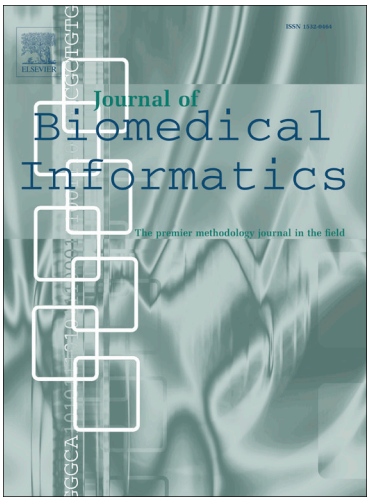


Benchmarking Deep Learning Models on Large Healthcare Datasets

Sanjay Purushotham, Chuizheng Meng, Zhengping Che, Yan Liu

PII: S1532-0464(18)30071-6
DOI: https://doi.org/10.1016/j.jbi.2018.04.007
Reference: YJBIN 2964

To appear in: *Journal of Biomedical Informatics*

Received Date: 17 October 2017
Revised Date: 22 March 2018
Accepted Date: 9 April 2018

Please cite this article as: Purushotham, S., Meng, C., Che, Z., Liu, Y., Benchmarking Deep Learning Models on Large Healthcare Datasets, *Journal of Biomedical Informatics* (2018), doi: https://doi.org/10.1016/j.jbi.2018.04.007

Benchmarking Deep Learning Models on Large Healthcare Datasets

Sanjay Purushotham[a,*], Chuizheng Meng[b,*], Zhengping Che[a], Yan Liu[a,**]

[a] *University of Southern California, Los Angeles, CA 90089, US*
[b] *Tsinghua University, Beijing 100084, China*

Abstract

Deep learning models (aka Deep Neural Networks) have revolutionized many fields including computer vision, natural language processing, speech recognition, and is being increasingly used in clinical healthcare applications. However, few works exist which have benchmarked the performance of the deep learning models with respect to the state-of-the-art machine learning models and prognostic scoring systems on publicly available healthcare datasets. In this paper, we present the benchmarking results for several clinical prediction tasks such as mortality prediction, length of stay prediction, and ICD-9 code group prediction using Deep Learning models, ensemble of machine learning models (Super Learner algorithm), SAPS II and SOFA scores. We used the Medical Information Mart for Intensive Care III (MIMIC-III) (v1.4) publicly available dataset, which includes all patients admitted to an ICU at the Beth Israel Deaconess Medical Center from 2001 to 2012, for the benchmarking tasks. Our results show that deep learning models consistently outperform all the other approaches especially when the 'raw' clinical time series data is used as input features to the models.

Keywords: deep learning models, super learner algorithm, mortality prediction, length of stay, ICD-9 code group prediction

1. Introduction

Quantifying patient health and predicting future outcomes is an important problem in critical care research. Patient mortality and length of hospital stay are the most important clinical outcomes for an ICU admission, and accurately predicting them can help with the assessment of severity of illness; and determining the value of novel treatments, interventions and health care policies. With the goal of accurately predicting these clinical outcomes, researchers have developed novel machine learning models [1, 2] and scoring systems [3] while measuring the

*Co-first authors.
**Corresponding author.
Email addresses: spurusho@usc.edu (Sanjay Purushotham), mengcz95thu@gmail.com (Chuizheng Meng), zche@usc.edu (Zhengping Che), yanliu.cs@usc.edu (Yan Liu)

improvement using performance measures such as sensitivity, specificity and Area under the ROC (AUROC). The availability of large healthcare databases such as Medical Information Mart for Intensive Care (MIMIC-II and III) databases [4, 5] has accelerated the research in this important area as evidenced by a lot of recent publications [6, 7, 8, 2, 9, 10, 11, 1, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22].

Severity scores such as SAPS-II [3], SOFA [23], and APACHE [24] have been developed with the objective of predicting hospital mortality from baseline patient characteristics, defined as the measurements obtained within the first 24 hours after ICU admission. Most of these scoring systems choose a small number of hand-picked explanatory predictors and use simple models such as logistic regression to predict mortality, while making linear and additive relationship assumptions between the outcome variable (mortality) and the predictors. Earlier studies [25, 26] have shown that such assumptions are unrealistic and that nonparametric methods might perform better than standard logistic regression models in predicting ICU mortality.

With the recent advances and success of machine learning and deep learning, many researchers have adopted these models for clinical prediction tasks for ICU admissions. Early works [27, 28, 29] showed that machine learning models obtain good results on mortality prediction and forecasting length of stay in ICU. Recently, Pirracchio [30] showed that a Super Learner algorithm [31]-an ensemble of machine learning models, offers improved performance for predicting hospital mortality in ICU patients and compared its performance to several severity scores on the MIMIC-II dataset. Johnson et al. [6] compared several published works against gradient boosting and logistic regression models using a simple set of features extracted from MIMIC-III dataset [5] for ICU mortality prediction. Harutyunyan et al. [2] empirically validated four clinical prediction benchmarking tasks on the MIMIC-III dataset using deep models. Even though some of these recent efforts have attempted to benchmark the machine learning models on MIMIC datasets, they do not provide a consistent and exhaustive set of benchmark comparison results of deep learning models for a variety of prediction tasks on the large healthcare datasets. Thus, in this paper, we report an exhaustive set of benchmarking results of applying deep learning models for MIMIC-III dataset and compare it with state-of-the art machine learning approaches and scoring systems. Table 1 shows the comparison of benchmarking works. We summarize the main contributions of this work below:

- We present detailed benchmarking results of deep learning models on MIMIC-III dataset for three clinical prediction tasks including mortality prediction, forecasting length of stay, and ICD-9 code group prediction. Our experiments show that deep learning models consistently perform better than the several existing machine learning models and severity scoring systems.

- We present benchmarking results on different feature sets including 'processed' and 'raw' clinical time series. We show that deep learning models obtain better results on 'raw' features which indicates that rule-based preprocessing of clinical features is not necessary for deep learning models.

The remainder of this paper is arranged as follows: in Section 2, we provide an overview of the related work; in Section 3, we describe MIMIC-III dataset and the pre-processing steps we employed to obtain the benchmark datasets; the benchmarking experiments is discussed in Section 4; and we conclude with summary in Section 5.

Table 1: Comparison of benchmarking works.

		Pirracchio 2016	Harutyunyan et al. 2017	Johnson et al. 2017	**This work**
Time Durations	24 hours	✓		✓	✓
	48 hours		✓	✓	✓
Number of Features	Smaller feature set	✓	✓	✓	✓
	Larger feature set				✓
Feature Type	Non-time series	✓		✓	✓
	Time-series		✓		✓
Databases	MIMIC-II	✓			
	MIMIC-III		✓	✓	✓
	MIMIC-III (CareVue)				✓
Scoring Systems	SAPS -II	✓			✓
	SOFA	✓			✓
Prediction Algorithms	Machine learning models	✓		✓	✓
	Deep learning models		✓		✓
Prediction Tasks	In-hospital mortality	✓	✓	✓	✓
	Short-term mortality				✓
	Long-term mortality				✓
	Length of stay		✓		✓
	Phenotyping		✓		
	ICD-9 code group		✓		✓

2. Related Work

We first provide a brief review of machine learning and deep learning models for healthcare applications, and then discuss the existing works on benchmarking healthcare datasets.

Early works [32, 33] have shown that machine learning models obtain good results on mortality prediction and medical risk evaluation. Physionet challenge[1] - a friendly competition platform - has resulted in development of machine learning models for addressing some of the open healthcare problems. With the recent advances in deep learning techniques, there is a growing interest in applying these techniques to healthcare applications due to the increasing availability of

[1] `https://physionet.org/challenge/`

large-scale health care data [34, 7, 35, 36]. For example, Che et al. [7] developed a scalable deep learning framework which models the prior-knowledge from medical ontologies to learn clinically relevant features for disease diagnosis. A recent study [37] showed that a neural network model can improve the prediction of several psychological conditions such as anxiety, behavioral disorders, depression, and post-traumatic stress disorder. Other recent works [38, 39, 40] have leveraged the power of deep learning approaches to model diseases and clinical time series data. These previous work have demonstrated the strong performance by deep learning models in health care applications, which significantly alleviates the tedious work on feature engineering and extraction.

The availability of deidentified public datasets such as Medical Information Mart for Intensive Care (MIMIC-II [4] and MIMIC-III [5]) has enabled researchers to benchmark machine learning models for studying ICU clinical outcomes such as mortality and length of hospital stay. Recent works such as Li-wei et al. [41] and Lee [42] have proposed and studied machine learning models for predicting mortality of ICU patients. Pirracchio [30] used MIMIC II clinical data [4] to predict mortality in the ICU and showed that the Super Learner algorithm - an ensemble of machine learning models, performs better than SAPS II, APACHE II and SOFA scores. Their work showed that machine learning models outperform the prognostic scores, but they did not compare their results with the recent deep learning models. Harutyunyan et al. [2] proposed a deep learning model called multi-task Recurrent Neural Networks to empirically validate four clinical prediction benchmarking tasks on the MIMIC-III database. While, their work showed promising benchmark results of deep learning models, they compared their proposed model only with a standard Logistic Regression model and a Long Short Term Memory Network [43], and omitted comparison with scoring systems (SAPS-II) or other machine learning models (such as Super Learner). Johnson et al. [6] studied the challenge of reproducing the published results on the public MIMIC-III dataset using a case-study on mortality prediction task. They reviewed 28 publications and then compared the performance reported in these studies against gradient boosting and logistic regression models using a simple set of features extracted from MIMIC-III dataset. They demonstrated that the large heterogeneity in studies highlighted the need for improvements in the way that prediction tasks are reported to enable fairer comparison between models. Our work advances the efforts of these previous benchmark works by providing a consistent and exhaustive set of benchmarking results of deep learning models on several prediction tasks.

3. MIMIC-III Dataset

In this section, we describe the MIMIC-III dataset and discuss the steps we employed to preprocess and extract the features for our benchmarking experiments.

3.1. Dataset Description

MIMIC III [5] is a publicly available critical care database maintained by the Massachusetts Institute of Technology (MIT)'s Laboratory for Computational Physiology. This database integrates deidentified, 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 (aged 15 years or above) and 7870 neonates admitted to an ICU at the BIDMC. The data covers 38 597 distinct adult patients with 49 785 hospital admissions. To obtain consistent benchmarking datasets, in this paper we only include the first ICU admission of the patients. Table 2 shows the statistics of our dataset, and Table 3 shows the baseline characteristics and outcome measures of our dataset. We observe that the median age of adult patients is 65.86 years (Quartile Q1 to Q3: 52.72 to 77.97) with 56.76 % patients are male, in-hospital mortality around 10.49 % and the median length of an hospital stay is 7.08 days (Q1 to Q3: 4.32 to 12.03).

Table 2: Summary statistics of MIMIC-III dataset.

Data	**Total**
# admissions in the MIMIC-III (v1.4) database	58 576
# admissions which are the first admission of the patient	46 283
# admissions which are the first admission of an adult patient (> 15 years old)	38 425
# admissions where adult patient died 24 hours after the first admission	35 627

3.2. Dataset Preprocessing

In this section, we describe in detail the cohort selection, data extraction, data cleaning and feature extraction methods we employed to preprocess our MIMIC-III dataset.

3.2.1. Cohort Selection

The first step of dataset preprocessing includes cohort selection. We used two sets of inclusion criterion to select the patients to prepare the benchmark datasets. First, we identified all the adult patients by using the age recorded at the time of ICU admission. Following previous studies [6], in our work, all the patients whose age was >15 years at the time of ICU admission is considered as an adult [2]. Second, for each patient, we only use their first admission in our benchmark datasets and for subsequent analysis, and dropped all their later admissions. This was done to prevent possible information leakage in the analysis, and to ensure similar experimental settings compared to the related works [6].

[2]Note that in MIMIC III (v1.4), all the patients under the age of 15 years are referred to as *neonates*.

Table 3: Baseline characteristics and in-hospital mortality outcome measures. Continuous variables are presented as *Median [InterQuartile Range Q1-Q3]*; binary or categorical variables as *Count (%)*.

	Overall	**Dead at hospital**	**Alive at hospital**
General			
# admissions	35627	3738	31889
Age	65.86 [52.72-77.97]	73.85 [60.16-82.85]	64.98 [52.04-77.21]
Gender (female)	15409 (43.24%)	1731 (46.31%)	13678 (42.88%)
First SAPS-II	33.00 [25.00-42.00]	48.00 [38.00-59.00]	32.00 [24.00-40.00]
First SOFA	3.00 [2.00-6.00]	6.00 [4.00-9.00]	3.00 [2.00-5.00]
Origin			
Medical	24720 (69.37%)	2969 (79.43%)	21751 (68.19%)
Emergency surgery	6134 (17.21%)	663 (17.74%)	5471 (17.15%)
Scheduled surgery	4783 (13.42%)	106 (2.84%)	4677 (14.66%)
Site			
MICU	12621 (35.42%)	1814 (48.53%)	10807 (33.88%)
MSICU	5821 (16.33%)	691 (18.49%)	5130 (16.08%)
CCU	5180 (14.54%)	523 (13.99%)	4657 (14.60%)
CSRU	7264 (20.38%)	245 (6.55%)	7019 (22.00%)
TSICU	4751 (13.33%)	465 (12.44%)	4286 (13.44%)
Lab Results			
HR (bpm)	84.00 [73.00-97.00]	90.00 [75.00-107.00]	84.00 [72.00-96.00]
MAP (mmhg)	76.00 [67.33-87.00]	74.00 [64.67-86.00]	77.00 [68.00-87.00]
RR (cpm)	18.00 [14.00-22.00]	20.00 [16.00-24.00]	18.00 [14.00-21.00]
Na (mmol/l)	138.00 [136.00-141.00]	139.00 [135.00-142.00]	138.00 [136.00-141.00]
K (mmol/l)	4.10 [3.80-4.60]	4.20 [3.70-4.70]	4.10 [3.80-4.60]
HCO_3 (mmol/l)	24.00 [21.00-26.00]	22.00 [18.00-25.00]	24.00 [21.00-26.00]
WBC ($10^3/mm^3$)	11.00 [7.90-14.90]	12.30 [8.00-17.20]	10.80 [7.90-14.60]
P/F ratio	257.50 [180.00-352.50]	218.66 [140.00-331.86]	262.50 [187.00-355.00]
Ht (%)	31.00 [26.00-36.00]	31.00 [27.00-36.00]	31.00 [26.00-36.00]
Urea (mmol/l)	1577.00 [968.00-2415.00]	1020.00 [518.50-1780.00]	1640.00 [1035.00-2470.00]
Bilirubine (mg/dl)	0.70 [0.40-1.70]	1.00 [0.50-3.50]	0.70 [0.40-1.50]
Outcomes			
Hospital LOS (days)	7.08 [4.32-12.03]	7.21 [3.31-14.44]	7.07 [4.40-11.88]
ICU death (%)	2860 (8.03%)	2860 (76.51%)	–
Hospital death (%)	3738 (10.49%)	3738 (100%)	–

3.2.2. Data Extraction

There are 26 tables in the MIMIC-III (v1.4) relational database. Charted events such as laboratory tests, doctor notes and fluids into/out of patients are stored in a series of 'events' tables. For the purpose of preparing benchmark datasets to predict clinical tasks, we extracted data for the selected cohort from the following tables: *inputevents (inputevents_cv/inputevents_mv)* (intake for patients monitored using Philips CareVue system/iMDSoft MetaVision system), *outputevents* (output information for patients while in the ICU), *chartevents* (all charted observations for patients), *labevents* (laboratory measurements for

patients both within the hospital and in outpatient clinics), and *prescriptions* (medications ordered, and not necessarily administered, for a given patient). We selected these tables as they provide the most relevant clinical features for the prediction tasks considered in this work. We obtained the following two benchmark datasets:

- MIMIC-III: This includes the data extracted from all the above tables for all the selected cohorts in the entire MIMIC-III database.
- MIMIC-III (CareVue): This includes the data extracted from all the above tables for the selected cohorts who are included in the *inputevents_cv* table (*inputevents* data recorded using Philips CareVue system) in the MIMIC-III database. MIMIC-III (CareVue) is a subset of MIMIC-III dataset and it roughly corresponds to the MIMIC-II [4] dataset.

3.2.3. Data Cleaning

The data extracted from MIMIC-III database has lots of erroneous entries due to noise, missing values, outliers, duplicate or incorrect records, clerical mistakes etc. We identified and handled the following three issues with the extracted data. First, we observed that there is inconsistency in the recording (units) of certain variables. For example, some of the prescriptions are recorded in 'dose' and in 'mg' units; while some variables in *chartevents* and *labevents* tables are recorded in both numeric and string data type. Second, some variables have multiple values recorded at the same time. Third, for some variables the observation was recorded as a range rather than a single measurement. We addressed these issues by these procedures:

- To handle inconsistent units: We first obtain the percentage of each unit appearing in the database for a variable. If there is only one unit, we do nothing. For variables with multiple and inconsistent units, if a major unit accounts for $\geq$90 % of the total number of records then we just keep all the records with the major unit and drop the other ones. For the rest of the variables/features which do not have a major unit, we convert all the units to a single unit based on accepted rules in literature [3]. For example, we convert the value of 'mg' to that of 'grams', and the value of 'dose' to that of 'ml' or 'mg' based on the variable.Table A.27 in the Appendix A.6 shows the conversion rules we employed for some of the features with inconsistent units. We drop the features for which we cannot find correct rules for conversion.
- To handle multiple recordings at the same time: For numerical features, we take the average of the multiple recordings present at the same time; and for categorical features, we only keep the value that appears first.
- To handle range of feature values: We take the median of the range to represent the value of the feature at a certain time point.

[3] `https://www.drugs.com/dosage/`

3.2.4. Feature Selection and Extraction

We process the extracted benchmark datasets to obtain the features which will be used for the prediction tasks. To enable an exhaustive benchmarking comparison study, we select three sets of features as described below.

Table 4: Feature Set A: 17 features used in SAPS-II scoring system.

Feature	Itemid	Item name	Table
glasgow coma scale	723	GCSVerbal	chartevents
	454	GCSMotor	chartevents
	184	GCSEyes	chartevents
	223900	Verbal Response	chartevents
	223901	Motor Response	chartevents
	220739	Eye Opening	chartevents
systolic blood pressure	51	Arterial BP [Systolic]	chartevents
	442	Manual BP [Systolic]	chartevents
	455	NBP [Systolic]	chartevents
	6701	Arterial BP #2 [Systolic]	chartevents
	220179	Non Invasive Blood Pressure systolic	chartevents
	220050	Arterial Blood Pressure systolic	chartevents
heart rate	211	Heart Rate	chartevents
	220045	Heart Rate	chartevents
body temperature	678	Temperature F	chartevents
	223761	Temperature Fahrenheit	chartevents
	676	Temperature C	chartevents
	223762	Temperature Celsius	chartevents
pao2 / fio2 ratio	50821	PO2	labevents
	50816	Oxygen	labevents
	223835	Inspired O2 Fraction (FiO2)	chartevents
	3420	FiO2	chartevents
	3422	FiO2 [Meas]	chartevents
	190	FiO2 set	chartevents
urine output	40055	Urine Out Foley	outputevents
	43175	Urine	outputevents
	40069	Urine Out Void	outputevents
	40094	Urine Out Condom Cath	outputevents
	40715	Urine Out Suprapubic	outputevents
	40473	Urine Out IleoConduit	outputevents
	40085	Urine Out Incontinent	outputevents
	40057	Urine Out Rt Nephrostomy	outputevents
	40056	Urine Out Lt Nephrostomy	outputevents
	40405	Urine Out Other	outputevents
	40428	Orine Out Straight Cath	outputevents
	40086	Urine Out Incontinent	outputevents
	40096	Urine Out Ureteral Stent #1	outputevents
	40651	Urine Out Ureteral Stent #2	outputevents
	226559	Foley	outputevents

Feature	Itemid	Item name	Table
	226560	Void	outputevents
	226561	Condom Cath	outputevents
	226584	Ileoconduit	outputevents
	226563	Suprapubic	outputevents
	226564	R Nephrostomy	outputevents
	226565	L Nephrostomy	outputevents
	226567	Straight Cath	outputevents
	226557	R Ureteral Stent	outputevents
	226558	L Ureteral Stent	outputevents
	227488	GU Irrigant Volume In	outputevents
	227489	GU Irrigant/Urine Volume Out	outputevents
serum urea nitrogen level	51006	Urea Nitrogen	labevents
white blood cells count	51300	WBC Count	labevents
	51301	White Blood Cells	labevents
serum bicarbonate level	50882	BICARBONATE	labevents
sodium level	950824	Sodium Whole Blood	labevents
	50983	Sodium	labevents
potassium level	50822	Potassium, whole blood	labevents
	50971	Potassium	labevents
bilirubin level	50885	Bilirubin Total	labevents
age	–	intime	icustays
		dob	patients
acquired immunodeficiency syndrome	–	icd9_code	diagnoses_icd
hematologic malignancy	–	icd9_code	diagnoses_icd
metastatic cancer	–	icd9_code	diagnoses_icd
admission type	–	curr_service	services
		ADMISSION_TYPE	admissions

- **Feature Set A**: This feature set consists of the 17 features used in the calculation of the SAPS-II score [3]. For these features, we drop outliers in the data according to medical knowledge and merge relevant features. For example, for the Glasgow Coma Scale score denoted as GCS score, we sum the GCSVerbal, GCSMotor and GCSEyes values; for the urine output, we sum the features representing urine; and for body temperature, we convert Fahrenheit to Celsius scale. Note that the SAPS II score features are hand-chosen and processed, and thus, we refer to them as 'Processed'

features instead of 'raw' features. Table 4 lists all the 17 processed features and their corresponding entries in the MIMIC-III database table. In our experiments, some of these features such as chronic diseases, admission type and age are treated as non-time series features, and the remaining features are treated as time series features.

- **Feature Set B**: This feature set consists of the 20 features related to the 17 features used in SAPS-II score. Instead of preprocessing the 17 features as done to obtain Feature set A, here we consider all the raw values of the 17 SAPS-II score features. In particular, we do not remove outliers and we only drop values below 0. For the GCS score, we treat GCSVerbal, GCSMotor and GCSEyes as separate features. We also consider PaO2 and FiO2 as individual features instead of calculating the PF-ratio (PaO2/FiO2 ratio). This feature set was built to study how the prediction models perform on the 'raw' clinical features.

- **Feature Set C**: This feature set consists of 136 raw features selected from the 5 tables mentioned in section 3.2.2 and includes the 20 features of Feature set B. These 136 features were chosen based on their low missing rate, from more than 20 000 features available in the 5 tables mentioned in section 3.2.2. Similar to feature set B, we did not preprocess this dataset (i.e. did not apply hand-crafted processing rules) and used the raw values of the features. It is worth noting that a few features appear multiple times as they were present in multiple tables. For example, Glucose appears in both Chart and Lab events, and was included in the feature set. This feature set was selected to study if the prediction models can automatically learn feature representations from a large number of raw clinical time series data and at the same time obtain better results on the prediction tasks. Table A.23 in the Appendix A.1 lists all the features of this feature set C.

We extract the above three feature sets from MIMIC-III and MIMIC-III (CareVue) datasets. After the feature selection, we obtained the non-time series and time series features which will be used in the experiments. We extracted the features from first 24 hours and first 48 hours after admission to ICU. Each time series feature is sampled every 1 hour. During the sampling, some features might have multiple readings during the same hour. Since we need to have a representative value for each feature at a particular time step (each hour for Feature C, each day for Feature A and B) we have to aggregate the multiple readings. Depending on the feature, we either take the average or the summation of the multiple recordings present at the same time step. For the feature sets, we take the summation of the multiple recordings generally for the fluids or medications into/out of patient, and average for other features. For example, in Feature set A, we sum up the urine output feature and take the average for the other features. To fill-in missing values, we performed forward and backward imputation. For some patients, certain features might be completely missing. We performed mean imputation for these cases during the training and validation

stage of the experiments. We obtain summary statistics of time-series features for models which are not capable of handling temporal data.

4. Benchmarking Experiments

In this section, we describe in detail the benchmark prediction tasks, the prediction algorithms and their implementation, and report the experimental results.

4.1. Benchmark Prediction Tasks

Here, we describe the benchmark prediction tasks which represent some of the important problems in critical care research. They have been well-studied in the medical community [24, 44, 26], and these tasks have been commonly used to benchmark machine learning algorithms [30, 2].

4.1.1. Mortality Prediction

Mortality prediction is one of the primary outcomes of interest of an hospital admission. We formulate mortality as a binary classification task, where the label indicates the death event for a patient. We define the following mortality prediction benchmark tasks:

- In-hospital mortality prediction: Predict whether the patient dies during the hospital stay after admitted to an ICU.
- Short-term mortality prediction: Predict whether the death happens within a short duration of time after the patient is admitted to the ICU. For this task, we define the 2-day and 3-day mortality prediction tasks where the patient dies within 2-days and 3-days respectively after admitted to ICU. For first 24-hour data, we can predict 2-day and 3-day mortality, while for the first 48-hour data we only predict 3-day mortality.
- Long-term mortality prediction: This task involves predicting if the patient dies after a long time since being discharged from the hospital. For this task, we consider the 30-day and 1-year mortality prediction tasks where the patient dies within 30-days or 1 year after being discharged from the hospital. Note that we still use only the first 24-hour data and first 48-hour data to predict 30-days and 1-year mortality.

Table 5 shows the mortality label statistics of the entire MIMIC-III dataset. The details about how the mortality labels are obtained from MIMIC-III database is explained in the Appendix A.2.

Table 5: Label statistics of mortality prediction task.

MIMIC-III datasource	Mortality label ratio w.r.t total admissions					# Admissions
	In-hospital	2-day	3-day	30-day	1-year	
Metavision (2008-2012)	0.096	0.015	0.014	0.124	0.232	15 376
CareVue (2001-2008)	0.111	0.014	0.015	0.134	0.261	20 261
All sources (2001-2012)	0.105	0.014	0.015	0.129	0.248	35 637

4.1.2. ICD-9 Code Group Prediction

In this benchmarking task, we predict the ICD-9 diagnosis code group (e.g. respiratory system diagnosis) for each admission. ICD (stands for International Statistical Classification of Diseases and Related Health Problems) codes are used to classify diseases and a wide variety of symptoms, signs, causes of injury or disease, etc. Nearly every health condition can be assigned an unique ICD-9 code group where each group usually include a set of similar diseases. In our work, we group all the ICD-9 codes for an ICU admission into 20 diagnosis groups[4] and treat this task as a multi-task prediction problem. Table 6 shows the ICD-9 code group label statistics of the MIMIC-III dataset. ICD-9 code group 760-779 was left out since this group corresponds to "Conditions Originating in the Perinatal Period. This ICD-9 code is usually assigned to newborns who are excluded from our benchmarking study as we are interested in only the adult cohort (discussed in section 3.2.1).

4.1.3. Length of Stay Prediction

In this benchmarking task, we predict the length of stay for each admission. We define the length of stay of an admission as total duration of hospital stay, i.e. the length of time interval between hospital admission and discharge from the hospital. We treat length of stay prediction task as a regression problem. Figure 1 shows the distribution of length of stay of the MIMIC-III benchmark datasets.

4.2. Prediction Algorithms

In this section, we describe all the prediction algorithms and the scoring systems that we have used for benchmarking tasks on MIMIC-III datasets.

4.2.1. Scoring Methods

SAPS-II. SAPS-II [3] stands for Simplified Acute Physiology Score and it is a ICU scoring system designed to measure the severity of the disease for patients admitted to an ICU. A point score is calculated for each of the 12 physiological features mentioned in Table 4 and a final SAPS-II score S is obtained as the sum of all point scores. Note that SAPS-II score is calculated using the data

[4] `http://tdrdata.com/ipd/ipd_SearchForICD9CodesAndDescriptions.aspx`

Table 6: ICD-9 code group label statistics. For each ICD-9 code group, the entry denotes the ratio of number of patients who have been assigned that ICD-9 code to the total number of patients in the dataset.

ICD-9 Code Group	**ICD-9 Code Range**	**MIMIC-III Metavision** (2008-2012)	**MIMIC-III CareVue** (2001-2008)	**MIMIC-III All Sources** (2001-2012)
1	001 - 139	0.302	0.225	0.258
2	140 - 239	0.201	0.151	0.172
3	240 - 279	0.765	0.629	0.688
4	280 - 289	0.445	0.311	0.369
5	290 - 319	0.416	0.244	0.318
6	320 - 389	0.424	0.195	0.294
7	390 - 459	0.846	0.820	0.831
8	460 - 519	0.504	0.468	0.484
9	520 - 579	0.461	0.339	0.391
10	580 - 629	0.461	0.352	0.399
11	630 - 679	0.003	0.005	0.004
12	680 - 709	0.119	0.090	0.102
13	710 - 739	0.266	0.133	0.190
14	740 - 759	0.042	0.032	0.036
15	780 - 789	0.411	0.251	0.320
16	790 - 796	0.115	0.064	0.086
17	797 - 799	0.050	0.016	0.030
18	800 - 999	0.453	0.448	0.450
19	V Codes	0.634	0.362	0.479
20	E Codes	0.429	0.263	0.335

collected within the first 24 hours of an ICU admission. After SAPS-II score is obtained, the individual mortality prediction can be calculated as [30]:

$$\log \frac{p_{\text{death}}}{1-p_{\text{death}}} = -7.7631 + 0.0737 \cdot S + 0.9971 \cdot \log(1+S)$$

SOFA. SOFA [23] is the Sepsis-related Organ Failure Assessment score (also referred to as the Sequential Organ Failure Assessment score) and it is used to describe organ dysfunction/failure of a patient in the ICU. The mortality prediction based on SOFA can be obtained by regressing the mortality on the SOFA score using a main-term logistic regression model.

New SAPS-II. A new SAPS-II scoring method was defined by Pirracchio [30]. It is a modified version of SAPS-II and is obtained by fitting a main-term logistic regression model using the same explanatory variables as those used in the

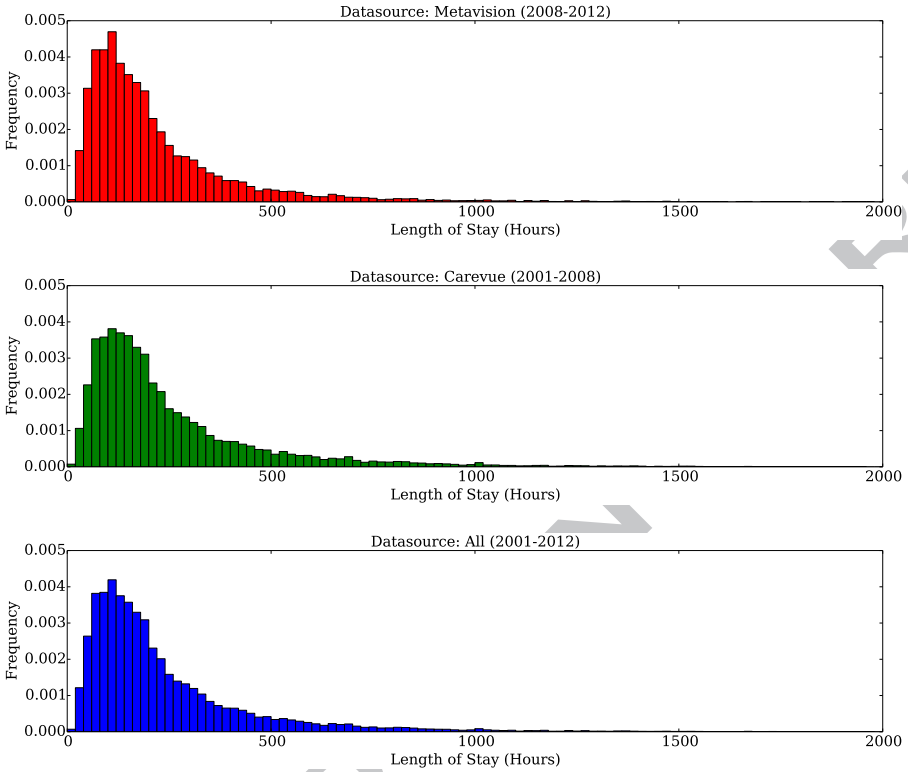


Figure 1: Distribution of length of stay. Values above 2000 hours are not shown in the figure.

original SAPS-II score calculation.

4.2.2. Super Learner Models

Super Learner [45, 31] is a supervised learning algorithm that is designed to find the optimal combination from a set of prediction algorithms. It represents an asymptotically optimal learning system and is built on the theory of cross-validation. This algorithm requires a collection of user-defined machine learning algorithms such as logistic regression, regression trees, additive models, (shallow) neural networks, and random forest. The algorithm then estimates the risk associated to each algorithm in the provided collection using cross-validation. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. From this estimation of the risk associated with each candidate algorithm, the Super Learner builds an aggregate algorithm obtained as the

optimal weighted combination of the candidate algorithms. Table 7 shows the algorithms used in the Super Learner algorithm [30] and their implementation available in the R and Python languages. In the experiments section, we will compare and discuss the results of Super Learner using these programming languages. Following Pirracchio [30], we consider two variants of Super Learner algorithm, namely, Super Learner I: Super Learner with categorized variables, and Super Learner II: Super Learner with non-transformed variables. Note that Super Learner-I is applicable only for Feature set A, while Super Learner-II algorithm can be used with all the Feature sets A, B and C.

Table 7: Algorithms used in Super Learner with corresponding R packages and Python libraries.

Algorithm	**R packages**	**Python libraries**
Standard logistic regression	SL.glm	sklearn.linear_model.LogisticRegression
Logistic regression based on the AIC	SL.stepAIC	sklearn.linear_model.LassoLarsIC
Generalized additive model	SL.gam	pygam.LinearGAM pygam.LogisticGAM
Generalized linear model with penalized maximum likelihood	SL.glmnet	sklearn.linear_model.ElasticNet
Multivariate adaptive polynomial spline regression	SL.polymars	pyearth.Earth
Bayesian generalized linear model	SL.bayesglm	sklearn.linear_model.BayesianRidge
Generalized boosted regression model	SL.gbm	sklearn.ensemble.GradientBoostingRegressor sklearn.ensemble.GradientBoostingClassifier
Neural network	SL.nnet	sklearn.neural_network.MLPRegressor sklearn.neural_network.MLPClassifier
Bagging classification trees	SL.ipredbagg	sklearn.ensemble.BaggingRegressor sklearn.ensemble.BaggingClassifier
Pruned recursive partitioning and Regression Trees	SL.rpartPrune	–
Random forest	SL.randomForest	sklearn.ensemble.RandomForestRegressor sklearn.ensemble.RandomForestClassifier
Bayesian additive regression trees	SL.bartMachine	–

4.2.3. Deep Learning Models

Deep Learning Models (also called Deep Neural Networks or Deep models) [46] have become a successful approach for automated extraction of complex data representations for end-to-end training. Deep models consist of a layered, hierarchical architectures of neurons for learning and representing data. The hierarchical learning architecture is motivated by artificial intelligence emulating the deep, layered learning process of the primary sensorial areas of the neocortex in the human brain, which automatically extracts features and abstractions from the underlying data [47, 48]. In a deep learning model, each neuron receives one or more inputs and sums them to produce an output (or activation). Each neuron in the hidden layers is assigned a weight that is considered for the outcome

classification, but this weight is itself learned from its previous layers. The hidden layers thus can use multidimensional input data and introduce progressively non-linear weight combinations to the learning algorithm.

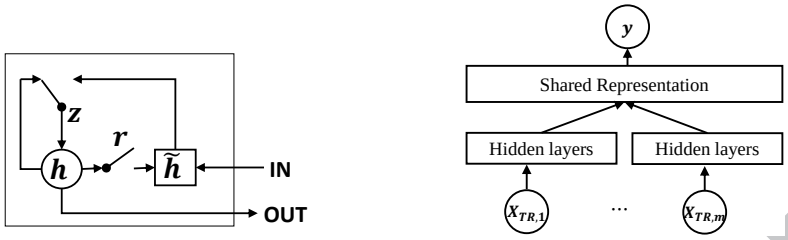
The main advantage of the deep learning approach is its ability to automatically learn good feature representations from raw data, and thus significantly reduce the effort of handcrafted feature engineering. In addition, deep models learn distributed representations of data, which enables generalization to new combinations of the values of learned features beyond those seen during the training process. Deep Learning models have yielded outstanding results in several applications, including speech recognition [49, 50], computer vision [51, 52, 53], and natural language processing [54, 55, 56, 57]. Recent research has shown that deep learning methods achieve state-of-the-art performance in analyzing health-related data, such as ICU mortality prediction [6], phenotype discovery [7] and disease prediction [36]. These works have demonstrated the strong performance by deep learning models in health care applications, which significantly alleviates the tedious work on feature engineering and extraction. Here, we will first briefly introduce two types of deep models namely Feedforward neural networks (FFN), which is a standard neural network structure, and Recurrent Neural Networks (RNN) which is used for modeling sequence and time series data. After that, we will describe our proposed Multi-modal deep learning model, a combination of FFN and RNN, which will be used in the benchmarking experiments.

Feedforward Neural Networks. A multilayer feedforward network [58] (FFN) is a neural network with multiple nonlinear layers and possibly one prediction layer on the top to solve classification task. The first layer takes $\boldsymbol{X}$ as the input, and the output of each layer is used as the input of the next layer. The transformation of each layer l can be written as

$$\boldsymbol{X}^{(l+1)} = f^{(l)}(\boldsymbol{X}^{(l)}) = s^{(l)}\left(\boldsymbol{W}^{(l)}\boldsymbol{X}^{(l)} + \boldsymbol{b}^{(l)}\right)$$

where $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ are respectively the weight matrix and bias vector of layer l, and $s^{(l)}$ is a nonlinear activation function, which usually is a *logistic sigmoid*, *tanh*, or *ReLU* [59] function. We optimize the cross-entropy or mean squared error prediction loss for classification or regression tasks respectively and get the prediction output from the topmost prediction layer.

Recurrent Neural Networks. Recurrent neural network (RNN) models have been shown to be successful at modeling sequences and time series data [60]. RNN with simple activations are incapable of capturing long term dependencies, and hence their variants such as Long Short Term Memory (LSTM) [43] and Gated Recurrent Unit (GRU) [61] have become popular due to their ability to capture long-term dependencies using memory and gating units. GRU can be considered as a simplified version of LSTM and it has been shown that GRU has similar performance compared to LSTM [62]. The structure of GRU is shown in Figure 2(a). Let $\boldsymbol{x}_t \in \mathbb{R}^P$ denotes the variables at time t, where $1 \leq t \leq T$. At each time t, GRU has a reset gate r_t^j and an update gate z_t^j for each of the



(a) Gated recurrent unit.

(b) Multimodal deep learning models.

Figure 2: Deep learning models. In multimodal deep models, $X_{(.)}$ represents the different inputs including temporal and non-temporal features, and y is the output.

hidden state h_t^j. The update function of GRU is shown as follows:

$$\boldsymbol{z}_t = \sigma\left(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right) \qquad \boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_r \boldsymbol{x}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right)$$
$$\tilde{\boldsymbol{h}}_t = tanh\left(\boldsymbol{W} \boldsymbol{x}_t + \boldsymbol{U}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}\right) \qquad \boldsymbol{h}_t = (\boldsymbol{1} - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t$$

where matrices $\boldsymbol{W}_z, \boldsymbol{W}_r, \boldsymbol{W}, \boldsymbol{U}_z, \boldsymbol{U}_r, \boldsymbol{U}$ and vectors $\boldsymbol{b}_z, \boldsymbol{b}_r, \boldsymbol{b}$ are model parameters.

Multimodal Deep Learning Model (MMDL). As our benchmarking datasets come from multiple tables and includes both temporal and non-temporal data, multimodal deep learning models [63] can be used to shared learn representations for the prediction tasks. Here, we propose a deep learning framework called as Multimodal Deep Learning Model (MMDL) to learn shared representations from multiple modalities using an ensemble of FFN and GRU deep learning models. The key idea is to use a shared representation layer to capture the correlations of modalities or to learn a similarity of modalities in representation space, which is beneficial when limited data is available from multiple modalities. Data from each of the different tables can be treated as a separate modality. For simplicity, in MMDL, we treat all the temporal features as one modality and all non-temporal features as another modality. Figure 2(b) shows an illustration of our MMDL framework with a common layer to learn the shared representations of modalities. MMDL uses FFN and GRU to handle non-temporal and temporal features respectively, and learns their shared latent representations for prediction tasks, i.e. the non-temporal data is fed into FFN and temporal data is input to GRU and then the outputs of the FFN and GRU are combined in a shared latent representation layer.

4.3. Implementation Details

We implement the Super Learner algorithm using R packages and Python libraries listed in Table 7. The deep learning models are implemented in Theano [64] and Keras [65] platforms. For all the prediction methods, we

conduct five-fold cross validation(by training on 3 folds, validation on 1 fold and report results on the remaining fold) and report the mean and standard error of performance scores of all 5 testing folds. We use Area under the ROC curve (AUROC) and Area under Precision-Recall Curve (AUPRC) as the evaluation metrics to report the prediction model's performance on classification tasks, and use Mean Squared Error (MSE) to report results on the regression task.

We use the default parameters (as listed in their R package) for each base algorithm in the SuperLearner algorithm since we found through grid search that the performance did not vary much with fine-tuning of base algorithms with different parameter settings. For example, we did the grid search of Gradient Boosting Classifier by varying learning rate [0.001, 0.01, 0.1], number of estimators [100,200,500] and max depth [3,6,9], and found that the AUC and AUPRC scores was best for the default parameters and their range was [0.841, 0.846] and [0.426, 0.435] respectively. Thus, we used default hyper parameters for the baseline algorithms of SuperLearner in all our experiments. Related work [30] which proposed SuperLearner algorithm for mortality prediction also used the default parameters to obtain their results.

All the deep learning models are trained with RMSProp optimizer method with learning rate of 0.001 on classification tasks and 0.005 on regression tasks. We used ReLU activation for all models since it is shown to achieve good performance in many classification tasks [66]. We also applied dropout with dropout rate set to 0.1 to all the deep learning models to avoid overfitting[67]. The batch size is chosen as 100 and the max epoch number is fixed at 250. Early stopping with best weight and batch normalization are applied during training. MMDL is a combination of FFN and GRU, in which FFN part handles non-temporal features and GRU part handles temporal features. The structure of MMDL used in our experiments is shown in Figure A.7 in the Appendix A.4.

All the data is divided to 5 folds with stratified cross validation, and standardization is done to the whole dataset with the mean and standard error of the training set. To enable reproducibility of our results, we have released our preprocessing codes and benchmark prediction task codes on Github[5].

For SuperLearner, we have considered the feature statistics such as maximum, minimum, and mean as the representative values for each of the time series data since its lacking of the capability of handling serial data.

4.4. Results

In this section, we report the benchmarking results of all the prediction algorithms on the MIMIC-III datasets. We answer the following questions: (a) How do the Deep Learning models compare to the Super Learner algorithm and scoring systems? (b) What is the performance of prediction methods on the different feature sets?

[5] `https://github.com/USC-Melady/Benchmarking_DL_MIMICIII`

4.4.1. Performance of Super Learner Algorithm Implementations

First, we compare the performance of Super Learner-R and Super Learner-Python softwares on in-hospital mortality prediction task using feature set A i.e. 17 processed features collected in the first 24 hours of ICU admission from MIMIC-III dataset. The result in Table 8 shows that Super Learner-Python performs slightly better than Super Leaner-R implementation. Moreover, Super Learner-Python can be evaluated significantly faster than Super Learner-R. Thus, in the following experiments, we will only report the results of Super Learner-Python version (unless otherwise stated) to evaluate and benchmark Super Learner algorithm on different tasks.

Table 8: Comparison of Super Learner-R and Super Learner-Python software versions on in-hospital mortality prediction task using Feature set A extracted from the first 24-hour data of MIMIC-III. Running time refers to total time taken to perform cross-validation evaluation.

		AUROC score	**AUPRC score**	**Running time**
SuperLearner-I	R version	0.8402 ± 0.0021	0.4304 ± 0.0130	36 hours
	Python version	**0.8448 ± 0.0038**	**0.4351 ± 0.0139**	30 minutes
SuperLearner-II	R version	0.8646 ± 0.0023	0.4917 ± 0.0093	28 hours
	Python version	**0.8701 ± 0.0053**	**0.4991 ± 0.0107**	25 minutes

4.4.2. Mortality Prediction Task Evaluation

Here, we report the performance of all methods described in Section 4.2 on the mortality prediction tasks for benchmark datasets MIMIC-III and MIMIC-III (CareVue). We report the mean and standard deviation of AUROC and AUPRC for all the tasks.

In-hospital Mortality Prediction. Tables 9 and 10 show the in-hospital mortality prediction task results of all the prediction algorithms on Feature Set A of MIMIC-III and MIMIC-III (CareVue) datasets for both 24 hour and 48 hour data. From these tables, we observe that deep learning models such as MMDL and RNN perform better than all the other models on 48-hour data. On 24-hour data, we observe that Super Learner II model obtains slightly better results than deep learning model.

Tables 11, 12, 13, and 14 show the in-hospital mortality prediction task results on Features set B and C of MIMIC-III and MIMIC-III (CareVue) datasets on the 24-hour and 48-hour data. We observe that: (i) Super Learner performs better than all algorithms used in SuperLearner library, (ii) On both the Feature Set B and Feature Set C, the deep learning model (MMDL) obtains the best results in terms of AUROC and AUPRC score, (iii) We can observe that the results on first 48-hour data are similar with those on first 24-hour data, showing that a longer record length helps little on the in-hospital mortality prediction task.

From the in-hospital mortality prediction task results, we make the following observations: (i) Deep learning models (MMDL) outperform all the other models when the raw features (Feature set B and C) are used for evaluation, (ii) All the models perform much better when more features are used for prediction, i.e. models perform better on Feature set C which has 136 raw features compared

Table 9: In-hospital mortality task on MIMIC-III using feature set A.

Method	Algorithm	Feature Set A, 24-hour data		Feature Set A, 48-hour data	
		AUROC Score	AUPRC Score	AUROC Score	AUPRC Score
Score Methods	SAPS-II	0.8035 ± 0.0044	0.3586 ± 0.0052	0.8046 ± 0.0083	0.3373 ± 0.0141
	New SAPS-II	0.8235 ± 0.0042	0.3989 ± 0.0120	0.8252 ± 0.0036	0.3823 ± 0.0119
	SOFA	0.7322 ± 0.0038	0.3191 ± 0.0085	0.7347 ± 0.0094	0.2852 ± 0.0167
Super Learner	SL.glm	0.8235 ± 0.0042	0.3987 ± 0.0120	0.8251 ± 0.0037	0.3828 ± 0.0112
	SL.gbm	0.8435 ± 0.0034	0.4320 ± 0.0125	0.8452 ± 0.0052	0.4163 ± 0.0121
	SL.nnet	0.8388 ± 0.0044	0.4200 ± 0.0135	0.8381 ± 0.0055	0.3989 ± 0.0131
	SL.ipredbagg	0.7556 ± 0.0064	0.3104 ± 0.0084	0.7510 ± 0.0078	0.2811 ± 0.0121
	SL.randomforest	0.7576 ± 0.0085	0.3104 ± 0.0084	0.7538 ± 0.0095	0.2830 ± 0.0121
	SuperLearner-I	0.8448 ± 0.0038	0.4351 ± 0.0139	0.8465 ± 0.0057	0.4190 ± 0.0124
	SL.glm	0.8024 ± 0.0043	0.3804 ± 0.0043	0.8013 ± 0.0021	0.3559 ± 0.0238
	SL.gbm	0.8628 ± 0.0037	0.4840 ± 0.0078	0.8518 ± 0.0049	0.4259 ± 0.0209
	SL.nnet	0.8490 ± 0.0079	0.4587 ± 0.0058	0.8383 ± 0.0058	0.4028 ± 0.0180
	SL.ipredbagg	0.8060 ± 0.0069	0.4087 ± 0.0110	0.7816 ± 0.0028	0.3455 ± 0.0159
	SL.randomforest	0.7977 ± 0.0079	0.3958 ± 0.0124	0.7813 ± 0.0059	0.3496 ± 0.0200
	SuperLearner-II	**0.8673 ± 0.0045**	**0.4968 ± 0.0097**	0.8595 ± 0.0035	0.4422 ± 0.0200
Deep Learning	FFN	0.8496 ± 0.0047	0.4632 ± 0.0074	0.8375 ± 0.0041	0.4090 ± 0.0169
	RNN	0.8544 ± 0.0053	0.4519 ± 0.0145	0.8618 ± 0.0059	0.4458 ± 0.0144
	MMDL	0.8664 ± 0.0056	0.4776 ± 0.0162	**0.8737 ± 0.0045**	**0.4714 ± 0.0176**

Table 10: In-hospital mortality task on MIMIC-III (Carvue) using feature set A.

Method	Algorithm	Feature Set A, 24-hour data		Feature Set A, 48-hour data	
		AUROC Score	AUPRC Score	AUROC Score	AUPRC Score
Score Methods	SAPS-II	0.8005 ± 0.0080	0.3625 ± 0.0065	0.8030 ± 0.0132	0.3448 ± 0.0219
	New SAPS-II	0.8217 ± 0.0047	0.4037 ± 0.0069	0.8226 ± 0.0129	0.3873 ± 0.0163
	SOFA	0.7263 ± 0.0100	0.3273 ± 0.0067	0.7309 ± 0.0105	0.2996 ± 0.0199
Super Learner	SL.glm	0.8212 ± 0.0052	0.4018 ± 0.0068	0.8227 ± 0.0132	0.3883 ± 0.0170
	SL.gbm	0.8405 ± 0.0056	0.4377 ± 0.0112	0.8414 ± 0.0111	0.4187 ± 0.0315
	SL.nnet	0.8332 ± 0.0041	0.4182 ± 0.0059	0.8309 ± 0.0061	0.3905 ± 0.0253
	SL.ipredbagg	0.7567 ± 0.0040	0.3063 ± 0.0098	0.7483 ± 0.0167	0.2921 ± 0.0211
	SL.randomforest	0.7553 ± 0.0058	0.3005 ± 0.0121	0.7538 ± 0.0150	0.2914 ± 0.0154
	SuperLearner-I	0.8417 ± 0.0052	0.4387 ± 0.0122	0.8415 ± 0.0096	0.4169 ± 0.0305
	SL.glm	0.8027 ± 0.0038	0.3931 ± 0.0105	0.8009 ± 0.0149	0.3683 ± 0.0209
	SL.gbm	0.8581 ± 0.0062	0.4810 ± 0.0126	0.8457 ± 0.0080	0.4349 ± 0.0233
	SL.nnet	0.8461 ± 0.0103	0.4674 ± 0.0173	0.8238 ± 0.0157	0.4017 ± 0.0412
	SL.ipredbagg	0.7921 ± 0.0077	0.3850 ± 0.0180	0.7782 ± 0.0052	0.3434 ± 0.0213
	SL.randomforest	0.7930 ± 0.0063	0.3830 ± 0.0091	0.7733 ± 0.0115	0.3455 ± 0.0253
	SuperLearner-II	**0.8651 ± 0.0075**	**0.4964 ± 0.0135**	0.8520 ± 0.0101	0.4493 ± 0.0246
Deep Learning	FFN	0.8488 ± 0.0082	0.4702 ± 0.0168	0.8326 ± 0.0112	0.4109 ± 0.0193
	RNN	0.8456 ± 0.0032	0.4505 ± 0.0091	0.8485 ± 0.0090	0.4246 ± 0.0214
	MMDL	0.8561 ± 0.0045	0.4764 ± 0.0144	**0.8564 ± 0.0107**	**0.4520 ± 0.0305**

Table 11: In-hospital mortality task on MIMIC-III using feature set B.

Method	Algorithm	Feature Set B, 24-hour data		Feature Set B, 48-hour data	
		AUROC Score	AUPRC Score	AUROC Score	AUPRC Score
Super Learner	SL.glm	0.7745 ± 0.0055	0.3134 ± 0.0112	0.7869 ± 0.0015	0.3103 ± 0.0212
	SL.gbm	0.8381 ± 0.0057	0.4059 ± 0.0156	0.8398 ± 0.0044	0.3932 ± 0.0155
	SL.nnet	0.8170 ± 0.0036	0.3650 ± 0.0124	0.8232 ± 0.0074	0.3591 ± 0.0135
	SL.ipredbagg	0.7641 ± 0.0070	0.3127 ± 0.0085	0.7627 ± 0.0118	0.3011 ± 0.0140
	SL.randomforest	0.7582 ± 0.0080	0.3100 ± 0.0116	0.7604 ± 0.0042	0.2895 ± 0.0138
	SuperLearner-II	0.8426 ± 0.0068	0.4160 ± 0.0136	0.8471 ± 0.0036	0.4055 ± 0.0155
Deep Learning	MMDL	**0.8730 ± 0.0065**	**0.4765 ± 0.0109**	**0.8783 ± 0.0037**	**0.4706 ± 0.0178**

Table 12: In-hospital mortality task on MIMIC-III using feature set C.

Method	Algorithm	Feature Set C, 24-hour data		Feature Set C, 48-hour data	
		AUROC Score	AUPRC Score	AUROC Score	AUPRC Score
Super Learner	SL.glm	0.8341 ± 0.0072	0.4045 ± 0.0164	0.8594 ± 0.0079	0.4254 ± 0.0196
	SL.gbm	0.8628 ± 0.0056	0.4705 ± 0.0138	0.8833 ± 0.0054	0.4954 ± 0.0223
	SL.nnet	0.7568 ± 0.0106	0.3424 ± 0.0139	0.7973 ± 0.0060	0.3690 ± 0.0154
	SL.ipredbagg	0.7895 ± 0.0077	0.3664 ± 0.0099	0.8074 ± 0.0100	0.3796 ± 0.0282
	SL.randomforest	0.7720 ± 0.0054	0.3427 ± 0.0045	0.7945 ± 0.0081	0.3616 ± 0.0143
	SuperLearner-II	0.8664 ± 0.0058	0.4821 ± 0.0142	0.8875 ± 0.0055	0.5059 ± 0.0214
Deep Learning	MMDL	**0.9410 ± 0.0082**	**0.7857 ± 0.0132**	**0.9401 ± 0.0099**	**0.7721 ± 0.0078**

to Feature Set B which has 20 features. This implies that deep models can learn better feature representations from multiple data modalities (instead of using hand-picked features as in Feature Set A) which results in obtaining better prediction results on the in-hospital mortality benchmark task. The comparisons of SuperLearner-II and MMDL on three feature sets shown in Figures 3 and 4 validate our observations. From these figues, we see that on Feature set C, deep learning models obtain around 7-8% and 50% improvement over SuperLearner models for AUROC and AUPRC respectively. Also, deep learning models obtain 8% improvement for Feature set C compared to Feature set A.

Short-term and Long-term Mortality Prediction. Tables 15, 16, 17, and 18 show the short-term and long-term mortality prediction task results on all the feature sets of MIMIC-III dataset on the 24-hour and 48-hour data. We observe that: (i) Super Learner-II and MMDL deep learning models have similar performance

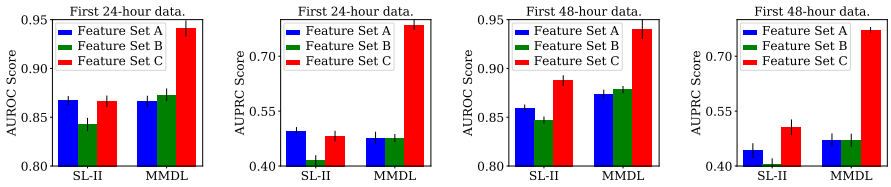


Figure 3: In-hospital mortality task on MIMIC-III data.

Table 13: In-hospital mortality task on MIMIC-III (CareVue) using feature set B.

Method	Algorithm	Feature Set B, 24-hour data		Feature Set B, 48-hour data	
		AUROC Score	AUPRC Score	AUROC Score	AUPRC Score
Super Learner	SL.glm	0.7724 ± 0.0129	0.3177 ± 0.0185	0.7872 ± 0.0139	0.3253 ± 0.0236
	SL.gbm	0.8284 ± 0.0064	0.4048 ± 0.0110	0.8298 ± 0.0089	0.3823 ± 0.0189
	SL.nnet	0.8133 ± 0.0060	0.3740 ± 0.0169	0.7976 ± 0.0201	0.3339 ± 0.0305
	SL.ipredbagg	0.7525 ± 0.0066	0.3158 ± 0.0152	0.7536 ± 0.0136	0.2991 ± 0.0197
	SL.randomforest	0.7497 ± 0.0118	0.3123 ± 0.0110	0.7502 ± 0.0094	0.2952 ± 0.0074
	SuperLearner-II	0.8347 ± 0.0062	0.4164 ± 0.0148	0.8340 ± 0.0099	0.3907 ± 0.0128
Deep Learning	MMDL	**0.8617 ± 0.0074**	**0.4612 ± 0.0245**	**0.8633 ± 0.0080**	**0.4425 ± 0.0166**

Table 14: In-hospital mortality task on MIMIC-III (CareVue) using feature set C.

Method	Algorithm	Feature Set C, 24-hour data		Feature Set C, 48-hour data	
		AUROC Score	AUPRC Score	AUROC Score	AUPRC Score
Super Learner	SL.glm	0.8282 ± 0.0089	0.3931 ± 0.0136	0.8497 ± 0.0088	0.4128 ± 0.0087
	SL.gbm	0.8550 ± 0.0049	0.4605 ± 0.0209	0.8753 ± 0.0026	0.4846 ± 0.0042
	SL.nnet	0.7412 ± 0.0067	0.3216 ± 0.0127	0.7946 ± 0.0132	0.3535 ± 0.0198
	SL.ipredbagg	0.7737 ± 0.0064	0.3216 ± 0.0127	0.7976 ± 0.0065	0.3696 ± 0.0127
	SL.randomforest	0.7690 ± 0.0080	0.3435 ± 0.0184	0.7900 ± 0.0101	0.3558 ± 0.0132
	SuperLearner-II	0.8592 ± 0.0055	0.4694 ± 0.0207	0.8808 ± 0.0030	0.4945 ± 0.0054
Deep Learning	MMDL	**0.9200 ± 0.0213**	**0.7546 ± 0.0297**	**0.9251 ± 0.0120**	**0.7451 ± 0.0093**

on the feature set A for both short-term and long-term mortality prediction, and both these algorithms perform better than all other prediction algorithms, (ii) On both the Feature Set B and Feature Set C, the MMDL deep learning model consistently obtains the best results in terms of AUROC and AUPRC score, (iii) all models obtain better AUPRC scores on the long-term mortality prediction task compared to the short-term mortality prediction task.

4.4.3. ICD-9 Code Prediction Task Evaluation

Tables 19 and 20 show the performance (AUPRC and AUROC scores) of all methods for the first 24-hour data of MIMIC-III on ICD-9 code prediction task. We observe that the MMDL deep models trained on Feature Set C outperforms Super Learner models trained on Feature Sets A, B, and C on almost all the ICD-9 Code prediction task, and on an average obtains 4-5% improvement.

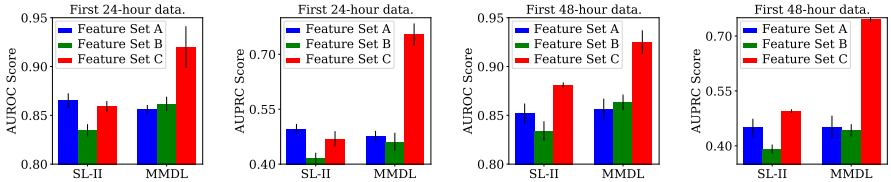


Figure 4: In-hospital mortality task on MIMIC-III (CareVue) data.

Table 15: AUROC scores of short-term and long-term mortality prediction tasks on MIMIC-III with 24-hour data.

Feature set	Algorithm	AUROC score			
		2-day Mortality	3-day Mortality	30-day Mortality	1-year Mortality
Feature Set A	SAPS-II Score	0.8453 ± 0.0088	0.8218 ± 0.0057	0.7921 ± 0.0051	0.7614 ± 0.0035
	New SAPS-II Score	0.8575 ± 0.0075	0.8370 ± 0.0053	0.8148 ± 0.0035	0.8042 ± 0.0013
	SOFA Score	0.7559 ± 0.0276	0.7412 ± 0.0076	0.7041 ± 0.0074	0.6611 ± 0.0036
	SuperLearner-I	0.8808 ± 0.0063	0.8627 ± 0.0079	0.8384 ± 0.0031	0.8260 ± 0.0019
	SuperLearner-II	0.8851 ± 0.0105	0.8770 ± 0.0094	0.8620 ± 0.0063	0.8467 ± 0.0022
	FFN	0.8673 ± 0.0069	0.8493 ± 0.0128	0.8475 ± 0.0050	0.8390 ± 0.0019
	RNN	0.8773 ± 0.0117	0.8612 ± 0.0083	0.8326 ± 0.0085	0.7958 ± 0.0026
	MMDL	0.8815 ± 0.0102	0.8725 ± 0.0063	0.8585 ± 0.0059	0.8450 ± 0.0019
Feature Set B	SuperLearner-II	0.8667 ± 0.0097	0.8535 ± 0.0128	0.8395 ± 0.0031	0.8347 ± 0.0046
	MMDL	0.8862 ± 0.0059	0.8769 ± 0.0107	0.8620 ± 0.0072	0.8452 ± 0.0008
Feature Set C	SuperLearner-II	0.8837 ± 0.0047	0.8746 ± 0.0073	0.8629 ± 0.0033	0.8589 ± 0.0032
	MMDL	**0.9084 ± 0.0207**	**0.9295 ± 0.0225**	**0.9169 ± 0.0054**	**0.8872 ± 0.0084**

Table 16: AUPRC scores of short-term and long-term mortality prediction tasks on MIMIC-III with 24-hour data.

Feature set	Algorithm	AUPRC score			
		2-day Mortality	3-day Mortality	30-day Mortality	1-year Mortality
Feature Set A	SAPS-II Score	0.1361 ± 0.0153	0.1730 ± 0.0214	0.4140 ± 0.0131	0.5084 ± 0.0154
	New SAPS-II Score	0.1587 ± 0.0226	0.1919 ± 0.0234	0.4589 ± 0.0125	0.5778 ± 0.0109
	SOFA Score	0.1027 ± 0.0278	0.1373 ± 0.0201	0.3497 ± 0.0167	0.4176 ± 0.0088
	SuperLearner-I	0.1967 ± 0.0205	0.2219 ± 0.0263	0.5053 ± 0.0173	0.6258 ± 0.0073
	SuperLearner-II	0.2463 ± 0.0111	0.2775 ± 0.0382	0.5652 ± 0.0186	0.6609 ± 0.0090
	FFN	0.2429 ± 0.0332	0.2449 ± 0.0315	0.5367 ± 0.0199	0.6453 ± 0.0081
	RNN	0.2491 ± 0.0293	0.2752 ± 0.0164	0.5028 ± 0.0178	0.5725 ± 0.0062
	MMDL	0.2529 ± 0.0338	0.2839 ± 0.0207	0.5483 ± 0.0187	0.6485 ± 0.0099
Feature Set B	SuperLearner-II	0.1767 ± 0.0319	0.2173 ± 0.0266	0.4926 ± 0.0090	0.6328 ± 0.0100
	MMDL	0.2475 ± 0.0364	0.1863 ± 0.0273	0.5458 ± 0.0231	0.6457 ± 0.0082
Feature Set C	SuperLearner-II	0.2048 ± 0.0085	0.2717 ± 0.0321	0.5530 ± 0.0096	0.6764 ± 0.0056
	MMDL	**0.3831 ± 0.0336**	**0.5139 ± 0.0193**	**0.7668 ± 0.0170**	**0.7690 ± 0.0077**

4.4.4. Length of Stay Prediction Task Evaluation

Table 21 shows the performance measured by Mean Squared Error (MSE) in minutes of all methods on the task of forecasting length of stay task with first 24-hour data and first 48-hour data of MIMIC-III dataset. We observe that (i) all deep learning models such as FFN, RRN and MMDL trained on Feature set C outperform Super Learner models trained on Feature sets A,B, and C. (ii) MMDL model obtains best performance in terms of mean squared error (in hours), and significantly outperforms Super Learner II by nearly 50%. Figures 5 and 6 show the distribution of length of stay predictions from our MMDL with respect to the ground-truth (i.e. actual length of stay) on Feature set C for first 24-hour and 48-hour data respectively. These plots show that our MMDL models predict the Length of stay quite accurately and in-fact our models' predictions are within 7.5 to 8.2 days of ground-truth for Feature set C.

Table 17: AUROC scores of short-term and long-term mortality prediction tasks on MIMIC-III with 48-hour data.

Feature set	Algorithm	AUROC score		
		3-day Mortality	30-day Mortality	1-year Mortality
Feature Set A	SAPS-II Score	0.8366 ± 0.0109	0.7841 ± 0.0072	0.7490 ± 0.0041
	New SAPS-II Score	0.8471 ± 0.0072	0.8104 ± 0.0047	0.7991 ± 0.0037
	SOFA Score	0.7465 ± 0.0179	0.6953 ± 0.0104	0.6454 ± 0.0052
	SuperLearner-I	0.8675 ± 0.0046	0.8364 ± 0.0033	0.8222 ± 0.0047
	SuperLearner-II	0.8706 ± 0.0095	0.8531 ± 0.0043	0.8409 ± 0.0031
	FFN	0.8466 ± 0.0186	0.8385 ± 0.0061	0.8309 ± 0.0048
	RNN	0.8633 ± 0.0116	0.8374 ± 0.0087	0.7966 ± 0.0036
	MMDL	0.8596 ± 0.0124	0.8612 ± 0.0059	0.8418 ± 0.0049
Feature Set B	SuperLearner-II	0.8448 ± 0.0162	0.8427 ± 0.0071	0.8360 ± 0.0057
	MMDL	0.8682 ± 0.0240	0.8628 ± 0.0111	0.8438 ± 0.0053
Feature Set C	SuperLearner-II	0.8473 ± 0.0114	0.8802 ± 0.0037	0.8673 ± 0.0051
	MMDL	**0.8713 ± 0.0494**	**0.9173 ± 0.0064**	**0.8702 ± 0.0054**

Table 18: AUPRC scores of short-term and long-term mortality prediction tasks on MIMIC-III with 48-hour data.

Feature set	Algorithm	AUPRC score		
		3-day Mortality	30-day Mortality	1-year Mortality
Feature Set A	SAPS-II Score	0.1082 ± 0.0150	0.3849 ± 0.0118	0.4845 ± 0.0092
	New SAPS-II Score	0.1307 ± 0.0225	0.4342 ± 0.0119	0.5647 ± 0.0090
	SOFA Score	0.0663 ± 0.0092	0.3156 ± 0.0143	0.3898 ± 0.0115
	SuperLearner-I	0.1344 ± 0.0247	0.4898 ± 0.0139	0.6171 ± 0.0088
	SuperLearner-II	0.1955 ± 0.0245	0.5255 ± 0.0152	0.6448 ± 0.0084
	FFN	0.1672 ± 0.0331	0.4962 ± 0.0153	0.6272 ± 0.0116
	RNN	**0.2371 ± 0.0336**	0.4974 ± 0.0149	0.5691 ± 0.0080
	MMDL	0.2131 ± 0.0344	0.5423 ± 0.0164	0.6421 ± 0.0116
Feature Set B	SuperLearner-II	0.1225 ± 0.0286	0.4892 ± 0.0197	0.6297 ± 0.0067
	MMDL	0.1659 ± 0.0434	0.5290 ± 0.0372	0.6444 ± 0.0133
Feature Set C	SuperLearner-II	0.0771 ± 0.0125	0.5479 ± 0.0079	0.6870 ± 0.0038
	MMDL	0.1510 ± 0.0246	**0.7314 ± 0.0149**	**0.7344 ± 0.0062**

4.4.5. Statistical tests to compare performance of models

From the above results in sections 4.4.2-4.4.4, we see that our deep learning model (MMDL) mostly outperforms the SuperLearner on all the tasks (Mortality, Length of Stay and ICD-9 prediction). In general, we observe that our deep models perform better than SuperLearner models when we train it with lot of data points. We also observe that our model performs similar or slightly worse than SuperLearner algorithms on two out of 20 ICD-9 tasks (task 11 and 19) as shown in Table 20 (as measured by AUROC score) and one out of 20 ICD-9 tasks (task 4) as shown in Table 19 (as measured by AUPRC score), because these tasks have smaller number of data points. To check if the performance

Table 19: ICD-9 code prediction AUPRC scores on MIMIC-III with first 24-hour data.

ICD-9 task	Super Learner				Deep Learning		
	Super Learner-I on Feature Set A	Super Learner-II on Feature Set A	Super Learner-II on Feature Set B	Super Learner-II on Feature Set C	FFN on Feature Set C	RNN on Feature Set C	MMDL on Feature Set C
1	0.5356 ± 0.0027	0.5861 ± 0.0090	0.5695 ± 0.0075	0.6273 ± 0.0077	0.5807 ± 0.0072	0.5978 ± 0.0211	**0.6491 ± 0.0121**
2	0.7290 ± 0.0141	0.7512 ± 0.0130	0.7478 ± 0.0097	0.7756 ± 0.0134	0.7422 ± 0.0064	0.4715 ± 0.0132	**0.8024 ± 0.0192**
3	0.8095 ± 0.0057	0.8235 ± 0.0054	0.8302 ± 0.0057	0.8631 ± 0.0042	0.8377 ± 0.0055	0.8614 ± 0.0022	**0.8690 ± 0.0115**
4	0.5454 ± 0.0072	0.5850 ± 0.0065	0.5836 ± 0.0037	0.6882 ± 0.0128	0.6556 ± 0.0162	0.6812 ± 0.0125	**0.7149 ± 0.0180**
5	0.4714 ± 0.0047	0.5168 ± 0.0073	0.5218 ± 0.0044	**0.5624 ± 0.0019**	0.5058 ± 0.0100	0.5220 ± 0.0048	0.5590 ± 0.0212
6	0.4112 ± 0.0046	0.4348 ± 0.0064	0.4507 ± 0.0067	0.5155 ± 0.0074	0.4566 ± 0.0175	0.5261 ± 0.0076	**0.5624 ± 0.0112**
7	0.9547 ± 0.0026	0.9596 ± 0.0025	0.9585 ± 0.0019	0.9701 ± 0.0018	0.9592 ± 0.0013	0.9552 ± 0.0040	**0.9729 ± 0.0021**
8	0.6960 ± 0.0061	0.7366 ± 0.0045	0.7306 ± 0.0079	0.7649 ± 0.0038	0.7493 ± 0.0050	0.8067 ± 0.0062	**0.8290 ± 0.0113**
9	0.5814 ± 0.0080	0.6350 ± 0.0107	0.6326 ± 0.0087	0.6961 ± 0.0091	0.6393 ± 0.0033	0.6748 ± 0.0106	**0.7034 ± 0.0148**
10	0.7213 ± 0.0034	0.7442 ± 0.0047	0.7328 ± 0.0063	0.7928 ± 0.0053	0.7715 ± 0.0036	0.8071 ± 0.0020	**0.8227 ± 0.0112**
11	0.0700 ± 0.0180	0.0965 ± 0.0193	0.0857 ± 0.0082	0.2303 ± 0.0345	0.1498 ± 0.0565	0.1623 ± 0.0504	**0.4888 ± 0.0631**
12	0.1629 ± 0.0106	0.1802 ± 0.0109	0.1862 ± 0.0138	0.2146 ± 0.0213	0.1930 ± 0.0117	0.2019 ± 0.0149	**0.3155 ± 0.0421**
13	0.2436 ± 0.0070	0.2551 ± 0.0062	0.2554 ± 0.0051	0.3144 ± 0.0074	0.2641 ± 0.0078	0.2945 ± 0.0155	**0.3435 ± 0.0293**
14	0.1318 ± 0.0177	0.1357 ± 0.0219	0.1370 ± 0.0198	0.1329 ± 0.0183	0.1134 ± 0.0225	0.0775 ± 0.0068	**0.1918 ± 0.0216**
15	0.4745 ± 0.0061	0.5036 ± 0.0032	0.4908 ± 0.0061	0.5343 ± 0.0096	0.4848 ± 0.0096	0.5182 ± 0.0079	**0.5564 ± 0.0195**
16	0.1164 ± 0.0057	0.1279 ± 0.0082	0.1264 ± 0.0080	0.1711 ± 0.0097	0.1488 ± 0.0093	0.1433 ± 0.0104	**0.2244 ± 0.0279**
17	0.0632 ± 0.0113	0.0649 ± 0.0028	0.0736 ± 0.0048	0.0913 ± 0.0101	0.0742 ± 0.0055	0.0604 ± 0.0072	**0.3700 ± 0.0577**
18	0.5934 ± 0.0069	0.6302 ± 0.0053	0.6300 ± 0.0059	0.6826 ± 0.0072	0.6430 ± 0.0133	0.6675 ± 0.0068	**0.7037 ± 0.0246**
19	0.5946 ± 0.0029	0.6205 ± 0.0053	0.6361 ± 0.0065	0.7138 ± 0.0076	0.6684 ± 0.0041	0.6951 ± 0.0049	**0.7184 ± 0.0086**
20	0.4820 ± 0.0069	0.5220 ± 0.0128	0.5382 ± 0.0054	0.6006 ± 0.0103	0.5506 ± 0.0066	0.5723 ± 0.0041	**0.6184 ± 0.0208**
Average	0.4694 ± 0.0076	0.4955 ± 0.0083	0.4959 ± 0.0073	0.5471 ± 0.0102	0.5094 ± 0.0111	0.5148 ± 0.0107	**0.6008 ± 0.0224**

Table 20: ICD-9 code prediction AUROC scores on MIMIC-III with first 24-hour data.

ICD-9 task	Super Learner				Deep Learning		
	Super Learner-I on Feature Set A	Super Learner-II on Feature Set A	Super Learner-II on Feature Set B	Super Learner-II on Feature Set C	FFN on Feature Set C	RNN on Feature Set C	MMDL on Feature Set C
1	0.7371 ± 0.0023	0.7757 ± 0.0049	0.7635 ± 0.0018	0.7971 ± 0.0044	0.7643 ± 0.0047	0.7879 ± 0.0133	**0.8194 ± 0.0078**
2	0.8342 ± 0.0091	0.8530 ± 0.0082	0.8472 ± 0.0056	0.8770 ± 0.0092	0.8467 ± 0.0062	0.7651 ± 0.0067	**0.8912 ± 0.0154**
3	0.6835 ± 0.0073	0.7022 ± 0.0070	0.7108 ± 0.0074	0.7532 ± 0.0061	0.7203 ± 0.0055	0.7460 ± 0.0059	**0.7604 ± 0.0158**
4	0.6714 ± 0.0052	0.7048 ± 0.0037	0.7038 ± 0.0023	0.7814 ± 0.0049	0.7575 ± 0.0067	0.7831 ± 0.0038	**0.8048 ± 0.0131**
5	0.6540 ± 0.0036	0.6833 ± 0.0047	0.6825 ± 0.0042	0.7163 ± 0.0036	0.6680 ± 0.0082	0.6921 ± 0.0032	**0.7184 ± 0.0169**
6	0.6222 ± 0.0028	0.6505 ± 0.0068	0.6526 ± 0.0027	0.7107 ± 0.0061	0.6651 ± 0.0071	0.7109 ± 0.0061	**0.7387 ± 0.0122**
7	0.8270 ± 0.0071	0.8457 ± 0.0067	0.8424 ± 0.0073	0.8753 ± 0.0060	0.8435 ± 0.0052	0.8339 ± 0.0068	**0.8942 ± 0.0045**
8	0.6976 ± 0.0049	0.7340 ± 0.0038	0.7315 ± 0.0066	0.7636 ± 0.0027	0.7447 ± 0.0029	0.7957 ± 0.0066	**0.8148 ± 0.0093**
9	0.6518 ± 0.0063	0.6980 ± 0.0093	0.6993 ± 0.0075	0.7512 ± 0.0107	0.7023 ± 0.0047	0.7298 ± 0.0102	**0.7527 ± 0.0105**
10	0.7807 ± 0.0024	0.7978 ± 0.0039	0.7881 ± 0.0041	0.8266 ± 0.0043	0.8089 ± 0.0016	0.8354 ± 0.0022	**0.8512 ± 0.0105**
11	0.9454 ± 0.0094	0.9635 ± 0.0060	0.9592 ± 0.0038	**0.9692 ± 0.0081**	0.8200 ± 0.0778	0.8268 ± 0.0322	0.9035 ± 0.0450
12	0.6340 ± 0.0151	0.6578 ± 0.0117	0.6638 ± 0.0161	0.6969 ± 0.0189	0.6670 ± 0.0068	0.6554 ± 0.0132	**0.7378 ± 0.0141**
13	0.5908 ± 0.0067	0.6078 ± 0.0049	0.6141 ± 0.0040	0.6727 ± 0.0055	0.6185 ± 0.0060	0.6409 ± 0.0088	**0.6873 ± 0.0133**
14	0.7170 ± 0.0172	0.7172 ± 0.0157	0.7204 ± 0.0211	0.7166 ± 0.0143	0.7003 ± 0.0128	0.6191 ± 0.0172	**0.7346 ± 0.0251**
15	0.6507 ± 0.0046	0.6774 ± 0.0048	0.6603 ± 0.0050	0.6938 ± 0.0028	0.6583 ± 0.0057	0.6897 ± 0.0063	**0.7148 ± 0.0135**
16	0.5901 ± 0.0156	0.6200 ± 0.0117	0.6176 ± 0.0144	0.6615 ± 0.0192	0.6243 ± 0.0121	0.6300 ± 0.0116	**0.7011 ± 0.0207**
17	0.6769 ± 0.0211	0.6917 ± 0.0094	0.6991 ± 0.0074	0.7530 ± 0.0236	0.6976 ± 0.0151	0.6144 ± 0.0223	**0.8104 ± 0.0381**
18	0.6405 ± 0.0076	0.6683 ± 0.0067	0.6676 ± 0.0066	0.7050 ± 0.0076	0.6722 ± 0.0115	0.7007 ± 0.0057	**0.7319 ± 0.0240**
19	0.6385 ± 0.0054	0.6619 ± 0.0068	0.6762 ± 0.0066	**0.7358 ± 0.0052**	0.6931 ± 0.0023	0.7127 ± 0.0019	0.7348 ± 0.0070
20	0.6263 ± 0.0071	0.6538 ± 0.0095	0.6656 ± 0.0043	0.7175 ± 0.0089	0.6719 ± 0.0053	0.7108 ± 0.0054	**0.7426 ± 0.0136**
Average	0.6935 ± 0.0080	0.7182 ± 0.0073	0.7183 ± 0.0069	0.7587 ± 0.0086	0.7172 ± 0.0104	0.7240 ± 0.0095	**0.7772 ± 0.0165**

ACCEPTED MANUSCRIPT

Table 21: Length of stay task on MIMIC-III with first 24/48-hour data. Mean squared error (MSE) shown is in minutes.

Model and feature set		First 24-hour data	First 48-hour data
Super Learner	Super Learner-I on Feature Set A	56 420.6077 ± 3739.0173	58 561.0081 ± 4223.1785
	Super Learner-II on Feature Set A	54 593.3317 ± 3265.8292	57 454.7028 ± 4349.3598
	Super Learner-II on Feature Set B	55 844.4209 ± 3248.3224	54 666.0875 ± 4859.4577
	Super Learner-II on Feature Set C	54 608.1099 ± 2923.9972	54 400.5845 ± 1582.4523
Deep Learning	FFN on Feature Set C	53 410.0918 ± 3207.9849	52 642.6508 ± 4373.4239
	RNN on Feature Set C	48 702.7641 ± 3768.5154	49 556.8024 ± 3794.0471
	MMDL on Feature Set C	**36 338.2015 ± 2672.3832**	**36 924.2312 ± 3566.4318**

of deep learning models are statistically significant we conducted parametric and non-parameteric statistical tests. First, we applied the normality test of Kolmogorov Smirnov (K-S) as suggested in [68], and found that our data did not pass the K-S test indicating that we cannot perform parametric tests. We note that since we only use MIMIC-III datasets, it is not possible to perform the rank-based non-parametric tests for comparing different algorithms on the mortality classification task since the rule of thumb to apply the non-parametric tests is at least 10 datasets and 5 algorithms [68]. Therefore, we performed rank-based non-parametric tests (such as Friedmans test [24]) followed by multiple comparison test (Bonferroni - Dunns procedure) for 20 ICD-9 classification tasks. Table 22 shows the pairwise comparison results using Friedman's test for multiple comparisons[6] with Bonferroni adjustment of p-values. From these tests, we found that the p-value was 1.248e-91 for rejecting the null hypothesis (null hypothesis was that all algorithms have similar performance), and we obtained a mean rank of 1.15 for MMDL and a mean rank of 4.32 for SuperLearner-II algorithm (lower rank means better performing algorithm). Thus, the rank-based non-parametric statistical tests show that our deep learning model namely MMDL is statistically better than the SuperLearner algorithm for ICD-9 classification tasks on all the feature sets.

4.4.6. Computation Time

Our Python implementation of the Super Learner algorithm took about 25-30 mins for evaluating the in-hospital mortality task using Feature Set A, and it took about 3 hours for the Feature Set C. A deep Feed forward neural (FFN) network implemented using Keras took around 90 and 100 minutes for evaluating the same mortality task using Feature sets A and C respectively, while the MMDL model (shown in Figure A.7in the Appendix A.4) took around 30 minutes and 1 hour for Feature sets A and C respectively. All our experiments was run on a 32-core Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz machine with NVIDIA TITAN-X GPU processor.

[6] We used Matlab's multcompare `https://www.mathworks.com/help/stats/multcompare.html` for multiple comparison test.

Table 22: Pairwise comparisons using Friedman's test for multiple comparisons with Bonferroni adjustment of p-values (Bonferroni-Dunn procedure). Here, SL-I and SL-II correspond to SuperLearner-I and SuperLearner-II respectively; FS A, FS B, and FS C respectively correspond to Feature Set A, B and C. All deep learning models are evaluated on Feature Set C.

Algorithm 1	Algorithm 2	Lower bound	Estimate	Upper bound	p-value for comparison
SL-I (FS A)	SL-II (FS A)	-2.588	-1.660	-0.732	1.16E-06
SL-I (FS A)	SL-II (FS B)	-2.588	-1.660	-0.732	1.16E-06
SL-I (FS A)	SL-II (FS C)	-5.358	-4.430	-3.502	2.52E-46
SL-I (FS A)	FFN	-3.258	-2.330	-1.402	5.06E-13
SL-I (FS A)	RNN	-3.898	-2.970	-2.042	5.12E-21
SL-I (FS A)	MMDL	-6.428	-5.500	-4.572	3.88E-71
SL-II (FS A)	SL-II (FS B)	-0.928	0.000	0.928	1.00E+00
SL-II (FS A)	SL-II (FS C)	-3.698	-2.770	-1.842	2.57E-18
SL-II (FS A)	FFN	-1.598	-0.670	0.258	5.94E-01
SL-II (FS A)	RNN	-2.238	-1.310	-0.382	3.79E-04
SL-II (FS A)	MMDL	-4.768	-3.840	-2.912	6.54E-35
SL-II (FS B)	SL-II (FS C)	-3.698	-2.770	-1.842	2.57E-18
SL-II (FS B)	FFN	-1.598	-0.670	0.258	5.94E-01
SL-II (FS B)	RNN	-2.238	-1.310	-0.382	3.79E-04
SL-II (FS B)	MMDL	-4.768	-3.840	-2.912	6.54E-35
SL-II (FS C)	FFN	1.172	2.100	3.028	1.31E-10
SL-II (FS C)	RNN	0.532	1.460	2.388	3.70E-05
SL-II (FS C)	MMDL	-1.998	-1.070	-0.142	9.68E-03
FFN	RNN	-1.568	-0.640	0.288	7.60E-01
FFN	MMDL	-4.098	-3.170	-2.242	6.68E-24
RNN	MMDL	-3.458	-2.530	-1.602	2.56E-15

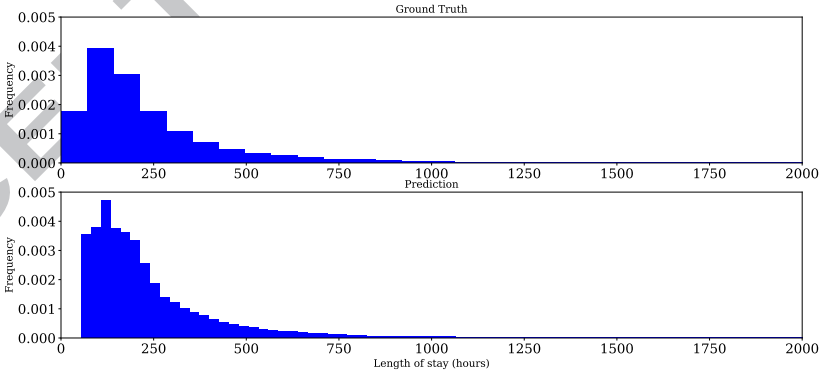


Figure 5: Comparision of the distribution with ground truth of lengths of stay predicted by MMDL on Feature Set C with first 24-hour data.

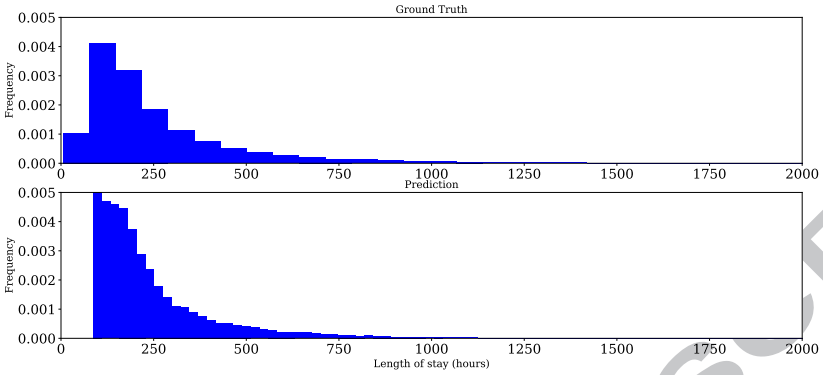


Figure 6: Comparision of the distribution with ground truth of lengths of stay predicted by MMDL on Feature Set C with first 48-hour data.

5. Summary

In this paper, we presented exhaustive benchmarking evaluation results of deep learning models, several machine learning models and ICU scoring systems on various clinical prediction tasks using the publicly available MIMIC-III datasets. We demonstrated that deep learning models consistently outperform all the other approaches especially when a large number of raw clinical time series data is used as input features to the prediction models.

Acknowledgments

This material is based upon work supported by the NSF research grants IIS-1134990, IIS-1254206, and Samsung GRO Grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

[1] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, arXiv preprint arXiv:1606.01865 (2016).

[2] H. Harutyunyan, H. Khachatrian, D. C. Kale, A. Galstyan, Multitask learning and benchmarking with clinical time series data, arXiv preprint arXiv:1703.07771 (2017).

[3] J.-R. Le Gall, S. Lemeshow, F. Saulnier, A new simplified acute physiology score (saps ii) based on a european/north american multicenter study, Jama 270 (1993) 2957–2963.

[4] J. Lee, D. J. Scott, M. Villarroel, G. D. Clifford, M. Saeed, R. G. Mark, Open-access mimic-ii database for intensive care research, in: Engineering in

Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, IEEE, 2011, pp. 8315–8318.

[5] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, R. G. Mark, Mimic-iii, a freely accessible critical care database, Scientific data 3 (2016).

[6] A. E. Johnson, T. J. Pollard, R. G. Mark, Reproducibility in critical care: a mortality prediction case study, Machine Learning for Healthcare (MLHC) (2017).

[7] Z. Che, D. Kale, W. Li, M. T. Bahadori, Y. Liu, Deep computational phenotyping, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 507–516.

[8] Z. Che, S. Purushotham, R. Khemani, Y. Liu, Interpretable deep models for icu outcome prediction, in: AMIA Annual Symposium Proceedings, volume 2016, American Medical Informatics Association, 2016, p. 371.

[9] K. L. Caballero Barajas, R. Akella, Dynamically modeling patient's health state from electronic medical records: a time series approach, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 69–78.

[10] J. Calvert, Q. Mao, A. J. Rogers, C. Barton, M. Jay, T. Desautels, H. Mohamadlou, J. Jan, R. Das, A computational approach to mortality prediction of alcohol use disorder inpatients, Computers in biology and medicine 75 (2016) 74–79.

[11] L. A. Celi, S. Galvin, G. Davidzon, J. Lee, D. Scott, R. Mark, A database-driven decision support system: customized mortality prediction, Journal of personalized medicine 2 (2012) 138–148.

[12] M. Ghassemi, M. A. Pimentel, T. Naumann, T. Brennan, D. A. Clifton, P. Szolovits, M. Feng, A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data., in: AAAI, 2015, pp. 446–453.

[13] M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, P. Szolovits, Unfolding physiological state: Mortality modelling in intensive care units, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 75–84.

[14] M. Hoogendoorn, A. el Hassouni, K. Mok, M. Ghassemi, P. Szolovits, Prediction using patient comparison vs. modeling: A case study for mortality prediction, in: Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the, IEEE, 2016, pp. 2464–2467.

[15] R. Joshi, P. Szolovits, Prognostic physiology: modeling patient severity in intensive care units using radial domain folding, in: AMIA Annual Symposium Proceedings, volume 2012, American Medical Informatics Association, 2012, p. 1276.

[16] J. Lee, D. M. Maslove, Customization of a severity of illness score using local electronic medical record data, Journal of intensive care medicine 32 (2017) 38–47.

[17] L.-w. Lehman, M. Saeed, W. Long, J. Lee, R. Mark, Risk stratification of icu patients using topic models inferred from unstructured progress notes, in: AMIA annual symposium proceedings, volume 2012, American Medical Informatics Association, 2012, p. 505.

[18] Y. Luo, Y. Xin, R. Joshi, L. A. Celi, P. Szolovits, Predicting icu mortality risk by grouping temporal trends from a multivariate panel of physiologic measurements., in: AAAI, 2016, pp. 42–50.

[19] S. Joshi, S. Gunasekar, D. Sontag, G. Joydeep, Identifiable phenotyping using constrained non-negative matrix factorization, in: Machine Learning for Healthcare Conference, 2016, pp. 17–41.

[20] J. Lee, D. M. Maslove, J. A. Dubin, Personalized mortality prediction driven by electronic medical data and a patient similarity metric, PloS one 10 (2015) e0127428.

[21] J. Lee, Patient-specific predictive modeling using random forests: An observational study for the critically ill, JMIR medical informatics 5 (2017).

[22] Y.-F. Luo, A. Rumshisky, Interpretable topic features for post-icu mortality prediction, in: AMIA Annual Symposium Proceedings, volume 2016, American Medical Informatics Association, 2016, p. 827.

[23] J.-L. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. Reinhart, P. Suter, L. Thijs, The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure, Intensive care medicine 22 (1996) 707–710.

[24] W. A. Knaus, J. E. Zimmerman, D. P. Wagner, E. A. Draper, D. E. Lawrence, Apache-acute physiology and chronic health evaluation: a physiologically based classification system., Critical care medicine 9 (1981) 591–597.

[25] R. Dybowski, V. Gant, P. Weller, R. Chang, Prediction of outcome in critically ill patients using artificial neural network synthesised by genetic algorithm, The Lancet 347 (1996) 1146–1150.

[26] S. Kim, W. Kim, R. W. Park, A comparison of intensive care unit mortality prediction models through the use of data mining techniques, Healthcare informatics research 17 (2011) 232–243.

[27] J. V. Tu, M. R. Guerriere, Use of a neural network as a predictive instrument for length of stay in the intensive care unit following cardiac surgery, Computers and biomedical research 26 (1993) 220–229.

[28] G. Doig, K. Inman, W. Sibbald, C. Martin, J. Robertson, Modeling mortality in the intensive care unit: comparing the performance of a back-propagation, associative-learning neural network with multivariate logistic regression., in: Proceedings of the Annual Symposium on Computer Application in Medical Care, American Medical Informatics Association, 1993, p. 361.

[29] C. W. Hanson III, B. E. Marshall, Artificial intelligence applications in the intensive care unit, Critical care medicine 29 (2001) 427–435.

[30] R. Pirracchio, Mortality prediction in the icu based on mimic-ii results from the super icu learner algorithm (sicula) project, in: Secondary Analysis of Electronic Health Records, Springer, 2016, pp. 295–313.

[31] E. C. Polley, M. J. Van der Laan, Super learner in prediction, U.C. Berkeley Division of Biostatistics Working Paper Series (2010).

[32] R. Caruana, S. Baluja, T. Mitchell, Using the future to" sort out" the present: Rankprop and multitask learning for medical risk evaluation, in: Advances in neural information processing systems, 1996, pp. 959–965.

[33] G. F. Cooper, C. F. Aliferis, R. Ambrosino, J. Aronis, B. G. Buchanan, R. Caruana, M. J. Fine, C. Glymour, G. Gordon, B. H. Hanusa, et al., An evaluation of machine-learning methods for predicting pneumonia mortality, Artificial intelligence in medicine 9 (1997) 107–138.

[34] T. A. Lasko, J. C. Denny, M. A. Levy, Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data, PloS one 8 (2013) e66341.

[35] A. Oellrich, N. Collier, T. Groza, D. Rebholz-Schuhmann, N. Shah, O. Bodenreider, M. R. Boland, I. Georgiev, H. Liu, K. Livingston, et al., The digital revolution in phenotyping, Briefings in bioinformatics (2015) bbv083.

[36] Z. Che, S. Purushotham, R. Khemani, Y. Liu, Distilling knowledge from deep networks with applications to healthcare domain, arXiv preprint arXiv:1512.03542 (2015).

[37] F. Dabek, J. J. Caban, A neural network based model for predicting psychological conditions, in: Brain Informatics and Health, Springer, 2015, pp. 252–261.

[38] N. Y. Hammerla, J. M. Fisher, P. Andras, L. Rochester, R. Walker, T. Plötz, Pd disease state assessment in naturalistic environments using deep learning, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 1742–1748.

[39] Z. C. Lipton, D. C. Kale, C. Elkan, R. Wetzell, Learning to diagnose with lstm recurrent neural networks, arXiv preprint arXiv:1511.03677 (2015).

[40] S. Purushotham, W. Carvalho, T. Nilanon, Y. Liu, Variational recurrent adversarial deep domain adaptation, International Conference on Learning Representations (ICLR) (2017).

[41] H. L. Li-wei, R. P. Adams, L. Mayaud, G. B. Moody, A. Malhotra, R. G. Mark, S. Nemati, A physiological time series dynamics-based approach to patient monitoring and outcome prediction, IEEE journal of biomedical and health informatics 19 (2015) 1068–1076.

[42] J. Lee, Patient-specific predictive modeling using random forests: An observational study for the critically ill, JMIR medical informatics 5 (2017).

[43] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[44] I. Silva, G. Moody, D. J. Scott, L. A. Celi, R. G. Mark, Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012, in: Computing in Cardiology (CinC), 2012, IEEE, 2012, pp. 245–248.

[45] M. J. Van der Laan, E. C. Polley, A. E. Hubbard, Super learner, Statistical applications in genetics and molecular biology 6 (2007).

[46] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[47] H. Larochelle, Y. Bengio, J. Louradour, P. Lamblin, Exploring strategies for training deep neural networks, Journal of Machine Learning Research 10 (2009) 1–40.

[48] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (2013) 1798–1828.

[49] G. Dahl, A.-r. Mohamed, G. E. Hinton, et al., Phone recognition with the mean-covariance restricted boltzmann machine, in: Advances in neural information processing systems, 2010, pp. 469–477.

[50] G. E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Transactions on audio, speech, and language processing 20 (2012) 30–42.

[51] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[52] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675–678.

[53] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[54] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, J. Černockỳ, Empirical evaluation and combination of advanced language modeling techniques, in: Twelfth Annual Conference of the International Speech Communication Association, 2011, pp. 605–608.

[55] A. Bordes, X. Glorot, J. Weston, Y. Bengio, Joint learning of words and meaning representations for open-text semantic parsing, in: Artificial Intelligence and Statistics, 2012, pp. 127–135.

[56] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations., in: hlt-Naacl, volume 13, 2013, pp. 746–751.

[57] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[58] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (1989) 359–366.

[59] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning (ICML), 2010, pp. 807–814.

[60] R. J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural computation 1 (1989) 270–280.

[61] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv:1409.1259 (2014).

[62] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).

[63] N. Srivastava, R. R. Salakhutdinov, Multimodal learning with deep boltzmann machines, in: Advances in neural information processing systems, 2012, pp. 2222–2230.

[64] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, Y. Bengio, Theano: new features and speed improvements, arXiv preprint arXiv:1211.5590 (2012).

[65] F. Chollet, Keras: Theano-based deep learning library, Code: https://github.com/fchollet. Documentation: http://keras. io (2015).

[66] B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network, arXiv preprint arXiv:1505.00853 (2015).

[67] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting., Journal of machine learning research 15 (2014) 1929–1958.

[68] J. Luengo, S. García, F. Herrera, A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests, Expert Systems with Applications 36 (2009) 7798–7808.

Appendix A. Appendix

Appendix A.1. Feature Set C

Table A.23 lists features in Feature Set C.

Table A.23: List of 136 features in Feature Set C.

Feature name	Table name
Albumin 5%	inputevents
Fresh Frozen Plasma	inputevents
Lorazepam (Ativan)	inputevents
Calcium Gluconate	inputevents
Midazolam (Versed)	inputevents
Phenylephrine	inputevents
Furosemide (Lasix)	inputevents
Hydralazine	inputevents
Norepinephrine	inputevents
Magnesium Sulfate	inputevents
Nitroglycerin	inputevents
Insulin - Regular	inputevents
Morphine Sulfate	inputevents
Potassium Chloride	inputevents
Packed Red Blood Cells	inputevents
Gastric Meds	inputevents
D5 1/2NS	inputevents
LR	inputevents
Solution	inputevents
Sterile Water	inputevents
Piggyback	inputevents
OR Crystalloid Intake	inputevents
PO Intake	inputevents
GT Flush	inputevents
KCL (Bolus)	inputevents
Magnesium Sulfate (Bolus)	inputevents
epinephrine	inputevents
vasopressin	inputevents
dopamine	inputevents
midazolam	inputevents
fentanyl	inputevents
propofol	inputevents
Gastric Tube	outputevents
Stool Out Stool	outputevents
Urine Out Incontinent	outputevents
Ultrafiltrate	outputevents
Fecal Bag	outputevents
Chest Tube #1	outputevents
Chest Tube #2	outputevents
Jackson Pratt #1	outputevents
OR EBL	outputevents

Feature name	Table name
Pre-Admission	outputevents
TF Residual	outputevents
urinary_output_sum	outputevents
HEMATOCRIT	labevents
PLATELET COUNT	labevents
HEMOGLOBIN	labevents
MCHC	labevents
MCH	labevents
MCV	labevents
RED BLOOD CELLS	labevents
RDW	labevents
CHLORIDE	labevents
ANION GAP	labevents
CREATININE	labevents
GLUCOSE	labevents
MAGNESIUM, TOTAL	labevents
CALCIUM	labevents
PHOSPHATE	labevents
INR(PT)	labevents
PT	labevents
PTT	labevents
LYMPHOCYTES	labevents
MONOCYTES	labevents
NEUTROPHILS	labevents
BASOPHILS	labevents
EOSINOPHILS	labevents
PH	labevents
BASE EXCESS	labevents
CALCULATED TOTAL CO2	labevents
PCO2	labevents
SPECIFIC GRAVITY	labevents
LACTATE	labevents
ALANINE AMINOTRANSFERASE (ALT)	labevents
ASPARATE AMINOTRANSFERASE (AST)	labevents
ALKALINE PHOSPHATASE	labevents
ALBUMIN	labevents
pao2	labevents
serum_urea_nitrogen_level	labevents
white_blood_cells_count_mean	labevents
serum_bicarbonate_level_mean	labevents
sodium_level_mean	labevents
potassium_level_mean	labevents
bilirubin_level	labevents
hgb	labevents
chloride	labevents
peep	labevents
Aspirin	prescriptions

Feature name	Table name
Bisacodyl	prescriptions
Docusate Sodium	prescriptions
Humulin-R Insulin	prescriptions
Metoprolol Tartrate	prescriptions
Pantoprazole	prescriptions
ArterialBloodPressurediastolic	chartevents
ArterialBloodPressuremean	chartevents
RespiratoryRate	chartevents
AlarmsOn	chartevents
MinuteVolumeAlarm-Low	chartevents
Peakinsp.Pressure	chartevents
PEEPset	chartevents
MinuteVolume	chartevents
TidalVolume(observed)	chartevents
MinuteVolumeAlarm-High	chartevents
MeanAirwayPressure	chartevents
CentralVenousPressure	chartevents
RespiratoryRate(Set)	chartevents
PulmonaryArteryPressuremean	chartevents
O2Flow	chartevents
Glucosefingerstick	chartevents
HeartRateAlarm-Low	chartevents
PulmonaryArteryPressuresystolic	chartevents
TidalVolume(set)	chartevents
PulmonaryArteryPressurediastolic	chartevents
SpO2DesatLimit	chartevents
RespAlarm-High	chartevents
SkinCare	chartevents
gcsverbal	chartevents
gcsmotor	chartevents
gcseyes	chartevents
systolic_blood_pressure_abp_mean	chartevents
heart_rate	chartevents
body_temperature	chartevents
fio2	chartevents
ie_ratio_mean	chartevents
diastolic_blood_pressure_mean	chartevents
arterial_pressure_mean	chartevents
spo2_peripheral	chartevents
glucose	chartevents
weight	chartevents
height	chartevents

Appendix A.2. Mortality Prediction Task Labels

The labels of in-hospital mortality are derived from table *ADMISSION*, in which the column *DEATHTIME* records either a valid death time of an admission

if the patient dies in hospital or a null value if the patient dies after discharge. Therefore, we assign the in-hospital mortality label of an admission to 1 if its *DEATHTIME* is not null, else we assign the label to 0.

The labels of short-term mortality are generated with values in column *INTIME* from table *ICUSTAY*, which are in-time records of icu stays, and values in column *DOD* from table *PATIENTS*, which are records of death time of patients. We calculate the length of time interval between *INTIME* and *DOD* of an admission and assign its labels by comparing it with pre-defined lengths.

The labels of long-term mortality are generated with values in column *DISCHTIME* from table *ADMISSION*, which are in-time records of icu stays, and values in column *DOD* from table *PATIENTS*, which are records of death time of patients. We calculate the length of time interval between *INTIME* and *DOD* of an admission and assign its labels by comparing it with pre-defined lengths.

Appendix A.3. Missing rates of feature sets

Tables A.24, A.25, and A.26 list the missing rates of features sets A, B, and C respectively.

Table A.24: Missing rates of Feature Set A for first 24 hours and 48 hours data.

Feature	24 hrs	48 hrs
glasgow_coma_scale	0.676	0.712
systolic_blood_pressure	0.196	0.245
heart_rate	0.177	0.225
body_temperature	0.662	0.697
pao2_fio2_ratio	0.930	0.944
urinary_output	0.413	0.430
serum_urea_nitrogen_level	0.899	0.920
white_blood_cells_count	0.902	0.924
serum_bicarbonate_level	0.903	0.923
sodium_level	0.868	0.901
potassium_level	0.829	0.871
bilirubin_level	0.973	0.980

Table A.25: Missing rates of Feature Set B for first 24 hours and 48 hours data.

Feature	24 hrs	48 hrs
gcsverbal	0.814	0.744
gcsmotor	0.814	0.744
gcseyes	0.813	0.743
systolic_blood_pressure_abp_mean	0.654	0.504
heart_rate	0.648	0.495
body_temperature	0.818	0.744
pao2	0.928	0.932
fio2	0.909	0.886
urinary_output	0.744	0.601
serum_urea_nitrogen_level	0.927	0.932
white_blood_cells_count_mean	0.930	0.936
serum_bicarbonate_level_mean	0.931	0.935
sodium_level_mean	0.907	0.917
potassium_level_mean	0.887	0.897
bilirubin_level	0.975	0.980

Table A.26: Missing rates of Feature Set C for first 24 hours and 48 hours data.

Feature	24 hrs	48 hrs
Gastric Gastric Tube	0.996	0.995
Stool Out Stool	0.998	0.997
Urine Out Incontinent	0.995	0.995
Ultrafiltrate Ultrafiltrate	1.000	0.999
Fecal Bag	1.000	0.999
Chest Tube 1	0.961	0.942
Chest Tube 2	0.997	0.995
Jackson Pratt 1	0.996	0.993
OR EBL	0.996	0.996
Pre-Admission	0.991	0.994
TF Residual	1.000	0.999
Albumin 5%	0.999	0.998
Fresh Frozen Plasma	0.995	0.996
Lorazepam (Ativan)	0.996	0.995
Calcium Gluconate	0.997	0.996
Midazolam (Versed)	0.995	0.995
Phenylephrine	0.979	0.971
Furosemide (Lasix)	0.998	0.996
Hydralazine	0.999	0.999
Norepinephrine	0.986	0.980
Magnesium Sulfate	0.989	0.990
Nitroglycerin	0.994	0.994
Insulin - Regular	0.989	0.984
Morphine Sulfate	0.993	0.991
Potassium Chloride	0.994	0.995
Packed Red Blood Cells	0.990	0.989
Gastric Meds	0.994	0.989
D5 1/2NS	0.998	0.998
LR	0.990	0.990
Solution	0.981	0.979
Sterile Water	0.998	0.997

Feature	24 hrs	48 hrs
Piggyback	0.997	0.995
OR Crystalloid Intake	0.992	0.994
PO Intake	0.969	0.941
GT Flush	0.998	0.996
KCL (Bolus)	0.997	0.996
Magnesium Sulfate (Bolus)	0.996	0.995
HEMATOCRIT	0.914	0.919
PLATELET COUNT	0.925	0.933
HEMOGLOBIN	0.929	0.937
MCHC	0.930	0.937
MCH	0.930	0.937
MCV	0.930	0.937
RED BLOOD CELLS	0.930	0.937
RDW	0.930	0.938
CHLORIDE	0.930	0.935
ANION GAP	0.933	0.938
CREATININE	0.927	0.933
GLUCOSE	0.934	0.938
MAGNESIUM	0.950	0.947
CALCIUM, TOTAL	0.956	0.954
PHOSPHATE	0.956	0.954
INR(PT)	0.938	0.948
PT	0.945	0.953
PTT	0.938	0.947
LYMPHOCYTES	0.968	0.981
MONOCYTES	0.968	0.981
NEUTROPHILS	0.968	0.981
BASOPHILS	0.968	0.981
EOSINOPHILS	0.968	0.981
PH	0.974	0.983
BASE EXCESS	0.964	0.966
CALCULATED TOTAL CO2	0.928	0.935

Feature	24 hrs	48 hrs
PCO2	0.928	0.935
SPECIFIC GRAVITY	0.975	0.984
LACTATE	0.954	0.965
ALANINE AMINOTRANSFERASE (ALT)	0.974	0.980
ASPARATE AMINOTRANSFERASE (AST)	0.974	0.980
ALKALINE PHOSPHATASE	0.975	0.981
ALBUMIN	0.981	0.986
Aspirin	0.994	0.996
Bisacodyl	0.993	0.994
Docusate Sodium	0.981	0.986
Humulin-R Insulin	0.994	0.995
Metoprolol Tartrate	0.991	0.991
Pantoprazole	0.991	0.993
ArterialBloodPressurediastolic	0.865	0.789
ArterialBloodPressuremean	0.866	0.790
RespiratoryRate	0.651	0.505
AlarmsOn	0.937	0.907
MinuteVolumeAlarm-Low	0.976	0.971
Peakinsp.Pressure	0.946	0.938
PEEPset	0.936	0.923
MinuteVolume	0.975	0.969
TidalVolume(observed)	0.945	0.936
MinuteVolumeAlarm-High	0.976	0.970
MeanAirwayPressure	0.941	0.929
CentralVenousPressure	0.977	0.964
RespiratoryRate(Set)	0.947	0.942
PulmonaryArteryPressuremean	0.977	0.965
O2Flow	0.924	0.890
Glucosefingerstick	0.947	0.914
HeartRateAlarm-Low	0.855	0.798
PulmonaryArteryPressuresystolic	0.957	0.939
TidalVolume(set)	0.950	0.945

Feature	24 hrs	48 hrs
PulmonaryArteryPressurediastolic	0.957	0.939
SpO2DesatLimit	0.968	0.964
RespAlarm-High	0.858	0.803
SkinCare	0.955	0.928
gcsverbal	0.814	0.745
gcsmotor	0.814	0.745
gcseyes	0.813	0.744
systolic_blood_pressure_abp_mean	0.654	0.508
heart_rate	0.648	0.499
body_temperature	0.818	0.747
pao2	0.928	0.935
fio2	0.909	0.891
urinary_output_sum	0.744	0.608
serum_urea_nitrogen_level	0.927	0.933
white_blood_cells_count_mean	0.930	0.937
serum_bicarbonate_level_mean	0.931	0.936
sodium_level_mean	0.907	0.919
potassium_level_mean	0.887	0.899
bilirubin_level	0.975	0.980
ie_ratio_mean	0.963	0.959
diastolic_blood_pressure_mean	0.654	0.508
arterial_pressure_mean	0.654	0.508
respiratory_rate	0.651	0.505
spo2_peripheral	0.656	0.513
glucose	0.883	0.843
weight	0.976	0.983
height	0.996	0.997
hgb	0.905	0.920
platelet	0.925	0.933
chloride	0.913	0.924
creatinine	0.927	0.933
norepinephrine	0.986	0.980

Feature	24 hrs	48 hrs
epinephrine	0.997	0.996
phenylephrine	0.979	0.971
vasopressin	0.998	0.997
dopamine	0.993	0.992
midazolam	0.990	0.983
fentanyl	0.984	0.973
propofol	0.952	0.941
peep	0.991	0.991
ph	0.924	0.931

Appendix A.4. MMDL model

Figure A.7 illustrates the structure of our proposed MMDL model for Feature Set B as the input.

Appendix A.5. Distribution of the duplicates present in the database

Figure A.8 shows the distribution of duplicates of all features present in the database. Most features appear as duplicates fewer than 50 times. All these duplicates where identified based on their time stamp and using simple natural language processing techniques, and they were removed if the entries were exactly similar, otherwise the latest entry is kept if there is a note in the database stating that the entry was updated. Otherwise, the duplicates were averaged or summed up depending on the feature.

Appendix A.6. Inconsistency of feature sets and unit conversion

For the features with inconsistency in units, we performed the unit conversion using a set of rules as shown in Table A.27. First, we searched the feature name at this URL `https://www.drugs.com/dosage/` to obtain the unit conversion rules as used in the medical community. For some of the features we were able to find the unit conversions at this URL. For *Fentanyl, TPA*, and *Ketamine* features, we obtained the range of values for each unit from the URL, and chose the unit conversion rule based on the statistics of the feature values, i.e. we compared the range of values for each unit and manually set proper conversion rules as listed in the Table. For some features such as *C-Reactive Protein*, and *DHEA-Sulfate*, we employed simple conversion rules since units of these features are comprised of standard units and can be converted based on standard rules. For example, 1 mIU/mL = 1000 mIU/L since 1000 mL = 1 Liter. For *Calcium Chloride* feature, we removed ml and only kept the mg unit.

Table A.27: Inconsistency of feature sets and unit conversion rules. The column - *Different units (% of major unit)* lists all the units for that feature along with the % of the major unit for that feature present in the database. For features with superscript *a*, the corresponding dosage was obtained from `https://www.drugs.com/dosage/`. For features with superscript *b*, a simple rules of unit conversion is employed. For features with *c*, only the majority unit values were kept. For features with superscript *d*, we compared the range of values in each unit and manually set proper conversion rules.

Feature name	Different units (% of major unit)	Conversion rule
Bivalirudin[a]	ml(52.13%), mg	1ml == 1mg
Xigris[a]	ml(66.47%), mg	5mg == 2.5ml
Azithromycin[a]	dose(66.47%), mg	250mg == 1dose
Isuprel[a]	ml(72.65%), mg	0.2mg == 1ml
Solumedrol[a]	mg(73.28%), ml	40mg == 1ml
Pantoprazole(Protonix)[a]	dose(75.12%), mg	1dose == 40mg
Fentanyl[d]	mcg(79.95%), mg	1000mcg == 1mg
Promod[a]	tsp(82.37%), gm	1tsp == 6.6g
Ketamine[d]	mg(84.20%), ml	1mg == 1ml
TPA[d]	mg(84.43%), ml	1mg == 1ml
Sodium Acetate[a]	mEQ/mEq(86.73%), U, ml	2mEq/mEQ == 1ml
		1U == 1mEq/mEQ
Factor Vlla[a]	mcg(87.16%), dose, pg	1dose == 5000mcg
		1pg == 1mcg
Coumadin(Warfarin)[a]	dose(88.26%), mg	1dose == 4mg
Calcium Chloride[c]	mg(89.86%), ml	Remove ml and only keep mg
Theophylline[b]	ug/ml(54.45%), ug/mL	Same unit, no conversion
Urobilinogen[b]	mg/dL(64.92%), EU/dL	1EU/dL == 1mg/dL
Uric Acid[a]	mg/dL(68.65%), MG/DL	Same unit, no conversion
Follicle Stimulating Hormone[b]	mIU/mL(75.86%), mIU/L	1mIU/mL == 1000mIU/L
Luteinizing Hormone[b]	mIU/mL(76.32%), mIU/L	1mIU/mL == 1000mIU/L
DHEA-Sulfate[b]	ug/dL(81.82%), nG/mL	1nG/mL == 0.1ug/dL
C-Reactive Protein[b]	mg/L(83.34%), mg/dL, MG/DL	1mg/dL == 10mg/L
Triglycerides[b]	mg/dL(85.67%), MG/DL	Same unit, no conversion
Troponin T[b]	ng/mL(86.29%), ng/ml	Same unit, no conversion
Testosterone[b]	pg/mL(86.96%), ng/dL	1pg/mL == 0.1ng/dL
RBC, Ascites[b]	#/uL(88.43%), #/CU MM	1uL == 1CU MM
Blasts[b]	#/uL(88.72%), #/CU MM	1uL == 1CU MM
Thyroxine (T4), Free[b]	ng/dL(88.89%), ng/dl	Same unit, no conversion
Thyroid Stimulating Hormone[b]	uIU/mL(89.25%), uU/ML	Typo, no conversion

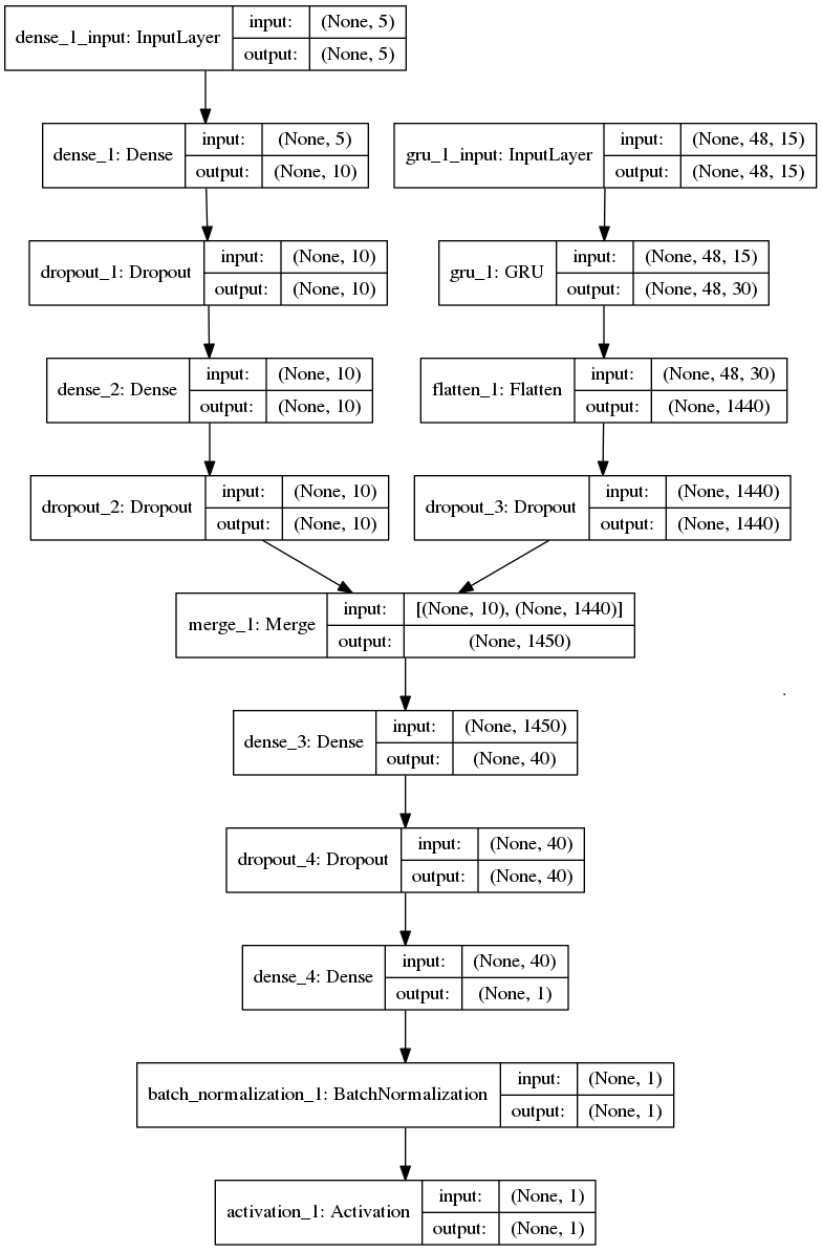


Figure A.7: Structure of the MMDL model with Feature Set B as input. Non-temporal features are fed to the feed forward network starting from "dense_1_input", and temporal features are fed to the RNN network starting from "gru_1_input" in the form of 2-D matrices (number of time steps X number of features). The word "None" in the figure refers to the batch size when training the deep learning models, which is 100 in our experiments. Numbers except "None" in one tuple describe the shape of the input/output tensor.

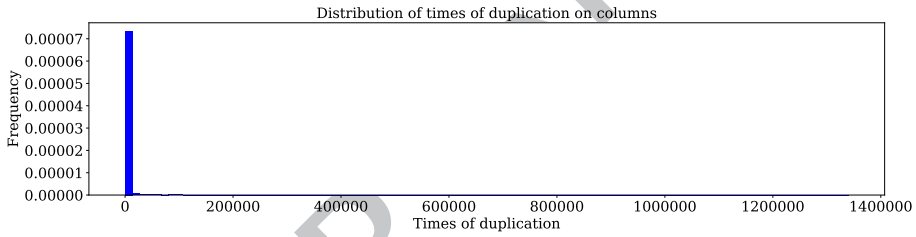
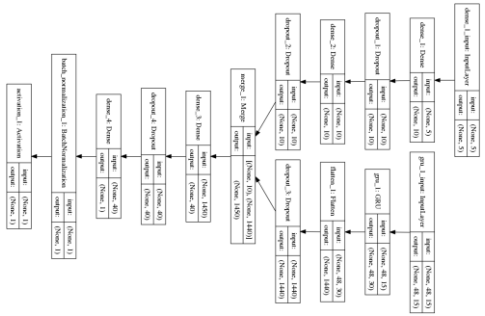
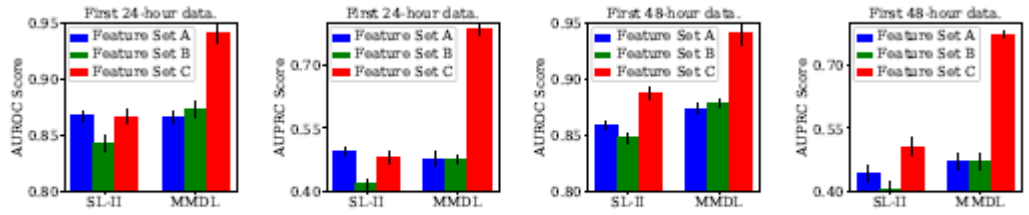


Figure A.8: Distribution of duplicates of all features. x-axis denotes the number of times duplicates appear in the database and y-axis represents the frequency of duplication of features.

Graphical Abstract



Deep Learning model



In-hospital mortality task on MIMIC-III data.

Benchmarking Comparison Results

Highlights

- Exhaustive benchmarking evaluation of deep learning models on MIMIC-III dataset
- Mortality, Length of stay, and ICD-9 code prediction tasks are used for evaluation
- Deep learning models achieve the best performance compared to all existing models
- Deep learning models perform well with raw clinical time series features