Journal of the American Medical Informatics Association, 24(2), 2017, 361–370
doi: 10.1093/jamia/ocw112
Advance Access Publication Date: 13 August 2016
Research and Applications

Research and Applications

Using recurrent neural network models for early detection of heart failure onset

Edward Choi,[1] Andy Schuetz,[2] Walter F Stewart,[2] and Jimeng Sun[1]

[1]Georgia Institute of Technology, Atlanta and [2]Sutter Health, Walnut Creek, California

Correspondence to Jimeng Sun, School of Computational Science and Engineering, Georgia Institute of Technology, 266 Ferst Drive Atlanta, GA 30313, USA; jsun@cc.gatech.edu; Tel: 404.894.0482

Received 27 February 2016; Revised 30 June 2016; Accepted 5 July 2016

ABSTRACT

Objective: We explored whether use of deep learning to model temporal relations among events in electronic health records (EHRs) would improve model performance in predicting initial diagnosis of heart failure (HF) compared to conventional methods that ignore temporality.

Materials and Methods: Data were from a health system's EHR on 3884 incident HF cases and 28 903 controls, identified as primary care patients, between May 16, 2000, and May 23, 2013. Recurrent neural network (RNN) models using gated recurrent units (GRUs) were adapted to detect relations among time-stamped events (eg, disease diagnosis, medication orders, procedure orders, etc.) with a 12- to 18-month observation window of cases and controls. Model performance metrics were compared to regularized logistic regression, neural network, support vector machine, and K-nearest neighbor classifier approaches.

Results: Using a 12-month observation window, the area under the curve (AUC) for the RNN model was 0.777, compared to AUCs for logistic regression (0.747), multilayer perceptron (MLP) with 1 hidden layer (0.765), support vector machine (SVM) (0.743), and K-nearest neighbor (KNN) (0.730). When using an 18-month observation window, the AUC for the RNN model increased to 0.883 and was significantly higher than the 0.834 AUC for the best of the baseline methods (MLP).

Conclusion: Deep learning models adapted to leverage temporal relations appear to improve performance of models for detection of incident heart failure with a short observation window of 12–18 months.

Key words: heart failure prediction, deep learning, recurrent neural network, patient progression model, electronic health records

OBJECTIVE

Before diagnosis of a disease, an individual's progression mediated by pathophysiologic changes distinguishes those who will eventually get the disease from those who will not. Detection of temporal event sequences that reliably distinguish disease cases from controls may be particularly useful in improving predictive model performance. We investigated whether recurrent neural network (RNN) models could be adapted for this purpose, converting clinical event sequences and related time-stamped data into pathways relevant to early detection of disease.

Electronic health record (EHR) data capture rich clinical and related temporal information. Patient health care encounters are well documented (eg, diagnoses, medications, and procedures) and time-stamped. However, EHR data are highly complex, given the structure and breadth of information captured (spanning provider behavior, care utilization, treatment pathways, and patient disease state) and irregular sampling frequency. To date, most predictive modeling work using EHR data rely on aggregate features (eg, event count and event average). Temporal relations among disaggregated features (eg, medication ordered at one time and procedure performed at another) are not captured using these methods.

We applied RNN models to heart failure (HF) cases and controls using longitudinal EHR data, and compared the model performance to traditional machine learning approaches. HF is one of the leading

© The Author 2016. Published by Oxford University Press on behalf of the American Medical Informatics Association.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

causes of morbidity and mortality among elderly individuals in developed economies and accounts for significant and growing health care expenditures.[1,2] Improved early detection could open new opportunities for delaying or preventing progression to diagnosis of HF and reduce cost.

BACKGROUND AND SIGNIFICANCE

Early detection of heart failure

Onset of HF is associated with a high level of disability, health care costs, and mortality (roughly 50% risk of mortality within 5 years of diagnosis).[1,2] There has been relatively little progress in slowing the progression of this disease, largely because it is difficult to detect before actual diagnosis. As a consequence, intervention has primarily been confined to the time period after diagnosis, with little or no impact on disease progression. Earlier detection of HF could lead to improved outcomes through patient engagement and more assertive treatment with angiotensin-converting enzyme inhibitors or angiotensin II receptor blockers, mild exercise, reduced salt intake, and possibly other options.[3–6]

Previous work on early detection of HF has relied on conventional modeling techniques, such as logistic regression or support vector machine (SVM), that use features representing the aggregation of events in an observation window and exclude temporal relations among events in the observation window.[7–9] In contrast, recurrent neural network (RNN) methods capture temporal patterns present in longitudinal data. RNN models have proven effective in many difficult machine learning tasks, such as image captioning[10] and language translation.[11] Extending these methods to health data is sensible.

Applications of deep learning

Deep learning methods have recently led to a renaissance of neural network–based models. Pioneering studies introduced stacked restricted Boltzmann machines[12] and stacked autoencoders,[13] which showed impressive performance in image processing, employing the layer-wise pretraining technique. Since then, variations of neural network application have explored deep architectures in computer vision,[14–16] audio processing,[17,18] and natural language processing (NLP),[11,19–21] among other fields.

RNN models are naturally suited to temporal sequenced data, and several variants have been developed for sequenced features. Hochreiter and Schmidhuber[22] proposed long short-term memory (LSTM), exhibiting impressive performance in numerous sequence-based tasks such as handwriting recognition,[23] acoustic modeling of speech,[24] language modeling,[25] and language translation.[26] Cho et al.[11] proposed the gated recurrent unit (GRU) model, structurally similar to but simpler than LSTM, and showed comparable, if not better, performance.[27] In the RNN work described herein, we used the GRU structure to model the temporal relations among health data from patient EHRs to predict the future diagnosis of HF.

Health care applications of deep learning

Researchers have recently started to apply deep learning methods to clinical applications. Lasko et al.[28] used autoencoders to learn phenotypic patterns from serum uric acid measurements. Che et al.[29] used deep neural networks with incremental learning on clinical time series data to discover physiologic patterns associated with known clinical phenotypes. Both works,[28,29] however, focused on learning patterns from clinical records rather than predicting a clinical event. Hammerla et al.[30] applied restricted Boltzmann machines on time series data collected from wearable sensors to predict the disease state of Parkinson's disease patients. Lipton et al.[31] used LSTM for multilabel diagnosis prediction using pediatric ICU time series data (eg, heart rate, blood pressure, glucose level, etc.). Both of these studies[30,31] used multivariate time series data from patients, which focused on very different clinical conditions, with continuous time series data. Our study focuses on early detection of HF for the general patient population based on widely available EHR data such as time-stamped codes (diagnosis, medication, procedure).

Deep learning techniques have been recently applied to clinical text data (eg, PubMed abstracts, progress notes) using Skip-gram[20,32,33] to learn relationships among clinical processes or unified medical language system (UMLS) concepts. Choi et al.[34] applied Skip-gram to longitudinal EHR data to learn low-dimensional representations for medical concepts such as diagnosis codes, medication codes, and procedure codes,[35] and to learn representations of medical concepts. We borrowed from this prior work to leverage similar representation of medical concepts through Skip-gram but focus on temporal modeling using RNN for predicting HF.

Time series analysis of EHRs

Traditional time series methods using linear models for low-dimensional data have been widely applied to EHRs: modeling the progression of chronic kidney disease to kidney failure using the Cox proportional hazard model,[36] the progression of Alzheimer's disease using the hidden Markov model[37] and fused group Lasso,[38] the progression of glaucoma using using a 2-dimensional continuous-time hidden Markov model,[39] the progression of lung disease using graphical models with the Gaussian process,[40] the progression of chronic obstructive pulmonary disease using the Markov jump process,[41] and the progression of multiple diseases using the Hawkes process.[42] These previous works were not able to model high dimensional non-linear relations as well as RNN. We focused on predicting the onset of HF using longitudinal structured patient data such as diagnosis, medication, and procedure codes. We used RNN, which provides a nonlinear improvement in model generalization and more scalability than many of the traditional methods, thanks to a more optimized software package and parallel architecture such as a graphics processing unit.

MATERIALS AND METHODS

GRU model for HF prediction

To represent clinical events in EHR data as computable event sequences, we adopted the one-hot vector format, often used for NLP tasks.[43] Figure 1A provides an example of how EHR events are represented as a set of one-hot vectors. Each of the N unique clinical events was represented as an N-dimensional vector, where one dimension is set to 1 and the rest are 0. Using these one-hot vectors, a sequence of clinical events (Figure 1B) can be converted to a sequence of one-hot vectors (Figure 1C). Such sequences were used to train the models, as described in the next section.

The proposed GRU model for HF prediction is an extension of the RNN framework, schematically depicted in Figure 2.

Given a sequence of clinical visits of length T (Figure 2A), the GRU accepts an input vector $\boldsymbol{x}_t$ (in our base case, one-hot vectors representing clinical codes) at each timestep t, while storing information in a single hidden layer $\boldsymbol{h}$ whose state changes over time $(\boldsymbol{h}_{t-1}, \boldsymbol{h}_t, \boldsymbol{h}_{t+1})$. After seeing the entirety of clinical events, we applied

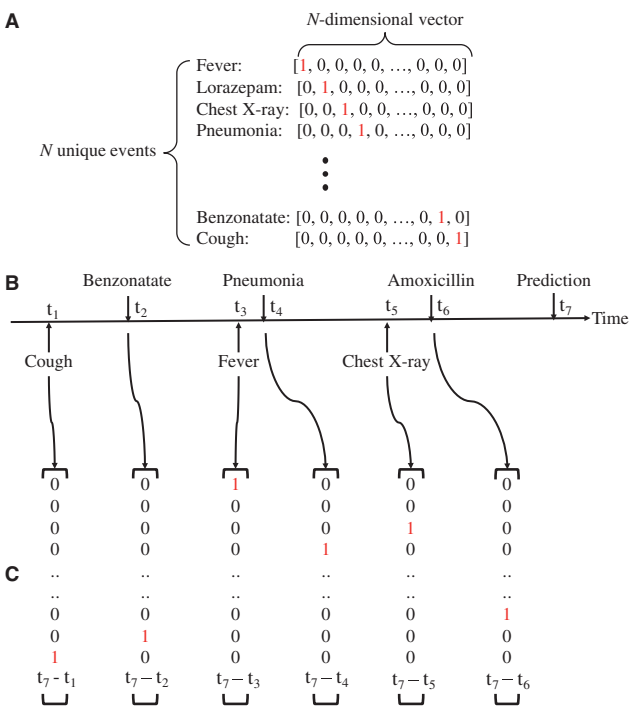

Figure 1. (**A**) One-hot vector encoding of clinical events. t in (**B**) indicates the time at which the event occurs, assuming we make the prediction at time t_7. We appended the time duration feature at the end of the one-hot vector, as shown in (**C**).

logistic regression to the final hidden state vector $\boldsymbol{h}_T$ and produced the scalar value y, which estimates the patient-specific risk score for future diagnosis of HF.

The dashed box of Figure 2A corresponds to the GRU model we used for HF prediction. GRU has 4 components (Figure 2B): $\boldsymbol{z}_t$, the update gate at timestep t; $\boldsymbol{r}_t$, the reset gate at timestep t; $\tilde{\boldsymbol{h}}_t$, the intermediate memory unit at timestep t; and $\boldsymbol{h}_t$, the hidden layer at timestep t. The mathematical formulation of Figure 2B is as follows:

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z\boldsymbol{x}_t + \boldsymbol{U}_z\boldsymbol{h}_{t-1} + \boldsymbol{b}_z)$$

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r\boldsymbol{x}_t + \boldsymbol{U}_r\boldsymbol{h}_{t-1} + \boldsymbol{b}_r)$$

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W}_h\boldsymbol{x}_t + \boldsymbol{r}_t \odot \boldsymbol{U}_h\boldsymbol{h}_{t-1} + \boldsymbol{b}_h)$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{z}_t) \odot \tilde{\boldsymbol{h}}_t$$

where σ() is the sigmoid function, and $\boldsymbol{W}$s, $\boldsymbol{U}$s, and $\boldsymbol{b}$s are the weight matrices and bias terms for calculating both gates $\boldsymbol{z}_t$ and $\boldsymbol{r}_t$ and the intermediate memory unit $\tilde{\boldsymbol{h}}_t$, respectively. Note that in GRU, the previous hidden layer $\boldsymbol{h}_{t-1}$ and the current input $\boldsymbol{x}_t$ do not directly change the value of the current hidden layer $\boldsymbol{h}_t$. Instead, they change the values of both gates $\boldsymbol{z}_t$, $\boldsymbol{r}_t$ and the intermediate memory unit $\tilde{\boldsymbol{h}}_t$. Then the current hidden layer $\boldsymbol{h}_t$ is determined by $\tilde{\boldsymbol{h}}_t$ and $\boldsymbol{z}_t$, where $\odot$ denotes element-wise multiplication (ie, Hadamard product) and both gates have values between 0 and 1. Therefore, if $\boldsymbol{r}_t$ is close to 0, $\tilde{\boldsymbol{h}}_t$ will disregard $\boldsymbol{h}_{t-1}$. If $\boldsymbol{z}_t$ is close to 1, $\boldsymbol{h}_t$ will disregard $\boldsymbol{x}_t$ and retain the same value as $\boldsymbol{h}_{t-1}$. Simply put, the reset gate allows the hidden layer to drop any information that is not found to be useful in predicting the future, and the updated gate controls how much information from the previous hidden layer should be retained in the current hidden layer. This characteristic of GRU is especially useful as it is not easy to identify information essential to predicting HF diagnosis, given the enormous volume of EHR data. GRU model training learns to keep or ignore particular inputs (eg, diagnoses, medications, procedures) at each timestep as it sees fit.

We also tested a GRU model to determine if feature representation of the time between an event and the index date in combination with the one-hot input vectors improved model performance. Specifically, the model was trained by appending to the input vector $\boldsymbol{x}_t$ an extra dimension that represents the time as a logarithm of the number of days between the event time t and the index date T. We applied logarithm transformation to minimize the skewed distribution of the durations. We also explored the time between consecutive visits or without logarithm transformation, but the above method provided the best result.

The logistic regression model applied to the final state of the hidden layer $\boldsymbol{h}_t$ is formulated as

$$y = \sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}_T + b)$$

where $\boldsymbol{w}$ and b are the weight vector and bias for logistic regression. To learn the parameters of the proposed model, we set the cross-entropy of the scalar outcome y as the loss function and tried to minimize it in terms of $\boldsymbol{W}_{\{z,r,h\}}$, $\boldsymbol{U}_{\{z,r,h\}}$, $\boldsymbol{b}_{\{z,r,h\}}$, $\boldsymbol{w}$, and b. The loss function is as follows:

$$\mathrm{Loss} = -\sum_{i=1}^{P}\left(c^{(i)}\log y^{(i)} + \left(1 - c^{(i)}\right)\log\left(1 - y^{(i)}\right)\right)$$

where P is the total number of patients, $c^{(i)}$ is the HF case indicator for the i-th patient where 1 indicates HF case and 0 control, and $y^{(i)}$ is the risk score of the i-th patient calculated by the model. Minimization can be performed through the back-propagation and mini-batch stochastic gradient descent.[44] The gradients are automatically calculated by Theano,[45] a deep learning software for Python.

Medical concept vectors

We constructed medical concept vectors that improve the one-hot vectors as the input $\boldsymbol{x}_t$. We leveraged an NLP embedding technique, Skip-gram,[20] to train vector representations of diagnosis codes, medication codes, and procedure codes. The resulting vector representation, denoted as a medical concept vector, is intended to capture the hidden relations among various codes.

We trained medical concept vectors by sliding a fixed-size window on a sequence of codes, maximizing the log probability at each step as follows:

$$\text{Maximize } \frac{1}{T}\sum_{t=1}^{T}\sum_{-w\le j\le w, j\ne 0}\log p\left(c_{t+j}|c_t\right),$$

$$\text{where } p\left(c_{t+j}|c_t\right) = \frac{\exp\left(\mathbf{v}\left(c_{t+j}\right)^{\mathrm{T}}\mathbf{v}(c_t)\right)}{\sum_{c=1}^{N}\exp\left(\mathbf{v}(c)^{\mathrm{T}}\mathbf{v}(c_t)\right)}$$

where T is the number of codes in all of a patient's visits, w is the size of the window, c_t is the code at position t, and $\mathbf{v}(c_t)$ is the vector representation of the code c_t. Simply put, Skip-gram tries to maximize the inner product of the vector representations of temporally proximal concepts. The size of the window and the dimensionality of the vector representation are hyperparameters generally set to 5 and 50–1000, respectively.[20] We trained the medical concept vectors using the encounter, medication order, procedure order, and problem list, with window size 5 and resulting dimensionality 100. Our

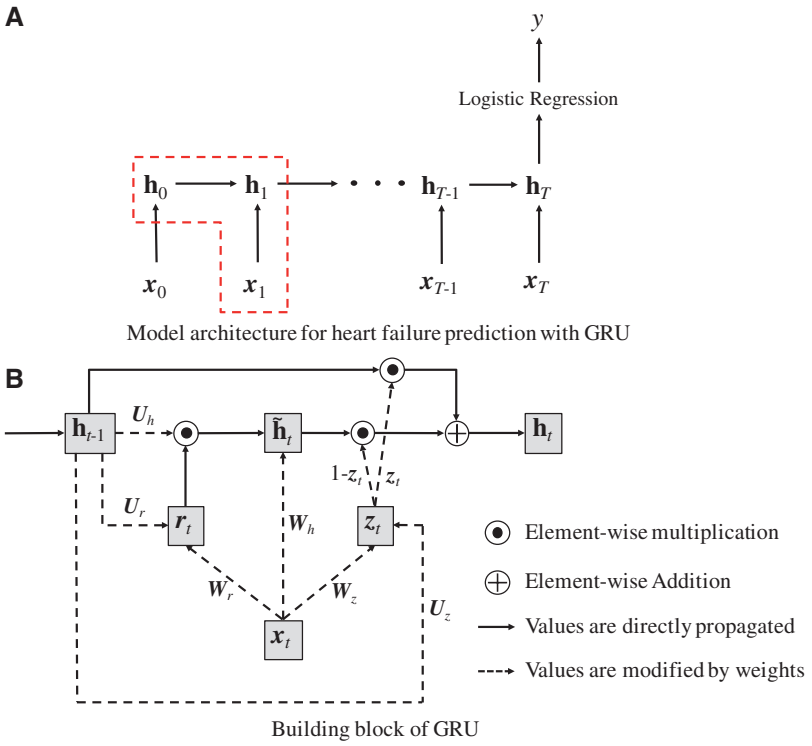

Figure 2. The GRU model architecture (**A**), and building blocks (**B**). Note that vectors are denoted in bold lowercase, matrices in bold uppercase, and scalars in plain lowercase letters.

other work presented the Skip-gram application for health care with a focus on interpretation of the resulting medical concept vectors.[35] This study, on the other hand, focuses on the sequential nature of the longitudinal EHR by using RNN for early detection of HF.

Data description

Data were from Sutter Palo Alto Medical Foundation (Sutter-PAMF) primary care patients. Sutter-PAMF is a large primary care and multispecialty group practice that has used an Epic Systems Corporation EHR for more than a decade. EHR data on primary care patients were extracted from encounters occurring between May 16, 2000, and May 23, 2013. The EHR dataset documented care delivered in the outpatient setting and included demographics, tobacco and alcohol consumption, clinical and laboratory values, International Classification of Disease version 9 (ICD-9) codes associated with encounters, orders, and referrals, procedure information in Current Procedural Terminology (CPT) codes, and medication prescription information in medical names. The dataset contained approximately 58 652 000 medical codes assigned to patients. Provider notes were not used in the present work, but could be included in a future effort.

Definitions of cases and controls

A density sampling design was used for the longitudinal records of the Sutter-PAMF patient population.[46] Cases met the criteria for incident onset of HF, described in Vijayakrishnan et al.[47] and adapted from Gurwitz et al.[48] Incident cases were of patients 40–85 years of age at the time of HF diagnosis. HF diagnosis (HFDx) was operationally defined as follows: (1) Qualifying ICD-9 codes for HF appeared as an encounter diagnosis or as the indication for a medication order. (Qualifying ICD-9 codes are listed in the supplementary material.) Qualifying ICD-9 codes with image and other related orders were excluded, because these orders often represented a suspicion of HF, where the results are often negative. (2) A minimum of 3 clinical encounters with qualifying ICD-9 codes had to occur within 12 months of each other, where the date of diagnosis was assigned to the earliest of the 3 dates. If the time span between the first and second appearance of the HF diagnostic code was >12 months, the date of the second encounter was used as the first qualifying encounter. The point upon each patient's timeline at which incident HF was established was denoted as HFDx.

Up to 10 eligible primary care clinic-, sex-, and age-matched (in 5-year intervals) controls were selected for each incident HF case, yielding an overall ratio of 9 controls per case. Each control was also assigned an index date, which was the HFDx timepoint of the matched case. Primary care patients were eligible to be controls if they did not meet the operational criteria for HF diagnosis prior to the HFDx timepoint plus 182 days of their corresponding case. Other details on matching are described in the supplementary section.

We extracted all records from the 18-month period before the HFDx, constituting an interval that could be partitioned into an observation window and a prediction window. The medical records within the observation window were used as the dataset for training models. Diagnosis, medication, and procedure codes assigned to each patient were temporally ordered. Multiple diagnoses/medications/procedures at a single visit were represented as multiple one-hot vectors with a random order.

Model evaluation

Baseline models for performance comparison

We trained 4 classification models—regularized logistic regression, multilayer perceptron (MLP), support vector machine (SVM), and K-nearest neighbor (KNN)—in addition to our proposed gated recurrent unit (GRU) model.

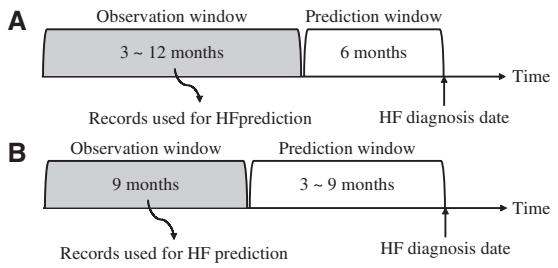

Figure 3. Two experimental settings where we alternately changed the length of the prediction window and the observation window. In (**A**), the prediction window was fixed at 6 months, while we varied the length of the observation window. In (**B**), the observation length was fixed at 9 months, while we varied the length of the prediction window.

Training strategy

All models were trained using the dataset created from the 18-month period before the HFDx. We iteratively divide the data into train, validation, and test with a ratio of 5:1:1, and report the model performance on the test set as the area under the ROC curve (AUC). Since probability is not explicitly estimated using the baseline SVM, we used the confidence score of SVM to calculate the AUC. Detailed training strategy is given in the supplementary material.

Input features and algorithms

We noticed that the sparseness and high-dimensionality of one-hot coded input vectors could be mitigated via 2 alternative approaches:

The traditional approach is to use standard medical concept groupers. ICD-9 diagnosis codes can be grouped by the Clinical Classification Software (CCS) diagnosis grouper[49] into 283 groups, medication codes can be grouped by the Generic Product Identifier[50] into 96 groups, and CPT procedure codes can be grouped by the CCS procedure grouper[51] into 244 groups. Alternatively, medical concept vectors based on Skip-gram can be used to capture relations among diagnoses, medications, and procedures.

Training data for the 2 GRU models (with and without time duration information) were constructed in the same fashion for all 3 types of input (one-hot encoding, grouped codes, and medical concept vectors based on Skip-gram).

Logistic regression, MLP, SVM, and KNN were also trained with these 3 types of aggregated feature vectors. In the aggregated one-hot vector, each dimension represents the total number of occurrences of a specific code in the observation window. Aggregated grouped code vectors and aggregated medical concept vectors are generated in the same fashion, except the one-hot vectors are replaced by grouped code vectors or pretrained medical concept vectors. All aggregated input vectors were normalized to zero mean and unit variance.

Evaluation strategy

The utility of a model is related, in part, to how much data are required for application and how far into the future an accurate prediction can be made. We conducted experiments to examine model performance for varying lengths of the prediction window (ie, time before HFDx) and the observation window (ie, length of time before the beginning of the prediction window), where features were only extracted from the defined observation window (Figures 3A and B). Note that we trained separate models for each observation window size so that models could learn optimal features from patient records of different lengths.

Implementation details

The GRU models, logistic regression, and MLP were implemented with Theano 0.7.[45] Adadelta[52] was used for model training because it does not depend strongly on the learning rate setting. SVM and KNN were implemented with Python Scikit-Learn 0.16.1. An Ubuntu machine with Xeon E5-2697, 128 GB memory, and Nvidia Tesla K80 was used to train all models. Hyperparameters used for training each model are described in the supplementary section. We have made our codes and synthetic data available on a public repository (https://github.com/mp2893/rnn_predict).

EXPERIMENT RESULTS

Data processing

From random samples of 265 336 Sutter-PAMF patients, 4178 incident HF cases and 29 139 control patients were identified. The average number of clinical codes assigned to each patient was approximately 72, and there were 18 181 unique clinical codes (6910 diagnosis codes, 6897 medication codes, and 4374 procedure codes) in total. The full sample of 265 336 was used for training the medical concept vectors, and the incident HF cases and controls were used for all other model training and evaluation tasks.

Performance of HF diagnosis prediction models

The average AUC of cross-validation for all models is shown in Figure 4, where all models were trained and tested using the dataset created with the 12-month observation window and the 6-month prediction window. The colors in Figure 4 represent different training input vectors, and the error bars indicate the standard deviation derived from the cross-validation. GRU models outperformed other models, as shown in Figure 4.

We increased the observation window to the full 18 months of patient history and zero prediction window, as shown in Figure 5. All models were again trained and tested for this experiment. The GRU model consistently outperformed all the other methods, with 0.883 AUC. Models trained using the medical concept vectors significantly outperformed models trained by one-hot vectors, and also outperformed models trained by grouped code vectors.

Prediction/observation length and prediction power

Tables 1 and 2 show the cross-validation AUCs resulting from experiments described in Figures 3A and B. These tables show that GRU outperformed all other models in all observation and prediction window sizes.

Prediction time

Table 3 shows the time required to make a prediction for a single patient for each model. We used models that were trained by medical concept vectors. The time was calculated by averaging the times the models took to make predictions on the test sets from the 6-folds using only the CPU.

Scalability of our approach

Figure 6 depicts the training time of 2 different GRU models for varying numbers of patients, with maximum gradient decent epoch set to 10.

Error analysis

We analyzed the incorrect predictions made by the GRU model using temporal data and trained on the 12-month observation window and the 6-month prediction window. We sampled 100 patients each

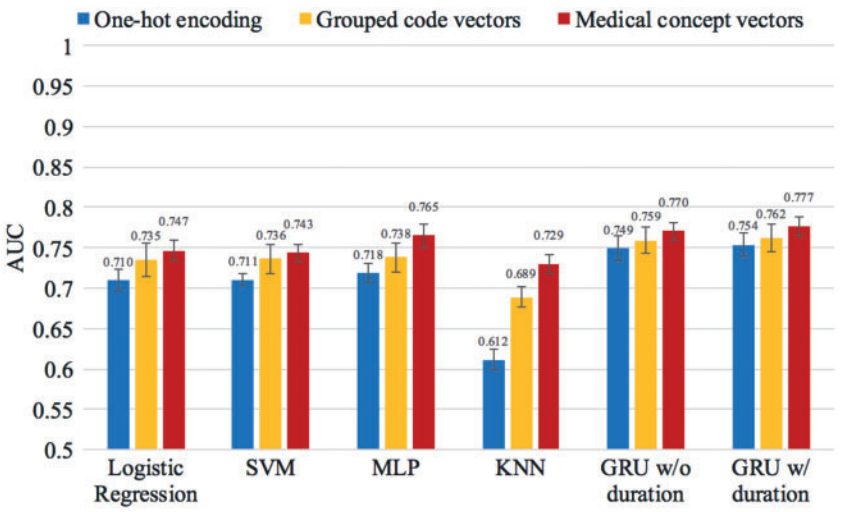

Figure 4. Heart failure prediction performance of the GRU and baseline models. All models were trained and tested using the dataset created from the 12-month observation window and the 6-month prediction window. The values of the AUC and the standard error are provided in the supplementary section.

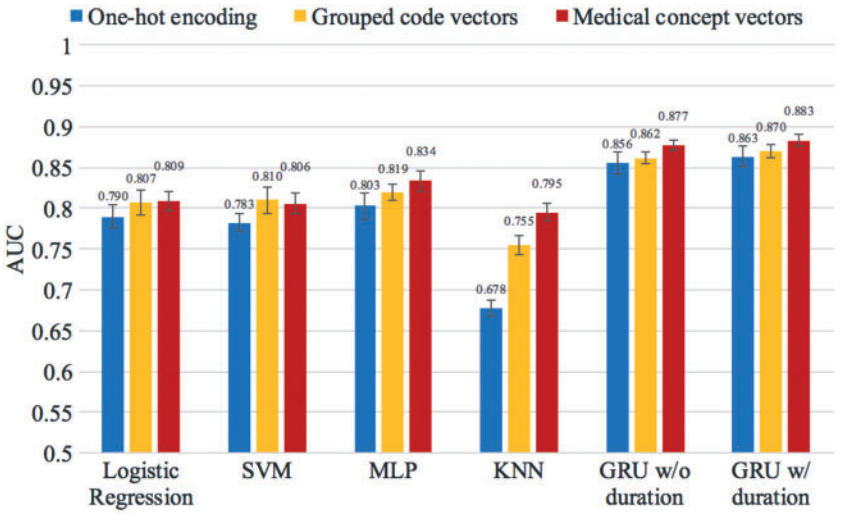

Figure 5. Heart failure prediction performance of the GRU and baseline models. All models were trained and tested using the dataset created from the 18-month observation window and 0-month prediction window. The values of the AUC and the standard error are provided in the supplementary section.

from false negatives (FNs), false positives (FPs), true positives (TPs), and true negatives (TNs). This procedure was performed for all 6-folds, which gave us 600 FN/TN/TP/FP samples. The characteristics of the samples are shown in Table 4.

Many of the incorrect predictions made by the GRU model can be attributed to visit frequency (Table 4). The second and third columns of Table 4 suggest that the GRU tends to give high prediction scores to patients who frequently visit the hospital. The fourth column, however, suggests that the number of codes assigned at each visit does not generally affect HF prediction.

The second aspect is the type of code. The last column shows that the GRU tends to give high prediction scores to patients with heart-related diseases such as atrial fibrillation and coronary atherosclerosis. Use of anticoagulants also seems to play an important role. Hypertension, an important factor for cardiovascular disease,[53] is also frequent in patients with high prediction scores. However, hypertension, being a very common disease among adults,[53] does not seem to qualify as an important feature in predicting HF, since it is also frequent among patients with low prediction scores. Overall, we can see that frequent codes are similar between FN samples and TN samples, and also between FP samples and TP samples.

The two most likely explanations for FN samples showing a similar pattern to TN samples are: (1) FN samples are related to acute HF (e.g., due to myocardial infarction) that shows little symptomatology before manifestation, and (2) there could be missing data of FN samples due to either lack of hospital visits or visits to hospitals that are not associated with Sutter. FP samples seem to be from patients who did not have HF even though their symptoms were very similar to those of patients with TP samples. Diseases that are highly related to HF, such as atrial fibrillation and coronary atherosclerosis,[54] do not always lead to HF. Overall, although the GRU shows impressive predictive performance, it seems to be confused by some cases and controls with similar patterns. We predict that using even richer information such as lab results or medical notes will help overcome this challenge and yield an even higher AUC.

Table 1. AUCs of models while varying the observation window length

Observation window (months)	Logistic regression	SVM	MLP	KNN	GRU w/duration
3	0.7210	0.7192	0.7312	0.7038	**0.7395**
4	0.7272	0.7256	0.7427	0.7084	**0.7463**
5	0.7314	0.7297	0.7455	0.7154	**0.7516**
6	0.7344	0.7327	0.7486	0.7165	**0.7529**
7	0.7388	0.7353	0.7530	0.7175	**0.7606**
8	0.7422	0.7398	0.7587	0.7253	**0.7641**
9	0.7441	0.7420	0.7603	0.7274	**0.7680**
10	0.7461	0.7434	0.7621	0.7311	**0.7713**
11	0.7462	0.7432	0.7639	0.7272	**0.7746**
12	0.7467	0.7435	0.7649	0.7293	**0.7768**

The length of the prediction window was fixed at 6 months. The highest AUCs among all models are shown in bold.

Table 2. AUCs of models that varied by length of prediction window between 3 and 9 months

Prediction window (months)	Logistic regression	SVM	MLP	KNN	GRU w/ duration
3	0.7511	0.7479	0.7650	0.7373	**0.7711**
4	0.7473	0.7439	0.7632	0.7303	**0.7678**
5	0.7458	0.7414	0.7620	0.7302	**0.7687**
6	0.7441	0.7420	0.7603	0.7274	**0.7680**
7	0.7426	0.7405	0.7617	0.7239	**0.7658**
8	0.7396	0.7366	0.7569	0.7197	**0.7651**
9	0.7341	0.7334	0.7558	0.7206	**0.7610**

The length of the observation window was fixed at 9 months. The highest AUCs are in bold.

DISCUSSION

The GRU model results in Figure 4 represent state-of-the-art prediction performance achieved by applying deep learning to discover complex relationships within EHR data. Of the models evaluated, the best performance was achieved with GRU that used temporal data and was trained with medical concept vectors. GRU models also significantly outperformed traditional machine learning models that rely on aggregate features (Figure 4) and trained with one-hot vectors or grouped code vectors. However, the baseline models trained with medical concept vectors showed comparable performance to the GRU models trained with one-hot vectors. This result suggests a potential benefit of medical concept vectors, which is especially useful when a good domain ontology is absent. Interestingly, the performance gain for GRU models achieved by using duration information was modest. We suspect this is due to the irregular pattern of patients' hospital visits. Although applying logarithm transformation improved the performance sightly, the innate irregularity of the visit pattern seems to make it difficult for the GRU to learn predictive duration features.

From Table 1, we see that GRU outperforms all the baseline models regardless of the observation window size. It seems that the longer the observation window, the more effectively all models performed. The GRU model produced the highest AUC when the prediction window was reduced to zero, and all data were used for the observation window. While this result is interesting from a methodological standpoint, such a model has limited clinical utility because it does not predict HF well in advance of a physician. We can see from Table 2 that all models performed better when predicting near-future HF cases. However, the prediction window size does not seem to affect the predictive performance as much as the observation window size. This suggests that access to duration of patient history is crucial for accurate HF prediction, perhaps because signs of incident disease manifest over a period of time.

Table 3. Prediction times of various models for a single patient

Performance metric	Logistic regression	SVM	MLP	KNN	GRU
Prediction time (seconds)	0.000002	0.000034	0.000259	36.66	0.020408

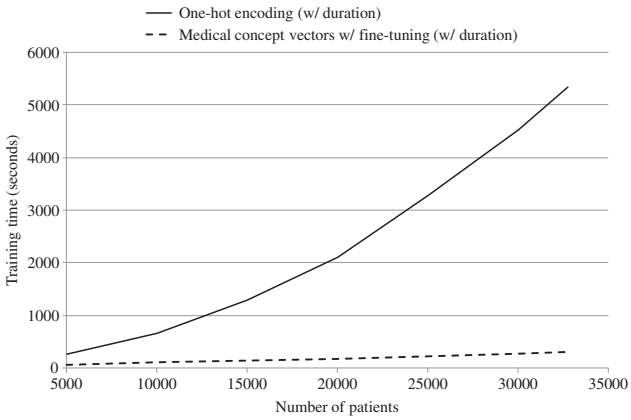

Figure 6. Training time vs number of patients.

Apart from KNN, which requires distance calculation between all data points at prediction time, GRU, being a sequential model, requires the longest time to make a prediction for a patient, as can be seen in Table 3. However, it is still able to compute HF risk for 1 million patients in 200 seconds using 100 central processing units, so cost will not pose a problem in real-world clinical application.

The GRU model using one-hot vectors displayed a super-linear relationship between training time and number of patients, as shown in Figure 6. On the other hand, the GRU model using medical concept vectors showed a linear increase. We attribute this to 2 causes. First, compared to medical concept vectors, whose dimensions are fixed, one-hot vectors increase their dimensionality as new patients are added to the training data, since new patients have new medical codes (diagnoses, medications, procedures). Second, one-hot vectors are very high-dimensional, and therefore use large amounts of VRAM, which leads to frequent garbage collection. Although the training time of the one-hot GRU models behaves near-linearly after 20 000 patients, it is still unfavorable to the medical concept vector case. Thus, we recommend using medical concept vectors to train GRU, as they both improve model performance and significantly reduce training time.

The GRU model that included temporal relations trained on medical concept vectors provided the best performance of all methods evaluated. So far our work has only taken a data-driven approach. We expect that model performance would benefit from using longer

Table 4. Characteristics of TN, FN, TP, and FP samples

Metric	Avg. no. of visits	Avg. visit frequency	Avg. no. of codes per visit	Top 10 most frequent codes
True negative	12.10	Every 20.6 days	3.08	1. Allergic rhinitis 2. Multiple immunotherapy 3. Hyperlipidemia 4. Routine medical examination 5. Screening mammogram 6. Psoriasis 7. Vaccine administration 8. Screening for cancer 9. Need for prophylactic influenza vaccination 10. Hypothyroidism
False negative	13.43	Every 19.0 days	2.94	1. Hypertension 2. Hyperlipidemia 3. Diabetes 4. Hypothyroidism 5. Keratosis 6. Need for prophylactic influenza vaccination 7. Routine medical examination 8. Esophageal reflux 9. Senile cataract 10. Screening mammogram
True positive	31.72	Every 9.7 days	3.05	1. Atrial fibrillation 2. Hypertension 3. Diabetes 4. Long-term use of anticoagulants 5. Hyperlipidemia 6. Chronic kidney disease 7. Anemia, unspecified 8. Coronary atherosclerosis 9. Edema 10. Chronic airway obstruction
False positive	29.95	Every 10.5 days	3.09	1. Atrial fibrillation 2. Hypertension 3. Long-term use of anticoagulants 5. Diabetes 4. Hyperlipidemia 6. Chronic kidney disease 7. Anemia 8. Coronary atherosclerosis 9. Edema 10. Therapeutic drug monitoring

observation periods and incorporating well-established expert medical knowledge, such as higher-level features and medical ontologies. Still, using only the raw medical order records, we were able to produce innovative, state-of-the-art results for HF diagnosis prediction. Note that in a real-world setting, however, the performance of the GRU model could be different depending on the nature of the actual cohort.

Although this work focused on HF, our approach is general and may be applied to a wide array of health care–related prediction problems. Further, the medical concept vectors we used to encode medical data were shown to generally improve the performance of both deep learning and conventional models, and thus may have utility in numerous health care applications where rich information needs to be succinctly represented.

Future work will focus on evaluating model performance for prediction windows beyond nine months, which may yield models with even greater clinical implication, and adding higher-level features and medical ontologies. Another possible enhancement is to consider using separate HF prediction models for different disease groups, such as hypertension and diabetes, which can potentially be more discriminative as the cohort is more homogeneous. However, we have to make sure a sufficient sample size is available for each disease group before developing separate RNN models. Visualizing the temporal dynamics of RNN models is another research direction, where currently only limited work with narrow application focus has been attempted.[55]

CONCLUSION

We proposed a novel predictive model framework for HF diagnosis using GRU deep learning methods. Compared to popular methods such as logistic regression, MLP, SVM, and KNN, GRU models exhibited superior performance in predicting HF diagnosis. By analyzing the results, we described the importance of respecting the sequential nature of clinical records. Future work will include incorporating expert knowledge into our framework and expanding our approach to additional health care applications.

CONTRIBUTORS

Edward Choi implemented the method and conducted all the experiments. All authors were involved in developing the ideas and writing the paper.

FUNDING

This work was supported by National Science Foundation grants IIS-1418511 and CCF-1533768 and National Heart, Lung, and Blood Institute grant 1R01H116832-01.

COMPETING INTERESTS

The authors have no competing interests to declare.

SUPPLEMENTARY MATERIAL

Supplementary material are available at *Journal of the American Medical Informatics Association* online.

REFERENCES

1. Roger VL, Weston SA, Redfield MM, *et al*. Trends in heart failure incidence and survival in a community-based population. *JAMA* 2004;292(3): 344–350.
2. Murphy SL, Xu J, Kochanek KD. Deaths: final data for 2010. *Natl Vital Stat Rep* 2010;61(4):1–117.
3. Investigators SOLVD. Effect of enalapril on mortality and the development of heart failure in asymptomatic patients with reduced left ventricular ejection fractions. *N Engl J Med* 1992;327:685–691.
4. Arnold J, Yusuf S, Young J, *et al*. Prevention of heart failure in patients in the Heart Outcomes Prevention Evaluation (HOPE) study. *Circulation* 2003;107(9):1284–1290.
5. Sciarretta S, Palano F, Tocci G, Baldini R, Volpe M. Antihypertensive treatment and development of heart failure in hypertension: a Bayesian network meta-analysis of studies in patients with hypertension and high cardiovascular risk. *Arch Int Med* 2011;171(5):384–394.
6. Wang C-H, Weisel R, Liu P, Fedak P, Verma S. Glitazones and heart failure critical appraisal for the clinician. *Circulation* 2003;107(10): 1350–1354.
7. Wang Y, Ng K, Byrd R, *et al*. Early detection of heart failure with varying prediction windows by structured and unstructured data in electronic health records. In *IEEE Engineering in Medicine and Biology Society* 2015:2530–2533.
8. Sun J, Hu J, Luo D, *et al*. Combining knowledge and data driven insights for identifying risk factors using electronic health records. In *American Medical Informatics Association* 2012;901–910.
9. Wu J, Roy J, Stewart W. Prediction modeling using EHR data: challenges, strategies, and a comparison of machine learning approaches. *Med Care* 2010;48(6):S106–S113.
10. Karpathy A, Li F. Deep visual-semantic alignments for generating image descriptions. *Computer Vision and Pattern Recognition (CVPR)* 2015:3128–3137. Boston, MA, USA.
11. Cho K, Van Merrienboer B, Gulcehre C, *et al*. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 2014:1724–1734. Doha, Qatar.
12. Hinton G, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. *Neural Comput* 2006;18(7):1527–1554.
13. Bengio Y. Learning deep architectures for AI. Foundations Trends Machine Learning. 2009;2(1):1–127.
14. Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)* 2012:1106–1114. Lake Tahoe, Nevada, United States.
15. Vincent P, Larochelle H, Bengio Y, Manzagol P-A. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine learning (ICML)* 2008:1096–1103. Helsinki, Finland.
16. Le Q, Ranzato M, Monga R, *et al*. Building high-level features using large scale unsupervised learning. In *International Conference on Machine Learning (ICML)* 2012, Edinburgh, Scotland, UK.
17. Lee H, Pham P, Largman Y, Ng A. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems (NIPS)* 2009;1096–1104. Vancouver, British Columbia, Canada.
18. Hinton G, Deng L, Yu D, *et al*. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *Signal Process Mag* 2012;29(6):82–97.
19. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In *arXiv preprint arXiv:1301.3781* 2013.
20. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)* 2013:3111–3119. Lake Tahoe, Nevada, United States.
21. Socher R, Pennington J, Huang E, Ng A, Manning C. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Empirical Methods in Natural Language Processing* (EMNLP). 2011:151–161. Edinburgh, UK.
22. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;9(8):1735–1780.
23. Grosicki E, El Abed H. ICDAR 2009 handwriting recognition competition. In *International Conference on Document Analysis and Recognition* 2009:1398–1402. Barcelona, Spain.
24. Sak H, Senior A, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *International Speech Communication Association* 2014;338–342. Singapore.
25. Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization. In *arXiv preprint arXiv:1409.2329* 2014.
26. Luong M-T, Sutskever I, Le Q, Vinyals O, Zaremba W. Addressing the rare word problem in neural machine translation. In *Association for Computational Linguistics (ACL)* 2015:11–19. Beijing, China.
27. Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning* (ICML). 2015:2342–2350. Lille, France.
28. Lasko T, Denny J, Levy M. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PloS One* 2013;8(6):e66341.
29. Che Z, Kale D, Li W, Bahadori M, Liu Y. Deep computational phenotyping. In *Knowledge Discovery and Data Mining* (KDD). 2015:507–516. Sydney, NSW, Australia.
30. Hammerla N, Fisher J, Andras P, Rochester L, Walker R, Plotz T. PD disease state assessment in naturalistic environments using deep learning. In *AAAI* 2015. 1742–1748. Austin, Texas, USA.
31. Lipton Z, Kale D, Elkan C, Wetzell R. Learning to diagnose with LSTM recurrent neural networks. In *arXiv preprint arXiv: 1511.03677* 2016.
32. Minarro-Gimenez J, Marin-Alonso O, Samwald M. Exploring the application of deep learning techniques on medical text corpora. *Stud Health Technol Inform* 2013;205:584–588.
33. De Vine L, Zuccon G, Koopman B, Sitbon L, Bruza P. Medical semantic similarity with a neural language model. In *International Conference on Information and Knowledge Management* (CIKM). 2014;1819–1822. Shanghai, China.
34. Choi Y, Chiu C, Sontag D. Learning low-dimensional representations of medical concepts. In *American Medical Informatics Association on Clinical Research Informatics* 2016. San Francisco, CA.
35. Choi E, Schuetz A, Stewart W, Sun J. Medical concept representation learning from electronic health records and its application on heart failure prediction. In *arXiv preprint arXiv:1602.03686* 2016.
36. Tangri N, Stevens L, Griffith J, *et al*. A predictive model for progression of chronic kidney disease to kidney failure. *JAMA* 2011;305(15): 1553–1559.

37. Sukkar R, Katz E, Zhang Y, Raunig D, Wyman B. Disease progression modeling using hidden Markov models. In *Engineering in Medicine and Biology Society* 2012:2845–2848.
38. Zhou J, Liu J, Narayan V, Ye J. Modeling disease progression via multi-task learning. *NeuroImage* 2013;78:233–248.
39. Liu Y-Y, Ishikawa H, Chen M, Wollstein G, Schuman J, Rehg J. Longitudinal modeling of glaucoma progression using 2-dimensional continuous-time hidden Markov model. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* 2013:444–451. Nagoya, Japan.
40. Schulam P, Saria S. A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *Advances in Neural Information Processing Systems (NIPS)* 2015:748–756. Montreal, Quebec, Canada.
41. Wang X, Sontag D, Wang F. Unsupervised learning of disease progression models. In *Knowledge Discovery and Data Mining (KDD)* 2014:85–94. New York, NY, USA.
42. Choi E, Du N, Chen R, Song L, Sun J. Constructing disease network and temporal progression model via context-sensitive Hawkes process. In *International Conference on Data Mining (ICDM)* 2015:721–726. Atlantic City, NJ, USA.
43. Goldberg Y. A primer on neural network models for natural language processing. In *arXiv preprint arXiv:1510.00726* 2015.
44. Bishop C. *Pattern Recognition and Machine Learning (Information Science and Statistics)* New York: Springer-Verlag; 2006.
45. Bergstra J, Breuleux O, Bastien F, *et al*. Theano: a CPU and GPU Math Expression Compiler. In *Python for Scientific Computing Conference* 2010.
46. Greenland S, Thomas D. On the need for the rare disease assumption in case-control studies. *Am J Epidemiol*. 1982;116(3):547–553.
47. Vijayakrishnan R, Steinhubl S, Ng K, *et al*. Prevalence of heart failure signs and symptoms in a large primary care population identified through the use of text and data mining of the electronic health record. *J Cardiac Failure* 2014;20(7):459–464.
48. Gurwitz J, Magid D, Smith D, *et al*. Contemporary prevalence and correlates of incident heart failure with preserved ejection fraction. *Am J Med* 2013;126(5):393–400.
49. Clinical Classifications Software (CCS) for ICD-9-CM. Agency for Healthcare Research and Quality. https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp. Accessed April 2016.
50. Medi-Span Electronic Drug File (MED-File) v2. Wolters Kluwer Clinical Drug Information. http://www.wolterskluwercdi.com/drug-data/medi-span-electronic-drug-file/. Accessed April 2016.
51. Clinical Classifications Software for Services and Procedures. Agency for Healthcare Research and Quality. https://www.hcup-us.ahrq.gov/toolssoftware/ccs_svcsproc/ccssvcproc.jsp. Accessed April 2016.
52. Zeiler M. ADADELTA: An adaptive learning rate method. In *arXiv preprint arXiv:1212.5701* 2012.
53. Nwankwo T, Yoon SS, Burt V, Gu Q. Hypertension among adults in the United States: National Health and Nutrition Examination Survey, 2011-2012. *NCHS Data Brief* 2013;113:1–8.
54. CDC. *Coronary Artery Disease (CAD)* Centers for Disease Control and Prevention. http://www.cdc.gov/heartdisease/coronary_ad.htm. Accessed July 2016.
55. Karpathy A, Johnson J, Li F. Visualizing and understanding recurrent networks. In *arXiv preprint arXiv:1506.02078*, 2015.