A CNN-LSTM for predicting mortality in the ICU

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Mohammad Hashir Khan

May 2019

Abstract

An accurate predicted mortality is crucial to healthcare as it provides an empirical risk estimate for prognostic decision making, patient stratification and hospital benchmarking. Current prediction methods in practice are severity of disease scoring systems that usually involve a fixed set of admission attributes and summarized physiological data. These systems are prone to bias and require substantial manual effort which necessitates an updated approach which can account for most shortcomings. Clinical observation notes allow for recording highly subjective data on the patient that can possibly facilitate higher discrimination. Moreover, deep learning models can automatically extract and select features without human input.

This thesis investigates the potential of a combination of a deep learning model and notes for predicting mortality with a higher accuracy. A custom architecture, called **CNN-LSTM**, is conceptualized for mapping multiple notes compiled in a hospital stay to a mortality outcome. It employs both convolutional and recurrent layers with the former capturing semantic relationships in individual notes independently and the latter capturing temporal relationships between concurrent notes in a hospital stay. This approach is compared to three severity of disease scoring systems with a case study on the MIMIC-III dataset. Experiments are set up to assess the CNN-LSTM for predicting mortality using only the notes from the first 24, 12 and 48 hours of a patient stay. The model is trained using K-fold cross-validation with k=5 and the mortality probability calculated by the three severity scores on the held-out set is used as the baseline. It is found that the CNN-LSTM outperforms the baseline on all experiments which serves as a proof-of-concept of how notes and deep learning can better outcome prediction.

Table of Contents

1 Introduction **1**

1.1 The intensive care unit . . . 1

1.2 The relevance of mortality prediction . . . 4

1.3 Current methods of mortality prediction . . . 6

1.4 The deficiencies with current methods . . . 7

1.5 Problem statement . . . 11

1.6 Remodeling mortality . . . 12

1.7 Proposal . . . 14

1.8 Organization of thesis . . . 16

2 Background **17**

2.1 Neural networks for text classification . . . 17

2.1.1 Recurrent neural networks . . . 18

2.1.2 Convolutional neural networks . . . 20

2.2 Word embeddings . . . 23

2.3 Related work . . . 25

3 Methodology **34**

3.1 Data preparation . . . 34

3.1.1 Preprocessing the notes in MIMIC-III . . . 35

3.1.2 fastText embeddings . . . 37

3.2 Experimental setup . . . 39

3.2.1 Design of experiments . . . 39

3.2.2 Cohort selection . . . 42
3.3 Modeling . . . 43
3.3.1 Model configuration . . . 43
3.3.2 Training . . . 51

4 Analysis and results **54**
4.1 Evaluation schema . . . 54
4.1.1 Baseline . . . 54
4.1.2 Metrics . . . 56
4.2 Results . . . 58
4.3 Discussion . . . 59

5 Conclusion **62**

Bibliography **65**

Vita **77**

List of Tables

1.1 Shortcomings of scoring systems 15

2.1 Cohort details of studies of interest 31
2.2 The kind of data used in studies of interest 32
2.3 Algorithms used in studies of interest 33

3.1 Columns in NOTEEVENTS 36
3.2 Experiment details 42
3.3 Cohort populations 43

4.1 Average AUROC per fold 58
4.2 Average AUPRC per fold 58

List of Figures

2.1 Unrolling an RNN 19
2.2 Unrolled LSTM 20
2.3 The convolution operation 22

3.1 The pipeline 34
3.2 t-SNE visualization at step 0 (a) and step 1300 (b) 38
3.3 Highlighting oxycodone in the t-SNE visualization 40
3.4 Highlighting fracture in the t-SNE visualization 41
3.5 Cohort selection criteria 44
3.6 Model components 45
3.7 Visualization of a single semantic feature extractor 50
3.8 Model architecture 52

4.1 Confusion matrices for CL-24 60

Chapter 1

Introduction

This chapter details the importance of the intensive care unit (ICU), the relevance of mortality prediction to the ICU and the current methods and their shortcomings. Then, it is discussed at length how the proposal of this thesis aims to improve mortality prediction all the while accounting for the shortcomings.

1.1 The intensive care unit

The intensive care unit (ICU), also known as the critical care unit (CCU), is a special department of a hospital catering to patients suffering from acute life-threatening illnesses. These patients are monitored round-the-clock by sophisticated equipment and specially trained staff. There can be different types of ICUs, such as neonatal and surgical, but most ICUs are multidisciplinary in terms of medical specialties.

The increasing relevance of the ICU

The intensive care unit is usually home to the sickest patients; as such, it harbors a highly critical atmosphere that requires quick and highly accurate decisions to be made by the care staff. Comprehensive care is provided in the ICU by a multi-professional team that can include physicians, nurses, pharmacists, respiratory therapists, etc. Consequentially, the ICU has a higher amount of resources allocated to it and is a source of considerable

expenditures for hospitals in the United States. ICU beds may be only 10% of the total beds but the ICU costs can be up to 22% of total hospital costs (Halpern et al., 1994). In fact, hospitals actually lose money on ICU stays: Cooper and Linde-Zwirble (2004) found that Medicare, on an average, reimburses the hospital only 83% of the costs of cases with an ICU stay, as compared to 105% of costs of cases without any ICU services. The ICU is expensive for the patient too: hospital discharges that had ICU services were charged $61,800 on an average, which is more than twice the amount of those without any ICU services ($25,200). Discharges with ICU services represented 47.5% of total hospital costs while being only 26.9% of total discharges (Barrett et al., 2014).

There have been three studies from Memorial Sloan Kettering Cancer Center that analyze the costs and use of critical care units in the USA from a national perspective over three somewhat consecutive time periods: Halpern et al. (2004) analyzes the trends from 1985 - 2000, Halpern and Pastores (2010) analyzes trends from 2000 - 2005 and Halpern et al. (2016) analyzes trends from 2000 - 2010 with a special focus on Medicare and Medicaid. On aggregating the results from these three studies, it is clear that in the 25 years between 1985 and 2010, the total cost of critical care medicine has increased by **465%** (from $19 billion in 1985 to $108 billion in 2010). The cost of a critical care bed per day has increased by approximately**263 %**(from $1,185 in 1985 to $4,300 in 2010).

However, there has been a downward trend in the number of hospitals offering acute care at the national level in the United States. From 2000 to 2009, the number of acute care hospitals decreased from 3,468 to 3,155. In the same time period, the change in the number of total hospital beds was incremental (only a 0.8% increase); but the total number of ICU beds has increased by 15% and the proportion of ICU beds of total hospital beds has increased from 10% to 12% (Wallace et al., 2015). Moreover, there is an increasing rate of intensive care unit admissions from emergency departments which is higher than the rate of population growth and overall emergency department visits. From 2002 to 2009, ICU admissions from emergency departments increased by 48.8% while overall emergency department visits have increased by only 18.2% (Mullins et al., 2013).

All in all, this presents a significant growing trend of the increasing role of the ICU in the national healthcare system and necessitates the improvement of quality of care provided in the ICU.

Outcome prediction for quality improvement

Healthcare outcomes are defined as the end results achieved by providing medical services to patients like mortality, readmissions and morbidity. Outcomes are an integral component of the quality improvement process. It is a part of the dominantly used Donabedian model (Donabedian, 1988) of quality improvement in care; this model has a 'structure-process-outcomes' approach for alleviating quality. Focusing on outcomes is also stipulated by the Society of Critical Care Medicine (Curtis et al., 2006) and European Society of Intensive Care Medicine (Thijs and Members of the Task Force European Society of Intensive Care Medicine, 1997) in their guidelines for improving quality of care.

Outcome prediction has garnered much interest as a potential factor for increasing the quality of treatment in the ICU. Proactive measures, based on predicted outcomes, tend to enhance medical decision making and reduce costs. Moreover, a higher-level assessment of healthcare quality is dependent on risk-adjusted outcome evaluation which involves predicted outcomes. Hence, it is imperative that outcome prediction is as accurate as possible.

Towards data driven medicine

Data-driven medicine is a rising interdisciplinary approach that aims to utilize all the clinical data available to augment current medical decision making and recognizes the value of incorporating personalized information into current medical practices. Methods for data-driven medicine are currently spearheaded by recent developments in artificial intelligence and machine learning. Artificial intelligence and machine learning have shown great success in healthcare from actualizing precision medicine through computational genomics (Menden et al., 2013) to aiding detection of pneumonia in X-rays better than individual radiologists (Rajpurkar et al., 2017).

Recent developments in technology have enabled capturing colossal amounts of data. As digital storage formats get cheaper and methods of capturing medical data become more efficient, the healthcare sector is on track to host a significant amount of data. Reddy and Sun (2013) have stated that healthcare data is expected to reach 25,000 petabytes on a global scale in 2020. The current deluge of the so-called "big data" is a severely underutilized asset that can have far-reaching effects in the healthcare industry. A study done by McKinsey Global Institute stated that this untapped resource, if used creatively and effectively, could reduce healthcare expenditures by eight percent and create more than $300 billion in value every year (Manyika et al., 2011).

The intensive care unit is one such setting in a hospital that has an abundance of data. The patients are monitored continuously by staff and equipment and all of their data is recorded diligently into the system. The data-rich environment of the ICU makes it an appropriate candidate to implement machine learning approaches for outcome prediction. Hence, this thesis is concerned with using recent developments in machine learning to increase the accuracy of outcome prediction in the ICU, specifically **the probability of mortality**.

1.2 The relevance of mortality prediction

Mortality rate is one of the most commonly used outcomes for measuring quality of care and as such, the task of mortality prediction has maintained a substantial presence in the healthcare industry. Mortality prediction in the ICU serves various functions for different stakeholders in healthcare as elaborated below.

At the individual level

It seeks to provide an unbiased empirical risk estimate for clinical decision making. Physicians equipped with quantitative estimates of the probability of mortality can prioritize patient care and prepare a targeted plan of action and prognosis. The probability of mortality can also provide a rough measure of the effectiveness of treatment: Afessa et al. (2004) found that a higher Acute Physiology Score on the third ICU day rather than the first day can identify potentially ineffective care in patients who entered with a high mortality probability.

It can also be used to evaluate the suitability of patients for novel treatments (Russell, 2015). Moreover, the probability of mortality provides the physician and the patients and/or their family with a basis for deciding on the aggressiveness of the treatment (Baird et al., 2008) or withdrawal of life support (Mendez-Tellez and Dorman, 2005).

At the organizational level

Mortality probability is used to quantify severity of illness for resource allocation (Russell, 2015). As the ICU is a significant source of costs, the mortality probability should be taken into account to ensure that the resource allocation is efficient. Mendez-Tellez and Dorman (2005) claim that identification of patients who may not survive their ICU stay will yield considerable cost savings.

It also aids in administrative tasks: Floege et al. (2015) states that a mortality risk score may help to categorize patients for registries or reimbursement systems and Park et al. (2009) even goes as far as to anticipate a role for it in accrediting individual critical care units.

The probability of mortality also serves as a basis for patient stratification. Risk stratification based on severity of disease score and mortality probability provides comparison standards between groups for quality assessments and treatment studies by research teams involved in the ICU (Russell, 2015).

At the regional and national health infrastructure level

Hospital quality is benchmarked and consequently rewarded by regional and national agencies using various measures including risk-adjusted mortality rates (RAMR). The RAMR of a hospital is its raw mortality rate that has undergone a statistical risk adjustment to account for any anomalies in the hospital's intended population. To calculate the statistical risk adjustment for mortality rate, a ratio of observed and predicted mortality is used.

The Agency for Healthcare Research and Quality (AHRQ) considers a set of 101 quality measures, of which 11 have been endorsed by the National Quality Forum (a not-for-profit that is concerned with health care quality measurement and reporting). As many as 8 of the 11 metrics were related to risk-adjusted mortality rates. Moreover, the Centers for Medicare and Medicaid Services (CMS) uses three kinds of risk-adjusted mortality rates as part of

its evaluation of healthcare quality in its Hospital Value-Based Purchasing program (which rewards hospitals for higher quality of care provided to Medicare beneficiaries).

This means that an accurate predicted mortality is pivotal to a *representative* risk-adjusted mortality rate. Mortality prediction, therefore, plays a role in the formulation of an illustrative indicator of quality of care which provides a fair and equitable means of comparison and evaluation of hospitals.

1.3 Current methods of mortality prediction

Mortality prediction may ostensibly seem like a hard binary classification problem but, realistically, it is more probabilistic. The probability of mortality is derived from a continuum of values that are positively correlated with the worsening condition of a patient and as such, most "severity of disease" scoring systems have proven to be a reliable measure of mortality probability. Severity scoring systems are methods rooted in evidence-based medicine and can be general or disease-specific. Some general scoring systems in use are APACHE II, APACHE III, SAPS II and MPM II. The severity of disease score is usually obtained by summing up certain features which are based on physiological measurements and attributes of the patient and/or their admission and the probability of mortality is derived from this final severity score. However, some of these systems are more statistically grounded with the weights for certain features being determined through multiple logistic regression.

The aforementioned systems are used primarily at the individual and organization level. Predicted mortality for reporting risk-adjusted mortality rates at higher levels is calculated using protocols set by the United States Department of Health and Human Services (the governing agency of both CMS and AHRQ). The protocols mostly involve systematically chosen risk factors in combination with logistic regression which were formulated through subjective empirical studies.

Assessing the performance of mortality prediction is usually described by discrimination and calibration. Discrimination refers to the ability of the predictor to differentiate between those who are going to die from those that will survive and calibration refers to how accurate the predicted probabilities are compared to the real probabilities (goodness of fit).

1.4 The deficiencies with current methods

The current methods of predicting mortality may seem effective but they suffer from a multitude of shortcomings.

Documented poor generalizeability

Generalizeability refers to the ability of a predictor to retain its performance across different populations. The scoring systems suffer from poor generalizeability as they are biased towards their specific patient cohorts and may not represent a general population of sick people. Moreover, this bias is exacerbated by the strict inclusion and exclusion criterias imposed on the original study.

This is highly notable when the scoring systems are applied in other countries. Pappachan et al. (1999) evaluated the performance of APACHE-III in the United Kingdom and found that the system overestimated mortality. It put forward the failure of APACHE-III to fit UK data as one of the possible reasons for the undesirable performance. Usually to adapt the scoring system to a new population, it requires re-calibration. Re-calibration involves re-calculating some of the coefficients of the equation that converts the raw severity score to a mortality probability. While Haaland et al. (2014) found that re-calibrating the SAPS II to a Norwegian population resulted in higher performance, Oliveira et al. (2013) discovered that SAPS 3 and APACHE III still performed poorly on transplant patients even after calibrating for Brazil.

There has been particular interest in evaluating the performance of the scoring systems in developing countries as most of the scoring systems were created and tested in Western countries. Haniffa et al. (2018) conducted an extensive review of the performance of APACHE III, SAPS II and MPM II in low and middle income countries and concluded that the performance of these systems in these countries is moderate, at best. It also theorized that the diagnostic categories in the scoring systems weren't suited to capture diseases that were common in the developing world like malaria and dengue. This was echoed in Aggarwal et al. (2006) as well; in its assessment of the performance of three scoring systems in an Indian ICU, it put forward the prevalence of tropical diseases in India as one of the factors

for the poor performance. Similarly, Ghorbani et al. (2017) attributed the poor performance of APACHE IV in an Iranian hospital to the development of the system with American healthcare data.

Effect of salience bias

Salience bias, or perceptual salience, is the bias that arises when the most common cases are perceived as the only ones that exist. The effect of salience bias on the development of the methods has been twofold.

Firstly, salience bias has affected the spectrum of diseases found in the ICU. There are some afflictions that have a higher prevalence in the ICU and as such, the current mortality prediction methods have become oriented more towards these common diseases. Consequently, there is no guarantee of these methods being successful for rare afflictions. Status epilepticus is a rare neurological disease that has been included in the National Organization for Rare Disorders database and has an incidence of only 9.9 - 41 cases per 100,000/year. Cheng (2017) found that APACHE II (the most common ICU scoring system in the US) performed poorly for predicting mortality of patients with status epilepticus. It further theorized that one of the reasons for the sub-par performance was because of few neurological patients being present in the original derivation study of APACHE II.

Similarly, Velissaris et al. (2012) concluded that commonly used severity scores do not seem to be useful in predicting mortality in severe leptospirosis, an extremely rare disease which had only 27 cases in the United States in 2015. The scoring systems also perform poorly for diseases that are generally common but were underrepresented in the derivation study like liver disease and HIV. Brown and Crede (1995) found that APACHE-II significantly underestimated the mortality rate in its study that involved only HIV-positive patients.

Secondly, salience bias can affect patient attribute selection. The features of the methods are designed around specific patient attributes. The initial set of attributes was selected manually by researchers which is prone to the risk of salience bias affecting the selection. There are certain attributes that might have a very high predictive power of mortality for a certain disease. But the methods are designed to be disease-agnostic and as such they only include common attributes like blood pressure, potassium levels etc. Oliveira et al.

(2013) attributed the low accuracy of SAPS 3 and APACHE II in transplant patients to the systems underscoring certain physiological variables such as creatinine level and liver function. Thrombocytopenia (low platelet count) is a significant predictor of mortality in sepsis (Lee et al., 1993) and Acute Respiratory Distress Syndrome (Wang et al., 2014) and was included as a physiological variable in APACHE-I but was later removed. Similarly, glucose level is a physiological variable that is very easy to measure and is associated with increased mortality in ICU patients (Park et al., 2013) but is not included as a physiological variable in SAPS-II. Liu et al. (2016) attempted to introduce glucose variability as a physiological variable in SAPS-II and found that it led to a superior performance in predicting mortality of non-diabetic ICU patients.

However, including more variables in the scoring systems may lead to lower performance: APACHE-I had 34 physiological variables but it was reduced to only 12 in APACHE-II. It may seem counterintuitive but increasing the number of variables increases the risk of missing data and leads to a systemic bias. This makes the patient attribute selection a double-edged sword. These methods trade off disease-specific attributes for higher generalizeability and thus, do not capture all the variability available which is crucial for mortality prediction.

Problems arising from patient data

It is possible that the patient attributes required may not be available at the time of calculation and can delay the assessment. Haniffa et al. (2018) speculated that the poor reproducibility of the scoring systems in low and middle income countries was also affected by the limited resources in the ICUs for recording variables required for calculating the score such as arterial oxygenation (which are routinely available in ICUs in high income countries). This phenomenon leads to missing variables which is detrimental: Afessa et al. (2005) found that missing Acute Physiology Score values may lead to underestimation of mortality predicted by APACHE III and the number and type of missing variables were independently associated with increased mortality. When such variables are missing, a general value is imputed in its place and this may cast a doubt on the accuracy of the prediction. Goldhill and Withington (1996) stated that incorrectly scoring a variable as

normal was the most common error in the ICU that it was analyzing and if this error was consistent, it would lead to an underestimated mortality probability.

Even if the data is available, there are issues with the acquisition and its reliability. Most of the components are physiological variables that can be easily measured but some components require interpretation which might vary from individual to individual. Chen et al. (1999) found that the reliability of data collection for APACHE II varied greatly. The kappa statistic ranged from 0.315 (for the Glasgow Coma Scale component) to 0.976 (for the age component). Féry-Lemonnier et al. (1995) theorized that this range of inter-observer variability was caused by several factors like loose definitions, ambiguities arising from translation into different languages, conversion into SI units and rounding up. Rounding up, in particular, is quite ambiguous as the components of the scoring systems use integers. Age is a component of most scoring systems and is usually not an integer, if calculated precisely. The APACHE II awards zero points if the patient is less than or equal to 44 and two points if between 45 and 54. The ambiguity arises when the patient is 44.5 years old: different physicians will round up to the closest integer differently which leads to a person with an age of 44.5 possibly getting an additional two points in their total APACHE II score. This is quite concerning as APACHE scores are very sensitive to changes: Goldhill and Withington (1996) showed that changing the APACHE score by two or four points can drastically alter the standardized mortality ratio.

The Glasgow Coma Scale (GCS) is another component of both APACHE and SAPS of which the interpretation has been problematic. GCS provides a quantitative measure of how conscious a patient is; confusion in the calculation of the GCS score arises when the patient is pharmacologically sedated (Aggarwal et al., 2006). There was a comparison done between the APACHE scores for sedated patients in two cases of GCS calculation in Livingston et al. (2000): one where the pre-sedation GCS score was considered and the other where a normal GCS score was imputed. It found that the performance of APACHE improved when pre-sedation score was used.

Moreover, these methods are designed to use only a single value from a defined period like 24 or 48 hours after admission. Several sequential recordings of observations exist in the period but these methods use either summarized data or the most abnormal observation in

their calculation and end up disregarding any existing temporal relationships. When data is not obtained at approximately the same time in the course of an acute illness or within a similar period, prognostic estimates will be inaccurate because physiological measures reflect different phases of critical illness (Mendez-Tellez and Dorman, 2005).

Summary of shortcomings

The current methods of predicting mortality —

1. Have poor generalizeability for both patients and diseases
2. Don't utilize all the data available for prediction
3. Select features manually
4. Require specific patient attributes that can have missing data and be ambiguous in interpretation
5. Don't account for any temporal relationships that may exist

1.5 Problem statement

The ICU is home to a significant source of high expenditures for a hospital. Moreover, critical care costs represent a significant portion of the American GDP; it was 0.56% in 2000 and increased to 0.72% in 2010, almost a 30% increase (Halpern et al., 2004, 2016). Mortality in the ICU is a particular concern as 30 - 50% of Americans spend time in an ICU during their final year of life (Barnato et al., 2004) and one in five Americans die in the ICU (Angus et al., 2004). As evidenced by Section 1.2, predicting mortality can alleviate care and is highly important to different stakeholders in the healthcare system.

This relevance points towards a need for highly accurate mortality prediction. However, due to the shortcomings in current prediction methods, the accuracy of the predicted probability seems questionable. Hence, the problem statement is "**how should a mortality prediction method be devised that accounts for all the shortcomings while increasing the accuracy of prediction?**"

1.6 Remodeling mortality

Addressing the shortcomings of current mortality prediction will require reconsideration of both the source data that derives the model and the model itself. A potential candidate for the source data and the model is free-text notes and a deep neural network (DNN), respectively. The following sections examine certain attributes of the two that specifically address the shortcomings.

The source data: free text notes

The intensive care unit is home to a plethora of data (which ranges from being underutilized to completely unused) that intrinsically possesses the following desirable attributes:

— Data acquisition in the ICU is immune from any confirmation or personal biases. As data is recorded on all the patients, there is no chance of either inclusion or exclusion criteria affecting the data available to derive a mortality prediction model.

— The ICU patients represent a large diversity of ailments compared to those in single system specialties. The constant data acquisition in the ICU ensures that every measurement is recorded regardless of how subtle or perceptible they are (rather than only the most common diseases). Moreover, the defining characteristics of all of these diseases are recorded which means that the ICU data possibly has a higher discriminative capability.

Hence, the data collected in the ICU is **more generalizeable** and **the effect of salience bias is reduced**. However, the question arises: *"the ICU has a surfeit of data, so which kind of source data should be chosen such that it embodies the aforementioned attributes in the best manner?"*

The ICU has two kinds of data: structured and unstructured. Structured ICU data often contains missing values due to irregular sampling or negligence in acquisition. As most prediction methods cannot work with missing data, the missing values are replaced with clinically imputed values. This means that there is some inaccurate data used when trying to predict from structured data. Moreover if the structured data were to be utilized, features

need to be extracted manually from it; which in turn, introduces the risk of salience bias affecting the selection of subjectively appropriate features.

In the domain of unstructured data, free-text observation notes (recorded by the ICU staff) pose as a feasible candidate for the source data. The lack of structure in free-text notes doesn't exhibit the "missing data" phenomenon and allows for recording highly subjective data on the patient. This facilitates the capture of most, if not all, defining features and nuances and thus reduces the risk of salience bias affecting the data acquisition. Also, some of the structured ICU data is additionally recorded in the observation notes.

Therefore, free-text notes seem the most appropriate candidate for the input data on the virtue of its unstructured format.

The prediction model: deep neural networks

Machine learning models pose as a feasible candidate for the prediction model, however, most of such models are designed for structured data and the chosen input data is unstructured in nature. While free-text notes can be converted into a more structured form through manual feature extraction techniques based in natural language processing, the temporal relationships between words is lost in doing so. Also, the risk of salience bias on feature selection isn't accounted for while using free-text notes. So the machine learning model chosen should be capable of working with unstructured data while still being able to extract and select features automatically without any bias.

Deep learning models, also called deep neural networks, are a subset of machine learning models that have demonstrated extraordinary success in predicting outcomes from unstructured data. Deep learning models exhibit the two following attributes that address the rest shortcomings —

— They are inherently able to learn which features to extract and select. The models, therefore, eliminate the step of manual and subjective feature extraction while accounting for any complex relationships present in the data nevertheless and thereby learn an accurate representation of the data for classifying into categories.

— A particular type of deep learning model called a recurrent neural network (RNN) has been adapted extensively to utilize free-text documents for prediction. It is inherently designed to predict from sequential data. They have mechanisms to incorporate the temporal change information into their prediction.

Hence, deep learning models have a **reduced effect of salience bias on the feature selection, don't require specific patient data to be available at the time of calculation and account for all temporal relationships** while trying to map features to outcomes. Therefore, deep learning models are deemed as the most appropriate prediction model for predicting mortality.

To summarize, the ways in which free-texts and/or deep learning models address the shortcomings associated with current methods are tabulated in Table 1.1.

1.7 Proposal

It has been established that free-text and a deep learning model in combination can potentially address all the shortcomings of current methods of mortality prediction and achieve higher accuracies. This thesis proposes an investigation and evaluation of such an approach that involves predicting mortality probability for a hospital stay using a "patient representation" learned from observation notes. The performance of the proposed approach is compared to traditional methods of calculating mortality probability; specifically by calculating SAPS-II (Simplified Acute Physiology Score II), APS III (Acute Physiology Score III) and OASIS (Oxford Acute Severity of Illness Score) scores for the patient.

The itemized technical approach is enumerated below:

1. Convolutional feature maps are used to extract features from the vectorized notes to create a document representation.

2. The document representation shall be sorted by hospital stay and charting time and fed into a recurrent neural network model with attention which learns a "patient representation" for each hospital stay.

Table 1.1: How notes and deep learning models address the shortcomings of scoring systems in predicting mortality

Shortcoming	Addressed by		How?
	Notes	DNN	
Have poor generalizeability for both patients and diseases	Yes		Every patient has notes recorded on them, regardless of what disease they are suffering from or any other kind of criteria. Hence, the notes in the ICU represent all types of patients
Don't utilize all the data available for prediction	Yes		Due to the unstructured format, free-text allows for recording highly subjective data. This ensures any attribute with a high discriminative power is recorded regardless of the caregivers' personal opinion of its predictive power
Extract and select features manually		Yes	Deep neural networks are capable of extracting features for prediction automatically.
Require specific patient attributes that can have missing data and ambiguity in intepretation	Yes	Yes	This shortcoming is addressed in both free text notes and RNNs. As the notes lack structure, any type of data can be recorded. Deep learning models extract features automatically so they don't require specific attributes either; hence, there is no ambiguity involved in interpretation. Moreover, RNNs have mechanisms to ignore missing timesteps.
Don't account for any existing temporal relationships		Yes	RNNs are innately designed to include temporal relationships in their prediction

3. The patient representation is then used as an input to a feedforward neural network that outputs the probability of mortality for that hospital stay.

1.8 Organization of thesis

The organization of the thesis is as follows.

— Chapter 2 introduces the specific concepts of deep learning involved in the approach and gives insights on how these mathematical operations work. It also provides a literature review that describes how deep learning has affected healthcare and mortality prediction in particular.

— Chapter 3 lays out a comprehensive pipeline for learning and evaluating the mapping of a set of patient notes to the mortality indicator.

— Chapter 4 elaborates the evaluation methods and the analytical results and a discussion on the performance.

— Chapter 5 concludes the thesis with suggested directions for future work.

Chapter 2

Background

This chapter first provides an overview on the background of the technical concepts used in this thesis. Then, a literature review on related work is detailed.

2.1 Neural networks for text classification

Neural networks are biologically-inspired machine learning algorithms which map the inputs to the outputs by the means of interconnected nodes ('neurons') arranged in layers. The simplest type of neural network is the feed-forward neural network or a **fully connected network**. It consists of a single input and output layer with one or more intermediate 'hidden' layers. An individual layer of this network performs an affine transformation on its inputs, i.e., it multiplies the input with a 'weight' term and then adds a 'bias' term. The transformed input is then passed through a non-linearity, called an 'activation', to get the output that may serve as the input to the next layer. This sequence of affine transformation and non-linearity can be generalized as a matrix multiplication.

$$\mathbf{y} = \sigma(\mathbf{W} \cdot \mathbf{x} + \mathbf{b})$$

where, $\mathbf{x}$ and $\mathbf{y}$ are the input and output vectors respectively, $\mathbf{W}$ is the weight matrix, $\mathbf{b}$ is the bias vector and σ is the activation function.

The set of all the weights and biases of the different neurons form the parameters of a neural network model. A neural network learns the mapping between inputs and output by

optimizing its set of parameters to minimize a loss function. This optimization is achieved through the backpropagation algorithm (a form of gradient descent), i.e., the parameters are iteratively adjusted using the gradient of the loss function to minimize it. In essence, the error is *backpropagated* through all the layers and the weights and biases are modified according to the gradient and this process is repeated till a minima is reached.

2.1.1 Recurrent neural networks

Recurrent neural networks (RNN) are a special class of neural networks that are designed to operate on sequential data that can also exhibit temporal and dynamic behavior, which makes it highly appropriate for modeling sentences. It consists of a feed-forward neural network for each time step with its hidden layer being connected to the hidden layer of the previous time step. Such connections between the hidden layers are known as **recurrent connections** and it forms a directed acyclic graph with the interconnected hidden layers forming a sort of 'memory'. The parameters are shared between time steps which reduces the computations required. RNNs are trained using a modification of backpropagation called *backpropagation through time*. As the name suggests, the algorithm backpropagates the error through the time steps in addition to the feed-forward connections.

Vanilla RNNs have major difficulties in learning long-range dependencies, i.e. the relationships between two time steps that are pretty far apart. This is caused by the vanishing gradients problem: as the gradients are multiplied using the chain rule, this may cause the gradients to become exponentially smaller and approach zero while being propagated back across time steps.

Similarly, RNNs are also vulnerable to the exploding gradient problem where the gradients can become significantly large. The exploding gradient problem can be dealt with by clipping the gradients above a certain value but there is no simple method for dealing with vanishing gradients.

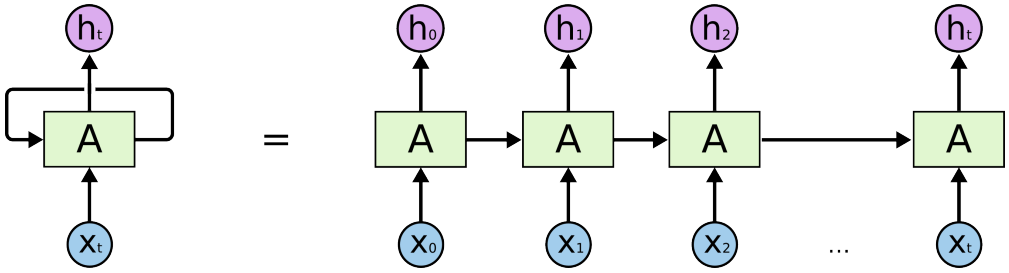


Figure 2.1: Unrolling a recurrent neural network. $\mathbf{x_t}$ and $\mathbf{h_t}$ are the input and hidden output respectively and **A** represents the hidden state. Image from Olah (2015).

Long short term memory

There have been several variants on the recurrent neural network architecture to solve the long-range dependency problem. One such architecture is the long short-term memory (LSTM) devised in Hochreiter and Schmidhuber (1997). The LSTM is a type of a recurrent neural network which has a sequence of inputs, an optional sequence of outputs, a hidden state and a cell state.

While a vanilla RNN cell consists of a fully-connected network with recurrent connections, an LSTM cell abstracts its mathematical operations into 'gates' which are a combination of weight multiplication and non-linearity. Information in the cell state, which is the main information highway of an LSTM, is modified according to the gates. There are three type of gates in an LSTM: forget, input and output gates. The forget gate decides how much of the previous cell state to forget, the input gate decides how much of the new input needs to modify the cell state and the output gate decides how much of the current cell state to add to the current hidden state. Mathematically, the gates are formulated as:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)
\end{aligned}
$$

where f_t, i_t and o_t represent the outputs of the forget, input and output gates respectively and σ represents the activation function.

The current cell state is modified by multiplying the forget gate output with the previous cell state and adding the previous hidden state determined by the input gate. Using an example in language modeling, if the input up to the $t-1$ time step had been about cars,

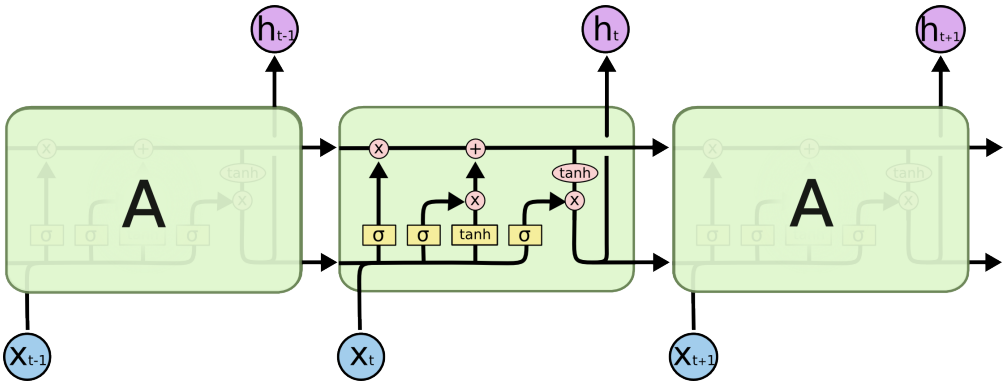


Figure 2.2: An unrolled LSTM depicting how the hidden state is calculated. Image from Olah (2015)

certain segments of the cell state would carry general information about automotives and certain segments would carry information specifically about cars. If the input at time step t is about trucks, then the forget gate will modify the previous cell state to 'forget' about cars and then the input gate will add information about trucks to that segment. As the input is still related to automotives, the forget and input gates will not modify its respective information.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

The new hidden state is the current cell state passed through a hyperbolic tangent non-linearity limited by the output gate.

$$h_t = o_t \cdot tanh(C_t)$$

If an output sequence is needed, it is obtained by passing the hidden state h_t through a user-chosen non-linearity such as sigmoid or softmax.

2.1.2 Convolutional neural networks

Convolutional neural networks (CNN) are a type of neural networks that have been inspired by the neurological concept of receptive fields. They have shown much success in the domain of computer vision and image classification. A CNN differs from a feed-forward neural network in the way that its input is spatial in nature, i.e., it is two-dimensional. Another way it differs is in its use of the convolution operation. The convolution operation is a

feature extraction technique that involves the use of a special matrix known as a kernel. A convolution takes a matrix as an input and outputs another matrix proportional to the shape of the input. The elements of the output matrix are produced by *convolving the kernel* across the input, i.e., summing up all the elements of the Hadamard product (element-wise multiplication) of the kernel with the corresponding region of the input. The convolution operation is demonstrated with the following example in Figures 2.3a and 2.3.

Let $\mathbf{X}$ be the input matrix and $\mathbf{K}$ be the kernel where,

$$\mathbf{X} = \begin{bmatrix} 3 & 3 & 2 & 1 & 0 \\ 0 & 0 & 1 & 3 & 1 \\ 3 & 1 & 2 & 2 & 3 \\ 2 & 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 2 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

Figure 2.3a is a visualization of how the convolution operation begins. The kernel is aligned with the top-left corner of the input and it calculates the sum of the Hadamard product of the overlapping region and the kernel. In the figure, this region is indicated by the shaded portion of $\mathbf{X}$ (corresponding values of $\mathbf{K}$ are shown in the subscript).

Let $\mathbf{Y}$ be the output of the convolution operation. Calculating the sum of the Hadamard product of this region, we get the first element of $\mathbf{Y}$:

$$\begin{aligned} y_{0,0} &= (3 \times 0) + (3 \times 1) + (2 \times 2) + (0 \times 2) + (0 \times 2) + (1 \times 0) + (3 \times 0) + (1 \times 1) + (2 \times 2) \\ &= 12 \end{aligned}$$

This is indicated by the shaded portion of the $\mathbf{Y}$ matrix depicted in Figure 2.3a. Figures 2.3b, 2.3c and 2.3d depict the subsequent steps of the convolution operation. With each step, the kernel slides one column to the right (called a stride) and calculates the sum of the Hadamard product and so on. Once the kernel reaches the right edge, it slides a row down and starts over from the left as seen in Figure 2.3d. The convolution operation is finished once the kernel reaches the bottom edge and $\mathbf{Y}$ is obtained. So in essence, the convolutional operation is a weighted moving sum of the input.

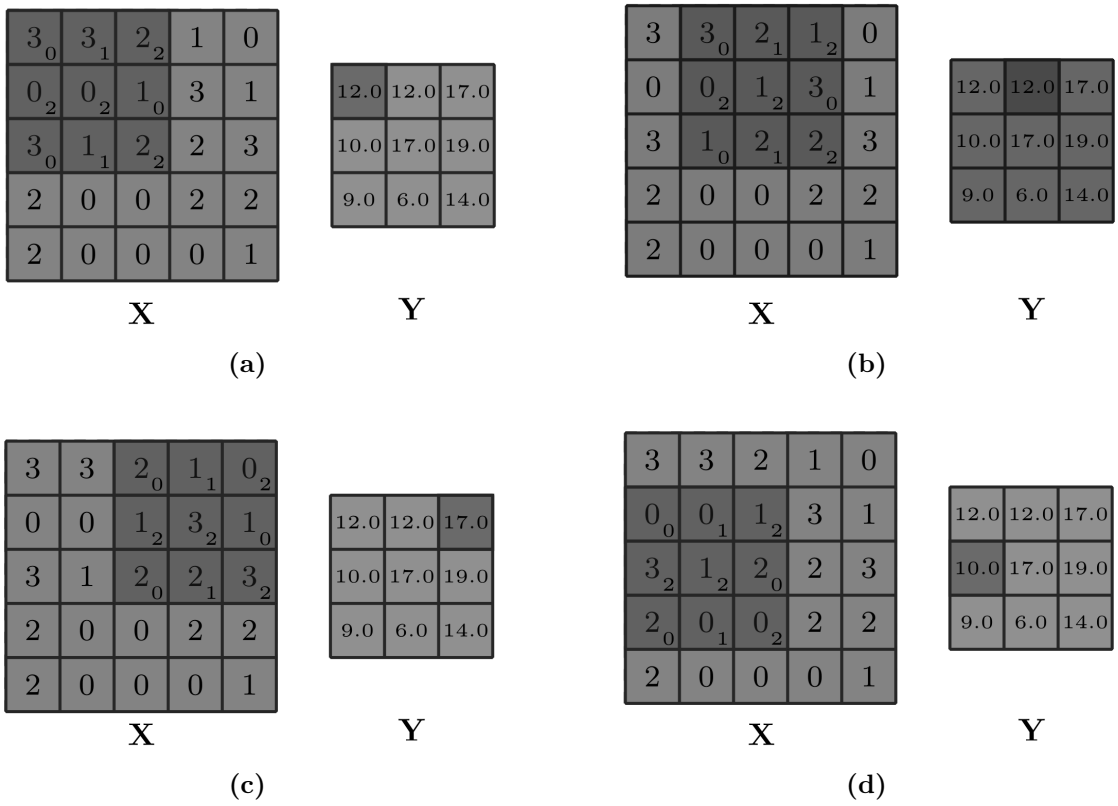


Figure 2.3: The convolution operation

In a CNN, the convolutional operation is usually followed by a max-pooling operation. Max pooling is similar to convolutions in the way that they both extract information from localized spatial regions of the input to produce a matrix. However, max pooling has no kernel associated with it. It merely slides a defined window across a 2D input and outputs the maximum number of that window.

In the context of image classification, the convolutional operation performs visual feature extraction. Convolving a kernel across an image basically entails looking for certain features in that region of the image such as edges and corners. When convolution layers are stacked, it is noticed that the higher convolutional layers start learning complex features like eyes, noses etc. This has led to much success for CNN's in the area of image classification, in which it is now state-of-the-art. Convolutional layers have been traditionally associated with visual imagery tasks but there has been a rise of their use in language models and are starting to show more advantage over recurrent networks as they are much less computationally expensive.

2.2 Word embeddings

As the thesis deals with language modeling, a way of representing words in a mathematical form that captures the meaning of the word and its relationships to other words needs to be devised. There have been several ways of such feature extraction from text such as bag-of-words and discrete encoding. But such methods are arbitrary and indicate none of the relationships that might exist between the words. Say, if the words dog, cat and car were encoded as simply 1, 2 and 3 respectively, the model will have no knowledge about the overarching attributes that dog and cat share (or lack thereof in the case of car and cat). Hence, there needs to be a way to capture all this information in forming word representations.

One way of doing so is through word embeddings. Word embeddings or word vectors (used interchangeably) are a type of word representation that allows words with a similar meaning to have similar numeric vectors (with the notion of similarity between vectors being a real valued number calculated using a function, commonly cosine similarity). Moreover,

another defining property of word embeddings is that word vector algebra is **meaningful**, i.e. addition or subtraction of word vectors results in a word vector that is semantically related to the addends/minuends. For example, if there are four words *man, woman, king* and *queen*, then the embeddings generated for these words should be capable of the following arithmetic:

$$\mathbf{x}_{man} - \mathbf{x}_{king} + \mathbf{x}_{woman} \approx \mathbf{x}_{queen}$$

where $\mathbf{x}_w$ denotes the vector for the word w in the subscript.

The primary intuition driving most methods to generate word embeddings is based on the *distributional hypothesis* which states that words that appear in the same contexts share semantic meaning. A lot of methods use context information and co-occurrence statistics to derive word embeddings. Some contextual embedding models are probabilistic language models and count-based methods like Latent Semantic Analysis.

Word2vec (Mikolov et al., 2013) is a neural probabilistic model created at Google Brain for learning word embeddings using a fully connected network. There are two models of word2vec: skipgram and continuous bag-of-words (CBOW). In both models, there is a shallow neural network with the hidden layer having the size of the desired dimension of the embedding. The difference is in the prediction task: the skipgram approach involves trying to predict the context given a word, whereas it is vice versa for the CBOW which tries to predict a word given the context. In the process of training these prediction tasks, the hidden layer learns the embedding of the words and when the training is over, the weight matrix of the hidden layer contains the embeddings of all the words.

Word2vec does have a shortcoming that it considers a word as an absolute atomic token and might not take into account the different parts of a word. For example, it won't be able to intuitively tell that guesstimate is made of guess and estimate. This shortcoming is addressed in fastText (Bojanowski et al., 2016). fastText is a word embedding model derived from word2vec and was developed at Facebook's AI Research (FAIR) lab. It differs from word2vec in the way that it uses n-grams of the characters in a word as its atomic tokens. For example, in a fastText model the word 'where' will be considered as being composed of smaller n-grams: ('whe', 'her', 'ere'). The fastText embedding for 'where' will be represented

by summarization of its constituent bag of n-gram embeddings; it will be the average of the embeddings of ('whe', 'her', 'ere'). This makes fastText superior to word2vec as it is able to understand how a word is built and able to infer the embedding for any out-of-vocabulary words by decomposing it into n-grams and averaging the embeddings.

2.3 Related work in deep learning for healthcare

Deep learning poses a great potential for application in medicine and healthcare with its superior capability of finding patterns. This potential has not gone unexplored in literature: a Google Scholar search for 'deep learning in healthcare' yields upwards of 250,000 results. It has found applications in various aspects of healthcare from drug discovery to phenotyping.

One area that gathered much interest is biomedical imaging and research in deep learning for medical imaging is dominated by convolutional neural network based architectures. CheXNet, a convolutional neural network adapted for radiology, detected pneumonia in chest X-rays better than practicing radiologists (Rajpurkar et al., 2017). A subset of 420 chest X-rays was used for analyzing the performance of CheXNet against four radiologists and it was found that the F1 score of CheXNet was higher than three of the individual F1 scores of the radiologists and the averaged F1 scores of all the radiologists. Hosseini-Asl et al. (2016) predicted Alzheimer's disease from structural brain MRI scans: it used a pre-trained 3D convolutional autoencoder to detect Alzheimer's disease and mild cognitive impairment from MRI scans and achieved a performance better than the state-of-the-art then. Similarly, Gulshan et al. (2016) detected diabetic retinopathy and diabetic macular edema from retinal fundus photographs. It utilized an InceptionV3 architecture and found that the deep learning approach had a very high sensitivity and specificity and the performance was comparable to ophthalmologists. Convolutional neural networks have shown promise in semantic segmentation of medical images. Ronneberger et al. (2015) devised a fully convolutional network (a CNN with no fully connected layer) called U-Net that could perform segmentation of neuronal structures in electron microscopy images of the brain.

Deep learning in healthcare has also utilized biomedical texts in electronic health records for various tasks. Using free-text radiology head CT reports, Shin et al. (2017) tried to detect

five different types of anomalies like intra-cranial bleed and stroke using a convolutional neural network with an attention mechanism. Culliton et al. (2017) tried to detect sepsis from patient notes by generating and summing all the word embeddings and classifying them. Yang et al. (2017) predicted which anti-hypertensive medications would be prescribed at the time of discharge using nursing notes and certain sections of discharge summaries. Deep learning models have also been used to perform triage in the emergency department: Gligorijevic et al. (2018) combined initial nursing assessment notes with structured data to predict which patients will require a large amount of resources. Deep learning methods have also been used to generate text from images in the medical domain. Shin et al. (2016) taught a multimodal CNN-RNN how to annotate lung X-rays with findings such as calcified granuloma etc.

Though, most deep learning methods utilize imaging or textual data, there have been applications of deep learning on other types of biomedical data: Fakoor et al. (2013) detected cancer using a stacked autoencoder and PCA on gene expression data and Fox et al. (2018) forecasted blood glucose levels using a multi-input model that used a recurrent network. Xu et al. (2015) tried to predict drug-induced liver injury from certain drugs wherein the drugs' molecular structure was encoded as an undirected graph and an RNN was used to map the encoded drugs as being positive or negative for inducing liver injury. Similarly, deep learning is also being used in chemo-informatics for drug discovery (Fleming, 2018).

The electronic health record has garnered much interest as the input data from which predictions should be made on the account of its highly multimodal format. The electronic health record (EHR) is a digital collection of all the health information that has been collected on a patient during their stay and occasionally outside their stay as well. It can include demographic data, vitals data, laboratory results, medical history etc. The integration of data from various sources and modalities ensures a true unbiased picture of the patient's health. The texts compiled in the EHR during a stay can serve as a summarized view of the patient. Boag et al. (2018) notes that due to the difficulty of analyzing data across different sources, the unstructured free text can contain important findings about the patient's state and trajectory. It tries to exploit this lack of structure for assessing how LSTMs would perform on two types of prediction tasks using notes: easy (predicting age, gender etc.) and

hard (diagnosis, readmission). It found that the LSTMs were superior to the baseline on the tasks that required clinical reasoning like diagnosis and length of stay.

However, writing notes can be tedious, especially for the physician. This is acknowledged in Liu (2018) which tries to simplify the process by using a language encoder-decoder model to automatically generate notes conditioned on patient characteristics. It uses demographic data and the past notes, medication and labs data from 24 hours prior to generate free text and finds a remarkable performance and suggests that the model be used for assisting note writing (through auto-completion and error detection) rather than replacing it entirely.

Mortality prediction has always been a prediction task of interest in the machine learning for healthcare community. Generally, machine learning approaches to mortality prediction have been specific to certain conditions like sepsis (Taylor et al., 2016), blood cancers (Verplancke et al., 2008), cardiovascular disease (Austin et al., 2012), prostate cancer (Ngufor et al., 2014), burn injury (Stylianou et al., 2015), coronary artery disease (Motwani et al., 2017) and others. Often, there'd be limitations on the studies deriving these machine learning models, hence the specific nature of these models. But there have been steps towards highly generalized mortality prediction models, especially with the release of de-identified public EHR datasets that can serve as the training data for the development. One such dataset has been Medical Information Mart for Intensive Care - III dataset (Johnson et al., 2016); most of the works cited below utilize MIMIC-III.

There have been multiple studies but 12 of these studies are given special focus as they are the most similar to this thesis. However, it would be disingenuous to compare the performance of these works as they vary significantly in the cohort selection criteria, which often is not fully reproducible. This sentiment is echoed in Johnson et al. (2017) which tries to reproduce 28 published works that deal with mortality prediction and use the MIMIC-III dataset and found that there was an at-most 25% discrepancy in the reproduced cohort from the original in around half the works.

Johnson and Mark (2017) compared the performance of traditional machine learning methods like regularized logistic regression and gradient boosting against the ICU scoring systems for predicting mortality. It extracted and summarized features from the patients vitals, labs and other charted values from two time windows, the first 24 hours after admission

and a random time period in the patient's stay. Its cohort selection served as the base for the cohort selection of this thesis. Tang et al. (2018) utilized a deep learning model to predict mortality from a combination of physiological data, demographic data and ICD coding data.

There has also been interest in assessing the ability of the different medical codes such as ICD codes to predict mortality. Stojanovic et al. (2017) tries to learn vectors for individual disease and procedure codes through word2vec by concatenating the sequential ICD and CPT codes of a patient to form a context 'sentence'. After learning the representations through multiple patient 'sentences', it evaluates the learned vectors for tasks of predicting Inpatient Quality Indicators such as mortality, length of stay and total charges by summing up the vectors of all the codes of a patient and using it as an input to a regression classifier. Sha and Wang (2017) have a similar hierarchical approach in which a single visit's ICD9 codes are mapped to a higher level visit representation and the visit representations are mapped to the outcome of mortality. It uses a gated recurrent unit, a type of recurrent neural network, to perform both the aforementioned mappings, with an attention mechanism on top of both the layers. The attention mechanism provides interpretability to the model; it provides a numeric measure of how important each ICD9 code is to the visit which gives an idea of which ICD9 code is the most responsible for the outcome. The cohort consists of patients with at least two hospital stays so that the higher-level visit representation can be calculated.

Training mortality prediction models as one of the tasks in a multitask learning setup has been gaining traction. Harutyunyan et al. (2017) was one of the first ones: it uses a feature set of 17 physiological time series variables to predict in-hospital mortality collected from the first 48 hours of the patient's stay. In addition to mortality, Harutyunyan et al. (2017) also evaluates the task of predicting decompensation, ICD codes and length of stay. Two types of models are implemented, logistic regression (serving as the baseline) and an LSTM model, and it is found that the LSTM-based model performs much better. Harutyunyan et al. (2017) presents itself as a benchmark and a lot of works build upon it by following the cohort selection at least, to even duplicating the entire experiment with a different proposed architecture such as Song et al. (2018) which uses a "multi-head attention mechanism" to evaluate the four different tasks. However, it has a different approach in the sense that rather than utilizing data from a set period from the patient's stay, it divides the patient

stay into 24 hour 'episodes.' Instead of mapping a patient to an outcome, it rather maps individual and series of episodes to the outcomes; where each episode is a 76-dimensional vector describing the vitals and other data from that episode.

Purushotham et al. (2017) is another work that presents itself as a benchmark and is more recent and more extensive work compared to Harutyunyan et al. (2017). It uses a feature set of 135 variables, both temporal and non-temporal, to predict not only in-hospital mortality, but also short-term (2 and 3 day) and long-term (30 day and 1 year) mortalities. The models used in benchmarking are of three general types: severity of disease scoring systems, Super Learner models and deep learning models with the deep learning models outperforming the rest; the best model architecture consisted of a multi-modal model of LSTM and a fully connected network for temporal and non temporal features respectively.

Multi-task learning is also utilized in Suresh et al. (2018): it trained an LSTM-based model for predicting mortality as a multi-task learning problem, where each task is predicting mortality for a different sub-group of the population. It uses an LSTM autoencoder to learn a dense representation of a sparse input of physiological time series and then finds sub-groups by using a Gaussian mixture model to find clusters in the learnt dense representations. It evaluates the performance of an LSTM to predict mortality across groups versus the entire population and finds that the multi-task LSTM is able to predict mortality better.

There have been very few works that have used notes to predict mortality. Ghassemi et al. (2014) combined static patient features with features derived from performing Latent Dirichlet Allocation on notes to serve as an input to a support vector machine for predicting mortality. Several notes, such as nursing notes and radiology notes, were extracted from a cohort of almost 20,000 and it was found that the static patient features didn't have a predictive power as compared to the notes. Kocbek et al. (2017) also uses notes (nursing notes, specifically) to predict mortality, however the mortality is 30 days after the first 24 hours rather than in-hospital and its cohort is limited to patients with a chronic kidney disease diagnosis (4,381 patients in total). It first extracts certain kinds of features, such Unified Medical Language System Concept Unique Identifiers (CUIs), TF-IDF and unigram/bigram and uses these as input to a regularized logistic regression to predict mortality.

Waudby-Smith et al. (2018) also utilized nursing notes: it explored how sentiment in nursing notes is associated with mortality. It assigned a real-valued sentiment score to each note using the TextBlob package in Python; the scores ranged between -1 and 1 with 1 representing truly positive sentiment, -1 representing negative sentiment and 0 representing neutral sentiment. It then used a logistic regression model to assess the predictive power of sentiment scores in calculating the probability of mortality and found that it was a significant predictor and led to increased performance of the mortality prediction model. Krishnan and Kamath (2018) assesses the power of electrocardiogram reports in the MIMIC-III database to predict mortality. From a cohort of 21,465 patients, it extracts the first ECG report from their stay and converts it into a summarized 100-dimension document embedding using Word2Vec. It then uses an Extreme Learning Machine, a type of feedforward neural network, to map the document embeddings to the outcome of mortality.

Grnarova et al. (2016) is the closest work to this thesis. It uses two hierarchical convolutional feature maps: one at the word level to create a sentence representation and the other on the generated sentence representations to create a patient vector for classification. It also appends an optional note information, such as type of note at the sentence level, and implements target replication to improve classification. However, there are certain aspects in which it differs from the thesis. First, the cohort considers only the first stay of any adult patient and subsequent readmissions are discarded; this leads to a cohort size of 31,244. Second, all the notes from a patient's stay are considered (except for discharge summaries) rather than from a defined time period. Third, all the notes collected in the patient stay are concatenated to form a single note, which causes loss of temporality. Sushil et al. (2017) follows a similar approach: it maps a patient representation to a mortality outcome using concatenated notes of patients with only one admission. Instead of using a convolutional network to learn the patient representation, it does so through either a stacked denoising autoencoder or the doc2vec algorithm. However, it doesn't achieve better performance than Grnarova et al. (2016) on the in-hospital mortality task.

Tables 2.1, 2.2 and 2.3 provide summarized technical details on these 12 studies.

Table 2.1: Cohort details of studies of interest

Study	Cohort size	Mortality
Ghassemi et al. (2014)	19,308	10.9%
Grnarova et al. (2016)	31,244	13.82%
Johnson and Mark (2017)	50,488	Not reported
Kocbek et al. (2017)	4,381	13.4%
Stojanovic et al. (2017)	35mil	4% at-most
Sha and Wang (2017)	7,537	19.3%
Sushil et al. (2017)	27,641	Not reported
Harutyunyan et al. (2017)	33,798	10.63%
Purushotham et al. (2017)	35,567	10.5%
Song et al. (2018)	21,049	13%
Suresh et al. (2018)	32,686	7%
Krishnan and Kamath (2018)	21,465	5%

Table 2.2: The kind of data used in studies of interest

Study	Type of data used	Window
Ghassemi et al. (2014)	Non-discharge notes and admission data	Entire stay
Grnarova et al. (2016)	Non-discharge notes	Entire stay
Johnson and Mark (2017)	Vitals, labs and admission data	First 24 hours
Kocbek et al. (2017)	Nursing notes	Entire stay
Stojanovic et al. (2017)	ICD codes	N/A
Sha and Wang (2017)	ICD codes	N/A
Sushil et al. (2017)	Non-discharge notes	Entire stay
Harutyunyan et al. (2017)	Vitals	First 48 hours
Purushotham et al. (2017)	Vitals, labs and admission data	First 24 hours
Song et al. (2018)	Vitals and labs	First 24 hours
Suresh et al. (2018)	Vitals and labs	First 24 & 48 hours
Krishnan and Kamath (2018)	ECG reports	First report

Table 2.3: Algorithms used in studies of interest

Study	Algorithms used
Ghassemi et al. (2014)	LDA for extracting features from notes and support vector machine for prediction
Grnarova et al. (2016)	Word2vec for pre-training embeddings and convolutional neural network for prediction
Johnson and Mark (2017)	Gradient boosting
Kocbek et al. (2017)	TF-IDF and logistic regression
Stojanovic et al. (2017)	Word2vec and logistic regression
Sha and Wang (2017)	GRU with hierarchical attention
Sushil et al. (2017)	Stacked denoising autoencoder and doc2vec
Harutyunyan et al. (2017)	Multi-task LSTM
Purushotham et al. (2017)	Multi-modal model for joint temporal and static features
Song et al. (2018)	Multi-head attention
Suresh et al. (2018)	Gaussian Mixture Model for clustering, seq2seq autoencoder for learning representations and LSTM for prediction
Krishnan and Kamath (2018)	Word2Vec for pre-training embeddings and Extreme Learning Machine for prediction

Chapter 3

Methodology

This chapter details the comprehensive pipeline for the mortality prediction task which is illustrated in Figure 3.1. It consists of three sequential components: data preparation, experimental setup and modeling. The data preparation module elaborates on preprocessing the input data into a more appropriate form for the model and pre-training the fastText vectors for the notes. The experimental setup describes the experiment design and cohort selection criteria and modeling describes the architecture for the CNN-LSTM model for calculating the mapping between input notes and the mortality outcome.

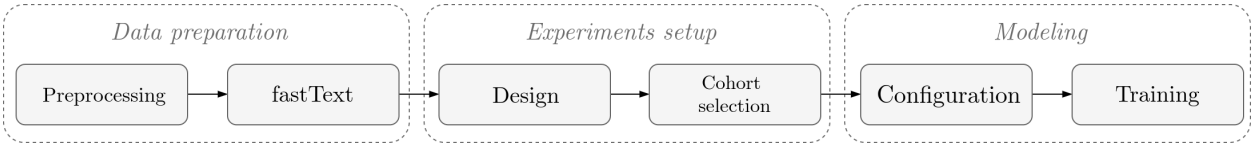


Figure 3.1: The pipeline

3.1 Data preparation

The input notes to the model need to be transformed into a computationally tangible structure that can be processed by the CNN-LSTM model. This involves extracting and preprocessing the notes to remove anomalous and infrequent tokens/characters.

The free-text notes, which serve as the data for this thesis, are sourced from the third iteration of the public medical relational database called "Medical Information Mart for Intensive Care" (MIMIC-III). MIMIC-III (Johnson et al., 2016) comprises information on critical care admissions at Beth Israel Deaconess Medical Center in Boston over a period of 2001-2015. It hosts data from six different types of ICU's: coronary, cardiac surgery, medical, neonatal, surgical and trauma. Data on patients includes vital signs, medications, laboratory measurements, observation notes charted by care providers, fluid balance, procedure codes, diagnostic codes, imaging reports and more. The data has been de-identified in accordance with the Health Insurance Portability and Accountability Act (HIPAA) to remove any personal information such as names, addresses, phone numbers etc. The dates have also been de-identified by producing a random shift to a year between 2100 and 2200, while still preserving seasonality such as time of day, day of week, etc. Access to the MIMIC-III dataset is gained after completion of a course on the ethics of human subjects research. The MIMIC-III dataset is composed of 23 tables stored as comma-separated value files and can be built into a relational database like PostgreSQL or MySQL.

MIMIC-III has 53,423 distinct adult admissions and 7,890 neonate admissions. The adult admissions comprised of 38,597 distinct patients (55.9% male) with a median age of 65.8 and median length of hospital stay of 6.9 days. The three most common International Classification of Diseases (ICD-9) codes for adult patients are related to arterial plaque buildup, sepsis and heart attack ('Coronary atherosclerosis of native coronary artery', 'Unspecified septicemia' and 'Subendocardial infarction, initial episode of care' to be specific). The in-hospital mortality is 11.5%, i.e., 1 in 9 patients died during their hospital stay which presents a significant class imbalance in the dataset.

3.1.1 Preprocessing the notes in MIMIC-III

In MIMIC-III, all the notes are present in the NOTEEVENTS table which has a total 2,083,180 notes from 58,362 unique hospital admissions. There are 11 columns in NOTEEVENTS (detailed in Table 3.1). Notes are of 15 different categories of *'Nursing/other', 'Radiology', 'Nursing', 'ECG', 'Physician', 'Discharge summary', 'Echo', 'Respiratory', 'Nutrition', 'General', 'Rehab Services', 'Social Work', 'Case Management',*

Table 3.1: Columns in NOTEEVENTS

Column	Description
ROW_ID, SUBJECT_ID, HADM_ID, CG_ID	Identifiers for row, patient, hospital stay and caregiver
CHARTDATE, CHARTTIME, STORETIME	Times of charting
CATEGORY, DESCRIPTION	Information on type of note.
ISERROR	If the note is erroneous as identified by a physician

'Pharmacy' and *'Consult'* in decreasing order of frequency. Any identifying information in each note has been replaced with a specific de-identified token, which is usually enclosed in square brackets.

First, any notes that had been identified as having an error (through the ISERROR column) were removed. The notes had three different type of time stamps: CHARTDATE, CHARTTIME and STORETIME. CHARTDATE and CHARTTIME were similar in the way that CHARTTIME narrowed the CHARTDATE down to the minute. The CHARTTIME column seemed the most accurate for basing the period of interest off of but it had a lot of missing data. Therefore, all of the missing data was replaced with the corresponding CHARTDATE with a time of midnight appended to it.

All the notes were then converted to lowercase and processed using regular expressions. First, any multiple line breaks and spaces were replaced by a single one. Then all the de-identified tokens (any token with square brackets and double asterisks on either side) were replaced with the special '<OOV>' token. All punctuation (except periods), tokens consisting of only numbers and other kind of special characters were removed. Hence, it was ensured that any note would contain only alphanumeric characters and periods.

A vocabulary was then defined on the basis of the frequency of the occurrence of the words: any words that occurred less than five times in the entire corpus were not included in

the vocabulary. The notes were processed again to remove any words that weren't present in the vocabulary to form the final corpus of patient files of notes.

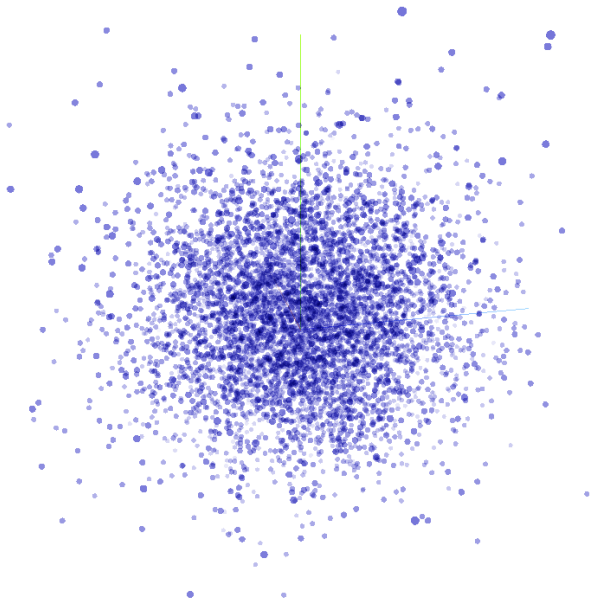
3.1.2 fastText embeddings

It is substantially better to pre-train a word embedding algorithm on the corpus rather than letting the network gradients calculate the embedding values. It saves a considerable amount of computing power during training time and captures relationships between words much better. The fastText algorithm was used to obtain the embeddings of each word in the vocabulary defined earlier, by using the notes corpus as the contexts. The source code for fastText was cloned from GitHub and built into a binary using the `gcc` compiler. The preprocessed notes were saved to a text file `processed_notes` with only the plain text contents of the note and no other meta-data; each note was separated by a line break. This file was supplied as an argument to the main compiled binary, which handled all the tokenization and one-hot encoding for context prediction required. The algorithm was trained with an embedding dimension of 200 with a window size of six for 20 epochs on 12 CPU threads which took about two hours to finish. The trained embeddings for each word were then saved to disk to be visualized for exploratory analysis and loaded into the model later. The following bash command was used to run the algorithm —

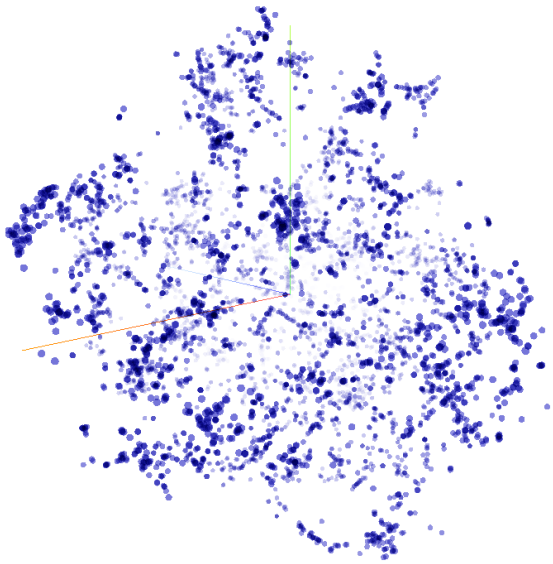
```
$ ./fasttext skipgram -input=processed_notes
-output=outputs/vectors -maxn=5 -dim=200 -ws=6 -epoch=10
```

The embeddings obtained by the fastText model were then assessed by visualizing in 3D space. The t-distributed Stochastic Neighbor Embedding technique for dimensionality reduction was used to plot the trained embeddings in 3D space with learning rate of 1 and perplexity of 43, only using the 6000 most common words (excluding stopwords).

After 1300 iterations, ill-defined local clusters started forming as seen in Figure 3.2. Some of these clusters had a discernible relationship: there was a 'pharmaceutical' cluster where all the words for drugs were clustered together; there was an 'interpersonal' cluster with



(a)



(b)

Figure 3.2: t-SNE visualization at step 0 (a) and step 1300 (b)

words like brother, father etc.; there was a 'dermatology' cluster with words describing skin conditions like bruised, purple, etc.

To demonstrate the quality of embeddings learned, the most similar words to a certain word were highlighted. In Figure 3.3, the word was oxycodone and it is apparent that the most similar words to oxycodone were related to painkillers. In the 3D projection, many of the similar words are clustered around oxycodone but there are some that are far from it. This is because the embedding space has been downsampled to produce a interpretable visualization, so those words may be closer to each other in 200-dimensional space but it isn't reflected in the t-SNE visualization

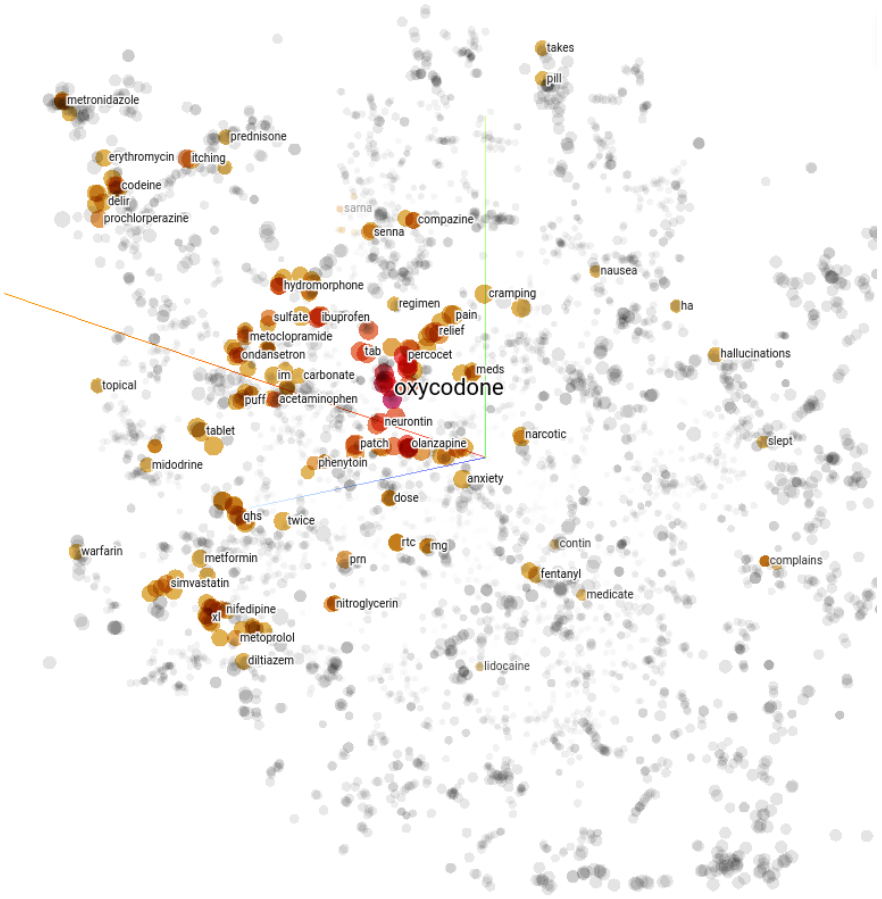
Similar was done for the word 'fracture', as shown in Figure 3.4. A lot of the similar words returned are related to orthopedics. On examining the Figure closely, it is noticeable that there are two clusters that are somewhat interpretable: one relating to the site of injury, such as hip and elbow, and the other relating to the cause like collision and fall.

3.2 Experimental setup

Setting up the experiments included designing experiments and cohort selection. Experiments were designed on the basis of how many hours of data was going to be used. Moreover, a well-defined cohort needed to be determined as MIMIC-III also hosts data on patients for whom the mortality prediction task is inappropriate (neonates) or futile (patients admitted with such severe trauma that death is certain).

3.2.1 Design of experiments

There was one main factor behind the design of experiments, **the period of interest**: it refers to a defined time period in the hospital stay from which the notes shall be extracted. The start of the period is always kept fixed as the admission time but the end point is where experiments differ. The notes corpus shall be built from only this period of interest, i.e., only the notes that had a chart time between the admission time and the end point are to be included in training and evaluating the CNN-LSTM for predicting mortality.



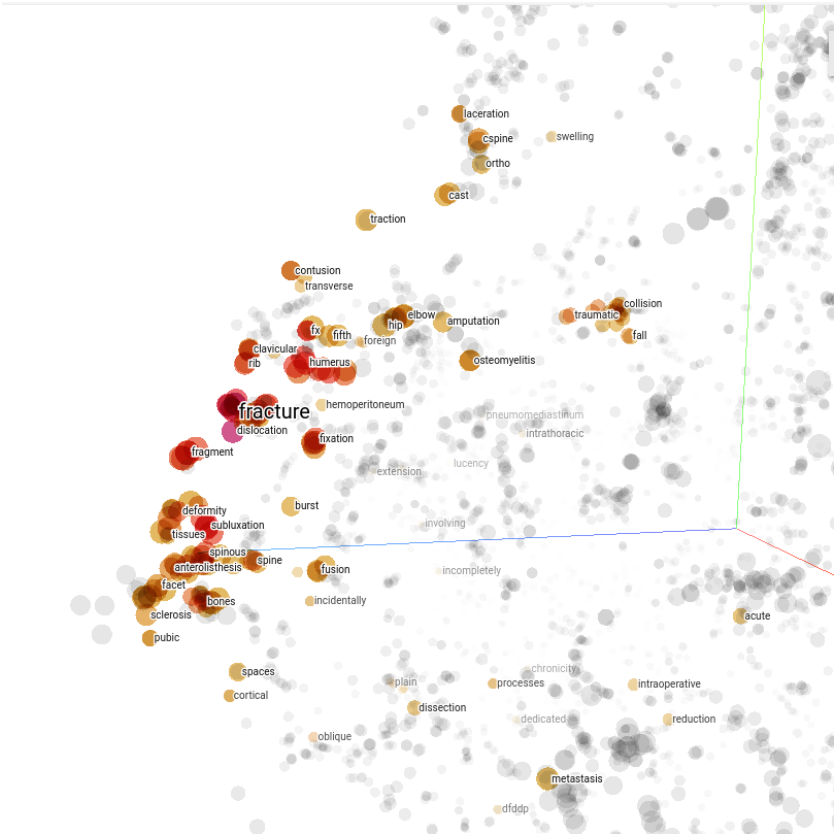
(a)

Nearest points in the original space:

oxycontin	0.196
tramadol	0.240
vicodin	0.277
ultram	0.278
percocet	0.302
morphine	0.307
roxicet	0.333
olanzapine	0.360
neurontin	0.378
methadone	0.393
ibuprofen	0.395
dilaudid	0.401
breakthrough	0.406
ambien	0.410
tab	0.411
acetaminophen	0.421

(b)

Figure 3.3: Highlighting the word **oxycodone** and its closest words in the t-SNE visualization



(a)

Nearest points in the original space:

fractures	0.139
comminuted	0.178
dislocation	0.189
nondisplaced	0.230
displaced	0.271
fractured	0.283
subluxation	0.294
fx	0.304
malalignment	0.326
humerus	0.327
radius	0.339
femur	0.342
humeral	0.348
acetabular	0.354
fibula	0.363
alignment	0.370
dens	0.370
fragment	0.374

(b)

Figure 3.4: Highlighting the word **fracture** and its closest words in the t-SNE visualization

There were three experiments defined: one main and two auxiliary experiments. The main experiment had a period of interest of 24 hours; most of the scoring systems (which shall be used as a baseline for evaluation) calculate mortality using data from only the first 24 hours of the patient stay, so this would ensure a fair comparison between the CNN-LSTM and scoring systems.

Due to the flexible nature of the data required for the CNN-LSTM, an auxiliary experiment with an end point of 12 hours after admission was also defined to see how the CNN-LSTM would perform with less data as compared to scoring systems. Conversely, the other auxiliary experiment had an end point of 48 hours after admission as one of the qualms with scoring systems is that the probability of mortality isn't updated with any data that has been collected after 24 hours.

The experiment names and descriptions are given in Table 3.2

Table 3.2: Experiment details

Name	Experiment type	Time period of interest
CL-24	Main	Within 24 hours of the admission time
CL-12	Auxiliary	Within 12 hours of the admission time
CL-48	Auxiliary	Within 48 hours of the admission time

3.2.2 Cohort selection

The cohort selection is done similar to Johnson and Mark (2017): it involves filtering out patients that had any of the following attributes: (i) An ICU stay of less than 4 hours, or (ii) An age less than 16 years at the time of admission, or (iii) Organ donation as the purpose of visit. This is the primary criteria for creating the **initial cohort**. There are also secondary criteria which is specific to each experiment; different cohorts are then defined for each experiment using the secondary criteria to filter out patients from the initial cohort. Secondary criteria for the experiment-specific cohorts were defined based on the time period of interest. It was based on only one condition: there should be at least one note collected

in the period of interest of that experiment. For example, the CL-12 experiment's cohort consisted of only the patients who had at least a single note charted in the **12 hour period** after their respective admission time. The entire cohort selection, including both primary and secondary criteria, is summarized in Figure 3.5

Table 3.3 details the number of patients in each experiment specific cohort. The cohorts were used to filter the notes as well. Using the initial cohort, the NOTEEVENTS table was reduced to 1,040,029 notes and 49,131 hospital admissions. It was then filtered by removing all discharge summaries and then keeping only the eight most frequent categories of notes, i.e., 'Nursing' ('Nursing/other' was merged into 'Nursing'), 'ECG', 'Radiology', 'Physician', 'Respiratory' and 'Echo'. Three different corpora were formed, one for the three experiments. Each corpus contained only the notes from the experiment's period of interest for only the patients that were in that experiment's cohort.

Table 3.3: Cohort populations

Experiment	Cohort size	In-hospital mortality
CL-24	44,410	4,910 (11.05%)
CL-12	40,254	4,525 (11.2%)
CL-48	46,452	5,126 (11.0%)

3.3 Modeling

In this section, the proposed model architecture for the CNN-LSTM is detailed. The input to the model is a 'patient file': it consists of all the preprocessed notes collected in the period of interest. The patient file is propagated through several layers to produce the output of the model, which is a real-valued number between [0,1] representing the probability of mortality.

3.3.1 Model configuration

The model is composed of four modules in succession, as visualized in Figure 3.6:

1. **An embedding layer**, which performs the embedding lookup

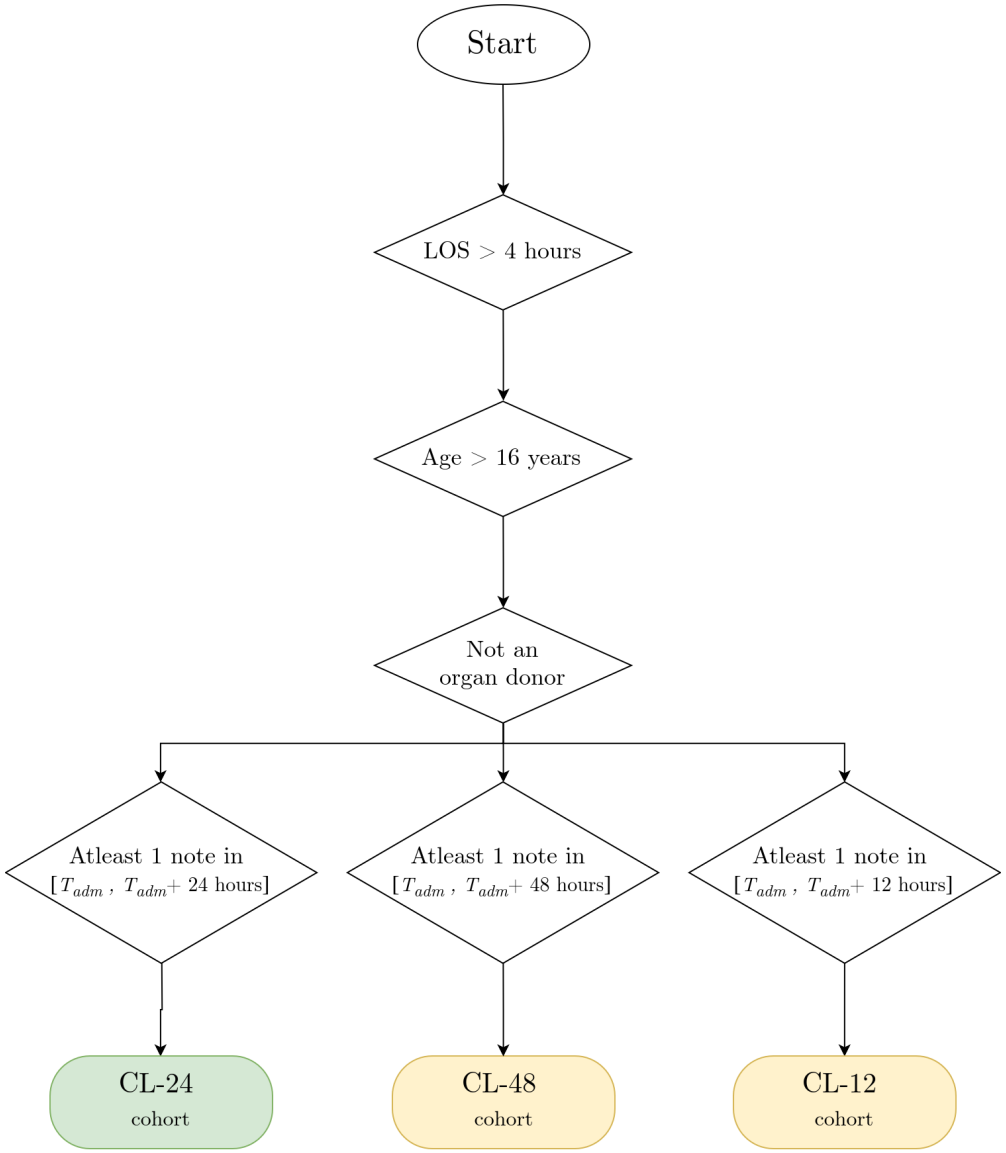


Figure 3.5: Flowchart demonstrating the cohort selection criteria. Any no conditions are discarded and aren't shown for clarity. T_{adm} refers to the admission time. The cohort for the main experiment is shown in green and the auxiliary experiments in yellow

2. **The semantic feature maps**, which consist of a one-dimensional convolutional layer followed by a global max pooling layer

3. **The temporal feature map**, which consists of a single LSTM layer

4. **The classifier**, which consists of fully-connected layers

A hierarchical approach is used to process the patient file: first, a document embedding is generated from applying the semantic feature maps on individual notes from a patient file. Then, the document embeddings are ordered by the ascending time of charting and fed sequentially into the LSTM. The LSTM generates a patient representation by processing the document embeddings one-by-one and assessing the temporal relationships. Finally, the patient representation is fed to the classifier which outputs the probability of mortality.

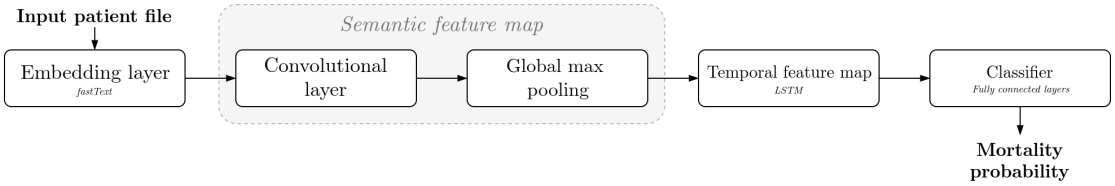


Figure 3.6: Model components

To represent the patient file mathematically, an indexing vocabulary is defined and each word in the vocabulary is mapped to a distinct whole number (called its 'word-index'). The entire patient file is transformed by replacing each word with its respective index and if a certain word is not found in the vocabulary, it is then represented by the special out-of-vocabulary token, '<OOV>' (which is mapped to 0). The '<OOV>' token is also used to pad the input to ensure a uniform data structure of the patient file. So, the input to the model is the matrix $\mathbf{X} \in \mathbb{R}^{T \times W}$ where x_{tw} is the w^{th} word-index of the t^{th} note, T and W are the total number of notes and words respectively and $t \in [0, 1, 2, \ldots, T], w \in [0, 1, 2, \ldots, W]$.

Hereon for the purpose of clarity:

1. *Dimension notation*

   — The words dimension (W) is referred to as the *lexical dimension* since it represents purely the semantics of a note with no information about the time of charting

   — The notes dimension (T) is referred to as the *temporal dimension* since the notes are collected at different points in time and represent temporal changes in the patient.

2. *Mathematical notation*

   — Tensors are denoted by an uppercase letter in italic boldface, e.g. $\boldsymbol{A}$. The usage of the word 'tensor' in this thesis refers to the same as the general mathematical definition of a tensor, but only with a rank of 3 or higher.

   — Matrices (rank-2 tensors) are denoted by an uppercase letter in boldface, e.g. $\mathbf{A}$

   — Vectors (rank-1 tensors) are denoted by a lowercase letter in boldface, e.g. $\mathbf{a}$

The structure of the model is explained in the sections below along with specifics on how each module has been implemented for this thesis.

The embedding layer

The embedding layer is the first layer of the model which performs a simple lookup operation: for each input word-index, it outputs its embedding of size E (which is the chosen dimension for the embedding). The embeddings for all the words in the indexing vocabulary are stored in its weight matrix $\mathbf{Z} \in \mathbb{R}^{V \times E}$, where V is the vocabulary size and each word-index's embedding is the x_{tw}^{th} row of the weight matrix. For example, if the word 'neurological' is mapped to a word-index of 23, then the embedding for 'neurological' will be the 23^{rd} row of $\mathbf{Z}$. This operation can be represented by:

$$\boldsymbol{V} = \ell(\mathbf{Z}, \mathbf{X})$$

$$v_{twe} = z_{\alpha e}$$

$$\text{where } \alpha = x_{tw}, t \in [0, 1, 2, \ldots, T], w \in [0, 1, 2, \ldots, W], e \in$$
$$[0, 1, 2, \ldots, E] \text{ and } \ell \text{ represents the embedding lookup function}$$

The embedding layer essentially replaces the scalar word-indices by their respective embedding, which is obtained through 'selecting' the specific row of the weight matrix $\mathbf{Z}$. Thus, it transforms the input $\mathbf{X}$ into $\boldsymbol{V}$ by adding another dimension of the embedding features: $\boldsymbol{V}$ is a three-dimensional tensor of embedded notes of the dimensions $T \times W \times E$.

Implementation: While the weight matrix of the embedding layer can be learned from the gradients flowing in the network, it is faster and computationally cheaper to learn the embeddings with a word vectorization algorithm beforehand and then load the embedding layer with the 'pre-trained' embeddings. For this thesis, the fastText algorithm was used for obtaining word embeddings with dimension $E = 200$. During the initialization of the model, the embedding layer is loaded with these pre-trained embeddings. The embedding layer is then frozen, i.e., no gradients flow into the layer during parameter updates so that computation during training the model is drastically reduced.

The semantic feature extractors: Conv1D + Global max-pooling

A semantic feature extractor is an abstraction over two subsequent layers, a convolutional layer and a global max pooling layer. After the patient file is embedded into vector space with the embedding layer, one or more semantic feature extractors are applied to it.

The processing of the input by a single feature extractor is described hereon. A one-dimensional convolutional (Conv1D) layer is first applied to the individual notes. A kernel is convolved along the lexical dimension to generate a single feature map, i.e. the kernel is slid along only one direction (as opposed to a two-dimensional convolution). This means that the kernel has the dimension of $K \times E$ where K is the kernel size and E is the embedding dimension. The Conv1D operation basically takes K sequential words at a time, performs the sum of the Hadamard product of their respective embeddings with the kernels and then slides the kernel one word ahead and so on.

Mathematically, it is given below for a single 2D slice of the input 3D tensor:

$$\mathbf{C}^{(1)} = \mathbf{K} * \mathbf{V}^{(1)}$$

$$c_i^{(1)} = \sum_{j=0}^{E} \sum_{i}^{i+K} k_{ij} \cdot v_{ij}^{(1)}$$

The $*$ represents the convolution operation with the slice index denoted by the parenthetical number in the superscript. $\mathbf{V}^{(1)}$ represents the first slice of the input, i.e., the first note and $\mathbf{C}^{(1)}$ is the result of the convolution operation on it. As the one-dimensional convolution operation is the 'sliding' sum of the Hadamard product of the input and the kernel, $\mathbf{C}^{(1)}$ is actually a vector of size $(W - K + 1)$ if there is no padding of the input.

A single vector isn't enough information, so there are multiple feature maps applied in a single convolutional layer with each map (also called a filter) having different kernel weights. The different filters are perceived to be looking for different features in the note but with the same size of grouping of words (the kernel size). So, the dimensions of the quantities involved in the operation change: with multiple filters, $\mathbf{C}^{(1)}$ becomes a matrix of size $(W - K + 1, F)$ where F is the number of filters applied. Each $c_{ij}^{(1)}$ represents a feature extracted from the i_{th} group of K words by the j_{th} feature map in the first note.

As there are multiple slices in the input embedded file, the convolutional layer is applied **time-distributed**: each Conv1D is applied independently on each matrix slice (each note) in the three-dimensional input tensor. Thus, the output of the convolution operation is the three-dimensional tensor $\boldsymbol{C}$ with dimensions $T \times C \times F$ where C is the reduced lexical dimension $(C = W - K + 1)$ and F is the chosen number of filters.

After the convolution operation, a global max-pooling layer is applied to the lexical dimension. Global max pooling is the same as max pooling, however the pooling window is the entire dimension rather than a sliding window. This is done to select the most salient feature present in all the filters to form a document representation. Similar to Conv1D, global max pooling is also applied time-distributed. Thus, this layer down-samples the three-dimensional tensor $\boldsymbol{C}$ to a two-dimensional matrix $\mathbf{D}$ of size (T, F) where each row

represents the document vector for each note. This operation is shown mathematically below:

$$\mathbf{D} = \max_{C} \boldsymbol{C}$$

$$d_{ik} = \max_{j \in C} c_{ijk}$$

$$\text{where } i \in [0, 1, \ldots, T], j \in [0, 1, \ldots, C] \text{ and } k \in [0, 1, \ldots, F]$$

This propagation of the input tensor $\boldsymbol{V}$ through the convolutional and global max pooling layers to get the resultant document representations matrix $\mathbf{D}$ is abstracted as a **single** semantic feature extractor.

Implementation: Three such semantic feature extractors are applied independently to each embedded note for this thesis. Kernel sizes of 3, 5 and 10 are used for each extractor with the feature extractor of kernel size 3 having 128 filters and the latter two having 64 filters. All the kernels are regularized with their respective L2 norms with $\lambda = 0.001$.

So, the function of this module can be interpreted as sequentially taking groups of 3, 5 or 10 words and extracting features from each group and then selecting the numerically largest feature. The outputs of the three feature extractors are concatenated along the filters axis to form a document representation of size 256 for each note.

$$\mathbf{D} = \mathbf{D}_3 \oplus \mathbf{D}_5 \oplus \mathbf{D}_3$$

Where $\oplus$ represents the concatenation operation and the subscript i of each feature extractor output matrix $\mathbf{D}_i$ represents the kernel size. Hence, the output of this module is a patient file of document representations, $\mathbf{D}$, a matrix of size $(T, 256)$.

The temporal feature extractor: LSTM

This module is composed of a long short-term memory RNN which learns the temporal changes, i.e., how the condition of the patient has evolved over time. The document representations produced by the previous layer are then arranged into the order in which they were recorded and each individual document representation is fed to the LSTM sequentially.

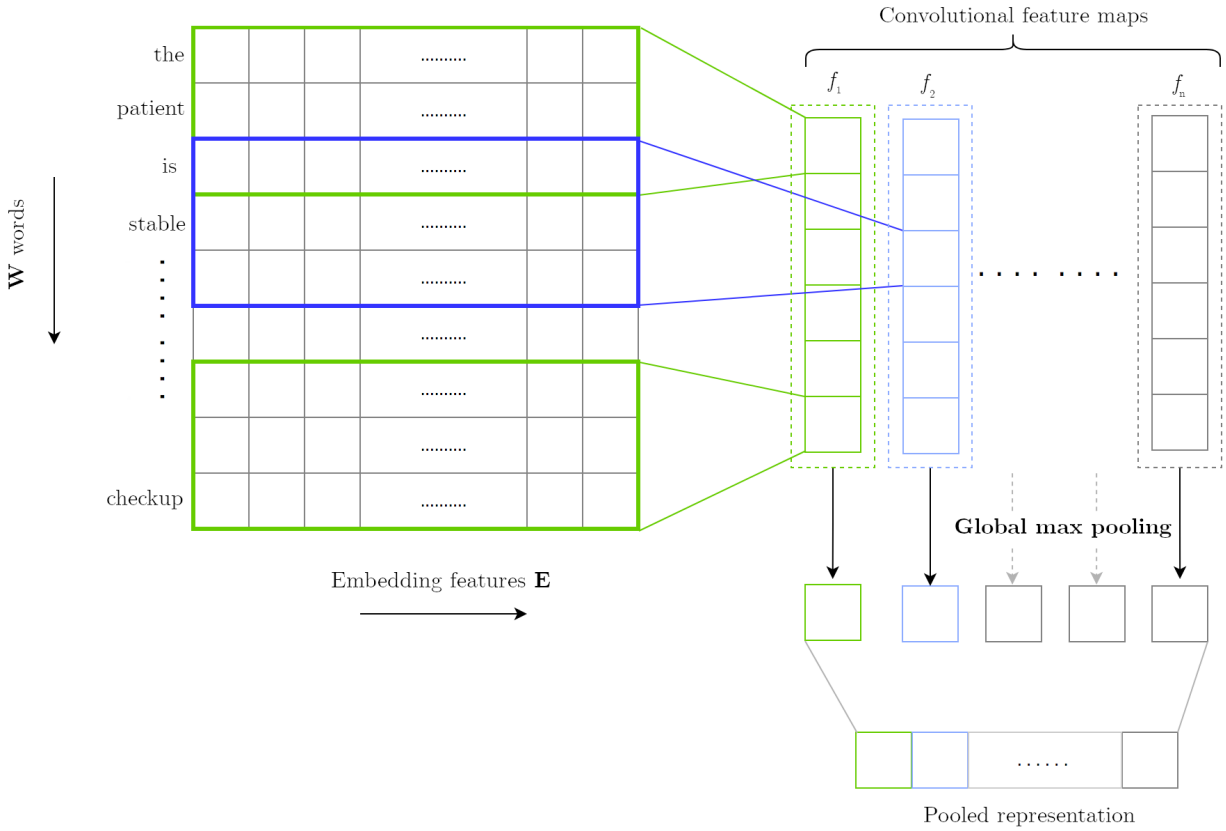


Figure 3.7: Visualization of a single semantic feature extractor. The different colors represent the different convolutional feature maps. The dotted box represents the dimension on which the global max pooling is performed, i.e. on individual feature maps.

The changes in temporal features are recorded in the cell state with each time step of the input. After the final document is processed, the cell state is used to generate a 'patient' vector that represents the condition of the patient at the end of the time period. Thus, this module transforms the input patient file $\mathbf{D}$ into a patient vector $\mathbf{p}$. The patient vector is of the size of the chosen parameter for the cell state of the LSTM.

Implementation: A single LSTM with a cell state size of 64 is used for modeling the hospital stay temporally. The input to the LSTM is regularized using dropout with a keep probability of 0.8.

Classifier: fully-connected layers

This module consists of only fully-connected layer(s) that perform classification on the features extracted by the previous layers. As mentioned in section 2.1, the fully connected layers basically perform a matrix multiplication between the input and their respective weight matrices, then an addition of the bias vector/scalar to the vector/scalar that results from the previous operation followed by a non-linearity. So, the input to this module is the patient representation generated by the prior LSTM and the output is a scalar $\hat{y}$, the probability of mortality.

$$\hat{y} = \sigma(\mathbf{W} \cdot \mathbf{p} + \mathbf{b})$$

Implementation: In this thesis, two sequential fully-connected layers are used with 16 and 1 neurons and activation functions ReLU and sigmoid respectively. Both of these layers are regularized by dropout on their respective inputs and L2 regularization on their weights.

The implementation of this particular configuration of the architecture is illustrated in Figure 3.8.

3.3.2 Training

The notes were sorted by the hospital stay and time and then grouped by hospital stay to form 'patient files' and transformed into numeric representation using the indexing dictionary. There were limits imposed on the patient files: only the first 500 words in a note and only

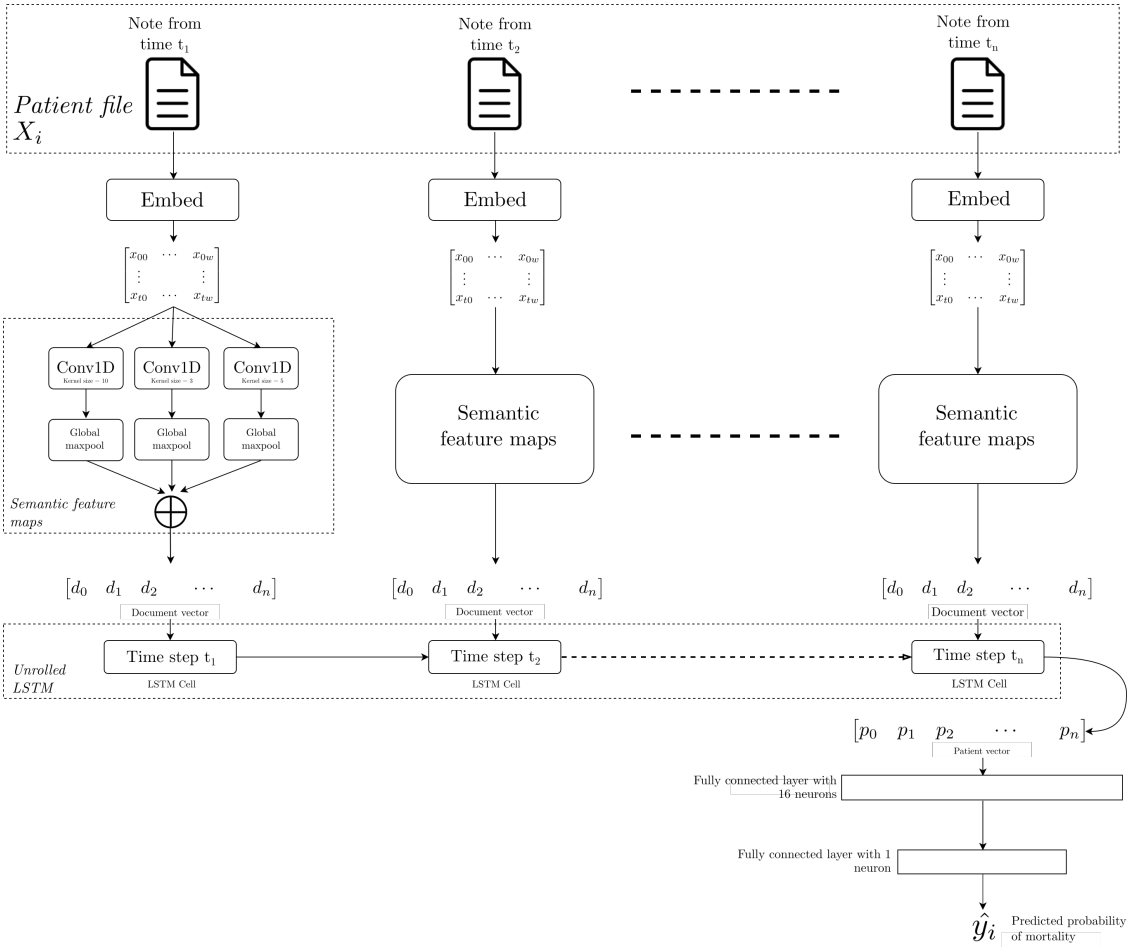


Figure 3.8: Model architecture

the first 8 notes in a file were considered. If there were less than 500 words and/or 8 notes, the patient file was padded with zeros appropriately. This was done to ensure a uniform data structure for the input.

The binary cross-entropy function was used as the loss function for training the model, which is given by:

$$\mathcal{F}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

where n is the number of training samples, θ is the set of all the parameters of the model, $\hat{y}_i$ is the predicted mortality and y_i is the actual mortality.

This loss function is highly suitable for binary classification tasks as it still facilitates parameter updates when the output of a neuron is saturated. The AMSGrad (Reddi et al., 2018) variant of the Adam (Kingma and Ba, 2014) mini-batch optimization algorithm was utilized to find the local optima of this cross entropy function averaged over the training set.

There is a significant class imbalance in the dataset with only 1 in 10 patients dying. Such a class distribution would lead the model to not learn any of the features associated with mortality and simply output 'survive' for all patients. So, the loss function is weighted with the loss for the 'mortal' class being penalized 2.8 times more than normal and the 'survive' class being penalized 0.6 times less than normal. These class weights were obtained from using the `compute_sample_weight` function from the `scikit-learn` package.

The notes were then split into training and test sets and the model was initialized and trained. Training was done on a Google Cloud instance with four vCPUs and an attached NVIDIA Tesla P100 GPU. The model was trained for 15 epochs with a batch size of 64 and an initial learning rate of 0.0002. Both early stopping and a learning rate schedule were implemented, with early stopping monitoring the validation metrics and the learning schedule decaying the learning rate every 10 epochs (since the model was trained only for 15 epochs, the learning rate was decayed only once).

Chapter 4

Analysis and results

The first section of this chapter elaborates the evaluation methods used for assessing the performance of the proposed model. The final two sections list and discuss the results obtained.

4.1 Evaluation schema

The evaluation of the performance of the proposed model on the mortality prediction task was done by comparing metrics of the proposed model to the baseline. This section provides information on the baseline and metrics used in this thesis.

4.1.1 Baseline

The baseline to which the CNN-LSTM model shall be compared to are the mortality probabilities calculated from three different severity-of-disease scoring systems: SAPS II, APS III and OASIS. The authors of Johnson et al. (2016) provide SQL code for calculating and saving these scores. For each experiment, only the scores of the patients in the cohort are extracted from the calculated scores. As these scores are calculated for ICU stays rather than hospital stays, only the score from the first ICU stay is considered if multiple are present. The scores are elaborated below:

SAPS-II

The Simplified Acute Physiology Score II, or SAPS-II (Le Gall et al., 1993), is a severity of disease scoring system that was developed in 1993. It utilized a cohort of around 13,000 patients from 137 ICU's in 12 countries in Europe and North America. The cohort excluded any patients under 18 years and patients undergoing burn treatment, coronary care or cardiac surgery. The score is calculated using 17 variables, 12 of which are physiological and the rest are patient attributes. The variables are: (i) Age (ii) Heart Rate (iii) Systolic Blood Pressure (iv) Temperature (v) Glasgow Coma Scale (vi) Mechanical Ventilation or CPAP (vii) PaO2 (viii) FiO2 (ix) Urine Output (x) Blood Urea Nitrogen (xi) Sodium (xii) Potassium (xiii) Bicarbonate (xiv) Bilirubin (xv) White Blood Cell (xvi) Chronic diseases (xvii) Type of admission.

All the continuous variables are discretized by binning and each bin has an associated whole number 'score'. The final SAPS-II score is produced by summing up all the scores. To calculate the mortality probability, first the logit needs to be calculated. It is calculated by a logistic regression equation whose coefficients were determined through the original study's validation set. The logit equation contains a logarithmic term as the SAPS-II score distribution was skewed. Finally, the probability of in-hospital mortality is calculated by passing the logit through a sigmoid function.

$$\hat{p}_{death} = \frac{e^{logit}}{1+e^{logit}}$$

where $logit = 7.7631 + 0.0737 \times (\text{SAPS II Score}) + 0.9971 \times [\ln(\text{SAPS II Score} + 1)]$

APS-III

The Acute Physiology Score III, or APS-III is a component of the APACHE-III score. Knaus et al. (1991) that represents the physiological state of the patient. APS-III is combined with other patient attributes to calculate final APACHE-III score and produce a mortality probability. The variables involved in calculating the APS-III score are: (i) Glasgow coma scale (ii) heart rate (iii) mean blood pressure (iv) temperature (v) respiration rate

(vi) ventilation/cpap (vii) chronic dialysis (viii) urine output (ix) PaO_2 (x) alveolar-arterial O_2 (xi) hematocrit (xii) WBC (xiii) creatinine (xiv) blood urea nitrogen (xv) sodium (xvi) albumin (xvii) bilirubin (xviii) glucose (xix) pH (xx) pCO_2.

Johnson (2014) calculated the logit required to solely map the APS-III to a hospital mortality probability, reducing the need to extract patient attributes separately. The mortality probability can now be calculated using only the APS-III score by -

$$\hat{p}_{death} = \frac{1}{1+e^{-logit}}$$

$$\text{where } logit = -4.4360 + 0.04726 \times (\text{APS III Score})$$

OASIS

The Oxford Acute Severity of Illness Score, or OASIS (Johnson et al., 2013), is a recently developed severity scoring system. It consists of the following components: (i) Heart rate (ii) GCS (iii) MAP (iv) Temperature (v) Respiratory rate (vi) Ventilation status (vii) Urine output (viii) Elective surgery (ix) Pre-ICU in-hospital length of stay (x) Age.

The mortality probability is obtained by the following equation:

$$\hat{p}_{death} = \frac{1}{1+e^{-logit}}$$

$$\text{where } logit = -6.1746 + 0.1275 \times (\text{OASIS Score})$$

4.1.2 Metrics

The baseline and proposed model are evaluated using two metrics: the area under the ROC curve (AUROC) and the precision-recall curve (AUPRC). To have a robust estimate of the model's performance, k-fold cross validation shall be done with k=5. For each fold, the model is to be evaluated on the test set after every epoch. The SAPS II, APS III and OASIS

mortality probabilities are also obtained only for the test set and their corresponding metrics are calculated. The metrics shall then be averaged per fold and compared.

AUROC

The ROC curve is a fundamental tool for diagnostic test evaluation. In an ROC curve, the true positive rate is plotted against the false positive rate for different cut-off points of a parameter. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The area under the ROC curve (AUROC) is a measure of how well a parameter can distinguish between two diagnostic groups (survival/mortality in this case). A model with high discrimination ability will have high sensitivity and specificity simultaneously, leading to an AUROC closer to 1; the lower the AUROC, the poorer the discrimination.

AUPRC

The area under the precision-recall curve (AUPRC) is a metric obtained by the summation of the integral of the precision-recall Curve. The precision-recall curve represents the trade-off between precision and recall for different thresholds. Precision is a ratio of true positives to predicted positives; in this context, it quantifies how many of the predicted deaths actually resulted in death. Whereas, recall is another ratio involving true positives and false negatives; it quantifies how many actual deaths were predicted as death. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall). The AUROC for an imbalanced dataset tends to be higher than if the dataset was balanced. However, the AUPRC doesnt suffer from this discrepancy, which makes it suitable for binary classification tasks involving highly imbalanced datasets.

4.2 Results

The evaluation results are tabulated below in Tables 4.1 and 4.2. It is noticeable that the CNN-LSTM outperforms all the three severity of scoring systems on predicting mortality in all the experiments.

AUROC

On the main experiment, it achieves an AUROC of 0.85 which is significantly higher than the rest. It achieves an even higher AUROC on the CL-48 experiment and is competitive on the CL-12 experiment (despite using data from a lesser period of time than the scoring systems).

Table 4.1: Average AUROC per fold

Experiment	CNN-LSTM	SAPS II	OASIS	APS III
CL-24	**0.85**	0.81	0.79	0.79
CL-12	**0.82**	0.81	0.78	0.79
CL-48	**0.87**	0.81	0.78	0.78

AUPRC

The AUPRC of the CNN-LSTM is significantly greater than the rest for all three experiments. As the dataset is pretty imbalanced (9 survival cases for every 1 mortal case), this metric demonstrates the higher discrimination and generalization capability of CNN-LSTM for predicting mortality.

Table 4.2: Average AUPRC per fold

Experiment	CNN-LSTM	SAPS II	OASIS	APS III
CL-24	**0.51**	0.41	0.39	0.36
CL-12	**0.45**	0.41	0.39	0.37
CL-48	**0.58**	0.40	0.39	0.36

The results are further examined using the confusion matrix plots in Figure 4.1. These plots were obtained by averaging out the confusion matrices from each fold of the CL-24 experiment and rounding to the closest integer. On assessing the averaged confusion matrix for APS III and OASIS predictions (Figure 4.1c and 4.1d), these scoring methods seem to predict survival very well but they underestimate mortality severely. More than 80% of the time, APS III and OASIS predicted that a fatal patient would survive. SAPS II seems to predict mortality much better (as seen in Figure 4.1b) than APS III and OASIS, however it still has a higher rate of false negatives. It is apparent that these three scoring methods are highly (moderately, in the case of SAPS II) biased towards predicting survival, which is very unhelpful in the task of mortality prediction. The CNN-LSTM (Figure 4.1a) has a lower discrimination for predicting survival but has a significantly higher discrimination for predicting mortality.

4.3 Discussion

A major advantage that the CNN-LSTM possesses in its approach is its ease of setup and adaptability. The severity of diseases scoring systems were all developed with expensive studies on cohorts sometimes spanning multiple countries. The development required manual acquisition of raw data and considerable effort for performing variable selection and ensuring generalizeability. The cost of developing the scores was also compounded with each iteration of updating the scoring system to a newer version. Even after much effort, the latest scoring systems produce instances of failure.

Implementing and adapting the scores is a resource-intensive task as well. Due to the nature of the studies being developed in a certain country, it almost requires calibration when adapted to a new country. The study required for calibrating the score poses another hurdle in the application of the scores for predicting mortality.

The CNN-LSTM approach is a relatively easier method to implement as it only requires notes as an input, as opposed to the varied collection of physiological variables and patient attributes required for the scoring systems. Developing a CNN-LSTM approach can be done by simply extracting the notes from a hospital's internal database while complying with

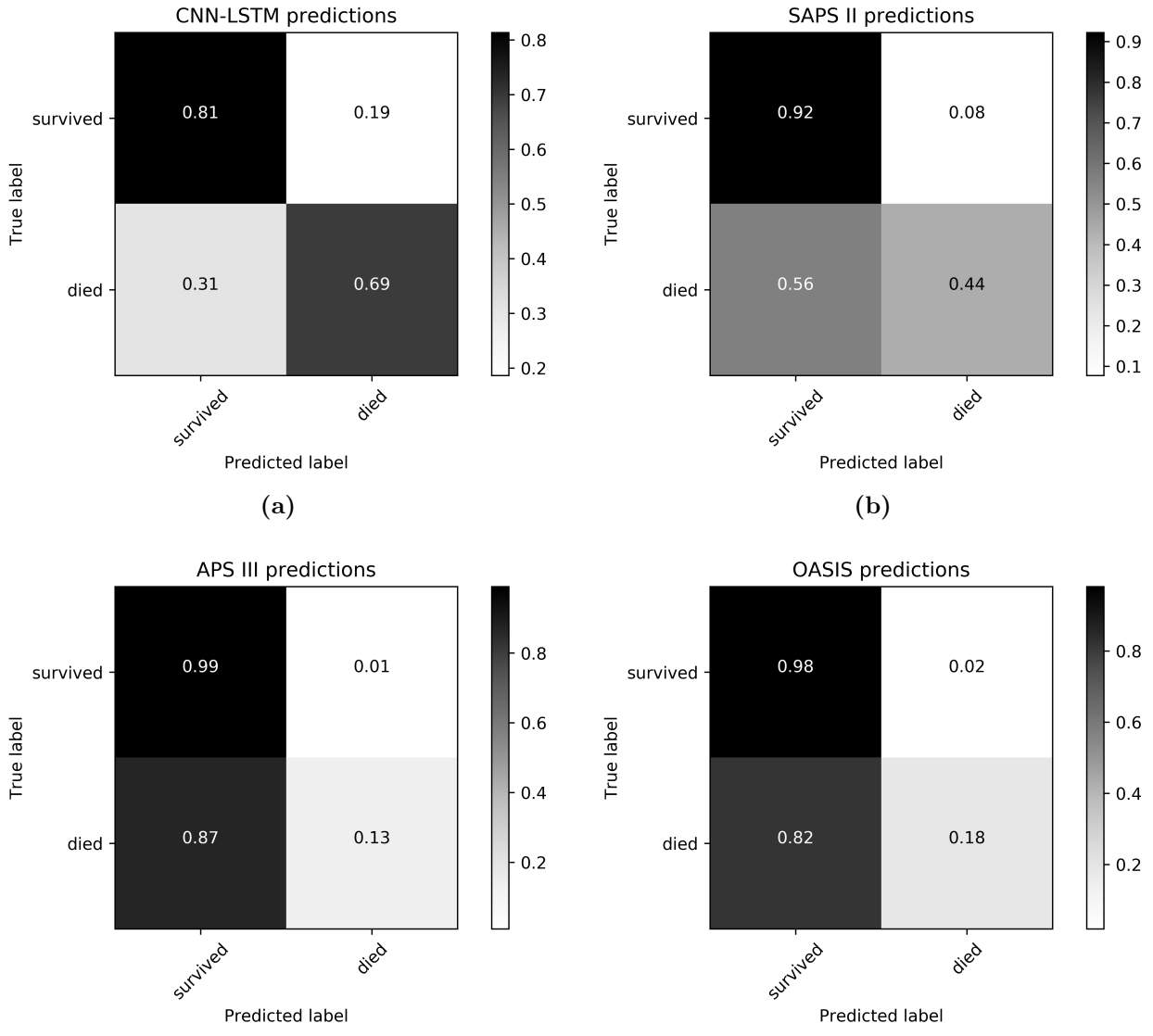


Figure 4.1: Standardized confusion matrices for the different methods for the CL-24 experiment, averaged per fold

patient privacy regulations. The big advantage is that the process for developing a CNN-LSTM can be a **retrospective** study; it would not require conducting an expensive study with complex selection criteria and regulatory compliance protocols. The development of a CNN-LSTM approach would not require extraction & processing of multivariate data which requires statistical specialists and has a risk of being error-prone.

Moreover, CNN-LSTM is highly capable of adapting to different situations through *transfer learning*: in this approach, a trained model is acquired and most of the layers' weights are frozen, except the final layers, and the model is fine-tuned on the new and different data. Transfer learning is a highly attractive attribute of most machine learning models. This means that a hospital can take a CNN-LSTM that has been trained on another hospital's data (in another city, state or even country) or use a public dataset to train their model and then simply fine-tune it on their data.

This is especially useful for hospitals with less data: Desautels et al. (2017) tested transfer learning for mortality prediction by training a boosted ensemble of decision trees on the MIMIC-III dataset with different proportions of private data from UCSF (UCSF was the destination where the model was to be applied). It was found that by using MIMIC-III in combination rather than only the private UCSF data, the amount of training data required to surpass 0.80 AUROC decreases from more than 4000 patients to fewer than 220 patients. Moreover, it decreases the clinical data collection time from approximately six months to less than 10 days.

However, the CNN-LSTM suffers from a lack of interpretability. It is highly unlikely to narrow down what part of the notes is affecting the predicted mortality. The black-box nature of the setup doesn't allow for determining any relationships that might exist between the patient's features and the predicted mortality. If any causal relationships were found, it can greatly augment in phenotyping the patient and help the physician even more in modifying the prognosis.

Chapter 5

Conclusion

A deep learning architecture, CNN-LSTM, for predicting mortality from the patient notes compiled in the initial period of their stay is presented in this thesis. It utilizes the structure-free format of free text which ensures that all the data available on the patient is used in the prediction, rather than limiting to certain pre-set physiological variables. The architecture is hierarchical in nature: a convolutional layer extracts features from embedded notes and feeds these features to a recurrent layer that accounts for temporality in the patient's state. The performance of the CNN-LSTM is compared to the baseline of traditional severity of disease scoring systems SAPS-III, APS-II and OASIS; these scoring systems use static patient features collected in the first 24 hours of a patient's stay.

For an equitable comparison between the CNN-LSTM and the baseline, the CNN-LSTM was trained on notes collected in the first 24 hours of a patient's stay. It achieves an AUROC and an AUPRC higher than the traditional severity of disease scoring systems and unlike the baseline, doesn't underpredict mortality. It achieves a competitive performance with only 12 hours of data without the rigorous extraction and preprocessing required for calculating the scores for the patient. Another experiment is performed with 48 hours of data to demonstrate the ability of the CNN-LSTM to update the mortality probability with newer data (which is a qualm associated with severity of disease scores) and the AUROC and AUPRC show a considerable increase.

The diverse nature of human population can affect prediction methods. ICUs across the world exhibit a broad spectrum of afflictions, ethnicities, ages and disease patterns and

any mortality prediction technique should account for all this variation. Severity of disease scoring systems are developed through studies involving a segment of the population which questions their ability to work for the other segments. Most of the these scoring systems have a localized approach to variables and a global approach to application, i.e., they try to include a finite amount of objective variables that are common and ignore any disease specific attributes to be able to generalize well across different populations. However, many studies that applied the scoring systems to their specific population (especially in Asia) are evidential of the lower than expected generalizeability of the scoring systems. This leads to an impasse: scoring systems need to be augmented with subjective features to improve the performance but those subjective features might degrade the performance for other segments of the population. Such features can also be difficult and/or expensive to obtain which can lead to missing data. When it comes to calculating any kind of outcome, all data on the patient should be taken into account without having to perform manual feature selection. Deep learning models are highly capable of prediction using such unstructured data. Moreover, deep learning models can also be fine-tuned on a hospital's data to reflect the attributes of its population without the need for an expensive study that is generally required for calibration of scoring systems. However, contemporary implementation of deep learning systems is hindered by the lack of interpretability in the models.

Directions for future work

It is possible to better the performance of the CNN-LSTM model on mortality prediction by making changes in the architecture and such changes need to be investigated. The model's interpretability can increase by adding an attention mechanism on top of the LSTM output, but this mechanism will only narrow down which document is important to the prediction. There needs to be a method of implementing attention on the semantic feature maps such that the prediction power can be localized to word-level. Concepts from the emerging area of explainable artificial intelligence (XAI) could be used for the application in this model.

The convolutional maps could be modified by using dilated convolutions which essentially look ahead rather than extracting features from immediate neighboring words. The LSTM

portion could be replaced with a temporal convolution to speed up computation and the use of residual and skip connections could be investigated as well.

Another way is to convert the CNN-LSTM to a multimodal setup by jointly modeling notes with other kinds of data available. This could include admission attributes, various medical imaging done and even the variables used in calculating the scores. However, there should be protocols set up to deal with missing data; some can even be incorporated into the model in the same manner as Lipton et al. (2016) and Che et al. (2016).

Bibliography

Afessa, B., Keegan, M. T., Gajic, O., Hubmayr, R. D., and Peters, S. G. (2005). The influence of missing components of the acute physiology score of APACHE III on the measurement of ICU performance. *Intensive Care Med.*, 31(11):1537–1543. 9

Afessa, B., Keegan, M. T., Mohammad, Z., Finkielman, J. D., and Peters, S. G. (2004). Identifying potentially ineffective care in the sickest critically ill patients on the third ICU day. *Chest*, 126(6):1905–1909. 4

Aggarwal, A. N., Sarkar, P., Gupta, D., and Jindal, S. K. (2006). Performance of standard severity scoring systems for outcome prediction in patients admitted to a respiratory intensive care unit in north india. *Respirology*, 11(2):196–204. 7, 10

Angus, D. C., Barnato, A. E., Linde-Zwirble, W. T., Weissfeld, L. A., Watson, R. S., Rickert, T., Rubenfeld, G. D., and Robert Wood Johnson Foundation ICU End-Of-Life Peer Group (2004). Use of intensive care at the end of life in the united states: an epidemiologic study. *Crit. Care Med.*, 32(3):638–643. 11

Austin, P. C., Lee, D. S., Steyerberg, E. W., and Tu, J. V. (2012). Regression trees for predicting mortality in patients with cardiovascular disease: What improvement is achieved by using ensemble-based methods? *Biom. J.*, 54(5):657–673. 27

Baird, R., MacNab, Y. C., Skarsgard, E. D., and Canadian Pediatric Surgery Network (2008). Mortality prediction in congenital diaphragmatic hernia. *J. Pediatr. Surg.*, 43(5):783–787. 5

Barnato, A. E., McClellan, M. B., Kagay, C. R., and Garber, A. M. (2004). Trends in inpatient treatment intensity among medicare beneficiaries at the end of life. *Health Serv. Res.*, 39(2):363–375. 11

Barrett, M. L., Smith, M. W., A, E., Honigman, L., and Pines, J. M. (2014). Utilization of intensive care services, 2011. HCUP statistical brief #185. Technical report, Agency for Healthcare Research and Quality. 2

Boag, W., Doss, D., Naumann, T., and Szolovits, P. (2018). What's in a note? unpacking predictive value in clinical note representations. *AMIA Jt Summits Transl Sci Proc*, 2017:26–34. 26

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*. 24

Brown, M. C. and Crede, W. B. (1995). Predictive ability of acute physiology and chronic health evaluation II scoring applied to human immunodeficiency virus-positive patients. *Crit. Care Med.*, 23(5):848–853. 8

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. (2016). Recurrent neural networks for multivariate time series with missing values. 64

Chen, L. M., Martin, C. M., Morrison, T. L., and Sibbald, W. J. (1999). Interobserver variability in data collection of the APACHE II score in teaching and community hospitals. *Crit. Care Med.*, 27(9):1999–2004. 10

Cheng, J. Y. (2017). Mortality prediction in status epilepticus with the APACHE II score. *Pediatr. Crit. Care Med.*, 18(4):310–317. 8

Cooper, L. M. and Linde-Zwirble, W. T. (2004). Medicare intensive care unit use: analysis of incidence, cost, and payment. *Critical care medicine*, 32(11):2247–2253. 2

Culliton, P., Levinson, M., Ehresman, A., Wherry, J., Steingrub, J. S., and Gallant, S. I. (2017). Predicting severe sepsis using text from the electronic health record. 26

Curtis, J. R., Cook, D. J., Wall, R. J., Angus, D. C., Bion, J., Kacmarek, R., Kane-Gill, S. L., Kirchhoff, K. T., Levy, M., Mitchell, P. H., and Others (2006). Intensive care unit quality improvement: A "how-to" guide for the interdisciplinary team. *Crit. Care Med.*, 34(1):211–218. 3

Desautels, T., Calvert, J., Hoffman, J., Mao, Q., Jay, M., Fletcher, G., Barton, C., Chettipally, U., Kerem, Y., and Das, R. (2017). Using transfer learning for improved

mortality prediction in a Data-Scarce hospital setting. *Biomed. Inform. Insights*, 9:1178222617712994. 61

Donabedian, A. (1988). The quality of care. how can it be assessed? *JAMA*, 260(12):1743–1748. 3

Fakoor, R., Ladhak, F., Nazi, A., and Huber, M. (2013). Using deep learning to enhance cancer diagnosis and classification. In *ICML Workshop on the Role of Machine Learning in Transforming Healthcare (WHEALTH '13)*. 26

Féry-Lemonnier, E., Landais, P., Loirat, P., Kleinknecht, D., and Brivet, F. (1995). Evaluation of severity scoring systems in ICUs—translation, conversion and definition ambiguities as a source of inter-observer variability in apache II, SAPS and OSF. *Intensive Care Med.*, 21(4):356–360. 10

Fleming, N. (2018). How artificial intelligence is changing drug discovery. *Nature*, 557(7707):S55–S57. 26

Floege, J., Gillespie, I. A., Kronenberg, F., Anker, S. D., Gioni, I., Richards, S., Pisoni, R. L., Robinson, B. M., Marcelli, D., Froissart, M., and Eckardt, K.-U. (2015). Development and validation of a predictive mortality risk score from a european hemodialysis cohort. *Kidney Int.*, 87(5):996–1008. 5

Fox, I., Ang, L., Jaiswal, M., Pop-Busui, R., and Wiens, J. (2018). Deep Multi-Output forecasting: Learning to accurately predict blood glucose trajectories. 26

Ghassemi, M., Naumann, T., Doshi-Velez, F., Brimmer, N., Joshi, R., Rumshisky, A., and Szolovits, P. (2014). Unfolding physiological state: Mortality modelling in intensive care units. *KDD*, 2014:75–84. 29, 31, 32, 33

Ghorbani, M., Ghaem, H., Rezaianzadeh, A., Shayan, Z., Zand, F., and Nikandish, R. (2017). A study on the efficacy of APACHE-IV for predicting mortality and length of stay in an intensive care unit in iran. *F1000Res.*, 6. 8

Gligorijevic, D., Stojanovic, J., Satz, W., Stojkovic, I., Schreyer, K., Del Portal, D., and Obradovic, Z. (2018). Deep attention model for triage of emergency department patients. 26

Goldhill, D. R. and Withington, P. S. (1996). Mortality predicted by APACHE II. the effect of changes in physiological values and post-ICU hospital mortality. *Anaesthesia*, 51(8):719–723. 9, 10

Grnarova, P., Schmidt, F., Hyland, S. L., and Eickhoff, C. (2016). Neural document embeddings for intensive care patient mortality prediction. 30, 31, 32, 33

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L., and Webster, D. R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410. 25

Haaland, O. A., Lindemark, F., Flaatten, H., Kvåle, R., and Johansson, K. A. (2014). A calibration study of SAPS II with norwegian intensive care registry data. *Acta Anaesthesiol. Scand.*, 58(6):701–708. 7

Halpern, N. A., Bettes, L., and Greenstein, R. (1994). Federal and nationwide intensive care units and healthcare costs: 1986-1992. *Critical care medicine*, 22(12):2001–2007. 2

Halpern, N. A., Goldman, D. A., Tan, K. S., and Pastores, S. M. (2016). Trends in critical care beds and use among population groups and medicare and medicaid beneficiaries in the united states: 2000-2010. *Crit. Care Med.*, 44(8):1490–1499. 2, 11

Halpern, N. A. and Pastores, S. M. (2010). Critical care medicine in the united states 2000-2005: an analysis of bed numbers, occupancy rates, payer mix, and costs. *Crit. Care Med.*, 38(1):65–71. 2

Halpern, N. A., Pastores, S. M., and Greenstein, R. J. (2004). Critical care medicine in the united states 1985-2000: an analysis of bed numbers, use, and costs. *Crit. Care Med.*, 32(6):1254–1259. 2, 11

Haniffa, R., Isaam, I., De Silva, A. P., Dondorp, A. M., and De Keizer, N. F. (2018). Performance of critical care prognostic scoring systems in low and middle-income countries: a systematic review. *Crit. Care*, 22(1):18. 7, 9

Harutyunyan, H., Khachatrian, H., Kale, D. C., and Galstyan, A. (2017). Multitask learning and benchmarking with clinical time series data. 28, 29, 31, 32, 33

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. 19

Hosseini-Asl, E., Gimel'farb, G., and El-Baz, A. (2016). Alzheimer's disease diagnostics by a deeply supervised adaptable 3D convolutional network. 25

Johnson, A. E. (2014). *Mortality prediction and acuity assessment in critical care*. PhD thesis, University of Oxford. 56

Johnson, A. E., Kramer, A. A., and Clifford, G. D. (2013). A new severity of illness scale using a subset of acute physiology and chronic health evaluation data elements shows comparable predictive accuracy. *Critical care medicine*, 41(7):1711–1718. 56

Johnson, A. E. and Mark, R. G. (2017). Real-time mortality prediction in the intensive care unit. In *AMIA Annual Symposium Proceedings*, volume 2017, page 994. American Medical Informatics Association. 27, 31, 32, 33, 42

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035. 27, 35, 54

Johnson, A. E. W., Pollard, T. J., and Mark, R. G. (2017). Reproducibility in critical care: a mortality prediction case study. In Doshi-Velez, F., Fackler, J., Kale, D., Ranganath, R., Wallace, B., and Wiens, J., editors, *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 361–376, Boston, Massachusetts. PMLR. 27

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 53

Knaus, W. A., Wagner, D. P., Draper, E. A., Zimmerman, J. E., Bergner, M., Bastos, P. G., Sirio, C. A., Murphy, D. J., Lotring, T., Damiano, A., et al. (1991). The apache iii prognostic system: risk prediction of hospital mortality for critically iii hospitalized adults. *Chest*, 100(6):1619–1636. 55

Kocbek, P., Fijačko, N., Zorman, M., Kocbek, S., and Štiglic, G. (2017). Improving mortality prediction for intensive care unit patients using text mining techniques. In *Proceedings of SiKDD 2017 Conference on Data Mining and Data Warehouses*. 29, 31, 32, 33

Krishnan, G. S. and Kamath, S. S. (2018). A supervised learning approach for ICU mortality prediction based on unstructured electrocardiogram text reports. In *Natural Language Processing and Information Systems*, pages 126–134. Springer International Publishing. 30, 31, 32, 33

Le Gall, J.-R., Lemeshow, S., and Saulnier, F. (1993). A new simplified acute physiology score (saps ii) based on a european/north american multicenter study. *Jama*, 270(24):2957–2963. 55

Lee, K. H., Hui, K. P., and Tan, W. C. (1993). Thrombocytopenia in sepsis: a predictor of mortality in the intensive care unit. *Singapore Med. J.*, 34(3):245–246. 9

Lipton, Z. C., Kale, D., and Wetzel, R. (2016). Directly modeling missing data in sequences with RNNs: Improved classification of clinical time series. In Doshi-Velez, F., Fackler, J., Kale, D., Wallace, B., and Wiens, J., editors, *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pages 253–270, Children's Hospital LA, Los Angeles, CA, USA. PMLR. 64

Liu, P. J. (2018). Learning to write notes in electronic health records. 27

Liu, W.-Y., Lin, S.-G., Zhu, G.-Q., Van Poucke, S., Braddock, M., Zhang, Z., Mao, Z., Shen, F.-X., and Zheng, M.-H. (2016). Establishment and validation of GV-SAPS II scoring system for Non-Diabetic critically ill patients. *PLoS One*, 11(11):e0166085. 9

Livingston, B. M., Mackenzie, S. J., MacKirdy, F. N., and Howie, J. C. (2000). Should the pre-sedation glasgow coma scale value be used when calculating acute physiology and chronic health evaluation scores for sedated patients? scottish intensive care society audit group. *Crit. Care Med.*, 28(2):389–394. 10

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Byers, A. H. (2011). *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. McKinsey. 4

Menden, M. P., Iorio, F., Garnett, M., McDermott, U., Benes, C. H., Ballester, P. J., and Saez-Rodriguez, J. (2013). Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties. *PLoS one*, 8(4):e61318. 3

Mendez-Tellez, P. A. and Dorman, T. (2005). Predicting patient outcomes, futility, and resource utilization in the intensive care unit: the role of severity scoring systems and general outcome prediction models. *Mayo Clin. Proc.*, 80(2):161–163. 5, 11

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119. 24

Motwani, M., Dey, D., Berman, D. S., Germano, G., Achenbach, S., Al-Mallah, M. H., Andreini, D., Budoff, M. J., Cademartiri, F., Callister, T. Q., Chang, H.-J., Chinnaiyan, K., Chow, B. J. W., Cury, R. C., Delago, A., Gomez, M., Gransar, H., Hadamitzky, M., Hausleiter, J., Hindoyan, N., Feuchtner, G., Kaufmann, P. A., Kim, Y.-J., Leipsic, J., Lin, F. Y., Maffei, E., Marques, H., Pontone, G., Raff, G., Rubinshtein, R., Shaw, L. J., Stehli, J., Villines, T. C., Dunning, A., Min, J. K., and Slomka, P. J. (2017). Machine learning for prediction of all-cause mortality in patients with suspected coronary artery disease: a 5-year multicentre prospective registry analysis. *Eur. Heart J.*, 38(7):500–507. 27

Mullins, P. M., Goyal, M., and Pines, J. M. (2013). National growth in intensive care unit admissions from emergency departments in the united states from 2002 to 2009. *Acad. Emerg. Med.*, 20(5):479–486. 2

Ngufor, C., Wojtusiak, J., Hooker, A., Oz, T., and Hadley, J. (2014). Extreme logistic regression: A large scale learning algorithm with application to prostate cancer mortality prediction. In *FLAIRS Conference*. aaai.org. 27

Olah, C. (2015). Understanding lstm networks. 19, 20

Oliveira, V. M. d., Brauner, J. S., Rodrigues Filho, E., Susin, R. G. A., Draghetti, V., Bolzan, S. T., and Vieira, S. R. R. (2013). Is SAPS 3 better than APACHE II at predicting mortality in critically ill transplant patients? *Clinics*, 68(2):153–158. 7, 8

Pappachan, J. V., Millar, B., Bennett, E. D., and Smith, G. B. (1999). Comparison of outcome from intensive care admission after adjustment for case mix by the APACHE III prognostic system. *Chest*, 115(3):802–810. 7

Park, B. S., Yoon, J. S., Moon, J. S., Won, K. C., and Lee, H. W. (2013). Predicting mortality of critically ill patients by blood glucose levels. *Diabetes Metab. J.*, 37(5):385–390. 9

Park, S.-K., Chun, H.-J., Kim, D.-W., Im, T.-H., Hong, H.-J., and Yi, H.-J. (2009). Acute physiology and chronic health evaluation II and simplified acute physiology score II in predicting hospital mortality of neurosurgical intensive care unit patients. *J. Korean Med. Sci.*, 24(3):420–426. 5

Purushotham, S., Meng, C., Che, Z., and Liu, Y. (2017). Benchmark of deep learning models on large healthcare MIMIC datasets. 29, 31, 32, 33

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M. P., and Ng, A. Y. (2017). CheXNet: Radiologist-Level pneumonia detection on chest X-Rays with deep learning. 3, 25

Reddi, S. J., Kale, S., and Kumar, S. (2018). On the convergence of adam and beyond. In *International Conference on Learning Representations*. 53

Reddy, C. K. and Sun, J. (2013). Big data analytics for healthcare. In *Tutorial presentation at the SIAM International Conference on Data Mining, Austin, TX*. achc.org.co. 4

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. 25

Russell, J. A. (2015). Assessment of severity of illness. In Hall, J. B., Schmidt, G. A., and Kress, J. P., editors, *Principles of Critical Care, 4e*, chapter 13. McGraw-Hill Education, New York, NY. 5

Sha, Y. and Wang, M. D. (2017). Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, ACM-BCB '17, pages 233–240, New York, NY, USA. ACM. 28, 31, 32, 33

Shin, B., Chokshi, F. H., Lee, T., and Choi, J. D. (2017). Classification of radiology reports using neural attention models. 25

Shin, H.-C., Roberts, K., Lu, L., Demner-Fushman, D., Yao, J., and Summers, R. M. (2016). Learning to read chest X-Rays: Recurrent neural cascade model for automated image annotation. 26

Song, H., Rajan, D., Thiagarajan, J. J., and Spanias, A. (2018). Attend and diagnose: Clinical time series analysis using attention models. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*. 28, 31, 32, 33

Stojanovic, J., Gligorijevic, D., Radosavljevic, V., Djuric, N., Grbovic, M., and Obradovic, Z. (2017). Modeling healthcare quality via compact representations of electronic health records. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 14(3):545–554. 28, 31, 32, 33

Stylianou, N., Akbarov, A., Kontopantelis, E., Buchan, I., and Dunn, K. W. (2015). Mortality risk prediction in burn injury: Comparison of logistic regression with machine learning approaches. *Burns*, 41(5):925–934. 27

Suresh, H., Gong, J. J., and Guttag, J. (2018). Learning tasks for multitask learning: Heterogenous patient populations in the ICU. 29, 31, 32, 33

Sushil, M., Šuster, S., Luyckx, K., and Daelemans, W. (2017). Unsupervised patient representations from clinical notes with interpretable classification decisions. 30, 31, 32, 33

Tang, F., Xiao, C., Wang, F., and Zhou, J. (2018). Predictive modeling in urgent care: a comparative study of machine learning approaches. *Jamia Open*, 1(1):87–98. 28

Taylor, R. A., Pare, J. R., Venkatesh, A. K., Mowafi, H., Melnick, E. R., Fleischman, W., and Hall, M. K. (2016). Prediction of in-hospital mortality in emergency department patients with sepsis: A local big Data-Driven, machine learning approach. *Acad. Emerg. Med.*, 23(3):269–278. 27

Thijs, L. G. and Members of the Task Force European Society of Intensive Care Medicine (1997). Continuous quality improvement in the ICU: general guidelines. *Intensive Care Med.*, 23(1):125–127. 3

Velissaris, D., Karanikolas, M., Flaris, N., Fligou, F., Marangos, M., and Filos, K. S. (2012). Commonly used severity scores are not good predictors of mortality in sepsis from severe leptospirosis: a series of ten patients. *Crit. Care Res. Pract.*, 2012:532376. 8

Verplancke, T., Van Looy, S., Benoit, D., Vansteelandt, S., Depuydt, P., De Turck, F., and Decruyenaere, J. (2008). Support vector machine versus logistic regression modeling for prediction of hospital mortality in critically ill patients with haematological malignancies. *BMC Med. Inform. Decis. Mak.*, 8:56. 27

Wallace, D. J., Angus, D. C., Seymour, C. W., Barnato, A. E., and Kahn, J. M. (2015). Critical care bed growth in the united states. a comparison of regional and national trends. *Am. J. Respir. Crit. Care Med.*, 191(4):410–416. 2

Wang, T., Liu, Z., Wang, Z., Duan, M., Li, G., Wang, S., Li, W., Zhu, Z., Wei, Y., Christiani, D. C., Li, A., and Zhu, X. (2014). Thrombocytopenia is associated with acute respiratory distress syndrome mortality: an international study. *PLoS One*, 9(4):e94124. 9

Waudby-Smith, I. E. R., Tran, N., Dubin, J. A., and Lee, J. (2018). Sentiment in nursing notes as an indicator of out-of-hospital mortality in intensive care patients. *PLoS One*, 13(6):e0198687. 29

Xu, Y., Dai, Z., Chen, F., Gao, S., Pei, J., and Lai, L. (2015). Deep learning for Drug-Induced liver injury. *J. Chem. Inf. Model.*, 55(10):2085–2093. 26

Yang, Y., Xie, P., Gao, X., Cheng, C., Li, C., Zhang, H., and Xing, E. (2017). Predicting discharge medications at admission time based on deep learning. 26

Vita

Mohammad Hashir Khan was born on the 18th of May, 1994 and raised in the Middle East. He completed his undergraduate from Delhi Technological University (formerly Delhi College of Engineering) in the summer of 2016 and later joined the University of Tennessee as a graduate student. He worked as a Graduate Research Assistant in the Center for Advanced Systems Research and Education (CASRE) under the supervision of Dr. Rapinder Sawhney. He will graduate with a Master of Science degree in Industrial Engineering with minors in Computational Sciences and Statistics.