Simple Recurrent Neural Network is All we Need for Clinical Events Prediction using EHR

Laila Rasmy  
*School of Biomedical Informatics  
University of Texas Health Science Center in Houston of organization*Houston, Texas, USA   
Laila.rasmy.gindybekhet@uth.tmc.edu

Yujia Zhou  
*School of Biomedical Informatics  
University of Texas Health Science Center in Houston of organization*Houston, Texas, USA

Yang Xiang  
*School of Biomedical Informatics  
University of Texas Health Science Center in Houston of organization*Houston, Texas, USA  
Jie Zhu  
*Department of Biostatistics, School of Public Health  
University of Texas Health Science Center in Houston of organization*Houston, Texas, USA

Hong Thoai Tran

*Department of Biostatistics, School of Public Health  
University of Texas Health Science Center in Houston of organization*Houston, Texas, USA

Hua Xu  
*School of Biomedical Informatics  
University of Texas Health Science Center in Houston of organization*Houston, Texas, USA

[Hua.Xu@uth.tmc.edu](mailto:Hua.Xu@uth.tmc.edu)

Zhiheng Li  
*School of Computer Science and Technology*

*Dalian University of Technology*

Dalian, China

Firat Tiryaki  
*School of Biomedical Informatics  
University of Texas Health Science Center in Houston of organization*Houston, Texas, USA

Degui Zhi  
*School of Biomedical Informatics  
University of Texas Health Science Center in Houston of organization*Houston, Texas, USA   
[Degui.Zhi@uth.tmc.edu](mailto:Degui.Zhi@uth.tmc.edu)

*Abstract*—*Recently it is of great interest to investigate the application of deep learning models for the prediction of clinical events using electronic health records (EHR) data. In EHR data, the longitudinal history of a patient is often represented as sequence data, where each patient can be represented as a sequence of visits, and each visit contains multiple events. As a result, deep learning models developed for sequence modeling, especially recurrent neural networks (RNNs), were developed for EHR-based clinical events prediction* [1]*.*

Keywords—Deep Learning, Patient-Specific Modeling, Electronic Health Records

# Introduction

Recently we witness tremendous efforts in applying deep learning models to predict clinical events as electronic health records (EHR) data are becoming increasingly available. Deep learning methods are being developed to predict mortality, early readmission, transfer to ICU ,extensive length of stay, and disease onset risks [2][3]–[5]. Even though deep learning models showing promising results to predict diagnoses given features from the EHR, leveraging the dependencies of multiple comorbidities in a heterogeneous population of patients remains a very challenging problem [6].

Because of the temporality of a patient’s health records, EHR data are often represented as sequence data, where each patient can be represented as a sequence of visits, and each visit contains multiple events. Therefore, the rich set of deep learning models developed for sequence modeling, mainly from NLP, audio, and video modeling community, can be applied to EHR data. Recurrent neural networks with gating mechanisms [7], [8] for passing long-term dependencies are extensively investigated in NLP tasks and are promising network architectures for EHR data modeling. Such networks include Long and Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) cell architectures. Indeed, multiple LSTM/GRU-based methods for EHR-based prediction have been developed [3], [5], [9], [10]

In order to move this field forward, a rigorous evaluation of various methods is needed. Although there are a few benchmarking studies using clinical data comparing deep RNN models with traditional methods [11]–[13], there is a lack of comprehensive comparison among these methods. In particular, while these methods are often with various architectural innovations, there is a lack of direct systematic benchmarking on the effectiveness of these methodological innovations. These methods are developed and evaluated on different datasets. Also, the choices of their hyper-parameters are not necessarily consistent. Moreover, there are a number of recent architectural innovations from NLP but have not been evaluated in the context of EHR-based prediction. The field of NLP has obviously leading in terms of methodological development and several extensive evaluations of various architectural variations has been conducted [cite]. We hope to bring the same rigor into the field of EHR-based prediction.

Here, we aim to offer a comprehensive evaluation of RNN-based methods in modeling EHR data. We evaluate 2 non-RNN models and 9 major types of RNN architectures. We engaged a thorough hyper-parameter tuning procedure using Gaussian Process-based Bayesian optimization. All models were evaluated against two prediction problems: the first is the prediction of the patient risk to develop Heart failure and the second is the risk of early readmission for patients with any disease of those publicly reported by CMS. We extracted our cohorts from the Cerner Health Fact database for the evaluation of RNN-based methods. Our Heart failure cohort data set includes 152,790 cases and 1,152,517 controls, with over 10 million encounters and 160 million events, and for one hospital 5,010 cases + 37,719 controls. we used one hospital data for the hyper-parameters optimization purposes. The readmission cohort is of 161,748 cases and 257,991 controls.

# Methods

## Problem Formulation

### Heart Failure Prediction

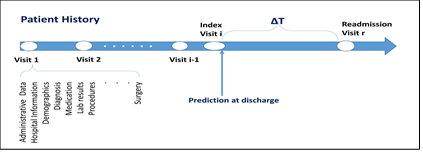
Cardiovascular diseases are the leading causes of mortality worldwide [[1]](https://www.sciencedirect.com/science/article/pii/S1532046418301175" \l "b0005). Among them, heart failure is a severe healthcare burden: per Centers of Disease Control and Prevention (CDC) and the American heart association (AHA), there were more than 5 million adult heart failure patients in the US in 2016, and that costs the nation more than $30 billion annually [[2]](https://www.sciencedirect.com/science/article/pii/S1532046418301175" \l "b0010). Estimating the risk for disease development can help early disease management and thus improving the health outcomes. Developing clinical event prediction models improves the quality of care especially when it is translated into real-time tools available at the bedside.

[need to add problem formulation]

### Risk of Early Readmission Prediction

Nearly 15 percent of hospitalized patients are readmitted within 30 days of discharge, the cost of unplanned readmissions is 15 to 20 billion dollars annually. In an effort to reduce the readmission rate and such associated costs. Center of Medicaid and Medicare Systems (CMS) began to publicly report 30-day early readmission for acute myocardial infarction (AMI), Heart Failure (HF), and pneumonia (PN) in 2009 and in 2014 they added Chronic obstructive pulmonary disease (COPD) and ischemic stroke (IS) on the CMS Hospital Compare website. Additionally, the Secretary of the Department of Health and Human Services (HHS) as per the Affordable Care Act established a Hospital Readmissions Reduction Program (HRRP) to reduce payments to hospitals for excess readmissions starting the Fiscal Year 2013. Therefore, Hospitals started to focus their efforts to reduce the early readmissions rate not only for the financial penalty but also for their effectiveness reputability. Identification of patients with high risk for early readmission at the time of discharge can help taking necessary measures to reduce such risk.

In this study we will predict the patient probability to readmit earlier than thirty days after their first inpatient admission with HF, AMI, PN, IS, or COPD, known as the index encounter. Fig 1. is an illustrative figure for the problem definition. Our main outcome is a binary variable that reflects if Δ T < 30 or not. We will train our models with all patients historical data available until the discharge of the index encounter. Patient data includes clinical information including diagnosis, medication, procedures and laboratory results, as well as patient demographics, hospital geographic information, and encounter level administrative information. Supplemental Table 1. includes a list of main covariates categories and total counts. We had in total over 15000 variables mainly clinical.



1. Problem definition

## Data Extraction

### Dataset

Cerner HealthFacts® database is derived from over 600 Cerner implementation in hospitals and clinics throughout the United States. It contains timestamped clinical information for over 50 million unique patients with more than 10 years of records. In total there are more than 110 million patient visits. These clinical data are mapped to the most common standards, for example, diagnoses and procedures are mapped to the International Classification of Diseases (ICD) codes, medications information include the national drug codes (NDCs), and laboratory tests are linked to their LOINIC codes. The data are de-identified and comply with the patient confidentiality requirements of the Health Insurance Portability and Accountability Act (HIPAA).

### Cohort Definition

#### Heart Failure Cohort

We followed the [case definition](https://www.sciencedirect.com/topics/medicine-and-dentistry/case-definition) and [case-control](https://www.sciencedirect.com/topics/medicine-and-dentistry/case-control-study) matching procedure described by Choi et al for heart failure prediction study [3] to construct our cohort dataset from Cerner Health Facts® EMR data. This is the same cohort we used for our previous work [14]. Cases were defined as patients who meet the following two criteria: (1) At least three heart failure related encounters had to occur within 12 months; (2) ≥50 years old at the time of the first HF diagnose. The date of the first diagnosis is designated as the index date of the patient. For each case, up to 10 controls were matched by a primary care hospital, sex, and age (five-year interval). Further, to match the time span of records in the Cerner Health Facts®, controls are required to have their first visit within one year of the first office visit of the matching case and have at least one visit a month before or any time after the diagnosis date of the matching case.

We further cleaned the extracted data to exclude all cases that showed any prior history of heart failure, as well as controls with any heart failure incidence before or after the index date with 180 days. In addition, we ensured that all the cases we used have at least one matched control and we cleaned up any redundant records. As a result, we obtained 152,790 cases and 1,152,517 controls.

#### Readmission Cohort

As previously described under the problem formulation, We consider the first inpatient encounter with any of the above-mentioned diseases as the index encounter. We followed the CMS definition of early in-hospital readmission within 30 days as per CMS guidelines [15]. We excluded any planned readmissions based on the encounter admission source or the index encounter discharge disposition, as well as recurring or transfer encounters. We define our cases as patients who readmitted within 30 days of their previous hospitalization and we define our controls as patients who get readmitted after 30 days. We also restricted our cohort to patients who had at least one encounter before their index encounter regardless of the encounter type. Supplementary Table II includes the logic used to define our cohort from the Cerner health facts® database using SQL queries.

## Baseline Models

### Logistic Regression

Logistic regression is the state-of-the-art model currently in use at clinical practice. For a baseline comparison, we had two versions of logistic regression. The first uses the sci-kit learn python package, logistic regression function, which directly take the aggregated one-hot vector of a patient as input. The second uses a single linear function after the embedding layer using the pytorch framework.

The difference between the two models is showing the impact of the use of the embedding layer on the model prediction accuracy.

### Random Forest

Another strong baseline is a random forest, so we used the sci-kit learn python package random forest classification package and tested the performance for our baseline.

## Recurrent Neural Network models

### Basic RNN models

RNNs are neural network models that take a sequence of vectors, one at a time step, as input, and output a sequence of vectors or numbers at each time step. For our prediction task, we commonly consider the output at the last time step as it represents the patient risk at the time of discharge of the index encounter. RNN map the dependency from a one-time step to the next through the sequence of hidden vectors used to represent information passed on from one time step to the next. Recurrent computations are conducted at the “RNN cell” which is the unit that maps the input and hidden vector from the previous time step to the output and the hidden vector to the next time step. There are three main types of RNN cells, namely vanilla RNN, Long and Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). The vanilla RNN cell [16], [17] consists of a single non-linear function. the major drawback for the vanilla RNN that it cannot keep the memory for longer sequences for vanishing gradient [18], [19]. LSTM is a special kind of RNN that is capable of learning long-time dependencies. It was first introduced in 1997 [7]. The LSTM uses the idea of “Constant Error Flow” for RNNs to create a “Constant Error Carousel” (CEC) which ensures that gradients don’t decay, and therefore prevents back-propagated errors from vanishing or exploding. The LSTM cell has three additional gates namely forget, input and output gates [20], a gate is mainly a sigmoid function. While GRU is one of the latest most popular modifications of basic RNN cell introduced in 2014 [8], GRU used similar gating mechanism to controls the flow of information similarly like LSTM, but without the output gate. That is, it exposes the full hidden content without any control. The performance of GRU is on par with LSTM, but computationally more efficient [21].

We evaluate the performance of each cell type using basic RNN functions available within PyTorch v.1 framework [22]

### Bidirectional RNN models

RNN is by default dealing with the sequence input in one direction. In 1997 the idea of using both past and future data to calculate the current time step hidden state was published on [23]. The outputs of two RNNs on both directions are concatenated together to produce output signals. This technique proved to be especially useful when combined with LSTM RNNs [18], [24], [25] as learning from future time steps to better understand the context and eliminate ambiguity. For the implementation of the Bidirectional RNN, we set the bidirectional option of different cell type to true, when defining the model and to calculate the final output we used the output of the last hidden layer from the first forward direction and the output at the first step for the reverse direction. We tested bidirectional RNN, LSTM and GRU cells.

### Dilated RNN models

Dilated RNN was introduced in 2017[26] inspired by the dilated causal convolutional neural network (CNN) architecture used by Wavenet. DRNN is a multi-layer cell-independent architecture characterized by multi-resolution dilated recurrent skip connections.

The main advantage of this architecture is to improve the ability of recurrent models to learn long-term dependency faster and with fewer parameters. For DRNN, we utilized the code available on <https://github.com/zalandoresearch/pt-dilate-rnn> after review.

### Time Aware LSTM (TLSTM)

T-LSTM is a variation of the LSTM cell, that proposed add an additional time-aware gate within the LSTM cell that degrades the memory based on the duration between two consecutive patient’s visits [27]. Therefore, the main advantage of T-LSTM that it captures the dependency of sequential data along with time irregularities and assumed to be associated with higher prediction accuracy.

### Quasi RNN (QRNN)

QRNN was developed by the salesforce research team in 2016 (Bradbury, et al., 2016), it alternates convolutional layers and a recurrent pooling function in across time steps. The main advantage of QRNN models is the fast speed on model training and evaluation, along with better predictive accuracy with stacked layers.

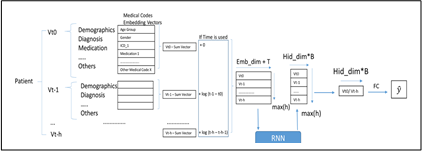
## Bayesian Optimization

For hyper-parameters selection for the models above, we tuned the parameters on a sample set of a single hospital using Bayesian Optimization package available on <https://github.com/fmfn/BayesianOptimization>. Bayesian optimization (BO) is a Sequential Model-based Global Optimization that uses a Gaussian Process to compute the expected improvement of any point in the search space that will lead to better results[29] [30]. As BO gives better results with continuous hyper-parameters, So we designed our hyperparameter tuning as follows:

1. Each model architecture we did grid search on categorical parameters like model parameters optimization function ‘optimizer’, and then run BO on other numerical hyper-parameters for each model optimizer combination.
2. For optimizer related hyper-parameters like Learning rate, L2 penalty, and Numerical tolerance (epsilon), BO search was set on the Log10 space, while for the embedding and hidden dimensions, we used the Log2 space.
3. For the Bi-directional RNN, we limited our initial search for a single layer for computational resources utilization, while we search multi-layers up to 3 layers for unidirectional architecture.
4. For Dilated RNN, the number of layers was searched as integer values between 1 and 4, and the dilation depth was calculated based on it.
5. For ‘Adadelta’ optimizer, we were overriding the learning rate to 1, as other values in the search space were associated with much lower performance.
6. In general, we used the Matern Kernel for the BO

## Evaluation Criteria

Our cohort is split into train, validation and test sets on the 7:1:2 ratio. We mainly use the area under the ROC curve (AUROC) as our main evaluation criteria among all models. We select the best model from the BO based on the validation set AUROC, and we report the Test AUC for the best models. Models comparison is based on Test AUC.



1. The proposed model training framework

# Results

Preliminary results from the BO search as appears in Table I showed that bidirectional LSTM or GRU models trained with RMSprop or any other adaptive optimizers is associated with the highest prediction accuracy with validation AUROC of around 72% and AUROC on test set of around 69% where baseline logistic regression and random forest showed AUROC around 60% and 64% respectively on the test set.

Retrain the models on the full cohort data set using the same hyperparameters associated with the best model (Bi-LSTM) as appear in Table I showed an AUROC of (xxx) on the validation set and AUROC of (xxx) on test set while LR showed an AUC of around 59% on test set and RF of (xxx).

# Discussion & Conclusion

We trained a large number of models with hyperparameter variations on each cohort. We identified the best model within a specific category based on the validation AUC. Figure 1 is showing the AUC on the Test set. Interestingly, the best performing model was the simple unidirectional GRU. In addition, we tested ensembling GRU with LR but that does not improve the model. Overall, we found our results inline with NLP researchers observations[31], [32]. Baseline RNN models such as GRU are often sufficient for predictive modeling tasks in EHR. One of the limitations of the current study is that we focused on a single hospital, due to constraints of computational resources. We plan to test and compare on the full cohort that includes hundreds of thousands of patients across multiple hospitals to further verify our conclusions.

Consistently the preliminary results of this study showed that RNN based models are outperforming the prediction accuracy of common machine learning models like LR and RF. Still, basic RNN models lack the interpretability advantage available with such models. Future work to consider adding attention layers over the RNN model for interpretability.

Although the key conclusion of this experiment shows is consistent with results from a previous experiment done on different cohort extracted from an older version of Cerner Healthfacts®, The results of the preliminary Bayesian optimization in this study experiments are somehow different from the results of a similar study previously. The previous study showed that a unidirectional GRU based model trained with an adagrad optimizer is associated with better prediction accuracy (reported AUROC on the test set was 75.5%). We can reflect that for two main reasons, the first is the number of covariates, the maximum numbers of covariates in the previous study was around 20,000 while the covariates in this experiment were around 85,000. The second reason is the criteria for the cohort definition, the previous study identified cases as patients with any incidence of early readmission during their full history in contrast to the criteria in this study where cases are patients who readmit before 30 days from their first inpatient hospitalization discharge date.

This work is still in progress, the results of Logistic regression on top of embedding is still pending. Additionally, we need to use SCIKITOPT for Bayesian optimization for baseline models. An additional consideration for the use of Auto-Keras for model search.

1. results of Bayesian optimization

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | |  | | |  | |  |  | |  | | Single Hospital Result | | | | | Full Cohort Result | |
| Cohort | Model | Emb  Size | | Hidden | | | | Optimizer | | LR | | | L2 | | EPs | Valid  AUC | @Epoch | Test  AUC | | | Valid  AUC | Test  AUC |
| Heart Failure  Prediction | GRU | | 256 | | 128 | | Adagrad | | | 0.05 | | | 0.0003 | | 1e-06 | 83.4 | 8 | | 84.8 | 82.2 | | 82.3 |
| Bi-GRU | | 256 | | 64 | | Adamax | | | 0.008 | | | 0.0002 | | 0.0001 | 83.4 | 4 | | 84.5 | 82.0 | | 82.1 |
| Bi-LSTM | | 256 | | 64 | | Adagrad | | | 0.1 | | | 0.0001 | | 0.0001 | 83.6 | 5 | | 84.4 | 82.2 | | 82.3 |
| LSTM | | 256 | | 64 | | Adagrad | | | 0.1 | | | 0.0001 | | 1e-06 | 83.2 | 6 | | 83.9 | 82.1 | | 82.2 |
| D-LSTM (2L) | | 64 | | 32 | | Adam | | | 0.002 | | | 0.0003 | | 1e-05 | 82.7 | 11 | | 83.3 | 81.7 | | 81.8 |
| D-GRU (2L) | | 64 | | 64 | | Adagrad | | | 0.025 | | | 2.4e-04 | | 1e-05 | 82.4 | 7 | | 83.3 | 82.0 | | 82.1 |
| QRNN | | 128 | | 32 | | Adagrad | | | 0.051 | | | 4e-06 | | 1e-06 | 83.1 | 3 | | 83.2 | 81.0 | | 81.2 |
| RETAIN | | 128 | | 128 | | Adadelta | | | 1 | | | 0.0001 | | - | 83 | 8 | | 83.8 | 82.2 | | 82.3 |
| Early Re-admission  Prediction | GRU | | 128 | | 128 | | Adagrad | | | 0.073 | | | 0.0043 | | 0.0001 | 75.4 | 22 | | 75.5 | 75.3 | | 74.7 |
| Bi-LSTM | | 64 | | 64 | | Adadelta | | | 1 | | | 0.003 | | 0.0001 | 75.1 | 17 | | 75.2 | 74.5 | | 73.9 |
| Bi-GRU | | 64 | | 64 | | Adadelta | | | 1 | | | 0.0027 | | 1e-06 | 76.2 | 13 | | 74.4 | 75.1 | | 74.6 |
| LSTM | | 128 | | 64 | | Adadelta | | | 1 | | | 0.0043 | | 1e-05 | 74.4 | 13 | | 73.8 | 73.9 | | 73.6 |
| QRNN | | 128 | | 128 | | Adagrad | | | 0.047 | | | 2.4e-05 | | 0.0001 | 73.5 | 1 | | 71.5 | 72.1 | | 71.9 |
| D-GRU (2L) | | 128 | | 64 | | Adagrad | | | 0.077 | | | 0.0013 | | 1e-07 | 74.5 | 6 | | 73.5 | 74.6 | | 74.2 |
| D-LSTM (2L) | | 128 | | 128 | | Adadelta | | | 1 | | | 0.0018 | | 0.0001 | 75.1 | 20 | | 72.8 | 70.2 | | 70 |
| RETAIN | | 128 | | 128 | | Adadelta | | | 1 | | | 0.0001 | | - | 71.9 | 9 | | 70.1 | 75.8 | | 76.3 |

##### Acknowledgment

Thanks to Dr.Luca Giancardo, Dr. Susan Fenton and Dr. Degui Zhi for their guidance and support. Special Thanks for SBMI Data Service Center team who facilitate the data extraction.

##### References

[1] C. Xiao, E. Choi, and J. Sun, “Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review,” *J. Am. Med. Informatics Assoc.*, Jun. 2018.

[2] A. Rajkomar *et al.*, “Scalable and accurate deep learning with electronic health records,” *npj Digit. Med.*, vol. 1, no. 1, p. 18, Dec. 2018.

[3] E. Choi, M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun, “RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism,” *Adv. Neural Inf. Process. Syst.*, pp. 3504–3512, 2016.

[4] T. Lagu *et al.*, “Validation and Comparison of Seven Mortality Prediction Models for Hospitalized Patients With Acute Decompensated Heart Failure.,” *Circ. Heart Fail.*, vol. 9, no. 8, p. e002912, Aug. 2016.

[5] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, “Doctor AI: Predicting Clinical Events via Recurrent Neural Networks.,” *JMLR Workshop Conf. Proc.*, vol. 56, pp. 301–318, Aug. 2016.

[6] X. Zhang, R. Henao, Z. Gan, Y. Li, and L. Carin, “Multi-Label Learning from Medical Plain Text with Convolutional Residual Models,” Jan. 2018.

[7] S. Hochreiter and J. J. Urgen Schmidhuber, “Long Short-Term Memory,” *Mem. Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.”

[9] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, “Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis,” *IEEE J. Biomed. Heal. Informatics*, pp. 1–1, 2017.

[10] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep learning for healthcare: review, opportunities and challenges,” *Brief. Bioinform.*, May 2017.

[11] H. Harutyunyan, H. Khachatrian, D. C. Kale, and A. Galstyan, “Multitask Learning and Benchmarking with Clinical Time Series Data,” Mar. 2017.

[12] S. Purushotham, C. Meng, Z. Che, and Y. Liu, “Benchmarking deep learning models on large healthcare datasets,” *J. Biomed. Inform.*, vol. 83, pp. 112–134, Jul. 2018.

[13] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent Neural Networks for Multivariate Time Series with Missing Values,” Jun. 2016.

[14] L. Rasmy *et al.*, “A study of generalizability of recurrent neural network-based predictive models for heart failure onset risk using a large and heterogeneous EHR data set,” *J. Biomed. Inform.*, vol. 84, 2018.

[15] “QualityNet - Readmission Measures.” [Online]. Available: http://www.qualitynet.org/dcs/ContentServer?c=Page&pagename=QnetPublic%2FPage%2FQnetTier2&cid=1219069855273. [Accessed: 27-Sep-2018].

[16] F. J. Pineda, “Generalization of back-propagation to recurrent neural networks,” *Phys. Rev. Lett.*, vol. 59, no. 19, pp. 2229–2232, Nov. 1987.

[17] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 79, no. 8, pp. 2554–8, Apr. 1982.

[18] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A Critical Review of Recurrent Neural Networks for Sequence Learning,” 2015.

[19] Y. Bengio, P. Simard, P. F. on neural networks, and undefined 1994, “Learning long-term dependencies with gradient descent is difficult,” *comp.hkbu.edu.hk*.

[20] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: continual prediction with LSTM.,” *Neural Comput.*, vol. 12, no. 10, pp. 2451–71, Oct. 2000.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” Dec. 2014.

[22] “PyTorch documentation — PyTorch master documentation.” [Online]. Available: https://pytorch.org/docs/stable/index.html. [Accessed: 28-Sep-2018].

[23] M. Schuster and K. K. Paliwal, “Bidirectional Recurrent Neural Networks,” *IEEE Trans. SIGNAL Process.*, vol. 45, no. 11, 1997.

[24] T. Thireou and M. Reczko, “Bidirectional Long Short-Term Memory Networks for Predicting the Subcellular Localization of Eukaryotic Proteins,” *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 4, no. 3, pp. 441–446, Jul. 2007.

[25] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM networks,” in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4, pp. 2047–2052.

[26] S. Chang *et al.*, “Dilated Recurrent Neural Networks,” Oct. 2017.

[27] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, “Patient Subtyping via Time-Aware LSTM Networks,” vol. 10, 2017.

[28] J. Bradbury, S. Merity, C. Xiong, and R. Socher, “Quasi-Recurrent Neural Networks,” Nov. 2016.

[29] R. P. Adams and O. Stegle, “Gaussian process product models for nonparametric nonstationarity,” in *Proceedings of the 25th international conference on Machine learning - ICML ’08*, 2008, pp. 1–8.

[30] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the Human Out of the Loop: A Review of Bayesian Optimization,” *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.

[31] G. Melis, C. Dyer, and P. Blunsom, “On the State of the Art of Evaluation in Neural Language Models,” Jul. 2017.

[32] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An Empirical Exploration of Recurrent Network Architectures.”

Supplemental Table I. List of Covariates used to train the models

|  |  |  |  |
| --- | --- | --- | --- |
| Category | Covariate | Abbreviation | categories count |
| Clinical | Diagnoses | D | 44086 |
| Laboratory results (in Cerner codes) | Lc | 356 |
| Laboratory results (in Loinic codes) | L | 10014 |
| Medication (generic names) | M | 2032 |
| Procedures | P | 26335 |
|  | | **82,823** |
| Demographic | Age | a | 89 |
| Gender | g | 3 |
| Marital Status | m | 7 |
| Race | r | 10 |
|  | | **109** |
| Hospital | Acute Status | HAS | 2 |
| Bed size | HBS | 7 |
| Census Division | CDv | 9 |
| Census Region | CRg | 4 |
| Health System id | HSi | 89 |
| Hospital id | Hid | 514 |
| Urban / Rural | HUR | 2 |
|  | | **627** |
| Administrative | Admission Source | Ads | 21 |
| Admission Type | Adt | 9 |
| Billing Indicator | Bil | 3 |
| Caresetting | Cst | 170 |
| Diagnoses related group (DRG) description | DRG | 1265 |
| Discharge Disposition | dsd | 35 |
| Encounter Type | Ptp | 40 |
| Insurance | Ins | 23 |
| Length of Stay | LOS | 7 |
| Medical Diagnosis Code (MDC) description | MDC | 26 |
| Physician Medical Speciality | PMs | 136 |
|  | | **1,735** |
| Total count of Covariates | | | 85,294 |

Supplemental Table II. List of Covariates used to train the models

| Pseudo-code for Cohort definition |
| --- |
| 1. Inpatient encounter at acute care hospital that last for at least 24 hours (i.e. discharged\_dt\_tm - admitted\_dt\_tm > 24hrs) 2. Age >=18 3. First inpatient encounter with any of the above diagnosis will be considered the index encounter 4. The next inpatient encounter regardless the hospital or the diagnoses i.e. (any inpatient admission) will be considered the readmission encounter, need to be at least 1 day apart from the index encounter 5. Cases are patients readmitted less than 30 days after the index encounter discharge date i.e. (readmission encounter admitted\_dt\_tm - index encounter discharged\_dt\_tm < 30 days and >= 1 day), and controls (readmission encounter admitted\_dt\_tm - index encounter discharged\_dt\_tm < 30 days 6. For data quality issue you might face the case of 7. missing admission dates --- need to exclude those encounters from the case scenario identification as we normally nvl null admission with the discharged\_dt\_tm and hence the duration of the encounter become 0 (less than 1 day). 8. encounters dates / duration overlapping ---- so you might find (readmission encounter admitted\_dt\_tm - index encounter discharged\_dt\_tm < 1 or 0). So, we exclude those patients from the study for simplicity. 9. Different encounters need to be at least 1 day apart. |

