Journal of Machine Learning Research 1 (2020) 1-48 Submitted 4/00; Published 10/00

Mortality Prediction using Electronic Health Records

Poornima Haridas POORNIMA.HARIDAS@CS.NYU.EDU
Department of Computer Science, Courant Institute of Mathematical Sciences
New York University
New York, NY 10003, USA

Yuan Ding YD1400@NYU.EDU
Center for Data Science
New York University
New York, NY 10003, USA

Editor: N/A

Abstract

Quantifying patient health by predicting the mortality is an important problem in critical care research. An accurate prediction of patient mortality can help with the assessment of severity of illness and guide decision making in terms of drastic measures required to save a patient. In this paper, we propose a simple Graph Neural Network based architecture for mortality prediction of patients in the ICU and compare it to a Recurrent Neural Network based benchmark model (GRU). GNNs are known to help extract information from data that can be inherently represented as a graph and this contributes to the explain-ability of the deep learning model which is becoming increasingly essential to all clinical prediction tasks. We find that, GNNs perform equivalent to RNNs and that deep learning models perform better when raw features are used.

Keywords: EHR, Mortality, GCNs, GRU

1. Introduction and Hypothesis

Deep learning models have been applied to various fields of research like natural language processing, computer vision and speech recognition to achieve outstanding results. The increasing availability of large healthcare databases like Medical Information Mart for Intensive Care (MIMIC-II and III), has spurred the use of these models for clinical healthcare applications as well. One of the most popular tasks in the domain is predicting patient mortality. It is an important clinical outcome for an ICU admission, and accurately predicting it helps assess the severity of illness and determining the value of various interventions and treatments. Several machine learning as well as deep learning models have been applied to the task with varying levels of success. The success and popularity of deep learning models could be attributed to the fact that these models require minimal feature engineering as they learn abstract representations of the data.

Graph Neural Networks (GNNs, Scarselli et al. (2008)) are a relatively new and upcoming type of neural network that operates on graph data structures and learns the structural features of the graph. Graph Convolutional Networks (GCNs) are a type of GNN that perform convolutions on a graph and produces meaningful node embeddings that can be

then used for node level tasks or aggregated to use for graph level tasks. GNN based architectures are being increasingly explored to be able to increase the explainability of the deep learning model and open the 'black-box' that most deep learning algorithms are known to be.

In this paper we compare the performance of a deep GCN model against a GRU model for the task of mortality prediction. GNNs are known to help extract information from data that can be inherently represented as a graph. We model patient records as fully connected graphs and use GCNs to create better embeddings of records that is then used for mortality prediction. Our **hypothesis** is that they would perform better than RNN based architectures and that deep learning models perform better with raw features than with processed features. To test our hypothesis, we evaluated a GRU model against a simple Graph Convolutional Network (GCN) based model on a couple of evaluation metrics. To the best of our knowledge, this is the first time GCNs have been used for the task for mortality prediction.

2. Related Work

Various machine learning models (Cooper et al. (1997), Calvert et al. (2016)) that predict the mortality rates have been developed over the years while deep learning models (Hammerla et al. (2015), Lipton et al. (2015)) have proven to be useful with the availability of large amounts of raw data. GRU-D developed by Che et al., is a GRU based architecture used for the mortality prediction task. It takes advantage of the multivariate time-series data format. GCNs (Kipf and Welling, 2016) are GNNs (Scarselli et al., 2008) where convolutions are embedded in GNNs from a spectral perspective. Zhu and Razavian (2019) used GNNs on EHR for predicting Alzheimer's Disease. Johnson et al. also discussed reproducibility of results for the mortality prediction task on MIMIC.

Purushotham et al. (2017) compiled a benchmark of many models for various clinical prediction tasks on the MIMIC-II and III dataset. We pre-processed our data in the format recommended by them to be able to benchmark our GRU and GCN model.

3. Methodology

3.1 Dataset

The MIMIC III (Johnson et al., 2016; Pollard, 2016) is a publicly available critical care database maintained by the Massachusetts Institute of Technology (MIT)'s Laboratory for Computational Physiology. It integrates de-identified, comprehensive clinical data of patients admitted to an Intensive Care Unit (ICU) at the Beth Israel Deaconess Medical Center (BIDMC) in Boston, Massachusetts during 2001 to 2012. MIMIC-III contains data associated with 53,423 distinct hospital admissions for adult patients whose age is above 15 and 7,870 admissions for neonates. For benchmarking purposes, we only included the records of the first ICU admission of adult patients (36,093). We built the relational database `mimic` with 26 raw tables using PostgreSQL (Johnson et al., 2017a).

Tables 1 and 2 show some descriptive statistics of our dataset.

Table 1: Baseline characteristics. Continuous variables are presented as $Mean[\max, \min]$, binary or categorical variables as $Count(\%)$.

Item	Overall	Dead at hospital	Alive at hospital
# of admissions	36,093	4029	32064
Age	65.86 [15.06, 89.09]	73.85 [15.18, 89.05]	64.98 [15.06, 89.09]
Gender (Female)	15607 (43.24%)	1866 (46.31%)	13749 (42.88%)
Origin: Medical	29,867	3330	26,537
Origin: Emergency	6226	699	5527

Table 2: Mortality label ratio w.r.t total admissions

	In-hospital	2-day	3-day	# of admissions
Mortality	0.112	0.018	0.014	36,093

3.2 Data Pre-processing

The data pre-processing was done in two parts: data cleaning and feature extraction. The original dataset is quite sparse and dirty. In order to be consistent with the benchmark paper (Purushotham et al. (2017)), we handled three main issues with the extracted data.

- **Inconsistent units/data types**: For example, some of the prescriptions were recorded both in 'dose' and in 'mg' units. For a specific variable, we select the unit with the highest frequency as the major unit and convert the rest. Some variables we used in *chartevents* and *labevents* tables are recorded in both numeric and string data type. These were unified as well.
- **Ambiguous recordings**: Some variables had multiple recordings at the same time, and we took the average of them; sometimes the observation was recorded as a range rather than a single measurement, in which case the median of the range to represent the value at a certain time point was taken.
- **Missing values**: Forward-backward imputation was performed to fill-in the missing values. For a few patients where some features were completely missing in the 24/48-hr time window, and we performed mean imputations during the training/validation procedure.

For the feature extraction, in order to compare the model performance using processed features and that using raw features, we prepared two feature sets.

- **Feature set A** contains 17 "processed" features (12 sequential + 5 static). "Processed" here means that we dropped outliers according to common domain/medical knowledge and merged the relevant features into one single feature.
- **Feature set B** is a raw set containing 15 sequential features and 5 static features. Actually the features we are using here is the same as set A, but do not remove outliers and use separate raw features. E.g. consider PaO2 and FiO2 as individual features instead of calculating the PF-ratio.

For the sequential features, we generate the time series in 24 and 48-hr windows by sampling the data once per hour. The specific features extracted are available in Table 4 in the appendix.

3.3 Models

We developed two models for our task, mortality prediction: - a Recurrent neural network (RNN) since the dataset contains a lot of time series features and a GNN based GCN model to model better patient embeddings.

For the RNN based model, we chose the GRU framework due to its ability to capture long-term dependencies using memory and gating units and computational efficiency compared to LSTM. The framework of GRU model is shown in Fig. 1. First, all the time series features were fed to the GRU model (the GRU output dimension is 25 for 2/3 day mortality prediction and 128 for the in-hospital mortality prediction). The output was concatenated with the 5 static features and fed to five fully connected layers to get the final output label. For regularization, we add a dropout layer in the middle with a dropout rate of 0.2, and ℓ_2 regularizer is also used.

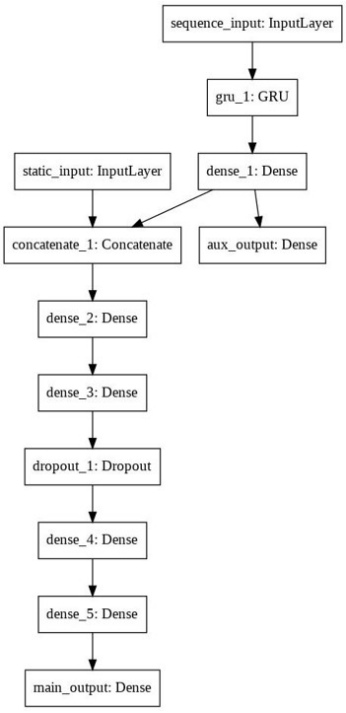


Figure 1: The GRU model

Our second model is a GNN based architecture (Fig.5) which comprises of two GCN layers followed by three fully connected layers. A third GNN model is in testing phase which replaces the fully connected layers with a global mean pooling layer on the node embeddings and adds a dropout layer to the first convolution result. The GNN models take in a fully connected graph whose nodes are time-series and static features (293 nodes). Each node has one feature, the value of the feature at a particular time interval.

All models were optimized for binary cross entropy loss with Adam optimizer. **Upsampling** was performed on the training set to deal with the extreme imbalanced mortality labels in the dataset. After upsampling, the ratio of positive/negative labels was roughly 0.4 vs 0.6.

4. Evaluation and Results

For each feature set, we use the 24-hr data to predict the 2-day mortality and in-hospital mortality, and 48-hr data to predict the 3-day mortality and in-hospital mortality respectively. We use Area under the ROC curve (AUROC) and Area under Precision-Recall Curve (AUPRC) as the **evaluation metrics** of the model's performance.

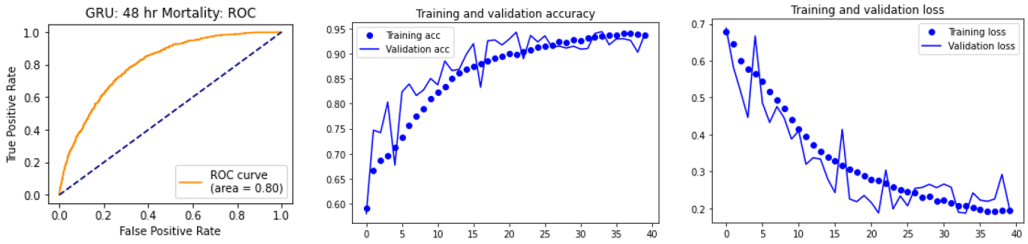
4.1 GRU model

We fine tuned three parameters: learning rate, dropout rate and parameter for ℓ_2 regularizer. The best results are shown in Table 3. We observe that: (i) Overall, GRU gives

a better prediction on long-term prediction task (in-hospital mortality) compared to the short-term prediction task (1-day after mortality); (ii) In all four cases, the prediction for set B outperforms the predictions for Set A; (iii) The performance using 24-hr data is even better than 48-hr data, indicating that a longer record length may not be necessary for the deep learning mortality prediction task. The best performance for both tasks are given by set B, 24 hrs data with AUROC of 0.82. This acceptable although the benchmark paper gives a higher score on RNN prediction (0.85-0.87). See Fig.2 and 3 for relevant plots.

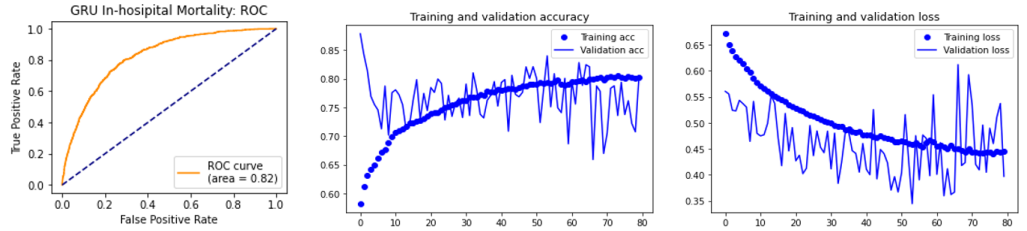
4.2 GNN model

The GNN model results are shown in Table 3 which was tested for only Set A, 24hrs. The ROC scores, accuracy on validation set and training loss plots for in-hospital predictions are shown in Fig.4. Our simple GCN architecture was able to achieve a 0.82 AUROC which is comparable to the GRU model. Limited experiments were performed on the GNN model due to time and resource constraints.



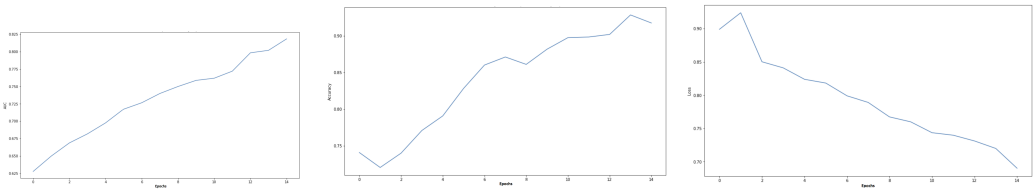
(a) The AUROC (b) The accuracy per epoch (c) The average loss per epoch

Figure 2: The plot for 2-day mortality prediction: set A, 24 hrs, GRU model



(a) The AUROC (b) The accuracy per epoch (c) The average loss per epoch

Figure 3: The plot for in-hospital mortality prediction: set A, 24 hrs, GRU model



(a) The ROC scores (b) The accuracy per epoch for validation set (c) The average loss per epoch for training set

Figure 4: Plots for in-hospital mortality prediction: set A, 24 hrs, GCN model

Table 3: The Results for Mortality Prediction (Benchmark results *)

Model	Feature Set	AUROC 1-day	AUROC In-hospital	AUPRC 1-day	AUPRC In-hospital
GRU	Set A, 24 hrs	0.79	0.81 (0.85*)	0.092	**0.343**
GRU	Set B, 24 hrs	**0.80**	**0.82 (0.87*)**	0.085	**0.342**
GRU	Set A, 48 hrs	0.77	0.78 (0.86*)	0.055	0.294
GRU	Set B, 48 hrs	0.78	0.79 (0.86*)	0.058	0.301
GNN	Set A, 24 hrs	0.5	**0.82 (0.87*)**	-	-

5. Conclusion and Discussion

Our tasks have verified our hypothesis that deep GNN models are an effective architecture for mortality prediction on EHR and that raw features help deep learning models learn better. Both models perform comparably for the in-hospital mortality prediction. The GRU model had better performance when raw features were used and in future experiments, we would like to test the same for the GNN models. A simple GCN architecture shows great promise with minimal training (45 epochs) when compared to several complex architectures. We believe that given more time, we could fine-tune the model to beat the current benchmark.

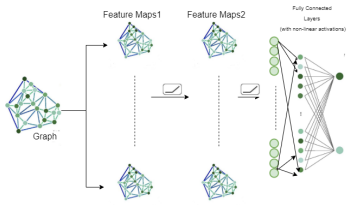


Figure 5: The GCN model

Due to time constraints, we only included 15 raw features in our model. In the future, we would like to train with more raw features to help increase the AUROC scores. We would also like to experiment with more complex GNN architectures (with concepts of attention) and explore packages that visualize graphs processed by the model in order to increase the explainability of the model and trust in the results being produced.

6. Team Work

Yuan & Poornima: Literature review / Data preprocessing / Presentation and report
Yuan: GRU implementation
Poornima: GNN implementation
Please go to our repository for the full version of codes used in this project:
`https://github.com/UTpH/dl_in_medicine`

Acknowledgments

We would like to thank our professors, Dr.Cem M. Deniz and Dr.Narjes Sharif Razavian for their guidance and support throughout the course.

Appendix A.

Table 4: 17 features used in Feature Set A

Items	**Table**	**Type**
Glasgow coma scale, Systolic blood pressure, Heart rate, Body temperature, and Pao2/fio2 ratio	chartevents	sequential
Serum urea nitrogen level, White blood cells count, Serum bicarbonate level, Sodium level, Potassium level, Bilirubin level	labevents	sequential
Urine output	outputevents	sequential
Acquired immunodeficiency syndrome (AIDS),Hematologic malignancy, and Metastatic cancer	diagnoses_icd	static
Age	icustays	static
Admission type	admissions	static

References

Jacob Calvert, Qingqing Mao, Angela J Rogers, Christopher Barton, Melissa Jay, Thomas Desautels, Hamid Mohamadlou, Jasmine Jan, and Ritankar Das. A computational approach to mortality prediction of alcohol use disorder inpatients. *Computers in biology and medicine*, 75:74–79, 2016.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

Gregory F Cooper, Constantin F Aliferis, Richard Ambrosino, John Aronis, Bruce G Buchanan, Richard Caruana, Michael J Fine, Clark Glymour, Geoffrey Gordon, Barbara H Hanusa, et al. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial intelligence in medicine*, 9(2):107–138, 1997.

Nils Yannick Hammerla, James Fisher, Peter Andras, Lynn Rochester, Richard Walker, and Thomas Plötz. Pd disease state assessment in naturalistic environments using deep learning. In *Twenty-Ninth AAAI conference on artificial intelligence*, 2015.

Johnson, Alistair EW, David J. Stone, Leo A. Celi, and Tom J. Pollard. The mimic code repository: enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association*, page ocx084, 2017a.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Alistair EW Johnson, Tom J Pollard, and Roger G Mark. Reproducibility in critical care: a mortality prediction case study. In *Machine Learning for Healthcare Conference*, pages 361–376, 2017b.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

Alistair EW Pollard, Tom J abd Johnson. The mimic-iii clinical database. `http://dx.doi.org/10.13026/C2XW26`, 2016.

Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmark of deep learning models on large healthcare mimic datasets. *arXiv preprint arXiv:1710.08531*, 2017.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20 (1):61–80, 2008.

Weicheng Zhu and Narges Razavian. Graph neural network on electronic health records for predicting alzheimer's disease. *arXiv preprint arXiv:1912.03761*, 2019.