2$ in Table 6. Basically, features with positive/negative score can be interpreted as positively/negatively correlated to the prediction target (long-term use in ST-LT, and opioid dependence in LT-OD). The score should only be compared within the same model and the same task. For both tasks, "Opioid Analgesics" prescription is selected as the most important indicators. "Non-opioid Analgesics" is also an important factor for long-term opioid use but not very useful to distinguish opioid-dependent user from long-term user. Several disorders diagnoses, such as "substance-related disorders", "anxiety disorders", and "other mental health disorders" (e.g., interview, evaluation, and consultation), are all highly related to opioid dependence. These findings are consistent with previous studies and most of the top features are also selected by LR and RF baselines. In addition, the scores for top features in LT-OD task are closer than those in ST-LT. This indicates that in *Setting A* identifying opioid-dependent users is a more challenging task which requires the exploit of more different features. The fact that all models had higher evaluation score on ST-LT than LT-OD in *Setting A* (Table 4 and 5) also supported the same claim. As we only did preliminary investigations, more details and validations will be discovered in the following work.

Table 6: Most Important features for long-term opioid patient (ST-LT, left) and opioid-dependent patient (LT-OD, right) identified from DNN-3hl model.

ST-LT Prediction				LT-OD Prediction			
Table	Code	Feature Name	$\mathcal{I}$	Table	Code	Feature Name	$\mathcal{I}$
RX	C8834	Opioid Analgesics	0.2287	RX	C8834	Opioid Analgesics	0.7784
RX	C8890	Amphetamine-like Stimulants	−0.0843	DX	CCS 661	Substance-related Disorders	0.6186
RX	C8838	Non-opioid Analgesics	0.0802	PR	CCS 182	Mammography	−0.3481
PR	CCS 227	Other Diagnostic Procedures	0.0272	DX	CCS 663	Substance Abuse/Mental Health History	0.3248
DX	CCS 258	Other Screening	−0.0218	DX	CCS 258	Other Screening	−0.2948
RX	C4859	Salicylates, Antirheumatic	−0.0204	PR	CCS 228	Prophylactic Vaccinations/Inoculations	−0.2796
DX	CCS 203	Osteoarthritis	0.0185	DX	CCS 651	Anxiety Disorders	0.2785
DX	CCS 205	Spondylosis	0.0179	RX	C8864	Anticonvulsants	0.2626
DX	CCS 98	Essential Hypertension	0.0126	RX	C8860	Benzodiazepine Derivatives	0.2382
RX	C2728	Vaccines/Toxoids, Other	−0.0120	DX	CCS 670	Miscellaneous Mental Health Disorders	0.2324

5 Summary

In this paper, we applied deep learning models for opioid user group predictions on a large-scale real-world EHR dataset. The deep learning models were able to achieve superior classification performance and identify useful feature indicators for opioid-dependent and long-term users. Our work demonstrated how novel deep learning models can be utilized to obtain state-of-the-art performance in practical clinical studies. In our future work, we plan to further investigate important features extracted from deep models, and incorporate numerical and unstructured EHR data along with code records into deep learning prediction models. We also plan to explore more fancy deep learning models to capture the temporal dependencies and evolutions for medical records of opioid users.

References

1. Dowell D, Haegerich TM, Chou R. CDC guideline for prescribing opioids for chronic pain in United States, 2016. Jama. 2016;315(15):1624–1645.

2. Thorson D, Biewen P, Bonte B, Epstein H, Haake B, Hansen C, et al. Acute pain assessment and opioid prescribing protocol. Institute for Clinical Systems Improvement. 2014;.
3. Paulozzi LJ, Mack KA, Hockenberry JM, et al. Vital signs: variation among states in prescribing of opioid pain relievers and benzodiazepinesUnited States, 2012. MMWR Morb Mortal Wkly Rep. 2014;63(26):563–8.
4. Frenk SM, Porter KS, Paulozzi LJ. Prescription opioid analgesic use among adults: United States, 1999–2012. NCHS data brief. 2015;189(189):1–8.
5. Shaheed CA, Maher CG, Williams KA, Day R, McLachlan AJ. Efficacy, tolerability, and dose-dependent effects of opioid analgesics for low back pain: A systematic review and meta-analysis. JAMA internal medicine. 2016;176(7):958–968.
6. Breivik H, Collett B, Ventafridda V, Cohen R, Gallacher D. Survey of chronic pain in Europe: prevalence, impact on daily life, and treatment. European journal of pain. 2006;10(4):287–333.
7. Chou R, Clark E, Helfand M. Comparative efficacy and safety of long-acting oral opioids for chronic non-cancer pain: a systematic review. Journal of pain and symptom management. 2003;26(5):1026–1048.
8. Kalso E, Edwards JE, Moore RA, McQuay HJ. Opioids in chronic non-cancer pain: systematic review of efficacy and safety. Pain. 2004;112(3):372–380.
9. for Disease Control C, (CDC P, et al. Vital signs: overdoses of prescription opioid pain relievers—United States, 1999–2008. MMWR Morbidity and mortality weekly report. 2011;60(43):1487.
10. Rudd RA, Aleshire N, Zibbell JE, Matthew Gladden R. Increases in drug and opioid overdose deathsUnited States, 2000–2014. American Journal of Transplantation. 2016;16(4):1323–1327.
11. of Justice D, of Public Affairs O. McKesson Agrees to Pay Record $150 Million Settlement for Failure to Report Suspicious Orders of Pharmaceutical Drugs. 2017;.
12. Boudreau D, Von Korff M, Rutter CM, Saunders K, Ray GT, Sullivan MD, et al. Trends in long-term opioid therapy for chronic non-cancer pain. Pharmacoepidemiology and drug safety. 2009;18(12):1166–1175.
13. Saunders KW, Von Korff M, Campbell CI, Banta-Green CJ, Sullivan MD, Merrill JO, et al. Concurrent use of alcohol and sedatives among persons prescribed chronic opioid therapy: prevalence and risk factors. The Journal of Pain. 2012;13(3):266–275.
14. Weisner CM, Campbell CI, Ray GT, Saunders K, Merrill JO, Banta-Green C, et al. Trends in prescribed opioid therapy for non-cancer pain for individuals with prior substance use disorders. Pain. 2009;145(3):287–293.
15. Braden JB, Sullivan MD, Ray GT, Saunders K, Merrill J, Silverberg MJ, et al. Trends in long-term opioid therapy for noncancer pain among persons with a history of depression. General hospital psychiatry. 2009;31(6):564–570.
16. Sun Y, Chen Y, Wang X, Tang X. Deep learning face representation by joint identification-verification. In: Advances in neural information processing systems; 2014. p. 1988–1996.
17. Parkhi OM, Vedaldi A, Zisserman A. Deep Face Recognition. In: BMVC. vol. 1; 2015. p. 6.
18. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473. 2014;.
19. Vinyals O, Toshev A, Bengio S, Erhan D. Show and tell: A neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015. p. 3156–3164.
20. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature. 2016;529(7587):484–489.
21. Che Z, Kale D, Li W, Bahadori MT, Liu Y. Deep computational phenotyping. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM; 2015. p. 507–516.
22. Lasko TA, Denny JC, Levy MA. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. PloS one. 2013;8(6):e66341.
23. Miotto R, Li L, Kidd BA, Dudley JT. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. Scientific reports. 2016;6.
24. Litjens G, Sánchez CI, Timofeeva N, Hermsen M, Nagtegaal I, Kovacs I, et al. Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. Scientific reports. 2016;6.
25. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin

cancer with deep neural networks. Nature. 2017;542(7639):115–118.

26. Ertosun MG, Rubin DL. Automated grading of gliomas using deep learning in digital pathology images: A modular approach with ensemble of convolutional neural networks. In: AMIA Annual Symposium Proceedings. vol. 2015. American Medical Informatics Association; 2015. p. 1899.
27. Sauver JLS, Grossardt BR, Yawn BP, Melton LJ, Rocca WA. Use of a medical records linkage system to enumerate a dynamic population over time: the Rochester epidemiology project. American journal of epidemiology. 2011;173(9):1059–1068.
28. Sauver JLS, Grossardt BR, Leibson CL, Yawn BP, Melton LJ, Rocca WA. Generalizability of epidemiological findings and public health decisions: an illustration from the Rochester Epidemiology Project. In: Mayo Clinic Proceedings. vol. 87. Elsevier; 2012. p. 151–160.
29. Nelson SJ, Zeng K, Kilbourne J, Powell T, Moore R. Normalized names for clinical drugs: RxNorm at 6 years. Journal of the American Medical Informatics Association. 2011;18(4):441–448.
30. Von Korff M, Saunders K, Ray GT, Boudreau D, Campbell C, Merrill J, et al. Defacto long-term opioid therapy for non-cancer pain. The Clinical journal of pain. 2008;24(6):521.
31. Hooten WM, St Sauver JL, McGree ME, Jacobson DJ, Warner DO. Incidence and risk factors for progression from short-term to episodic or long-term opioid prescribing: a population-based study. In: Mayo Clinic Proceedings. vol. 90. Elsevier; 2015. p. 850–856.
32. Sauver JS, Buntrock J, Rademacher D, Albrecht D, Gregg M, Ihrke D, et al. Comparison of Mayo Clinic Coding Systems. 2010;.
33. Elixhauser A, Steiner C, Palmer L. Clinical classifications software (CCS). Book Clinical Classifications Software (CCS). 2008;.
34. Smith GI. Basic CPT/HCPCS Coding 2007. American Health Information Management Association; 2006.
35. Murphy KP. Machine learning: a probabilistic perspective. MIT press; 2012.
36. Boureau YL, Ponce J, LeCun Y. A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th international conference on machine learning (ICML-10); 2010. p. 111–118.
37. Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural networks. 1989;2(5):359–366.
38. Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10); 2010. p. 807–814.
39. Goller C, Kuchler A. Learning task-dependent distributed representations by backpropagation through structure. In: Neural Networks, 1996., IEEE International Conference on. vol. 1. IEEE; 1996. p. 347–352.
40. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735–1780.
41. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:14091259. 2014;.
42. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints. 2016 May;abs/1605.02688.
43. Chollet F. Keras. GitHub; 2015.
44. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 2014;15(1):1929–1958.
45. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:150203167. 2015;.
46. Tieleman T, Hinton G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning. 2012;4(2).
47. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, et al. Intriguing properties of neural networks. arXiv preprint arXiv:13126199. 2013;.
48. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825–2830.