Risk Prediction in the General Internal Medicine Ward at St. Michael's Hospital

Preprint · April 2019

DOI: 10.13140/RG.2.2.27695.55205

3 authors, including:

Vin Bhaskara
University of Toronto
14 PUBLICATIONS **164** CITATIONS

SEE PROFILE

Risk Prediction in the General Internal Medicine Ward at St. Michael's Hospital

Vineeth Bhaskara, Yingying Fu, Sindhu Gowda

University of Toronto
{bhaskara,yingying}@cs.toronto.edu, sindhu.gowda@mail.utoronto.ca

April, 2019

Abstract

Prediction of clinical interventions remains an open challenge in health-care. This task is complicated by data sources that are sparse, noisy, heterogeneous and outcomes that are imbalanced. An early warning system (EWS) making use of initial patient information can be helpful to physicians and hospitals to make clinically actionable planning. In this work, we aim to make a prediction regarding a patient's need for critical care. We explore various deep learning architectures in order to exploit different possible underlying structure in provided data. In particular, we explore architectures without recurrence, with recurrence, and graph neural networks. We also propose a data-driven regularization layer to incorporate diagnosis information into the model without requiring them during inference. Our models try to learn better patient representation by exploiting additional information from ICD codes and learning the underlying graphical structure in the dataset.

1 Introduction

Health care data is complex and heterogeneous. It is helpful to have an early warning system (EWS) built on patient data to assist physicians in decision making. The goal for the project is to assess the risk of a General Internal Medicine (GIM) patient to experience severe outcomes. This project aims to evaluate different models to predict GIM ward patient outcome, given only data from the first 24 hours after admission. To this end, we take advantage of the success of deep learning models to capture rich representations of data with little hand-engineering by domain experts.

2 Background and Related Works

Studies on EWS systems have been performed by multiple hospitals, for example the HEWS in [12], NEWS in [13]. Both systems measure the physiological parameters from patients and score the risk on a scale of 0 to 3, where 0 indicates normal behaviours. The physiological parameters typically include heart rate, respiratory rate, temperature, blood pressure, oxygen saturation, and level of consciousness. These initial assessments have shown to be effective to predict unanticipated admission to the ICU or death within 24 hours of a NEWS measurement [13] with an AUC of over 0.85. In our work, additional features, such as labs, medication orders and LDA coefficients over clinical notes, are also available to build the prediction model.

References to the code and training parameters are available at the private GitHub repository `https://github.com/vinbhaskara/ML4H-GIM-Risk-Assessment`. Please email your GitHub username for access.

Another recent work by Nestor *et. al.* [9] uses similar features from MIMIC to predict mortality. They evaluate the effect of masking the year of treatment in de-identified data, suggesting that existing baselines on MIMIC might over-estimate the true performance due to the random train-validation-test splits. In contrast to MIMIC, our dataset includes the patient admission times to the GIM ward that can be utilized in making proper validation splits to estimate the true performance.

Work by Esteva *et. al.* [5] on predicting skin cancer from images shows that a model trained on a finer disease partition often performs better than the one trained directly on the target classes. Though our dataset doesn't comprise of images, we evaluate this idea by exploiting the multiple diagnosis labels we have at our disposal.

3 Data

The data has been extracted from the St. Michael Hospital Enterprise Data Warehouse (EDW) and the patient electronic record (Soarian) by the clinical collaborators. The dataset includes structured variables (e.g. treatment orders, lab results, vitals). The clinical collaborators had processed the raw data into DataFrames sampled at 8 hour time intervals. They found that shorter time intervals resulted in many missing values. Missing data was processed similarly to [14]. The data set is from 22,000 patient encounters of 14,000 unique patients from 2011 to 2019. The dataset is split into train, validation and test sets in 80%, 10% and 10% respectively. The oldest data starting from 2011 is used as training data; while the newest data is used as test data.

The binary outcome labels are very imbalanced with 8.53% positive outcome only. The positive outcome labels are further divided into 4 types: transfer to ICU, transfer to palliative care, voluntary transfer to palliative care and death. In addition, ICD-10 codes are also in the dataset. There are many possible ICD codes for specific diseases. To avoid large number of codes with low frequencies, we used the first letter of the ICD-10 code as an intermediate diagnosis code. This results in 21 possible codes for all encounters. These ICD-10 diagnoses codes can be used as training labels only, because ICD-10 codes are available months after the outcome occurs.

4 Methods

In the subsequent sub-sections, we summarize the baseline models such as logistic regression, boosted tree-based classifier (XGBoost [3]), and GRU-D [2], and introduce our proposal of a data-driven regularization term that better utilizes ICD information. Following this, we discuss the proposed methods across three wide classes of models, namely, models without recurrent connections (feed-forward), with recurrent connections, and graph convolutional neural networks. Finally, we show how ensembling the diverse classes of trained models helps in improving performance across most metrics like AUC-ROC, AUC-PRC, PPV, Recall, etc.

4.1 Baselines

For non-neural baseline models, we used penalized logistic regression and XGBoost after running a randomized grid search over the hyperparameters for the best validation performance. As a neural baseline, we trained a GRU-D [2] model that handles missingness per feature dimension in addition to capturing the recurrent timeseries information across the three timesteps. For all the models other than XGBoost, class weights were chosen based on the fraction of encounters with positive or negative outcome labels to compensate the extreme label imbalance. For the XGBoost model, hyperparameters such as `min_child_weight`, that correct for the class imbalance, were picked based on a randomized grid search using validation AUC.

Two different input representations were used to train the non-neural baselines where the model architectures aren't inherently designed for timeseries data unlike GRU-D. The first variant treats each of the three timesteps in an encounter independently where the label for each time step is taken to be the same as the last label for the encounter. This input representation does explicit credit assignment by assigning the final outcome label to *each* timestep of that encounter. Therefore, the aggregate statistics when training a model with the entire data would allow it to learn how a feature dimension correlates with the final target. The second representation concatenates the features across the three

time steps, thus, increasing the feature dimensions to 2571. This allows timestep dependent weights to be learned since each feature per timestep is treated as a separate dimension.

Both these representations used for training non-neural baselines do not fully capture the inductive bias of the data as they don't involve weight sharing across identical features over different timesteps. As a baseline that captures the input data pattern well, we train a GRU-D model after making minor changes (such as increasing the dropout, etc.) to prevent overfitting.

4.2 Incorporating ICD-10 diagnosis codes that are not available during inference

Many EHR data sources such as MIMIC [7] and the GIM dataset described previously include rich fine-grained information about the final diagnosis in the form of ICD codes. Since these codes are obtained months after the patient outcome occurs, they cannot be used as input features to the model. We propose a general technique that we call *data-driven regularization* to better incorporate such information into the model without requiring them during inference. Our proposal involves utilizing the ICD codes as intermediate training labels, in contrast to a previous work by Choi et. al. [4] that attempts exploiting the diagnosis information using multi-level embeddings (MiME).

As shown in Fig 1b, the softmax output from the ICD codes are concatenated with the output of the last fully connected layer. The concatenated vector is used as the input to the final softmax layer of the model. The loss from the ICD code layer is added as a regularization term to the target loss function.

$$\mathcal{L}(\vec{\theta}) = \mathcal{L}_{CE}(\vec{y}, \vec{t}) + \lambda \, \mathcal{L}_{CE}(\vec{y}_{ICD}, \vec{t}_{ICD}),$$

where λ is a tunable regularization hyperparameter (we use $\lambda = 0.5$).

Moreover, this regularization term motivates the network to learn better disentangled internal representations, and also helps in maintaining sufficient gradient signal using the extra labels available.

4.3 Feed-forward Networks without Recurrence

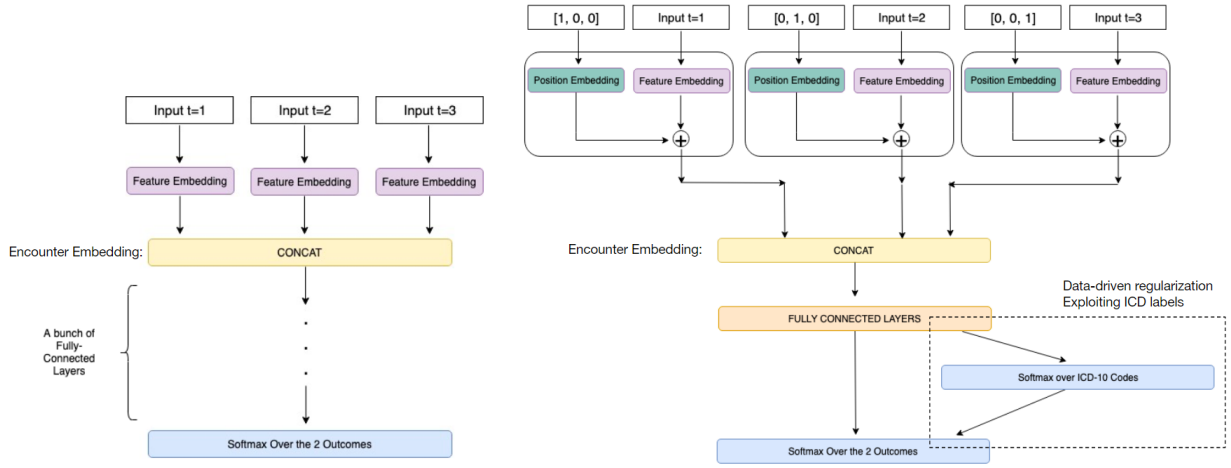
Recent works based on attention [15, 10, 11] that do not employ recurrent connections have proven to be extremely successful, especially, for sequence-to-sequence prediction tasks such as Neural Machine Translation. These models can be thought of as extensions to the feed-forward architecture proposed original by Bengio et. al. [1] in the context of language modeling and word-embeddings.

Since our task involves a constant number (three) of timesteps (truncated to 8-hr interval) per encounter, the issue of long-term time dependency is minimal for us. Additionally, since the number of timesteps considered are few (only 3 per encounter), instead of attention, we could afford fully-connected layers "attending" to each timestep instead of having the computationally cheaper attention module that uses a shared layer to compute contextual weights for the general cases of long and variable length sequences.

We trained a vanilla feed-forward (FNN) architecture with an input embedding layer that is shared across the three timesteps (see Fig.1a), very similar to the original language model architecture proposed by Bengio et. al. [1]. But, unlike their architecture, the inputs in our case are continuous feature values as opposed to one-hot vectors. Therefore, the embedding layer in our architecture is a fully-connected layer with a $\texttt{tanh}(\cdot)$ non-linearity (to keep the embeddings reasonably bounded), but not a simple dictionary look-up.

But with the architecture in Fig.1a the feature embeddings are forced to encapsulate the timestep information as well, in addition to the input feature characteristics, since the weights are constrained to be the same. Therefore, we factor the feature embedding (see Fig.1b) into positional and input embeddings that help in partly disentangling the time dimension from the input features. The additive interaction used between the positional and input embeddings has been inspired by the Transformer Encoder [15].

We used two layers of fully-connected residual blocks as the intermediate layers. The final hidden layer also receives the embeddings from the initial layer via a skip-connection to minimize the information lost across the depth. We heavily employ tricks such as Batch Normalization [6] to reduce the internal covariate shift (since our features are heterogenous), and Dropout to prevent overfitting. Our architectures have about half-a-million parameters each.



(a) Vanilla FNN with shared feature embeddings

(b) FNN with shared feature and positional embeddings (with and without the data-driven regularization layer)

Figure 1: FNN architectures trained.

Based on the observations by Esteva et. al. [5] that finer labels help improving the target prediction performance, we also evaluate models trained on 5 outcome categories where the original outcome 1 (death or ICU) is split into 4 sub-cases depending on the type of ICU care or death. The final binary label probabilities are retrieved by summing up the probabilities for the 4 positive outcome labels to get the total probability for severe outcome.

4.4 Neural Networks with Recurrent Connections

In addition to GRU-D, we train a standard LSTM architecture with a shared input embedding layer similar to the one discussed above. We also compare the performance of the LSTM with and without the data-driven regularization layer that incorporates the ICD-10 codes in the training.

Since recurrent connections are radically different than feed-forward connections in terms of the optimization and decision surfaces, these add to the diversity of the trained models, ultimately helping to create a better ensemble model.

4.5 Graph ConvNets (GCN)

The analysis with graph convolutional networks is based on the simple intuition that there could be shared features between the patients, and that viewing the database as a network with some shared connections can help the model learn a better feature embedding for individual patient representation. Graph convolutional networks have seen recent success as efficient methods for node level classification [16]. We wanted to explore if each patient representation can be improved by aggregating features across the neighbours.

We train the graph convolutional networks on two different learnt embeddings. The first is the learnt embedding of a 3-step time series data that is passed through a common embedding layer and then through a vanilla LSTM network. The second is the learnt encounter embedding, where a unique feature embedding is learnt for each time step and then concatenated together to obtain an encounter-level embedding. The output of the these models are taken as the input features for the GCN. The first model is refereed to as the GCN-LSTM-Embed model and the later as GCN-FNN-Embed model. Graphs are formed on these features based on the minimum-distance criteria in the Euclidean space. The hyper-parameters for minimum distance and number of neighbours are tuned to ensure that the networks are not too sparse or too dense. The GCN architecture is based on a vanilla GCN proposed by Kipf *et. al.* [8]. We also add the data-driven regularization layer as described previously.

4.6 Ensemble

We ensemble the XGBoost baseline model with the FNN model having positional encodings using equally-weighted Geometric Mean showing how this improves performance. We also train an ensemble XGBoost model on the validation predictions generated by each of the individual trained models (including the baselines) using XGBoost. We choose XGBoost over Logistic Regression for ensembling because the probabilities across models are not calibrated.

5 Experiments and Results

5.1 Evaluation

The objective is to predict the binary outcome for a patient based on data collected from only the first 24 hours after admission. We report AUC-ROC and AUC-PRC for all the trained models. We also report the PPV, FPR score at a target Recall of 75%, and Recall at a target PPV of 20% based on the production requirements at St. Michael's Hospital.

Additionally, we also stratify performance of our models across the GIM encounter timestamp in 2 month intervals, gender, length of stay and ICD diagnosis codes to identify the regions where each model excels. We also summarize the generalization capability by looking at the differences between Train and Test AUC-ROC scores. The standard deviations reported are based on 3 random seeds.

Table 1: Model performances measured by AUC-ROC score ($mean \pm std$), FPR at 0.75 TPR, and Generalization Gap for ICU/death prediction based on features per encounter.

	Models	Train	Validation	Test	FPR on Test	Train–Test Gap
Non-Neural Baselines	LR	0.8624	0.7696 ± 0.000007	0.7843 ± 0.00002	0.3022 ± 0.0001	0.0781
	XGBoost	0.9573	**0.8031**$\pm$0.0021	**0.8060** $\pm$0.0006	**0.2810**$\pm$0.0111	**0.1513**
Non-Recurrent (Feed-forward)	2 Outcomes	0.8392	0.8024 ± 0.0015	**0.8002** $\pm$0.0052	0.3205 ± 0.0179	0.039
	2 Outcomes + ICD10	0.8397	0.8035 ± 0.0022	0.7941 ± 0.0050	0.3376 ± 0.0331	0.0456
	5 Outcomes	0.8252	0.8073 ± 0.0011	0.7955 ± 0.0019	0.3217 ± 0.0219	0.0297
	5 Outcomes + ICD10	0.8203	0.8100 ± 0.0002	0.7953 ± 0.0030	0.3154 ± 0.0131	0.025
	5 Outcomes + ICD10 + Pos Enc	0.8223	**0.8101**$\pm$0.0006	0.7981 ± 0.0047	**0.2916**$\pm$0.0051	**0.0242**
	XGBoost over Embeddings	0.9906	0.8040 ± 0.0044	0.7999 ± 0.0021	0.3049 ± 0.0085	0.1907
Recurrent	GRU-D	0.8415	0.7965 ± 0.0036	**0.7958** $\pm$0.0041	0.3046 ± 0.0158	0.0457
	LSTM	0.8292	0.8076 ± 0.0058	**0.7944**$\pm$0.0036	0.3099 ± 0.0154	**0.0216**
	LSTM + ICD10	0.8187	0.8023 ± 0.0047	0.7758 ± 0.0063	0.3232 ± 0.0141	0.0429
GCN	GCN-FNN-Embed	0.8034	0.8056 ± 0.0013	0.7779 ± 0.0027	0.3415 ± 0.0012	**0.0255**
	GCN-LSTM-Embed	0.8252	0.7820 ± 0.0006	0.7794 ± 0.0003	0.3525 ± 0.0012	0.0459

5.2 Results

Table 1 shows the performance of the models measured by AUC-ROC score, FPR scores at a target Recall of 75%, and the Generalization Gap (Train–Test AUC). The two non-neural baselines both show significant overfitting with the largest gap. Neural models have significantly lower generalization gap compared to the baseline models. The FNN model with a 5-way output softmax, ICD-10 regularization, and positional encoding achieves the highest validation AUC with the lowest generation gap of all the models.

Table 2 shows the performance of the models measured by area under precision-recall curve. XGBoost significantly overfits to the training data. Out of the rest of the models, FNN with 2-way softmax and LSTM models have the highest test score. FNN with 5-way softmax + ICD10 layer + Pos Enc, and GRU-D have the highest PPV at 75% recall.

The models are also evaluated by stratifying on gender and length of stay (LOS). Table 3 shows the AUC-ROC, AUC-PRC and PPV when stratified over gender. The dataset is imbalanced in terms of gender with 8510 female and 11348 male patients. However, both genders have the same percentage of positive outcome in the training set. Despite this imbalance, female patients have a

Table 2: Model performances measured by Area Under Precision-Recall curve ($mean \pm std$), Test Recall at 20% PPV, and Test PPV at 75% Recall for ICU/death prediction based on features per encounter.

Models		Train	Validation	Test	PPV	Recall
Non-Neural Baselines	LR	0.3022	0.2986 ± 0.00002	0.3022 ± 0.0001	0.1636 ± 0.0002	0.7542 ± 0.0
	XGBoost	0.8214	**0.3535**± 0.0015	**0.3625**± 0.0084	**0.1944** ± 0.0058	**0.7654**± 0.0079
Non-Recurrent (Feed-forward)	2 Outcomes	0.3706	0.3284 ± 0.008	**0.3184**± 0.0081	0.1738 ± 0.0082	0.7542 ± 0.0
	2 Outcomes + ICD10	0.3698	0.3173 ± 0.0098	0.2982 ± 0.0139	0.1671 ± 0.0132	0.7542 ± 0.0
	5 Outcomes	0.3453	**0.348**± 0.0037	0.2998 ± 0.0066	0.1738 ± 0.0098	0.7561 ± 0.0026
	5 Outcomes + ICD10	0.3288	0.3272 ± 0.0106	0.2891 ± 0.006	0.1759 ± 0.0061	0.7542 ± 0.0
	5 Outcomes + ICD10 + Pos Enc	0.323	0.3322 ± 0.0148	0.2842 ± 0.0149	**0.1874** ± 0.0027	0.7542 ± 0.0
	XGBoost over Embeddings	0.9562	**0.3522**± 0.0242	**0.3254**± 0.0034	0.1815 ± 0.0037	**0.7579**± 0.0026
Recurrent	GRU-D	0.3909	0.3461 ± 0.0027	0.3174 ± 0.0063	**0.1815** ± 0.0077	0.7561 ± 0.0026
	LSTM	0.3519	**0.3518**± 0.01	**0.3187**± 0.0029	0.1793 ± 0.0077	**0.7579**± 0.0026
	LSTM + ICD10	0.3052	0.3166 ± 0.0144	0.2655 ± 0.0295	0.1728 ± 0.0062	0.7561 ± 0.0026
GCN	GCN-FNN-Embed	0.2951	0.2921 ± 0.0061	0.2603 ± 0.0055	0.1648 ± 0.0002	0.7561 ± 0.0026
	GCN-LSTM-Embed	0.2920	0.2840 ± 0.0003	0.2845 ± 0.0003	0.1608 ± 0.0009	**0.7579**± 0.0026

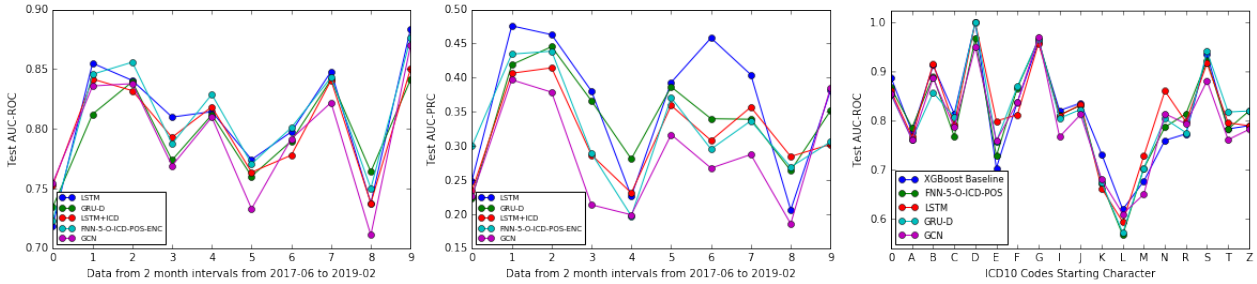
Table 3: Model performance stratified by Gender for AUC-ROC, Area Under Precision score and FPR at 0.75 TPR for ICU/death prediction based on features per encounter on Test set.

Models		AUC-ROC		AUC-PR		PPV @ 75% Recall	
		Female	**Male**	**Female**	**Male**	**Female**	**Male**
Non-Neural Baselines	LR	0.8358	0.7601	0.3731	0.2649	0.1853	0.1648
	XGBoost	**0.8339**	**0.7954**	0.3872	**0.3684**	0.1897	**0.1959**
Non-Recurrent (Feed-forward)	2 Outcomes	0.8429	0.7676	0.3753	0.2776	0.2341	0.1696
	2 Outcomes + ICD10	0.8499	0.7760	0.3670	0.2874	**0.2388**	0.1641
	5 Outcomes	**0.8511**	0.7734	0.3472	0.2994	0.2017	0.1575
	5 Outcomes + ICD10	0.8338	0.7771	0.2988	0.2768	0.2192	**0.1734**
	5 Outcomes + ICD10 + Pos Enc	0.8484	0.7714	0.3330	0.2891	0.2275	0.1713
	XGBoost over Embeddings	0.8446	0.7766	0.3794	0.2943	0.2182	**0.1811**
Recurrent	GRU-D	0.8464	0.7778	0.3769	0.3028	0.2152	**0.1737**
	LSTM	**0.8507**	0.7643	0.3829	0.2852	0.1839	0.1533
	LSTM + ICD10	0.8293	0.7635	0.3265	0.2939	**0.2330**	0.1584
GCN	GCN-FNN-Embed	0.8010	0.7591	0.2625	0.2498	0.1672	0.1552
	GCN-LSTM-Embed	0.8177	0.7639	0.30545	0.2739	0.1702	0.1676
Ensemble	GM of XGB Baseline and FNN with Pos Enc	**0.8548**	**0.7978**	**0.3973**	**0.3423**	**0.2307**	**0.1802**
	Ensembling all models using XGB	**0.8545**	**0.7987**	**0.3928**	**0.3426**	**0.2187**	**0.1869**

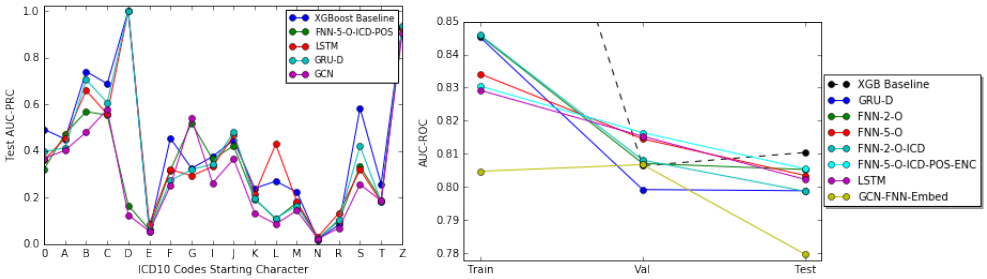
higher AUC-ROC, AUC-PRC and PPV than male patients across all models. The GCN model has the lowest difference between the two genders.

The predictions of the models stratified on length of stay is shown in Table 4. Length of stay is grouped into three bins: less than 3 days, between 3 to 7 days and over 7 days. Out of all encounters, approximately 30% encounters are less than 3 days, 38% are between 3-7 days and 32% are over 7 days. Across all models, the AUC-ROC, AUC-PRC and PPV are all significantly higher for length of stay less than 3 days, and lowest for length of stay over 7 days. This is expected because only data from the first 24 hours are considered.

Since each encounter only has three time steps, models based on treating each time step as an independent input were also evaluated but were found to perform poorly. LR model treating each time step independently results in AUC-ROC, AUC-PRC, and PPV of 0.7751, 0.2844 and 0.161, respectively.

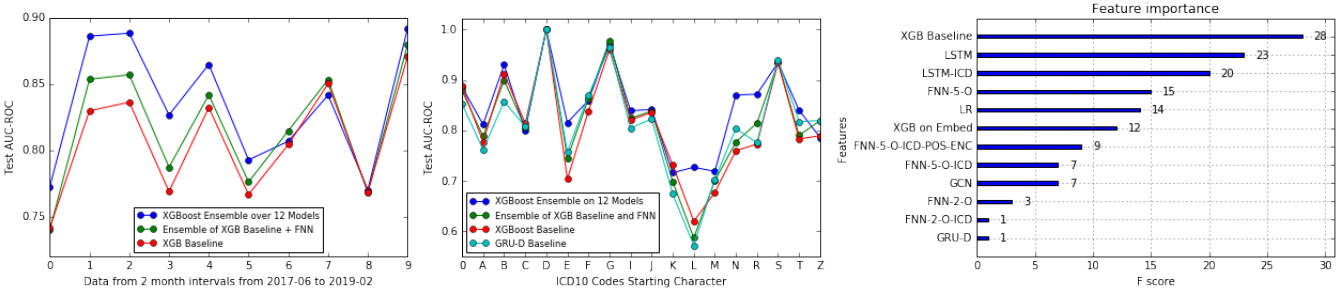


(a) AUC-ROC across months (b) AUC-PRC across months (c) AUC-ROC across ICD codes

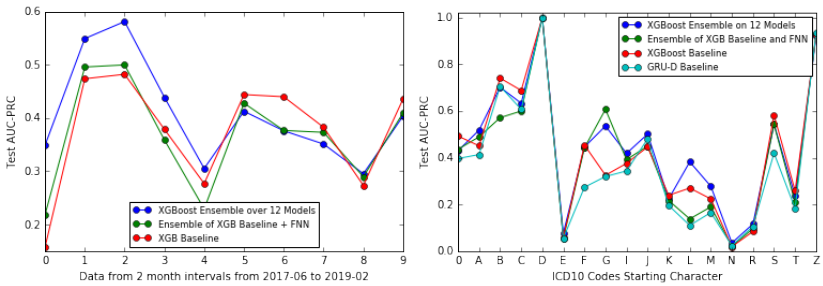


(d) AUC-PRC across ICD codes (e) Generalization across Train–Val–Test based on AUCs on mean prediction across 3 random seeds per model.

Figure 2: Performance of models stratified across months and ICD Codes in the Validation/Test data combined.



(a) AUC-ROC across months (b) AUC-ROC across ICD codes (c) Importance of models in the ensemble



(d) AUC-PRC across months (e) AUC-PRC across ICD codes

Figure 3: Performance of the ensemble models compared to the baseline models.

Table 4: Model performance stratified by length of stay in days for AUC-ROC, Area Under Precision score and FPR at 0.75 TPR for ICU/death prediction based on features per encounter on Test set.

Models		AUC-ROC			AUC-PR			PPV @ 75% Recall		
		<3	**3<day<7**	**>7**	**<3**	**3<day<7**	**>7**	**<3**	**3<day<7**	**>7**
Non-Neural	LR	0.8226	0.7978	0.7144	0.4226	0.3451	0.2251	0.1923	0.1469	0.1485
Baselines	XGBoost	0.8462	**0.8200**	0.7384	0.4285	**0.4339**	**0.3027**	0.2672	0.1801	0.1646
Non-Recurrent (Feed-forward)	2 Outcomes	0.8491	0.7863	0.7278	**0.4867**	0.3638	0.2348	0.2229	0.1347	0.1568
	2 Outcomes + ICD10	**0.8509**	0.7940	0.7402	0.4334	0.3512	0.2461	0.2365	0.1358	0.1828
	5 Outcomes	0.8459	0.7938	0.7355	0.4048	0.3149	0.2556	**0.2756**	0.1469	0.1592
	5 Outcomes + ICD10	0.8055	0.8015	0.7565	0.3484	0.2975	0.2447	0.1584	0.1643	0.1767
	5 Outcomes + ICD10 + Pos Enc	0.8394	0.7768	0.7618	0.4134	0.3048	0.2569	0.2059	0.1309	**0.2046**
	XGBoost over Embeddings	**0.8540**	0.7852	0.7438	0.4853	0.3313	0.2187	**0.2713**	0.1187	0.1773
Recurrent	GRU-D	0.8273	0.8130	0.7431	0.4412	0.3502	0.2506	0.2333	0.1478	0.1853
	LSTM	**0.8507**	0.7735	0.7358	**0.4949**	0.3414	0.2261	0.2333	0.1169	0.1773
	LSTM + ICD10	0.8236	0.7701	0.7351	0.3984	0.3321	0.2320	0.1852	0.1347	**0.1920**
GCN	GCN-FNN-Embed	0.8185	0.7695	0.7168	0.3441	0.2442	0.2183	0.2035	0.1298	0.1611
	GCN-LSTM-Embed	0.8343	0.7559	0.7292	0.3291	0.3248	0.2209	0.2333	0.1190	0.1645
Ensemble	GM of XGB Baseline and FNN with Pos Enc	0.8454	**0.8164**	**0.7692**	0.4562	0.4065	**0.2745**	0.2380	0.1432	0.1941
	Ensembling all models using XGB	**0.8641**	0.8085	0.7639	0.4559	**0.4125**	0.2633	0.2611	**0.1802**	0.1631

Table 5: Ensemble Model performances measured by AUC-ROC, AUC-PRC, and FPR, PPV at 0.75 Recall, and Recall at 20% PPV for ICU/death prediction on Test set.

Models		Test AUC-ROC	FPR on Test	Test AUC-PRC	PPV	Recall
Ensemble Models	Geo. Mean of XGB Baseline and FNN with Pos Enc	0.8165	**0.2773**	**0.3598**	**0.1951**	0.7542
	Geo. Mean of all models	0.8135	0.3008	0.3592	0.1827	0.7542
	Ensembling all models using XGB	**0.81687**	0.28087	0.3537	0.1931	0.7542

6 Discussion

Tables 1, 2, 3, 4, and Figure 2 clearly show that neural network models excel over the XGBoost baseline across different regions of various stratifications. Therefore, one expects the ensemble model to perform even better. Table 5 and Figure 3 also evidently show how the ensemble models perform better than any individual model in most of the comparisons.

We observe that the performance of a single model saturates around 0.80 AUC on Test/Validation without exception across the diverse set of models we trained. This can be attributed to the limited data available for training. Moreover, since many features were binned in the training dataset and many others binary, the variance of individual feature dimensions across the training cases were not significant. Such a redundancy in the training data further reduces the effective size of the dataset. To demonstrate this, we train the XGBoost baseline model on concatenated features across the timesteps with an incrementally increasing size of the training dataset starting from 100 training samples to quantify approximately how many of the training examples are really helping.

We also analyze the cumulative variance captured by top n features after PCA both for features per timestep and concatenated features across timesteps (i.e. features per encounter) to approximately quantify the redundancy across feature dimensions. Figure 4 shows the learning curve and the PCA plots.

The learning curve clearly reveals how the test and the validation performances start plateauing around 6000–8000 samples of the training dataset, reducing the effective size of the total examples available. The PCA plots also reveal the redundancy of feature dimensions. For instance, we found that 90% and 99% of the total variance of the training dataset (across timesteps) can be captured with only 20% (177) and 51.6% (443) of features, respectively, out of 857 total raw dimensions per timestep. Similarly, PCA of features per encounter (i.e. concatenated features across the 3 timesteps) reveals that roughly 90% and 99% of the total variance can be captured by a mere 7.5% (193) and

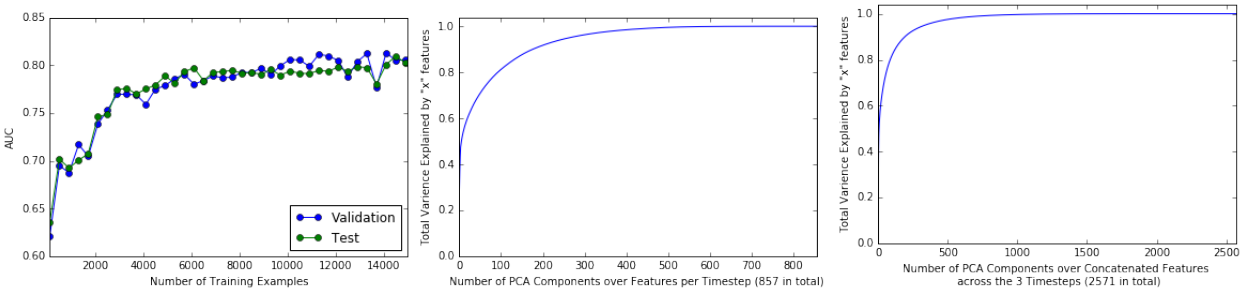


Figure 4: Learning curve for the baseline XGBoost model trained on concatenated features across timesteps with incrementally larger training dataset, and Variance explained by top PCA components per and across timesteps.

28% (730) of total features (2571). Therefore, there is a significant redundancy, both along the columns and the rows of the training dataset.

The models with ICD-10 diagnosis codes as additional intermediate labels do not show expected improvements across different metrics (although models with it achieve the best validation AUC in Table 1. This may be caused by the imbalance in the distribution of the ICD-10 codes. Over half of the encounters were in 4 out of the 21 possible ICD codes (binned by the first character). In addition, since the test data are from 2018-2019, approximately 20% of the test data have not been assigned an ICD-10 code. As a result, they are binned into NaN category, which is not a part of the training dataset.

7 Limitations

As discussed above, based on our analysis in Fig. 4, we conclude that not only was the size of the training dataset small, but also the rows and the columns were highly correlated, making the effective size of the dataset further smaller. Secondly, the target labels were highly disproportionate in number (10:1 ratio). This was also the case for the ICD diagnoses codes that were used as intermediate labels.

As per the agreement with the collaborators, the team only had access to data at the LKS-CHART office. We were not given any remote access to the data. On some of the days, the team members had to share a single laptop to access data and train models.

Acknowledgments

We would like to thank our collaborators at LKS-CHART: Joshua Murray and Chloe Pou-Prom for their guidance, and Michaelia Young and Kasthuri Karunanithi for their assistance.

References

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[2] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *CoRR*, abs/1606.01865, 2016.

[3] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[4] Edward Choi, Cao Xiao, Walter F. Stewart, and Jimeng Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. *CoRR*, abs/1810.09593, 2018.

[5] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature 542 (7639)*, 542:115–118, 2017.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[7] Shen L Lehman L Feng M Ghassemi M Moody B Szolovits P Celi LA Johnson AEW, Pollard TJ and Mark RG. Mimic-iii, a freely accessible critical care database.

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

[9] B. Nestor, M.B.A. McDermott, G. Chauhan, T. Naumann, M.C. Hughes, A. Goldenberg, and M Ghassemi. Rethinking clinical prediction: Why machine learning must consider year of care and feature aggregation. *Machine Learning for Health (ML4H) Workshop at NeurIPS*, 2018.

[10] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[11] Alec Radford. Improving language understanding by generative pre-training. 2018.

[12] S. Skitch, B. Tam, M. Xu, L. McInnis, A. Vu, and A. Fox-Robichaud. Examining the utility of the hamilton early warning scores (hews) at triage: Retrospective pilot study in a canadian emergency department. *CJEM*, 20(2):266–274, 2018.

[13] G. Smith, D.R. Prytherch, P. Meredith, P. Schmidt, and P. Featherstone. The ability of the national early warning score (news) to discriminate patients at risk of early cardiac arrest, unanticipated intensive care unit admission, and death. *Resuscitation*, 84, 01 2013.

[14] H. Suresh, N. Hunt, A. Johnson, L.A. Celi, P. Szolovits, and M. Ghassemi. Clinical intervention prediction and understanding with deep neural networks. *Proceedings of Machine Learning for Healthcare, JMLR*, 2017.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[16] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.