UNIVERSITY OF CALIFORNIA
Los Angeles

Predicting In-Hospital Mortality
from Intensive Care Admissions Records
with Recurrent Neural Networks

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics

by

Asa Wilks

2021

ABSTRACT OF THE THESIS

Predicting In-Hospital Mortality
from Intensive Care Admissions Records
with Recurrent Neural Networks

by

Asa Wilks
Master of Applied Statistics
University of California, Los Angeles, 2021
Professor Guido F. Montufar, Chair

This study explores the implications of different modeling choices when predicting mortality during intensive care visits using recurrent neural networks. Using the MIMIC-III database, models were trained and tested with varying memory cells, architectures, and other hyperparameters. Performance gains from incorporating information from unstructured clinical notes was tested as well. The study finds that a range of relatively shallow networks with varying memory cells and architectures can perform well and produce similar results, all of which outperform traditional mortality risk scores such as SAPS II. Adding information from clinical notes boosts model performance even with a simple natural language processing algorithm. Although methodological differences make direct comparisons complicated, the most accurate model presented here achieves an AUROC score of 0.943 which represents a slight improvement over similar prior work.

The thesis of Asa Wilks is approved.

Yingnian Wu

Frederic R. Paik Schoenberg

Guido F. Montufar, Committee Chair

University of California, Los Angeles

2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Machine learning applications in healthcare have increased dramatically in recent years with varying degrees of success and adoption. One promising area of study is prediction of clinical events. Not only does a great deal of data exist – in the form of Electronic Health Records (EHR) and clinical images – but clinicians and researchers were making use of risk scores derived from the predictions of more traditional statistical models already, before the rise of modern machine learning algorithms such as deep learning. Recent work has demonstrated the ability of deep learning algorithms to improve upon the accuracy of traditional risk scores, and this study seeks to explore the degree to which different network architectures, memory layers, and model inputs affect that accuracy using the publicly available MIMIC-III database of EHR records from the Intensive Care Unit (ICU) at a major American hospital.

There are four broad objectives of this study. The first goal is to replicate a specific benchmarking study to allow for direct comparisons between models derived in that work, comparing the accuracy of both traditional risk scores and deep learning approaches. A second objective is to explore the impact of choices between memory cells, network architecture, and hyperparameters on the accuracy of ICU mortality prediction tasks. The third objective is to test accuracy improvements from the inclusion of features derived from unstructured clinical notes. A fourth and final objective is to explore the extent to which the choice of evaluation metric impacts the selection of preferred models.

This study produces models of comparable accuracy to prior published work using similar data, with the most accurate model presented here achieving an AUROC score of 0.943. It provides evidence that the additional retention provided by the third gate in Long Short-Term Memory (LSTM) layers yield modest benefits over the simpler and less computationally expensive Gated Recurrent Unit (GRU) cells with two gates. The study provides no evidence that including information from future time steps with bidirectional recurrent layers increases performance. Unlike most prior work, some of these models include both the structured medical data from the EHR record as well as the unstructured clinical notes consisting of free text recorded by clinicians. Although overly simplistic, models show that using Term Frequency-Inverse Document Frequency (TF-IDF) scores to incorporate this free text results in accuracy improvements. While this study provides evidence of modeling choices that result in more accurate mortality predictions in the ICU, it also shows that most of the accuracy improvements of deep learning approaches over traditional risk scores can be achieved with shallow and relatively simple Recurrent Neural Networks (RNNs). Further it is shown that the performance of these networks is relatively robust to a range of different network architectures. Finally, comparing the model performance over a range of evaluation metrics underscores the importance of making modeling goals explict before selecting the preferred model.

CHAPTER 2

Prior Work

Possibly the most widely used model for ICU mortality prediction based on traditional linear and generalized linear models is an updated Simplified Acute Physiology Score (SAPS II) developed by Le Gall, Lemeshow, and Saulnier [1]. The SAPS II score is based on regression methods using 17 features of the ICU admission and the patient over the first 24 hours of the admission. An improved version of SAPS II was developed by Pirracchio [2] using a modified logistic regression approach with the same predictors. These scores form the most commonly used baselines for judging machine learning prediction of mortality in the ICU, but they are not the highest performing overall. The most accurate system based on traditional statistical approaches alone appears to be the Acute Laboratory Risk of Mortality Score (ALaRMS) [3], which combines patient characteristics with results from clinical and comorbidity software (highest AUROC 0.90-0.91).

Particularly after the release of the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) database, there have been numerous studies focused on predicting ICU mortality with various machine learning algorithms. For example Lee [4] uses the MIMIC database to predict ICU mortality with random forests, while Li-wei et al. [5] explore predictions based specifically on cardiovascular variables. In early 2020, Yu and colleagues [6] published work using Recurrent Neural Networks to predict mortality using a dynamic framework (highest AUROC 0.885) which produces predictions at each time point instead of once at the end of a 24-hour period. When direct comparions have been published all of these approaches have produced better accuracy than SAPS II, but directly comparing deep learning approaches in the literature is complicated by authors making a variety of different decisions – for example analyzing only subsets of the data or patients with specific conditions. Further, published work is not always accompanied by computer code that can be used to replicate the results. Purushotham and coauthors [8] work to solve this problem by benchmarking a variety of methods for multiple different prediction problems and making their code available on GitHub (highest AUROC= 0.9410). In an effort to add to their work, the present study adopts nearly all of Purushotham et al.'s data processing decisions in order to draw direct comparisons, but tests different modeling arrangements.

Perhaps the most successful and commercially viable work predicting in-hospital mortality, as well as other clinical events, was led by researchers at Google AI [9]. Unlike most other work, they use all inpatient hospital records as opposed to the smaller subset of data from the ICU that exists within MIMIC. The researchers also include all unstructured clinical notes. Their predictions are generated by an ensemble of three algorithms, including a Long Short-Term Memory Recurrent Neural Network (LSTM RNN), examining all prior patient records rather than limiting to the single admission. While some modeling details are available, much work was done on proprietary systems at Google from which no code is available. The choice of Natural Language Processing (NLP) algorithm is also unclear. Unlike much prior work, the Google researchers do not use the MIMIC-III data and instead use data from UCSF and University of Chicago medical centers, reporting AUROC accuracy on in-hospital mortality prediction of 0.93 and 0.95 for the two hospital systems separately. Of note, the prediction task the Google researchers faced was more challenging than most other prior work because their population included non-ICU hospital visits, which typically produce less data for models to consider.

One of the objectives of this study is to replicate the Purushotham et al. study, allowing direct comparisons between models. The relevant results from that work are presented in Table 2.1.

The SAPS II and New SAPS II represent the baseline traditional ICU mortality risk measures. Included as a deep learning based alternative to SAPS II, the feed forward SAPS II is a simple feed forward neural network only using the SAPS II inputs as features. The Multimodal Deep Learning model, the best performing Purushotham et al. model, consists of applying two dense layers to the time invariant features and one Gated Recurrent Unit

Table 2.1: Purushotham et al. (2018) Mortality Prediction Accuracy Results using traditional models SAPS II, New SAPS II and neural network models Feedforward SAPS II and Multimodal Deep Learning

Model	AUROC
SAPS II	0.8035
New SAPS II	0.8235
Feed Forward Network SAPS II	0.8496
Multimodal Deep Learning	0.9410

(GRU) layer for the time varying features before merging and applying two more dense layers. As shown in Table 2.1, the shallow feed forward network outperforms the traditional scores when given the same information as features. The Multimodal Deep Learning model uses all the structured clinical information (but not the unstructured notes) as inputs, and significantly outperforms the traditional scores.

CHAPTER 3

Data

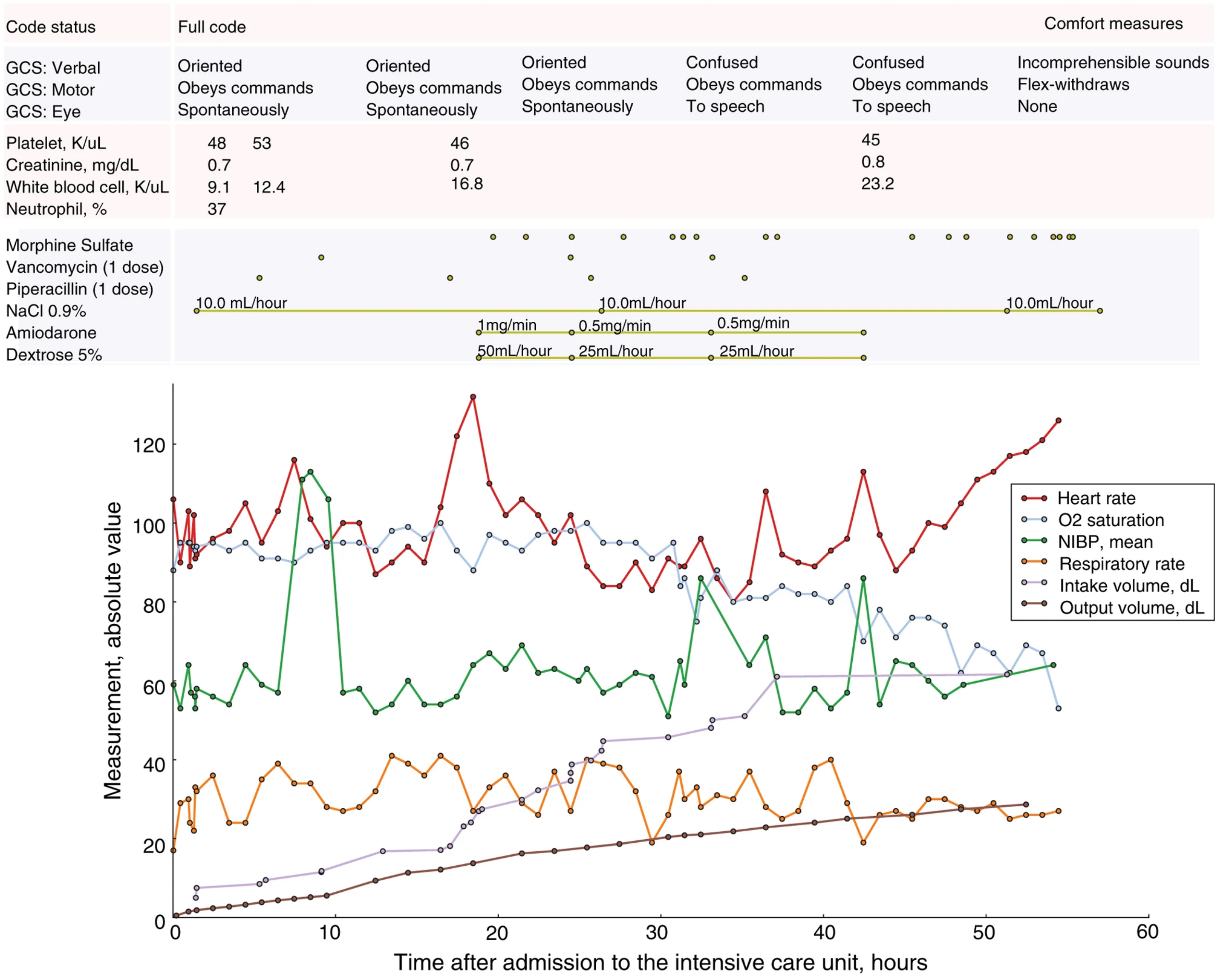
3.1 MIMIC-III

The Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-III) database [10] was developed by the MIT Lab for Computational Physiology and contains data on 61,532 ICU stays from Beth Israel Deaconess Medical Center in Boston from June 2001 through October 2012. The 26-table relational database includes all the standard components of EHR records including demographics of the patient, diagnoses, procedures, prescription drugs, microbiological events, lab results, chart events, and free-text clinical notes. The structured data included in this analysis is taken from the input events, output events, lab events, chart events, and prescription drug tables. The unstructured free-text doctor's notes data comes from the note events table. All time-varying data elements include timestamps in seconds. Data is deidentified but MIT requires several data security trainings to obtain a Data Use Agreement and download the files. Table 3.1 displays a summary of the tables in MIMIC-III along with a brief description. Using MIMIC-III requiers constructing a relational database from these tables. Tables in the database are linked with a variety of identifiers, including identifiers for patients, ICU stays, admissions, and caregivers. Other tables include a temporal component to the linking stategy. Figure 3.1 shows a summmary of a selection of clinical variables for a hypothetical patient. Data points within an admission are timestamped and various time series can be constructed as in this example.

Table 3.1: MIMIC-III Table Names and Descriptions, from MIMIC-III Summary [10]

Table	Description
ADMISSIONS	Every unique hospitalization for each patient in the database (defines HADM_ID).
CALLOUT	Information regarding when a patient was cleared for ICU discharge and when the patient was actually discharged.
CAREGIVERS	Every caregiver who has recorded data in the database (defines CGID).
CHARTEVENTS	All charted observations for patients.
CPTEVENTS	Procedures recorded as Current Procedural Terminology (CPT) codes.
CPT	High level dictionary of Current Procedural Terminology (CPT) codes.
ICD DIAGNOSES	Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD-9) codes relating to diagnoses.
ICD PROCEDURES	Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD-9) codes relating to procedures.
ITEMS	Dictionary of local codes ('ITEMIDs') appearing in the MIMIC database, except those that relate to laboratory tests.
LABITEMS	Dictionary of local codes ('ITEMIDs') appearing in the MIMIC database that relate to laboratory tests.
DATETIMEEVENTS	All recorded observations which are dates, for example time of dialysis or insertion of lines.
DIAGNOSES ICD	Hospital assigned diagnoses, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system.
DRGCODES	Diagnosis Related Groups (DRG), which are used by the hospital for billing purposes.
ICUSTAYS	Every unique ICU stay in the database (defines ICUSTAY_ID).
INPUTEVENTS CV	Intake for patients monitored using the Philips CareVue system while in the ICU, e.g., intravenous medications, enteral feeding, etc.
INPUTEVENTS MV	Intake for patients monitored using the iMDSoft MetaVision system while in the ICU, e.g., intravenous medications, enteral feeding, etc.
OUTPUTEVENTS	Output information for patients while in the ICU.
LABEVENTS	Laboratory measurements for patients both within the hospital and in outpatient clinics.
MICROBIOLOGYEVENTS	Microbiology culture results and antibiotic sensitivities from the hospital database.
NOTEEVENTS	Deidentified notes, including nursing and physician notes, ECG reports, radiology reports, and discharge summaries.
PATIENTS	Every unique patient in the database (defines SUBJECT_ID).
PRESCRIPTIONS	Medications ordered for a given patient.
PROCEDUREEVENTS MV	Patient procedures for the subset of patients who were monitored in the ICU using the iMDSoft MetaVision system.
PROCEDURES ICD	Patient procedures, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system.
SERVICES	The clinical service under which a patient is registered.
TRANSFERS	Patient movement from bed to bed within the hospital, including ICU admission and discharge.

Figure 3.1: Sample Patient Stay, from MIMIC-III Summary [10]



3.2 Structured Clinical Data Processing

This research nearly replicates the initial data preprocessing from Purushotham et al., which is discussed in detail in their paper, and major points are presented here. Like Purushotham et al., this study excludes patients under age 16 at the time of admission. In cases where the patient was admitted multiple times between 2001 and 2012, only one admission is included. Unlike the Purushotham et al. work, which keeps only the first record for patients with multiple admissions, this study selects an admission at random. Keeping only the first admission in an analysis of mortality has the effect of classifying all patients with multiple admissions as survivors, which is a departure from any real-world data these models might encounter. This decision results in in a slightly higher death rate – 11.15 percent compared to 10.49 percent in Purushotham et al. Table 3.2 summarizes the outcome variable, in-hospital deaths, and shows that there were 3,974 deaths among 35,644 admissions. From the data available within MIMIC-III it is possible to track survival after the patient is released from the hospital in order to construct measures of 30-day or 60-day mortality, but those measures are beyond the scope of this study.

Table 3.2: MIMIC-III Sample Summary: Admissions and Deaths

Admissions	35,644
In-hospital Deaths	3,974
In-hospital Death Rate	0.1115

As is common in medical data, MIMIC includes a nontrivial amount of data errors, missingness, and other general noise, and some basic data cleaning was undertaken. Values were multiplied by factors of ten to convert to common units - typically grams, milligrams, or milliliters – unless more than 90 percent of the records were the same unit, in which case records with other units were set to missing. For events of the same type occurring at exactly the same time, values were averaged (for continuous variables) or the first ordered value was selected (for categorical variables). Finally, some variables had values that represented a range at a given timestamp and in these cases the median was used.

Variables with repeated observations were set up in a time series of one record per hour through the first 24 hours of the ICU stay. Within each ICU admission, if multiple observations were recorded for a given variable during the hour these were either summed (for input/output events like fluids or medications) or averaged (for other variables such as heart rate, body temperature, etc). If an admission has gaps in these variables, forward and backward imputation was undertaken within the admission using hourly records before or after the gap. Variables that were completely missing during the stay were imputed with the mean within training and validation samples separately.

The SAPS II score is derived from a logistic regression model predicting mortality from 17 variables, including 12 physiological variables, presence of drug overdose, length of stay prior to being admitted to the ICU, clinical category, source of admission to the ICU, age, and sex. The values of the physiological predictors in the model are determined by the worst value for each during the first 24 hours of the ICU stay. The SAPS II model produces both a raw score and a predicted probability of death. The predicted probability of death is typically used as a risk control in broader analyses of patient health rather than a clinical tool to alert clinicians of elevated risk. Table 3.3 summarizes the hourly average of the raw numeric inputs required for computation of the SAPS II score, among the most important features for predicting mortality. The full list of 136 features derived from the structured clinical data is shown in Appendix A.

3.3 Unstructured Text Processing

Clinical notes records, not considered in the Purushotham et al. study, were processed separately. Within each admission, all notes occurring within the same hour for the first 24 hours were appended together such that each admission has 24 “bags of words” corresponding to each hour. In the bag of words conceptualization, rules of grammar, punctuation, and the sequence of words are ignored, and only the frequencies are stored [17]. Hourly note records were then stripped of stop words, punctuation, digits, and made lower case. As no other cleaning was performed on the notes, they contain all manner of abbreviations, typographical

Table 3.3: SAPS II Inputs from Structured Clinical Data

Feature	**Mean**	**Standard Deviation**
Age (days)	27,280.87	19,953.74
Glasgow Coma Scale - Verbal	2.93	1.90
Glasgow Coma Scale - Motor	5.24	1.44
Glasgow Coma Scale - Eyes	3.24	1.09
Systolic Blood Pressure	121.82	144.13
Heart Rate (beats per min)	89.17	4,677.05
Body Temperature (Celsius)	37.04	2.38
Partial Pressure of Oxygen (PaO2)	143.19	94.77
Fraction Inspired Oxygen (FiO2)	27.27	32.73
Urine Output	131.25	2,920.01
Serum Urea Nitrogen Level (mmol/l)	29.00	23.37
White Blood Cell Count (103/mm3)	11.42	9.80
Serum Bicarbonate Level (mmol/l)	25.28	4.86
Sodium Level (mmol/l)	138.51	5.40
Potassium Level (mmol/l)	4.12	0.94
Bilirubin Level (mg/dl)	3.67	7.20

errors, and other noise. Term frequency - inverse document frequency (TF-IDF) was then implemented using the scikit-learn vectorizer, requiring each word to be in a minimum of 5 records and to be present in less than 50 percent of records to receive a score. Results were saved for the top 250, 1,000, 2,500, and 4,000 words. The natural language processing is disussed in more detail in section 4.1.5 of the methods chapter.

3.4 Processed Data Structure

After the various data preparation steps outlined above, the structured data takes the form of a tensor with dimensions 35,644 (number of admissions), 24 (number of hours per admission), and 136 (number of features from the structured data). For models including the clinical notes the number of features rises, reaching 4,136 for the 4,000 word case. This includes the 136 features and Term Frequency-Inverse Document Frequency (TF-IDF) scores for the top 4,000 words. Prior to analysis this data is split into training (60 percent), validation (20 percent), and test (20 percent) sets. All results are reported on the test set, which was not used for training or validation.

CHAPTER 4

Methods

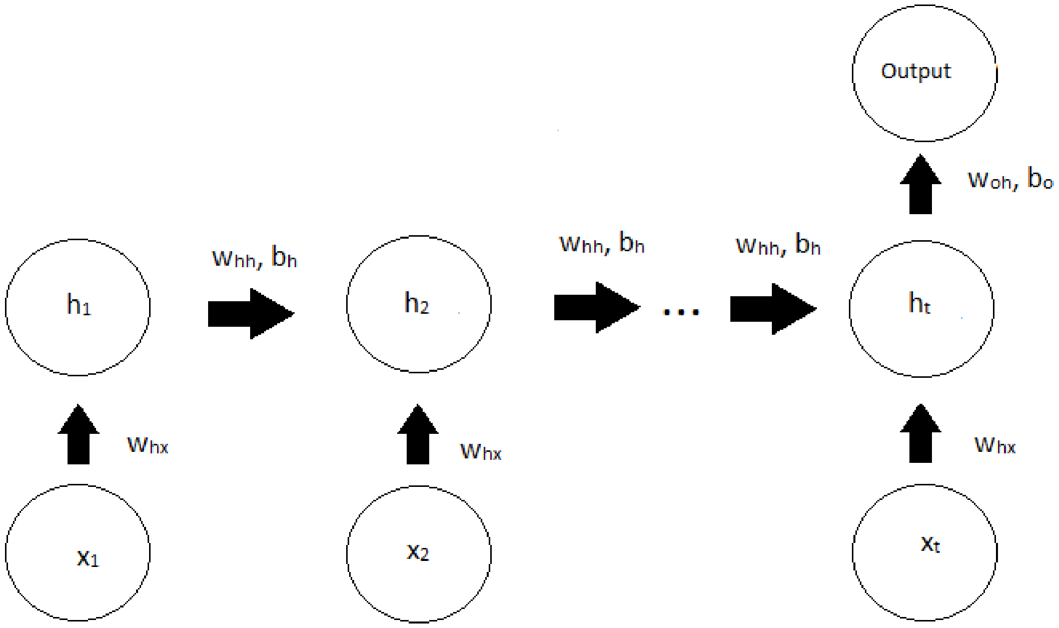
4.1 Network Architecture

4.1.1 Recurrent Neural Networks

All neural networks explored were Recurrent Neural Networks (RNNs) designed for sequence data [14]. Because many features are used to make a single classification prediction, the RNNs in this study have many-to-one network architectures. RNNs are a class of deep neural networks that make information from prior time steps in the sequence available to predict current and future time steps. The RNN accomplishes this by maintaining a hidden state that is updated after each time step. Figure 4.1 illustrates a deep many-to-one recurrent neural network such as those used in this study. In this case, the inputs x represent the set of features measured at each of 24 hourly time steps with k hidden layers. Each hidden layer is a collection of neurons that performs a nonlinear transformation on inputs from the prior layer and passes output to the subsequent layer.

The basic memory cells in simple RNNs, which control the information passed forward in the hidden state, often result in a vanishing or exploding gradient problem that makes it difficult if not impossible for the network to learn dependencies more than a few time steps away. Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) cells are popular solutions that allow for signal to be effectively carried forward as the sequence is processed. In general, GRUs are less computationally expensive to train but LSTMs can have somewhat better performance.

Figure 4.1: Many-to-One Deep Recurrent Neural Network. Inputs x and hidden state variable h produce a single prediction as an output.



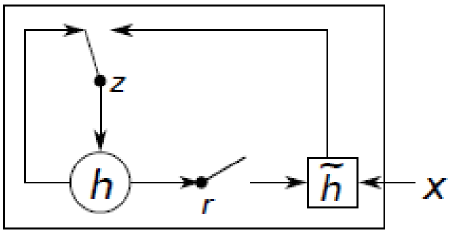
4.1.2 Gated Recurrent Units

Gated Recurrent Units, developed by Kyunghyun Cho et al. [7], include an update gate (Γ_u) and a reset gate (Γ_r) to carry information across time steps. The update gate determines the information passed to future steps, while the reset gate determines what information is discarded. The structure of the GRU memory cell is summarized in the Figure 4.2. In this single cell, the update gate z determines whether to update the hidden state h and reset gate r determines whether to ignore the prior hidden state.

4.1.3 Long Short-Term Memory

Long short-term memory cells have additional forget and output gates [15]. The forget gate (Γ_f) determines how much information from the previous hidden state is retained and the output gate (Γ_o) determines the hidden state for the next time step. A value for a given time step represents the activation of the inputs for that time step multiplied by their weights,

Figure 4.2: GRU Memory Cells, with Update Gate z, Reset Gate r, and Hidden State h, from Cho et al. [7].



plus the hidden state value from the prior time step multiplied by its own weights. The structure of the LSTM memory cell is summarized in Figure 4.3, showing two LSTM cells in a sequence.

4.1.4 Bidirectional Layers

Bidirectional RNN layers connect layers processing the time series both in order and then in reverse order [16]. The outputs from both directions are combined for each time step, allowing the network to include information from both the past and the future during the current time step. The resulting network therefore has more parameters than a similar unidirectional layer. Figure 4.4 illustrates the flow of information forwards as well as backwards through the timesteps.

4.1.5 Term Frequency-Inverse Document Frequency

The unstructured clinical notes were processed using the term-frequency inverse document frequency (TF-IDF) algorithm. For each admission, notes were combined hourly for the first 24 hours and each hourly record was considered a document. Term frequency scores are first computed for each word by dividing each word's frequency in a document by the total number

Figure 4.3: LSTM Memory Cells, from Hochreiter and Schmidhuber [15].

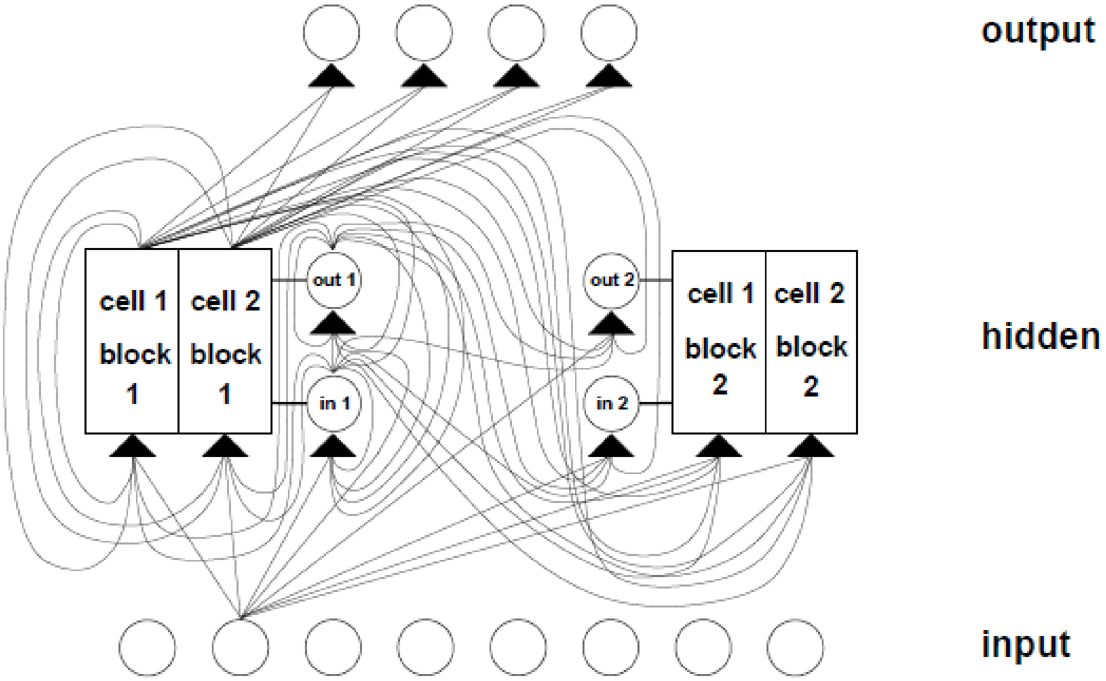
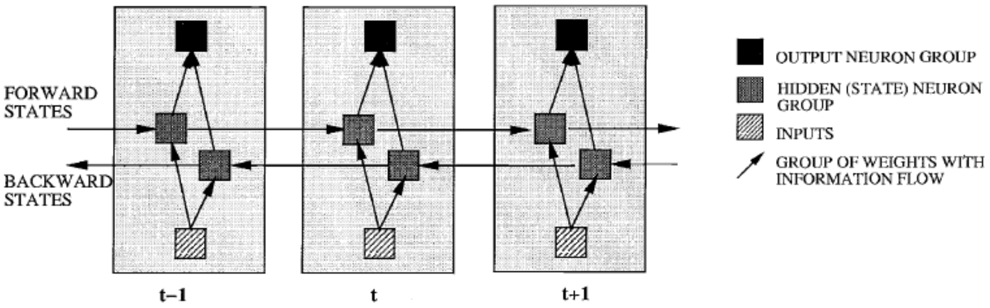


Figure 4.4: Bidirectional Recurrent Neural Network, Adapted from Shuster and Paliwal [16]


FORWARD STATES
BACKWARD STATES
t−1
t
t+1
OUTPUT NEURON GROUP
HIDDEN (STATE) NEURON GROUP
INPUTS
GROUP OF WEIGHTS WITH INFORMATION FLOW

of words in that document. The inverse document frequency is computed as the log of the total number of documents divided by the number of documents containing the word. The TF-IDF score is computed as the product of the term frequency and the inverse document frequency and provides a measure of how common a word is in a document compared to other documents. Equations 4.1 - 4.3 below capture the TF-IDF scoring algorithm.

$$TF_t = \frac{\text{Occurences of Term } t \text{ in a Document}}{\text{Number of Terms in the Document}} \tag{4.1}$$

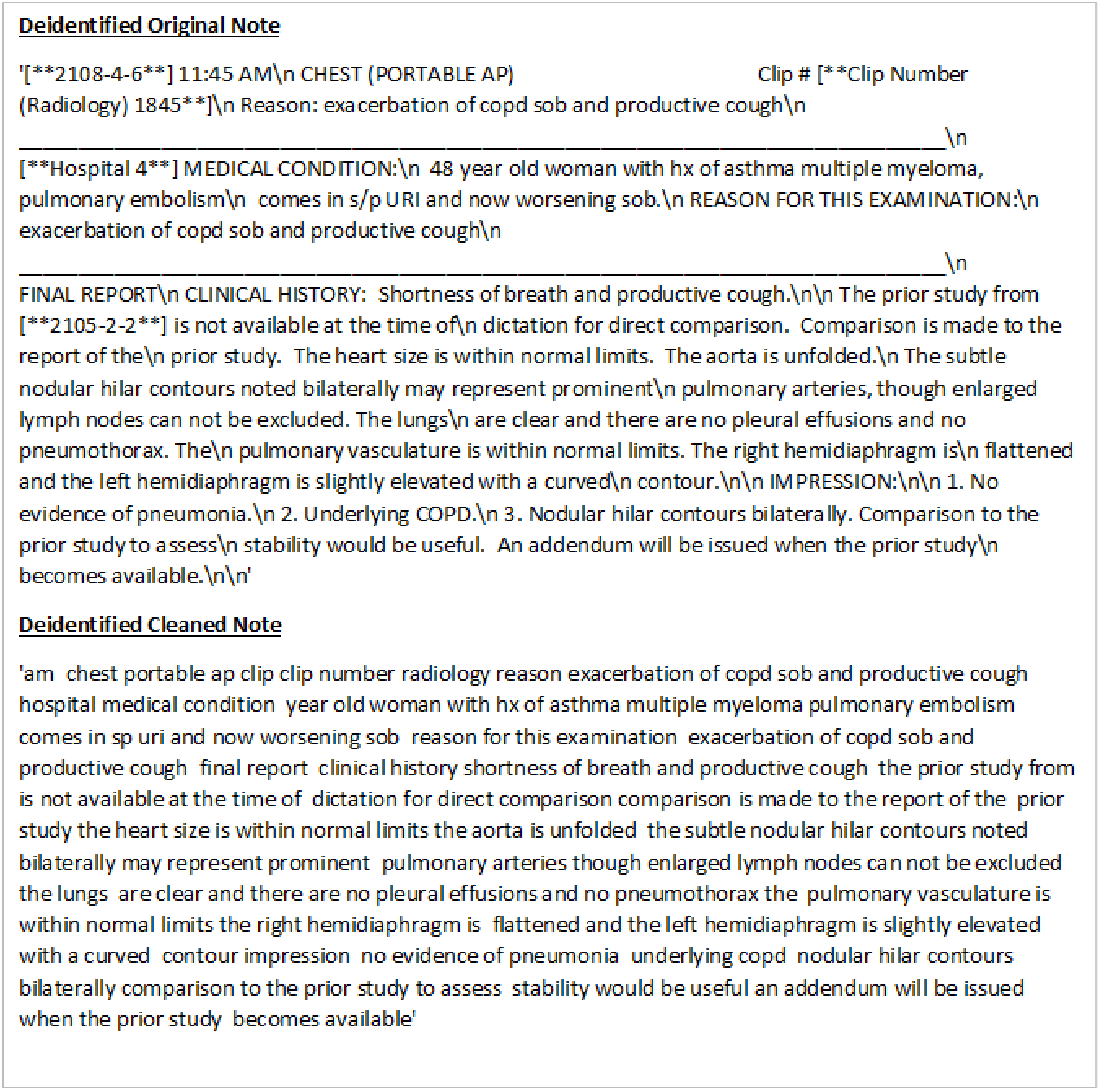
$$IDF_t = log\left(\frac{\text{Number of Documents}}{\text{Number of Documents including Term } t}\right) \tag{4.2}$$

$$TFIDF_t = TF_t(IDF_t) \tag{4.3}$$

A high TF-IDF score for a term means the term had a combination of high frequency within that hour's notes and a low prevalence across all other hours of notes and all other admissions. In order to limit to relevant features, words were not scored if they appeared less than 5 times across all notes. Words that appeared in more than half of notes were similarly excluded.

Figure 4.5 shows a prototypical example of the note processing (with segments truncated and scrambled to avoid using an actual data point as an example), which also demonstrates

Figure 4.5: Scrambled and De-Identified Example of Note Processing



Deidentified Original Note

'[**2108-4-6**] 11:45 AM\n CHEST (PORTABLE AP)   Clip # [**Clip Number (Radiology) 1845**]\n Reason: exacerbation of copd sob and productive cough\n

____\n

[**Hospital 4**] MEDICAL CONDITION:\n 48 year old woman with hx of asthma multiple myeloma, pulmonary embolism\n comes in s/p URI and now worsening sob.\n REASON FOR THIS EXAMINATION:\n exacerbation of copd sob and productive cough\n

____\n

FINAL REPORT\n CLINICAL HISTORY: Shortness of breath and productive cough.\n\n The prior study from [**2105-2-2**] is not available at the time of\n dictation for direct comparison. Comparison is made to the report of the\n prior study. The heart size is within normal limits. The aorta is unfolded.\n The subtle nodular hilar contours noted bilaterally may represent prominent\n pulmonary arteries, though enlarged lymph nodes can not be excluded. The lungs\n are clear and there are no pleural effusions and no pneumothorax. The\n pulmonary vasculature is within normal limits. The right hemidiaphragm is\n flattened and the left hemidiaphragm is slightly elevated with a curved\n contour.\n\n IMPRESSION:\n\n 1. No evidence of pneumonia.\n 2. Underlying COPD.\n 3. Nodular hilar contours bilaterally. Comparison to the prior study to assess\n stability would be useful. An addendum will be issued when the prior study\n becomes available.\n\n'

Deidentified Cleaned Note

'am chest portable ap clip clip number radiology reason exacerbation of copd sob and productive cough hospital medical condition year old woman with hx of asthma multiple myeloma pulmonary embolism comes in sp uri and now worsening sob reason for this examination exacerbation of copd sob and productive cough final report clinical history shortness of breath and productive cough the prior study from is not available at the time of dictation for direct comparison comparison is made to the report of the prior study the heart size is within normal limits the aorta is unfolded the subtle nodular hilar contours noted bilaterally may represent prominent pulmonary arteries though enlarged lymph nodes can not be excluded the lungs are clear and there are no pleural effusions and no pneumothorax the pulmonary vasculature is within normal limits the right hemidiaphragm is flattened and the left hemidiaphragm is slightly elevated with a curved contour impression no evidence of pneumonia underlying copd nodular hilar contours bilaterally comparison to the prior study to assess stability would be useful an addendum will be issued when the prior study becomes available' why TF-IDF scores are almost surely an overly simplistic tool for this problem. The algorithm is simply comparing frequency and not discerning meaning. For example, part of the cleaned note reads "the lungs are clear and there are no pleural effusions." This note would receive the same TF-IDF scores for 'pleural' and 'effusions' regardless of whether the doctor indicated their presence or absence, or whether these words were even directly adjacent to one another in the note. More sophisticated NLP processing could potentially combine algorithms to consider phrases as opposed to words. Further, more sophisticated work could employ algorithms that take advantage of an existing corpus of clinical notes and perform sentiment analyses to classify notes' tone as positive, negative, or nuetral with respect to a patient's health. These techniques were beyond the scope of this study, which simply tests the value of the TF-IDF scores in prediction as a baseline.

4.2 Computation

4.2.1 Optimization

Neural networks are trained iteratively, with parameters being adjusted to minimize prediction error in the training set. Stochastic gradient descent is by far the most commonly used algorithm used to tackle this optimization problem, and there are many extensions [14]. Two popular stochastic gradient descent optimization algorithms were tested in the course of this study. The RMSProp, or Root Mean Square Propagation, algorithm was suggested by George Hinton in a Coursera lecture [11] and was initially used in this study in keeping with the Purushotham et al. work. The RMSProp algorithm automatically adjusts the learning rate and allows rates to vary across parameters. In RMSProp each parameter is updated according to the following equations. George Hinton's suggested default is $\beta = 0.9$.

$$m_t = \beta m_{t-1} + (1 - \beta) g_t^2 \tag{4.4}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{m_t} + \epsilon} g_t \tag{4.5}$$

Some experimentation with optimizer choice made it clear that the Adam algorithm was superior over a range of different model architectures. Adam, or Adaptive Moment Optimization, also calculates adaptive learning rates for each parameter but adds a momentum component [12]. The following equations (4.6) and (4.7) represent the first and second moments of the gradient, respectively:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{4.6}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{4.7}$$

The bias corrected equations are:

$$\hat{m_t} = \frac{m_t}{1 - \beta_1^t} \tag{4.8}$$

$$\hat{v_t} = \frac{v_t}{1 - \beta_2^t} \tag{4.9}$$

And the parameter update equation is:

$$w_{t+1} = w_t - \frac{\hat{m_t}\eta}{\sqrt{\hat{v_t}} + \epsilon} \tag{4.10}$$

After some experimentation with Adam's hyperparameters β_1, β_2, and ϵ, the default values proposed by the authors of Adam (0.9, 0.999, and 10^{-8} respectively) were selected for results presented here. Through evaluation of the accuracy on the validation set of learning rate parameters 0.001, 0.0075, 0.0005, and 0.00025 a learning rate of 0.0005 was selected.

4.2.2 Regularization

Both L2 regularization, dropout, and a combination of the two were tested. L2 regularization reduces overfitting by applying a penalty for large weights in the loss function, based on the norm of the weights [14]. Weights regularized with an L2 penalty may shrink dramatically but, unlike L1 regularization, will not typically shrink to zero [18]. L2 regularization was applied to the input weights on the RNN units using the kernel regularizer in Keras within

TensorFlow [19]. Dropout regularization works by randomly dropping units from the network to avoid overfitting by too heavily relying on a small subset of the weights. Regularization for the RNNs included both dropout and recurrent dropout, where units are dropped from the recurrent state. While L2 regularization was effective in some cases, especially when combined with a low level of dropout, it proved difficult to tune. The level of L2 regularization that was effective for one network often resulted in either overfitting or underfitting when applied to another model. The combination of dropout and recurrent dropout, however, proved effective for most models with minimal tuning. Therefore, final models presented here do not include L2 regularization and instead include dropout of 0.2, meaning that one fifth of units were dropped out.

4.2.3 Performance Metrics

The primary accuracy metric used in this study is the area under the reciever operating characteristic (AUROC), also known as the AUC or c-statistic. The reciever operating characteristic curve is a plot of the sensitivity against the specificity, and the AUROC metric provides balance when comparing model performance in terms of false positives and false negatives [18]. In mortality prediction this reflects both the percent of deaths correctly classified and the percent of survivals correctly classified [13]. An AUROC score of 1.0 corresponds to a perfect classifier and a score of 0.5 corresponds to a random classifier. Model loss was used as the metric for model selection. A TensorFlow callback was written which tracked validation loss after each epoch and saved an updated model each time the validation loss improved. The AUROC is perhaps the most widely used metric for binary classification tasks, and it was selected to allow for comparisons with other work. However, other performance measures were computed as well.

The overall accuracy is reported throughout, reflecting the percent of correct predictions overall. This metric can be useful when comparing models, but it does not account for the fact that the data is unbalanced, with survival far more common than death. Recall, a measure of sensitivity, is computed as the True Positive Rate, which in this case is the percentage of ICU deaths that were correctly classified as such. Precision, a measure of specificity, can be measured by computing the Positive Predictive Value, or the number of true positives divided by the sum of true and false positives. In other words, precision represents the share of all predicted deaths that were actual deaths. In probability terms, recall in this study is the probability of predicted death conditional on an actual death, while precision is the probability of an actual death conditional on a predicted one. Finally, the F1 score is also reported. Seeking to balance precision and recall the F1 score is computed as:

$$F1 = \frac{1}{\frac{1}{2}(\frac{1}{recall} + \frac{1}{precision})} \tag{4.11}$$

4.2.4 Statistical Computing

The scale of computation required for both data preparation and analysis was too great for a standard personal machine, and this study used cloud-based services for all statistical computing. For full functionality the MIMIC-III data must be deployed as a relational database with 26 tables. In this case, the database was stood up from zipped .csv files using PostgreSQL 12 on Amazon's Relational Database Service (RDS) within Amazon Web Services (AWS). Raw data was loaded into tables, applying indexes to speed processing and running completeness checks. The RDS instance was of class db.m5.2xlarge with 8 CPUs and 32GB of RAM, allowing for parallelization during the data preparation step by ensuring the database was capable of efficiently processing multiple queries simultaneously.

Primary data preparation computation was done within an AWS Elastic Computing (EC2) instance. After configuring and assigning appropriate permission groups, the EC2 instance was able to pass queries to the RDS instance and store the results. The EC2 instance was of class c5d.12xlarge, designed for high performance computation, networking and storage, with 48 CPUs and 96GB of RAM running on Amazon Linux. All data preparation code was written in Python 3. Queries were passed to the PostgresSQL database instance using the 'psycopg2' package, and parallelization was done using the 'multiprocessing' package. Using these packages tandem, each of the 48 CPUs in the EC2 instance could act as an independent worker node sending distinct PostgreSQL queries to the RDS instance and retrieving results.

Once ready for analysis, the data was moved from Amazon's Elastic Cloud Compute (EC2) instance to an Amazon Simple Storage Service (S3) bucket and transferred to the RAND Corporation's in-house platform known as the Analytic Cloud Computing Service (ACCS). The RAND ACCS platform is a secure and compliant solution running on Amazon Web Services (AWS) cloud infrastructure, which is only accessible from within the RAND private network. Besides access restrictions, the RAND ACCS platform generally functions as AWS does, with the same products available at the same prices. Models were trained with Deep Learning Machine Images (AMIs) on Ubuntu Linux using the P2 and G3 class GPU instances, scaled up depending on network size.

4.3 Summary of Tested Models

To aid interpretation of the results presented in this study, a concise summary the neural network attributes that are fixed and those that vary is provided here. Models of varying depth and with varying types of recurrent layers are presented, but all hidden layers contain 136 nodes. This number was chosen to match the number of inputs from the structured clinical data. The number of nodes per hidden layer was kept constant and the number of overall parameters was instead varied by adding or subtracting layers in the network. Some experimentation provided no reason to believe that shallower networks with greater numbers of nodes, or the opposite, would have a meaningful effect on the results. After some tuning, the Adam optimizer was used throughout with a learning rate of 0.0005 and Beta 1 and Beta 2 were set at 0.9 and 0.999, respectively. Batch size was set at 100 in keeping with the Purushotham et al. work after some experimentation provided no support for changing it. All recurrent layers used the TensorFlow defaults of *tanh* activation and sigmoid recurrent activation functions. After experimentation with L2 regularization combined with varying levels of dropout, a fixed dropout rate of 0.2 was selected across all models presented here. A TensorFlow callback was written to preserve the model with the lowest validation loss, and all results reported here are based on the 20 percent test set which was held out completely. Finally, models excluding clinical notes features were trained for 250 epochs (passes through the entire dataset) while models including the notes were trained for 500 epochs, as it appeared they were still learning after 250. Its also worth noting that while the data preparation decisions were chosen to nearly match the Purushotham study the model attributes were not - that study used the RMSProp optimizer with a higher learning rate and a lower dropout rate, and only used GRU recurrent layers without exploring LSTM or bidirectional layers.

CHAPTER 5

Results

The results presented here are organized according the objectives of the study. The first section demonstrates replication of the Purushotham et al. work, compares results from shallow RNNs to traditional methods like SAPS II, and explores implications of modeling choices with shallow networks. The second section shows the results of experimenting with deeper networks. The third section tests gains in accuracy from including information from unstructured clinical notes. Finally, the fourth section explores the sensitivity of model selection to the use of different evaluation metrics.

5.1 Shallow Recurrent Neural Networks

Table 5.2 shows the initial set of results for this study, presenting models with one or two recurrent layers using all features except the unstructured notes. A primary objective of this study is to compare results to the Purushotham et al. work, and the feed forward version of SAPS II is designed to exactly replicate their result from Table 2.1. Running the same Feed Forward model with nothing but the SAPS II predictors yields and AUROC score of 0.8401 compared to 0.8496 in the comparison work. As mentioned previously, this study made use of the same data with nearly identical data processing protocols. The single exception is that for patients with multiple admissions, the Purushotham et al. work selected the first admission while here a single admission was selected randomly. This means there are slightly more deaths in this study, and the somewhat lower AUROC score for the replicated feed forward network suggests the randomly selected admissions were slightly harder to classify.

Another key objective of this study was to compare the accuracy of various RNN struc-

Table 5.1: Shallow Networks without Clinical Notes

Model	**Hidden Layers**	**AUROC**	**Accuracy**	**F1 Score**
FFN SAPS-II	4	0.8401	0.8964	
GRU	1	0.9318	0.9299	0.6449
GRU	2	0.9309	0.9266	0.6082
LSTM	1	0.9311	0.9289	0.6366
LSTM	2	0.9394	0.9313	0.6392
Bidirectional LSTM	1	0.9353	0.9331	0.6600
Bidirectional LSTM	2	0.9384	0.9324	0.6471

tures for mortality prediction with EHR data to more established clinical risk scores. Its clear from comparing the results of Table 2.1 to those from Table 5.1 that that AUROC performance of the shallow RNNs is much higher relative to the traditional SAPS II scores. The traditional methods in Table 2.1 show AUROC scores of 0.80 and 0.82 for SAPS II and New SAPS II, respectively, but all of the shallow RNNs acheived an AUROC of at least 0.93. Another important result is that perfoemace is quite similar across the RNN models, with AUROC between 0.93 and 0.94 on all the shallow RNNs despite different memory cells and network architectures. Despite their similarities, some notable patterns were apparent. The models generally benefitted slightly from an additional recurrent layer. Comparing the networks with two recurrent layers across memory cell types, it also appears that the additional gate in the LSTM memory cells adds some benefit when compared to the simpler GRU cells. Finally, although it seems plausible that incorporating information from future events might aid in prediction, that does not appear to be the case. The networks with bidirectional LSTM layers had very similar but slightly inferior AUROC scores compared the unidirectional networks, which is particularly notable since each bidirectional layer has more parameters than the unidirectional layers. Because models were selected based on the

Table 5.2: Deeper RNNs

RNN	**Hidden Layers**	**AUROC**	**Accuracy**	**F1 Score**
LSTM	3	0.9406	0.9338	0.6478
Bidir. LSTM/LSTM/GRU	3	0.9365	0.9308	0.6383
LSTM/Dense	3	0.9374	0.9315	0.6504
Bidir. LTSM/LSTM/LSTM	3	0.9382	0.9311	0.64853
Bidir. LSTM/Dense	3	0.9344	0.9264	0.59522
LSTM	4	0.9389	0.9324	0.64507

lowest validation loss, this result is not due to overfitting the networks with bidirectional layers. Finally, allthough much better performing than traditional methods (or the feed forward network with only the traditional predictors as features) the best of these models also have slightly lower AUROC scores than the most accurate Purushotham et al. model which acheived an AUROC score of 0.941.

5.2 Deeper Recurrent Nerual Networks

Table 5.2 shows results from a selection of deeper recurrent networks. A variety of combinations were tried, including architectures which added a dense layer prior to the output layer. All models showed good – and similar – performance but the only model that resulted in better performance than the model with two unidirectional LSTM layers was the model with three such layers. This model has a nearly identical AUROC score to the best Multimodal model from the Purushotam et al. study (AUROC of 0.9406 and 0.9410, respectively), shown in Table 2.1.

All other configurations of GRU, bidirectional, or dense layers had slightly worse AUROC scores. Notably, a model with four unidirectional LSTM layers slightly underperformed the three-layer version. While far from an exhaustive list of all possible configurations, the results broadly confirm that unidirectional LSTM is the best choice for the problem even though good performance was obtained from all models. Again, the lack of performance gains from the bidirectional layers is especially notable since these layers include more parameters overall.

5.3 Networks Including Clinical Notes

Another primary goal of the study was to explore potential performance gains from adding information from clinical notes. Results from the models without clinical notes suggested no added benefit from a bidirectional layer, but with TF-IDF scores for words now included as features that may no longer be the case. Table 5.3 shows results for a variety of network architectures including the same 136 features from the structured clinical data and an additional 2,500 features from the clinical notes, trained for 250 epochs. These new features represent words with the top 2,500 TF-IDF scores. If one of these words was present in the notes during the hour, the feature takes the value of the TF-IDF score, otherwise it takes a value of zero. The three-layer unidirectional LSTM network, the best performing model without the notes features, also has the highest AUROC score when including them and the score has risen, although only slightly. The model with two bidirectional LSTM layers has a similar score but the fact that this model has more parameters without better performance suggests that the bidirectional part of the layer is not important to the model. As in the models without including notes, models with GRU layers had similar but slightly worse performance and adding a dense layer before the output layer was not helpful. Notably inclusion of the notes has led the best models here to perform with higher accuracy than the best model from the Purutosham et al. study which did not include the notes.

By tracking training and validation loss, its was also unclear that the three-layer unidirectional model was done training after 250 epochs. Figure 5.1 plots loss and AUROC score over the epochs for the three-layer unidirectional LSTM including no clinical notes features, 1,000, and 2,500 clinical notes features trained for 500 rather than 250 epochs. While progress had mostly flattened out, training continued slowly and the selected best model (as defined by smallest loss) occurred in epoch 268 with no clinical notes features, epoch 345

Table 5.3: Networks with 2,500 Clinical Notes Features, Trained for 250 Epochs

	Hidden Layers	**AUROC**	**Accuracy**	**F1 Score**
Bidir. LSTM	2	0.9414	0.9325	0.6653
Bidir. LSTM (2), GRU	3	0.9403	0.9327	0.6537
Bidir. LSTM (2), Dense	3	0.9357	0.9303	0.6775
LSTM	3	0.9418	0.9341	0.6676

with 1,000, and epoch 414 with 2,500 notes features. As a final test of the value of adding clinical note TF-IDF scores, Table 5.5 shows the results of this model, along with results of three-layer unidirectional LSTM models trained for 500 epochs using 0, 250, 1,000, and 4,000 clinical notes features.

Figure 5.1: Training and Validation Plots for RNNs with Increasing Clinical Notes Features

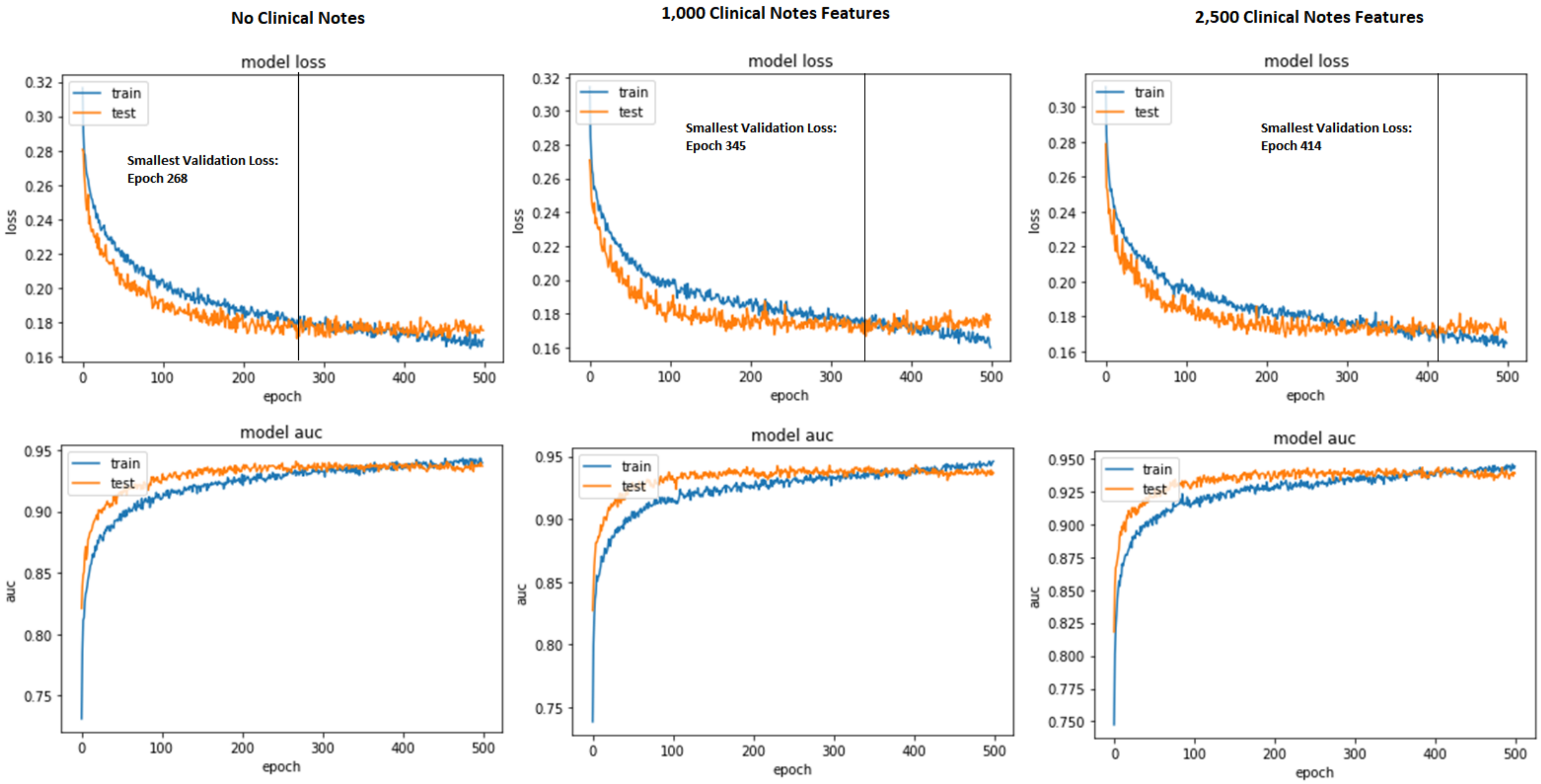


Table 5.4: Networks with a Range of Clinical Notes Features, Trained for 500 Epochs

RNN	Hidden Layers	Note Features	AUROC	Accuracy	F1 Score
LSTM	3	0	0.9408	0.9327	0.6486
LSTM	3	250	0.9400	0.9310	0.6506
LSTM	3	1,000	0.9415	0.9328	0.6576
LSTM	3	2,500	0.9432	0.9321	0.6361
LSTM	3	4,000	0.9425	0.9320	0.6457

Training the model with three unidirectional LSTM layers and no clinical notes features for 500 epochs barely increased the AUROC score. This was expected as the validation loss plots suggested the model was done training, but the result is important in that this model with no note features achieves almost exactly the same AUROC score as the highest performing model from the Purushotham et al. study – 0.9408 compared to 0.9410, despite using different models – and models including clinical notes show higher AUROC scores than the comparison study. As more clinical notes features are included the scores generally rise, although not dramatically, until the 4,000 word model which does not show improvement. While relatively minor, the higher accuracy of models using a simple natural language processing algorithm to generate features from the notes suggest this information should not be routinely ignored in attempts to build predictive algorithms for EHR data. The fact that the relatively simplistic TF-IDF metric can improve performance suggests that a more complex representation of the notes, which captures words' meaning rather than simply their frequency, may result in an even bigger performance increase.

5.4 Comparison of Evaluation Metrics

The AUROC score was used as throughout as the primary performance metric by which models were compared, but there are important reasons to pay attention to additional performance metrics as well. Table 5.5 shows the full set of performance metrics for a selection of the higher performing models of various types. For the model with two hidden GRU lay-

Table 5.5: Full Evaluation Metrics for Selected Models

RNN	**Note Features**	**AUROC**	**F1 Score**	**Recall**	**Precision**
GRU (2)	0	0.9311	0.6082	0.4921	0.7961
LSTM (3)	0	0.9406	0.6478	0.5261	0.8427
LSTM (2), Dense	0	0.9396	0.6775	0.5806	0.8132
LSTM (3)	250	0.9400	0.6506	0.5552	0.7856
LSTM (3)	1000	0.9415	0.6576	0.5576	0.8014
LSTM (3)	2500	0.9432	0.6361	0.5127	0.8376
LSTM (3)	4000	0.9425	0.6457	0.5358	0.8125

ers, the performance measures are all in agreement that the model is inferior to the others. However for the higher performing deeper LSTM networks the metrics show some ambiguity in which model is highest performing. Based on the primary AUROC metric, the network with three hidden LSTM layers and 2,500 clinical note features is the highest performing. However, the model with the highest F1 score - if only by a small margin - is the model with two LSTM layers and a dense layer with no clinical notes included. Examining the performance of these models based on precision and recall separately highlights the importance of defining a model's purpose before selecting the "best" model. If the goal of the model is to estimate patient risk to use as a covariate in the context of a broader study, model builders might prefer the model with highest precision, as this metric summarizes the probability of death conditional on a predicted death. On the other hand, if the model were to be used in a clinical setting alerting doctors to a patient in a dire situation, model builders might prefer the model with the highest recall since this metric captures the share of true deaths that were accurately predicted as such, and thereby minimizes highly undesirable cases where a patient who is predicted to survive actually dies. It is instructive that the model with the highest precision rates relatively poorly on recall, and vice versa, while these two models look relatively similar in terms of the balanced AUROC and F1 score metrics.

CHAPTER 6

Conclusion

This study implemented many variations of Recurrent Neural Networks for predicting mortality in the ICU and documented the implications for model performance. One general conclusion is that a range of popular choices of RNN architecture, memory cells, and hyperparameters can achieve good (and similar) results. However the results here also show that some modeling choices consistently led to better performance in this context. Throughout the study, LSTM layers generally performed somewhat better than GRU layers which suggests that the additional gates in the LSTM cells provide performance gains at the cost of additional computing power. Secondly, although it seems intuitive that using information from future timesteps to predict a current timestep in the ICU whould be beneficial, this study found no evidence that using bidirectional RNN layers yields performance gains. The RNNs with bidirectional layers tended to be no more accurate, and in some cases slightly worse performing, than unidirectional layers, despite the fact that these bidirectional layers include more parameters. Additionaly, this study concludes that the Adam optimizer outperforms the RMSProp optimizer and found no evidence to support deviating from the authors' proposed default parameter of β_1, β_2, and ϵ as 0.9, 0.999, and 10^{-8} respectively.

Another contribution of this study is exploring the addition of unstructured free-text clinical notes in combination with the more commonly used structured clinical data. The TF-IDF algorithm used here for natural language processing represents more of a first step, or proof of concept, approach than something that would be implemented in a production environment. The TF-IDF algorithm is overly simplistic for these purposes because it only measures the relative frequency of individual words in the note without considering phrases of multiple words or even attempting to determine the presence or absence conditions or treatments that are mentioned. A more sophisticated approach might consider phrases rather than just words, attempt to locate a previously assembled clinical notes corpus, and perform a sentiment analysis to classify each note as postiive, negative, or nuetral with respect to patient prognosis. Still, that even a very basic NLP approach resulted in higher accuracy, even by relatively small margins, suggests that this data source should not be ignored, and additional performance gains could result from more sophisticated text processing.

Placing the results of this study in the context of prior work, all models investigated here had non-trivially higher accuracy than the widely used SAPS II score [1]. This is a result that has been found consistently in the existing research on mortality prediction in the ICU. With respect to the Purushotham et al. work [8], from which this study borrows most data processing decisions, the best performing model presented here without considering clinical notes achieves nearly identical performance compared to the best performing Purushotham model despite significantly different modeling choices. Models that included processed clinical notes features, not considered by Purushotham et al., slightly outperformed the models in that study, although no performance gains were found after 2,500 TF-IDF score features. Although the models here are less comparable to other previous work due to methodological and data differences, models in this study generally have comparable AUROC performance to other published work. For example, the accuracy of the higher performing models in this study is within the range of the results reported by Google AI [9]. Google AI processes the notes with more sophisticated NLP algorithms and uses an ensemble of multiple prediction techniques but also faces a more challenging prediction task in that predictions are made for all hospital admissions not just those in the ICU.

Finally, examining alternative performance metrics underscores the extent to which model selection is sensitive to the choice of metric. The AUROC statistic is clearly the most widely used metric in current mortality prediction work, but another balanced metric – the F1 score – could also be chosen and would sometime yield different results. For some purposes a balanced metric may not be the most appropriate choice at all. For example models with the goal of generating accurate risk controls to be used as inputs in another model – as is often the case with SAPS II – may prefer to maximize precision. Models such as those developed by Google AI, which are intended to be clinical tools, may be better off focusing on recall since most clinicians would likely have a higher tolerance for false alarms than for undetected fatal conditions. The results here show that models chosen to maximize recall may do so at the expense of precision, and vice versa, so carefully selecting metrics at the outset is important.

APPENDIX A

Appendix

Table A.1: Full list of inputs from structured clinical data and the table from which they were drawn

Data Element	MIMIC-III Table
AlarmsOn	CHARTEVENTS
arterial_pressure_mean	CHARTEVENTS
ArterialBloodPressurediastolic	CHARTEVENTS
ArterialBloodPressuremean	CHARTEVENTS
body_temperature	CHARTEVENTS
CentralVenousPressure	CHARTEVENTS
diastolic_blood_pressure_mean	CHARTEVENTS
fio2	CHARTEVENTS
Gcseyes	CHARTEVENTS
Gcsmotor	CHARTEVENTS
Gcsverbal	CHARTEVENTS
Glucose	CHARTEVENTS
Glucosefingerstick	CHARTEVENTS
heart_rate	CHARTEVENTS
HeartRateAlarm-Low	CHARTEVENTS
Height	CHARTEVENTS
ie_ratio_mean	CHARTEVENTS

Continued on next page

Table A.1 – full list of structured inputs continued from previous page

Data Element	MIMIC-III Postgres Table
MeanAirwayPressure	CHARTEVENTS
MinuteVolume	CHARTEVENTS
MinuteVolumeAlarm-High	CHARTEVENTS
MinuteVolumeAlarm-Low	CHARTEVENTS
O2Flow	CHARTEVENTS
Peakinsp.Pressure	CHARTEVENTS
PEEPset	CHARTEVENTS
PulmonaryArteryPressurediastolic	CHARTEVENTS
PulmonaryArteryPressuremean	CHARTEVENTS
PulmonaryArteryPressuresystolic	CHARTEVENTS
RespAlarm-High	CHARTEVENTS
RespiratoryRate	CHARTEVENTS
RespiratoryRate(Set)	CHARTEVENTS
SkinCare	CHARTEVENTS
spo2_peripheral	CHARTEVENTS
SpO2DesatLimit	CHARTEVENTS
systolic_blood_pressure_abp_mean	CHARTEVENTS
TidalVolume(observed)	CHARTEVENTS
TidalVolume(set)	CHARTEVENTS
weight	CHARTEVENTS
Albumin 5%	INPUTEVENTS
Calcium Gluconate	INPUTEVENTS
D5 1/2NS	INPUTEVENTS
dopamine	INPUTEVENTS
epinephrine	INPUTEVENTS
fentanyl	INPUTEVENTS

Continued on next page

Table A.1 – full list of structured inputs continued from previous page

Data Element	MIMIC-III Postgres Table
Fresh Frozen Plasma	INPUTEVENTS
Furosemide (Lasix)	INPUTEVENTS
Gastric Meds	INPUTEVENTS
GT Flush	INPUTEVENTS
Hydralazine	INPUTEVENTS
Insulin - Regular	INPUTEVENTS
KCL (Bolus)	INPUTEVENTS
Lorazepam (Ativan)	INPUTEVENTS
LR	INPUTEVENTS
Magnesium Sulfate	INPUTEVENTS
Magnesium Sulfate (Bolus)	INPUTEVENTS
midazolam	INPUTEVENTS
Midazolam (Versed)	INPUTEVENTS
Morphine Sulfate	INPUTEVENTS
Nitroglycerin	INPUTEVENTS
Norepinephrine	INPUTEVENTS
OR Crystalloid Intake	INPUTEVENTS
Packed Red Blood Cells	INPUTEVENTS
Phenylephrine	INPUTEVENTS
Piggyback	INPUTEVENTS
PO Intake	INPUTEVENTS
Potassium Chloride	INPUTEVENTS
Propofol	INPUTEVENTS
Solution	INPUTEVENTS
Sterile Water	INPUTEVENTS
Vasopressin	INPUTEVENTS

Continued on next page

Table A.1 – full list of structured inputs continued from previous page

Data Element	MIMIC-III Postgres Table
ALANINE AMINOTRANSFERASE (ALT)	LABEVENTS
ALBUMIN	LABEVENTS
ALKALINE PHOSPHATASE	LABEVENTS
ANION GAP	LABEVENTS
ASPARATE AMINOTRANSFERASE (AST)	LABEVENTS
BASE EXCESS	LABEVENTS
BASOPHILS	LABEVENTS
bilirubin_level	LABEVENTS
CALCIUM	LABEVENTS
CALCULATED TOTAL CO2	LABEVENTS
CHLORIDE	LABEVENTS
Chloride	LABEVENTS
CREATININE	LABEVENTS
EOSINOPHILS	LABEVENTS
GLUCOSE	LABEVENTS
HEMATOCRIT	LABEVENTS
HEMOGLOBIN	LABEVENTS
Hgb	LABEVENTS
INR(PT)	LABEVENTS
LACTATE	LABEVENTS
LYMPHOCYTES	LABEVENTS
MAGNESIUM, TOTAL	LABEVENTS
MCH	LABEVENTS
MCHC	LABEVENTS
MCV	LABEVENTS
MONOCYTES	LABEVENTS

Continued on next page

Table A.1 – full list of structured inputs continued from previous page

Data Element	MIMIC-III Postgres Table
NEUTROPHILS	LABEVENTS
pao2	LABEVENTS
PCO2	LABEVENTS
peep	LABEVENTS
PH	LABEVENTS
PHOSPHATE	LABEVENTS
PLATELET COUNT	LABEVENTS
potassium_level_mean	LABEVENTS
PT	LABEVENTS
PTT	LABEVENTS
RDW	LABEVENTS
RED BLOOD CELLS	LABEVENTS
serum_bicarbonate_level_mean	LABEVENTS
serum_urea_nitrogen_level	LABEVENTS
sodium_level_mean	LABEVENTS
SPECIFIC GRAVITY	LABEVENTS
white_blood_cells_count_mean	LABEVENTS
Chest Tube #1	OUTPUTEVENTS
Chest Tube #2	OUTPUTEVENTS
Fecal Bag	OUTPUTEVENTS
Gastric Tube	OUTPUTEVENTS
Jackson Pratt #1	OUTPUTEVENTS
OR EBL	OUTPUTEVENTS
Pre-Admission	OUTPUTEVENTS
Stool Out Stool	OUTPUTEVENTS
TF Residual	OUTPUTEVENTS

Continued on next page

Table A.1 – full list of structured inputs continued from previous page

Data Element	MIMIC-III Postgres Table
Ultrafiltrate	OUTPUTEVENTS
urinary_output_sum	OUTPUTEVENTS
Urine Out Incontinent	OUTPUTEVENTS
Aspirin	PRESCRIPTIONS
Bisacodyl	PRESCRIPTIONS
Docusate Sodium	PRESCRIPTIONS
Humulin-R Insulin	PRESCRIPTIONS
Metoprolol Tartrate	PRESCRIPTIONS
Pantoprazole	PRESCRIPTIONS

REFERENCES

[1] 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, pp. 2957-2963, 1993.

[2] R. Pirracchio. *Mortality prediction in the ICU based on mimic-ii results from the super ICU learner algorithm (SICULA) project.* Secondary Analysis of Electronic Health Records, Springer, pp. 295-313, 2016.

[3] Y.P. Tabak, X. Sun, C.M. Nunez, R.S. Johannes. *Using electronic health record data to develop inpatient mortality predictive model: Acute Laboratory Risk of Mortality Score (ALaRMS.* Using electronic health record data to develop inpatient mortality predictive model: Acute Laboratory Risk of Mortality Score (ALaRMS). J Am Med Inform Assoc., 21(3):455-463, 2014.

[4] J. Lee. *Patient-specific predictive modeling using random forests: an observational study for the critically ill.* JMIR Med. Inf., 5, 2017.

[5] 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 J. Biomed. Health Inf., 19, 2015.

[6] K. Yu, M. Zhang, T. Cui, M. Hauskrecht. *Monitoring ICU Mortality Risk with A Long Short-Term Memory Recurrent Neural Network.* Pac Symp Biocomput. 25:103-114, 2020.

[7] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.* ArXiv 1406.1078, 2014.

[8] S. Purushotham, C. Meng, Z. Che, Y. Liu. *Benchmarking deep learning models on large healthcare datasets.* Journal of Biomedical Informatics, Volume 83, ISSN 1532-0464, 2018.

[9] A. Rajkomar, E. Oren, K. Chen, et al. *Scalable and accurate deep learning with electronic health records.* npj Digital Med 1, 18, 2018.

[10] A.E.W. Johnson, T.J. Pollard, L. Shen, L. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, and R.G. Mark. *MIMIC-III, a freely accessible critical care database.* Scientific Data. DOI: 10.1038/sdata.2016.35, 2016.

[11] T. Tieleman and G. Hinton. *Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude* Coursera: Neural Networks for Machine Learning, 2012.

[12] D.P. Kingma, J.L. Ba . *Adam: a Method for Stochastic Optimization.* International Conference on Learning Representations, 1–13, 2015.

[13] G.L. Grunkemeier, R. Jin *Receiver operating characteristic curve analysis of clinical risk models.*Ann Thorac Surg., 72(2):323-6, 2001.

[14] I. Goodfellow and Y. Bengio and A. Courville. *Deep Learning.* MIT Press, www.deeplearningbook.org, 2016.

[15] S. Hochreiter and J. Schmidhuber. *Long short-term memory.* Neural Computation, 9(8), 1735–1780, 1997.

[16] M. Schuster and K. Paliwal. *Bidirectional recurrent neural networks.* IEEE Transactions on Signal Processing 45(11):2673 - 2681, 1997.

[17] W. Pu, N. Liu, S. Yan, J. Yan, K. Xie and Z. Chen. *Local Word Bag Model for Text Categorization.* Seventh IEEE International Conference on Data Mining, 2007.

[18] T. Hastie, R. Tibshirani,and J.H. Frieldman. *The elements of statistical learning: Data mining, inference, and prediction.* New York: Springer, 2001.

[19] F. Chollet. *Keras.* keras.io, 2015.