Patient data representation for outcome prediction of congestive heart failure patients

Computer science and engineering, 30 credits

Halmstad 2019-10-29
Nandhini Subramanyan, Ranjani Subramanyan
Nandhini Subramanyan
Ranjani Subramanyan: Patient data representation for outcome prediction of congestive heart failure patients ©
ABSTRACT

Artificial Intelligence (AI) has its roots in every field in present scenario. Healthcare is one of the sectors where AI is reaching considerable growth in recent years. Tremendous increase in healthcare data availability and considerable growth in big data analytic methods has paved way for success of AI in healthcare and research is being driven towards improvement in quality of service. Healthcare data is stored in the form of Electronic Health Records (EHR) which consists of temporally ordered patient information. There are many challenges with EHR data like heterogeneity, missing values, biases, noise, temporality etc. This master thesis focuses on addressing the problem of visit level irregularity which refers to irregular timing between events (patient’s visits).

In order to handle visit level irregularity, a multi layer perceptron (MLP) model with gating mechanisms (highway MLP) has been used. With the help of experiments conducted on Medical Information Mart for Intensive Care-III (MIMIC-III) dataset and results obtained, it is shown that visit level irregularity influences the clinical outcome prediction. It is shown that for handling two visits of a patient, highway MLP performed almost same as MLP models with time information used as feature. However, highway MLP model turns out to be a simpler model than MLP models in terms of computational complexity.
ACKNOWLEDGEMENTS

We would like to thank our supervisors Slawomir Nowaczyk, Awais Ashfaq and Alexander Galozy for their deep insights. We extend our special gratitude to Awais Ashfaq for his regular supervision meetings and guiding us in right direction.

We thank our family and friends, who have supported us through all odds. We would also like to extend our thanks to our study group (team beku) who have made this two years of Master’s degree journey into a memorable one.

Thank You,
Nandhini Subramanyan,
Ranjani Subramanyan
# CONTENTS

List of Figures ix  
List of Tables xi  
Listings xii  
Acronyms xiii  

## 1 INTRODUCTION  
1.1 Motivation and purpose 1  
1.2 Problem definition 2  
1.3 Research question and contribution 3  

## 2 BACKGROUND  
2.1 Literature Review 5  
2.1.1 Patient data representation 5  
2.1.2 Models 6  
2.1.3 Evaluation 6  
2.1.4 Addressing irregularity in EHR data 6  
2.2 Theory 7  
2.2.1 Random forest 7  
2.2.2 XGBoost 7  
2.2.3 Multi layer perceptron 7  
2.2.4 Long short term memory 8  
2.2.5 Highway networks 9  
2.2.6 Med2Vec architecture 10  
2.2.7 Class imbalance 11  

## 3 DATASET - MIMIC-III  
3.1 Overview of MIMIC-III 13  
3.2 CHF patients statistics 14  

## 4 METHODOLOGY  
4.1 Setting up MIMIC-III 17  
4.2 Input features 17  
4.2.1 Hand-picked features (based on literature) 17  
4.2.2 Readmission prediction features 19  
4.2.3 Diagnosis and medication codes 20  
4.2.4 Time gap 20  
4.3 Data representation 22  
4.4 Prediction outcomes 22  
4.4.1 Hospital readmission 22  
4.4.2 Length of stay 22  
4.4.3 In-hospital mortality 23  
4.5 Single visit models 23  
4.5.1 Random forest classifier 23  
4.5.2 XGBoost classifier 24  
4.5.3 Neural network 24  
4.5.4 Channel-wise neural network 24
4.6 Dimension reduction  25
4.7 Two visit models  26
  4.7.1 Channel-wise neural network  26
  4.7.2 Highway models  28
  4.7.3 Long short term memory  30
4.8 Handling class imbalance  31
4.9 Neural network parameters and learning  32
4.10 Evaluation criteria  32
5 RESULTS AND DISCUSSION  35
  5.1 Model performance  35
    5.1.1 Single visit model  35
    5.1.2 Two visit models  37
    5.1.3 Time significance  38
    5.1.4 Time input in different number of categories  39
  5.2 Model interpretability  40
    5.2.1 Highway model learning  40
    5.2.2 Role of time input in prediction outcomes  41
    5.2.3 Model reliability  45
    5.2.4 Mean of features to learn about model’s decision  47
  5.3 Computational complexity  49
  5.4 Summary  50
  5.5 Research answer  50
6 CONCLUSION  53
  6.1 Conclusion  53
  6.2 Limitations  53
  6.3 Discussion  54
  6.4 Future work  54

BIBLIOGRAPHY  57
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Multi layer perceptron</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2</td>
<td>LSTM unit</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Med2Vec architecture</td>
<td>11</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Distribution of various CHF disease codes</td>
<td>14</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Age and gender distribution of CHF patients</td>
<td>16</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Correlation plot for hand picked features</td>
<td>18</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Prediction point for different predictions</td>
<td>19</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Input to the model from MIMIC-III</td>
<td>21</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Channel-wise neural network</td>
<td>25</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Channel-wise models for readmission prediction (after feature selection)</td>
<td>26</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Channel-wise models for readmission prediction (forced temporal)</td>
<td>27</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Highway temporal model</td>
<td>28</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Highway single gate</td>
<td>29</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Highway MLP</td>
<td>30</td>
</tr>
<tr>
<td>Figure 15</td>
<td>LSTM model for full patient</td>
<td>31</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Single visit models ROC AUC before feature selection</td>
<td>35</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Single visit models ROC AUC after feature selection</td>
<td>36</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Single visit models ROC AUC after input change</td>
<td>36</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Two visit models after feature selection</td>
<td>37</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Two visit models after input change</td>
<td>38</td>
</tr>
<tr>
<td>Figure 21</td>
<td>ROC AUC scores for 15 fold cross validation in case of readmission prediction in highway MLP model</td>
<td>39</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Different time categories for MLP concatenated and Highway MLP models</td>
<td>40</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Carry gate output</td>
<td>41</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Categorical risk change in simple neural network model</td>
<td>43</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Categorical risk change in MLP concatenated model</td>
<td>43</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Categorical risk change for length of stay (multiclass) in different models</td>
<td>45</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Reliability curve for 30 days readmission</td>
<td>46</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Reliability curve for 60 days readmission</td>
<td>46</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Reliability curve for 90 days readmission</td>
<td>46</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Reliability curve for in-hospital mortality</td>
<td>47</td>
</tr>
</tbody>
</table>
List of Figures

Figure 32 Mean of features 48
LIST OF TABLES

Table 1 Top three ICD-9 code distribution in MIMIC-III 14
Table 2 CHF ICD9 codes 15
Table 3 Range of values for measurements 17
Table 4 Readmission features 19
Table 5 Time gap as number of days into categories 20
Table 6 Length of stay in number of days to categorical 23
Table 7 Number of inputs in prediction outcomes after feature selection 25
Table 8 Time gap as temporal factor 27
Table 9 Class distribution (number of visits) for different predictions 31
Table 10 Behaviour of different models based on change in categorical time input 41
Table 11 Behaviour of different models based on change in categorical time input for length of stay 42
Table 12 Computation time and epochs for MLP concatenated model 49
Table 13 Computation time and epochs for HW MLP model 49
Table 14 Total computation time for MLP concat and HW MLP models 49
Table 15 Results summary 50
Table 16 Results summary 50
<table>
<thead>
<tr>
<th>Listing</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listing 1</td>
<td>Length of stay statistics</td>
<td>22</td>
</tr>
<tr>
<td>Listing 2</td>
<td>Parameter tuning for random forest</td>
<td>23</td>
</tr>
<tr>
<td>Listing 3</td>
<td>Parameter tuning for XGBoost</td>
<td>24</td>
</tr>
</tbody>
</table>
ACRONYMS

AI    Artificial Intelligence
NLP   Natural Language Processing
EHR   Electronic Health Records
CHF   Congestive Heart Failure
ICU   Intensive Care Unit
CNN   Convolutional Neural Network
RNN   Recurrent Neural Network
GRU   Gated Recurrent Units
SDA   Stacked Denoising Autoencoders
MIMIC-III Medical Information Mart for Intensive Care-III
HIPAA Health Insurance Portability and Ac-countability Act
ICD   International Classification of Diseases
NDC   National Drug Code
SMOTE Synthetic Minority Over-sampling Technique
ROC   Receiver Operating Characteristic Curve
AUC   Area Under Curve
LSTM  Long Short-Term Memory Network
MLP   Multi Layer Perceptron
T-LSTM Time-aware Long Short-Term Memory Network
RMSE  Root Mean Squared Error
LOS   Length of Stay
RF    Random Forest
OOB   Out of Bag
XGB   XGBoost
ReLU  Rectified Linear Unit
TPR   True Positive Rate
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPR</td>
<td>False Positive Rate</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
</tbody>
</table>
INTRODUCTION

1.1 MOTIVATION AND PURPOSE

Artificial Intelligence (AI) systems in healthcare can be of many forms. For instance, it could be in the form of robots that are used for purposes like drug delivery to treat patients, teaching children with special needs or to aid the elderly people (care robots) or it could also be in the form of virtual approach to build decision support systems or many more [1]. These kind of decision support systems designed with medical data can be used to help doctors in making informed clinical decisions by providing insights about a patient’s conditions to predict outcomes such as readmission, in-hospital mortality, length of stay and many other applications like these. The healthcare data is available in many forms, for instance, images, text, time-series (ECGs), sounds, categorical, numerical, text and so on.

In general, EHR data comprises of information such as diagnoses made on each patient, lab results, procedures followed, medications prescribed, vital measurements for each visit made by patients to a hospital and their demographic information such as age, gender, ethnicity and many more. The real challenge lies in representing this data for predictive modelling because of its high dimensionality, temporal nature and sparsity. High dimensionality of EHR data leads to longer computation time, need for larger storage space and also leads to sparsity in data. When there are large number of diagnoses, this could lead to sparsity in data for rarely occurring diseases. Hence, need for more data arises in case of high dimensional data. Temporal nature of EHR data is the order in which diagnoses occur and time interval between diagnoses or time interval between visits. These aspects should be considered when handling EHR data. In order to address the above mentioned aspects, state-of-the-art deep learning models are being used [2].

Potential use of these predictive models covers addressing a wide range of problems in the field of healthcare such as hospital readmission prediction, hospital length of stay prediction, in-hospital mortality prediction, phenotype classification, drug recommendation, decompensation and so on [3]. These predictive models can also be helpful for hospitals to manage their resources efficiently, for patients to know more about future risks, for doctors to have more understand-
2 Introduction

ing about a patient’s condition and on the whole to improve the qual-
ity of medical service.

1.2 Problem Definition

EHRs are temporally ordered high dimensional data with sequential
relationship between each visit made by a patient, demographic infor-
mation, clinical notes, lab results, clinical diagnosis and medications.
Data representation plays an important role in the performance of a
predictive model [4]. A patient can be represented in multiple ways
with help of EHR data. For example, each patient may be represented
as one feature vector or a patient may be represented as multiple fea-
ture vectors where each vector gives information about a particular
visit.

Data representation plays a vital role in converting medical data or
text to a form which is understandable (numbers) for machines to
process and learn from the data. The main purpose of data repre-
sentation lies in mapping high dimensional medical data to lower
dimensional space and learning latent relationship in the data. By la-
tent relationship, we mean the relationship between domain concepts.
Deep learning approaches that have been proposed recently capture
this relationship between data in an efficient way [5]. As said earlier,
temporal nature of the data should also be taken into consideration
when creating a representation for EHR data, that is, order in which
diagnoses has been made or order in which patient’s visits occurred
should be taken into account.

Apart from the temporal nature of EHR data and its high dimension-
ality, there are also other challenges with EHR data. Some of them are
biases, missing values, irregularities in the data like visit level irregu-
larity, feature level irregularity and many more. Visit level irregularity
refers to varying timegap between each visit made by a patient. Fea-
ture level irregularity refers to progression of different diseases at
varying time intervals.

But why is this irregularity important to consider in healthcare? Ac-
cording to authors of the book [6], time is one of the important factors
in diagnostic process. Diseases progress through time and there can
be time elapse between onset of disease and symptoms showing up
in a patient. Or, there may be delay in recognizing actual symptoms
as diagnosis. When handling medical data, the rate at which a dis-
ease progresses, that is, irregularity between visits may give more
information about a patient’s condition.
The main research question that we would like to address is:

- Are visit level irregularities in EHR data important to consider for predicting clinical outcomes?

In [7], visit level irregularity has been handled using Long Short Term Memory (LSTM) and it is due to the fact that number of visits differ for each patient and LSTMs can handle inputs of variable length. In case of Medical Information Mart for Intensive Care - III (MIMIC-III) with Congestive Heart Failure (CHF) cohort, average number of visits is two in a time window of 90 days. Hence, we have considered two visits of patient (previous and current visit) as input. The contribution of this thesis lies in the fact that we have shown that visit level irregularity can also be addressed with two visits (like partial history) rather than complete patient history as being done in literatures [7], [8], [9] widely.

Initially, to prove the significance of adding time as an input feature, single visits models (models with input as current visit information) were tested. To improve further, two visit models (models with previous and current visit) along with time as an input feature were tested. In order to use time as more than just a feature, inspired by works in language modelling and speech recognition [10], we have tried simpler networks with memory to handle dependencies within medical data instead of LSTMs. In other words, visit level irregularity problem has been addressed using Multi Layer Perceptron (MLP) with channel inputs and gating mechanisms.

To evaluate the proposed model, different clinical outcomes like hospital readmission, in-hospital mortality and length of stay have been predicted. Unplanned readmissions [11] are seen as an indicator of hospital’s quality of service. From patients point of view, increase in healthcare costs can be stressful for them. Prediction of length of stay and mortality prediction [12] are important as they help hospitals to provide better services for patients and also to manage resources of a hospital efficiently. The performance of the models have been reported using ROC AUC scores. In addition to evaluation metrics, learning of the models, risk change when timegap is changed between the visits and model reliability have been investigated.

This thesis will answer the following questions as well:

- For what kind of predictions does visit level irregularity play a significant role?
• How should this visit level irregularity be incorporated? Should it be given as a feature or use time information to decide the importance of the previous visit?

• If time gap should be given as a feature, should it be given as categorical input or continuous input?

In this report, Chapter 2 gives an overview of existing works, its advantages and disadvantages, Chapter 3 gives a brief description of the dataset that we have used for the project, Chapter 4 gives a detailed description of how database has been set and also about the models that we have implemented, in Chapter 5, we talk about the results obtained and Chapter 6 includes discussion, conclusion and future work.
BACKGROUND

2.1 LITERATURE REVIEW

2.1.1 Patient data representation

Simplest method for data representation is one-hot encoding. But one-hot encoding fails to capture the latent relationship between data [13]. By latent relationship, we mean that representation should be similar for closely related diseases. One more limitation of one-hot encoding is that if there are N different features, such as diagnosis, procedure and medication, then the resulting vector is N-dimensional. As number of features increases, dimension of the vector will also increase, thus introducing sparsity in data.

The proposed method in [13] and [9] uses skip-gram to learn latent relationship between the codes occurring within same visit. By combining the code representations and summing the resultant vectors from each visit, representation for a patient was obtained. But temporal sequence of the visits was ignored. However, by using skip-gram, dimensions of the vector was reduced to D where D is generally chosen by user ranging between 50 and 1000. This D-dimensional vectors are used for prediction later.

In [8] and [14], same representation as above has been used. In [8], authors have used pre-trained embedding layer to detect pattern in medical records. Giordano et al.[14] proposed a model to create patient representation by concatenating medical events from EHR data of a patient in sequential order. In this model, authors have mapped the words into a vector space by taking semantic nature of events into account. A dynamic window has been introduced by authors to handle temporal sequence of medical events which is an extension of Word2Vec and they have also added a time decay factor to give more importance to most recent diagnosis.

Choi et al.[15] proposed a model that aims in learning and interpreting the representations for both medical codes and visits. There are two types of information that can be obtained from EHR data according to the author - one is the relationship between medical codes occurring within a visit and other is sequence in which visits occur. Visit level representation helps us to know more about diagnoses that has been made on a patient and helps us to predict future diagnosis.
The authors have used MLP to generate visit level representation. In [7], authors have projected high dimensional vector to a lower dimensional space to obtain a representation for each patient.

In [16] and [17], autoencoders has been used for data representation. Miotto et al.[16] proposed a model that makes use of unsupervised data representation for feature learning. The proposed neural network uses Stacked Denoising Autoencoders (SDA) which learns the regularities and dependencies in data to generate a patient representation. Denoising Autoencoders prevents overfitting as it reconstructs output from noisy input. These denoising autoencoders are trained to fill missing information in patient records. In the proposed model, same structure has been used in all autoencoders that are trained layer by layer. SDA learns pattern in data which when grouped gives patient representation. A single vector or sequence of vectors from temporal window represents a single patient.

2.1.2 Models

Both supervised and unsupervised machine learning approaches have been applied to EHR data. Deep neural networks like Gated Recurrent Units (GRU) [9], Convolutional Neural Networks (CNN) [8], LSTMs [7], [17] have been used for predictive modelling with EHR. Once EHR data representation is done, it is then utilised to predict clinical outcomes of a patient.

2.1.3 Evaluation

Evaluation has been done by predicting next diagnoses and medication in succeeding visits and time until next visit in [9] and in [15], final visit level representation was evaluated by predicting medical code in current visit and in next visit. Unplanned readmission has been used for evaluation in [8] and in [7]. In [14] and in [17], authors have evaluated their models by clustering medical events based on their type to show their model’s ability to capture relationship between medical diagnoses. Probability of a patient being diagnosed with a particular disease in a time span of one year has been used for evaluation in [16].

2.1.4 Addressing irregularity in EHR data

Pham et al [7] have addressed the problem of visit level irregularity using LSTM networks. Visit level irregularity has been handled by giving elapsed time between two visits as input. This information modifies the forget gate to control if previous visit made by a patient is important or not. The authors have used larger time window of
about 12 months, 24 months and whole history of a patient to give importance to previous visits.

Baytas et.al [17] have also used LSTM to address visit level irregularity in EHR data. With the help of elapsed time between visits, authors have modified LSTM to time-aware LSTM network(T-LSTM). The elapsed time between two visits influences the forget gate of LSTM. The authors have decomposed previous memory into two components - long term and short term components. Authors have given importance to short term components taking both long and short term dependencies into consideration using LSTM based on elapsed time information.

2.2 THEORY

2.2.1 Random forest

Random forest (RF) [18] is an ensemble learning method. RF is based on bagging technique. It has multiple decision trees and it can be used for both classification and regression. Multiple trees are built in parallel. The output of one tree is not dependent on another. In classification, final output is obtained by majority voting. Random forest takes n samples of data by bootstrapping where n is the number of trees in a forest. For each sample, a classification tree is grown. Each node split is based on random set of features. Because of this randomness, random forest are less prone to overfitting. Error estimate for each iteration is done on an out-of-bag (OOB) sample.

2.2.2 XGBoost

XGBoost (XGB) [19] is an ensemble learning library that uses gradient boosting decision tree algorithm. Gradient boosting technique uses ensemble of weak learners to model a strong learner. Models are built sequentially rather than in parallel, in such a way that new models created are used to reduce error from previous models. This process is repeated until error is minimized as much as possible. In such boosting technique, overfitting can be controlled by choosing right number of trees. Output of a boosting model is weighted average of all the models.

2.2.3 Multi layer perceptron

MLP [20] is a feed forward neural network that tries to approximate a function which can map input to a target. A simple MLP consists of three layers - input layer, hidden layer and output layer. The number of hidden layers can vary in MLP. In a fully connected network, each
neuron in a layer will be connected to all neurons in the successive layer with a weight $w_{ij}$. Here, i and j represents a specific neuron in previous and given layer respectively. Output of each neuron, $a_j$, can be seen in the following Equation 1,

$$a_j = g \sum_{i=0}^{n} w_{ij} a_i$$

(1)

where n is the number of neurons in previous hidden layer, $w_{ij}$ is the weight of a $i^{th}$ neuron in previous layer to a $j^{th}$ neuron in current layer, g is an activation function, $a_i$ is an input to a neuron and $a_j$ is output of the neuron. Activation functions are used to establish a non-linear relationship between input and target. MLPs are trained using backpropagation algorithm [21].

Figure 1: Multi layer perceptron. taken from [1]

Figure 1 illustrates Equation 1. The inputs to model are shown as $x_1$, $x_2$ and weights $w_{ij}$ in Equation 1 are shown as $w_{1j}$, $w_{2j}$ which are the weights connected from previous layer to current layer.

2.2.4 Long short term memory

LSTM [22] has been introduced to handle vanishing gradient problem in vanilla RNN. The gates in LSTM namely, input, forget and update

---

To be found online at
gate are used to handle long time dependencies and vanishing gradient problem. The equations of the gates are as follows:

\[
i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \quad (2)
\]

\[
f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \quad (3)
\]

\[
o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \quad (4)
\]

\[
c_t = f_t c_{t-1} + i_t \tanh(W h_{t-1} + U x_t + b) \quad (5)
\]

\[
h_t = o_t \tanh(c_t) \quad (6)
\]

where Equation 2, Equation 3, Equation 4 are input, forget and output gates respectively, Equation 5 is current memory cell and Equation 6 is current output. The output of current unit depends on current input \(x_t\), previous memory \(c_{t-1}\) and previous output \(h_{t-1}\). When the value in forget gate is 1, everything is remembered and current memory output is obtained by summing old and new memory. The current output \(h_t\) is obtained by element-wise multiplication of output gate and current memory. The depiction of a cell describing the equations is shown in Figure 2.

![Figure 2: LSTM unit. taken from 2](https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg)

Figure 2 shows a single LSTM unit with input gate \(I_t\), forget gate \(F_t\) and output gate \(O_t\) from which current output \(h_t\) and current memory cell \(c_t\) is calculated.

### 2.2.5 Highway networks

Highway networks [23], which are inspired by LSTM networks, has a gating function that can be used to bypass information through the network. These networks were initially used to optimize training in deep neural networks. However, in language modelling tasks, these

---

To be found online at [https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg](https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg).
highway networks are used alongside with LSTM as an additional memory [10]. The authors have introduced two non linear transforms in the highway layer - transform gate and carry gate. A highway layer can be explained with the following Equation 7,

\[ y = H(x, W_H).T(x, W_T) + x.C(x, W_C) \]  

where C is the carry gate, T is the transform gate and H is the transform function. In Equation 7, x is the input and W is the weight. If C = 1 - T, Equation 7 can be modified to Equation 8 as follows:

\[ y = H(x, W_H).T(x, W_T) + x.(1 - T(x, W_T)) \]  

The output values are defined by particular T values which can be seen in Equation 9,

\[ y = \begin{cases} 
  x, & \text{if } T(x, W_T) = 0 \\
  H(x, W_H), & \text{if } T(x, W_T) = 1 
\end{cases} \]  

From Equation 9, it can be seen that when the value of transform gate is zero, the output is same as input. If the transform gate value is not zero, non-linear transformation of input is the output. Also, in the above equation, T is a sigmoid function.

\[ T(x) = \sigma(W_T^T x + b_T) \]  

where \( W_T \) and \( b_T \) are weight matrix and bias vector of the transform gate respectively. By learning \( W_T \) and \( b_T \), networks passes the input to next layer.

2.2.6 Med2Vec architecture

The representation used in this thesis is Med2Vec architecture [15] which is shown in Figure 3. In each visit, there are number of diagnoses which is represented as \( x_i \in 0,1 \). The intermediate visit representation is formed using Rectified Linear Unit (ReLU) activation function to which demographic information \( d_i \) is added to form final visit representation \( v_i \). This final visit representation is used for predictions.
2.2.7 Class imbalance

Some of the techniques to handle class imbalance are oversampling (adding samples to existing ones), undersampling (removing samples) techniques and cost-sensitive learning [24]. The disadvantages with sampling techniques are as follows:

- Undersampling techniques lead to loss of valuable information.
- Random Oversampling replicates minority class samples.
- Other oversampling methods like Synthetic Minority Over-Sampling Technique (SMOTE) [25] generates synthetic samples of minority class but consumes more time because actual data consists of small number of minority class samples.

Cost sensitive learning method tries to minimise the misclassification cost with an assumption that real world applications will not have uniform costs for misclassifications. Cost sensitive learning shifts the bias towards minority class [26].
3.1 OVERVIEW OF MIMIC-III

The dataset provided for this master thesis project is Medical Information Mart for Intensive Care-III (MIMIC-III) [27]. MIMIC-III is a freely accessible critical care database. This database contains information of patients collected from Beth Israel Deaconess Medical Center in Boston, Massachusetts. This database has 26 tables such as admissions, chart events, diagnoses, Intensive Care Unit (ICU) stays, lab events, patients, transfers and so on. Each patient has an unique subject ID and also unique admission ID for each admission. There are around 56000 unique admissions and 46000 unique patients in this database. MIMIC-III contains demographic information about a patient such as their date of birth, date of death, ethnicity, marital status, gender. There are also other information in MIMIC-III such as an expiry flag to indicate death of a patient in a hospital, clinical measurements taken from patients, procedures done on a patient, diagnosis made for each patient for all the admissions, hospital length of stay and so on [28]. More detailed description of the information available in each table and details of tables that has been selected has been discussed in Chapter 4.

MIMIC-III is a collection of information from critical care information systems, hospital database and Social Security Administration Death Master File. Information from two critical care information systems, namely CareVue and Meta Vision, were merged in this database. According to the standards of Health Insurance Portability and Accountability Act (HIPAA), data deidentification process was carried out before forming MIMIC-III database. Information such as patient’s name, address were removed. The date related information were shifted to sometime in future (between the year 2100 and 2200) with a random offset but without disturbing the actual intervals available in original information. Date of birth of patients who were aged above 89 were shifted to sometime in order to range their date of birth to 300 years according to HIPAA standards.

The tables ADMISSIONS, PATIENTS, ICUSTAYS, SERVICES and TRANSFERS can be used to track patients. There are tables prefixed with D_to find out the definitions of codes such as procedures, diagnoses and items which were used to take measurements from a patient. Other tables in MIMIC-III gives information about measurements, ob-
servations and billing information for each patient.

Diagnosis are represented in International Classification of Diseases (ICD-9) format in the database. The top three ICD-9 code from MIMIC-III database can be seen in Table 1.

<table>
<thead>
<tr>
<th>ICD-9 CODE</th>
<th>DISEASE</th>
<th>% OF ADMISSIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>401.9</td>
<td>Hypertension</td>
<td>31.8</td>
</tr>
<tr>
<td>428.0</td>
<td>Congestive Heart Failure</td>
<td>2.01</td>
</tr>
<tr>
<td>427.31</td>
<td>Atrial fibrillation</td>
<td>1.98</td>
</tr>
</tbody>
</table>

3.2 CHF PATIENTS STATISTICS

In this thesis, only CHF patients are considered. The ICD-9 code for CHF corresponds to class 428. There are around 15 codes within class 428 corresponding to different types of CHF. Distribution of CHF disease alone can be seen in Figure 4.

The percentage shown in Figure 4 corresponds to number of admissions made for each CHF code in the database. The codes in the legend of Figure 4 is listed in Table 2.
3.2 CHF Patients Statistics

Table 2: CHF ICD9 codes

<table>
<thead>
<tr>
<th>ICD-9 Code</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>4280</td>
<td>Congestive heart failure (unspecified)</td>
</tr>
<tr>
<td>42832</td>
<td>Chronic diastolic heart failure</td>
</tr>
<tr>
<td>42833</td>
<td>Acute on chronic diastolic heart failure</td>
</tr>
<tr>
<td>42823</td>
<td>Acute on chronic systolic heart failure</td>
</tr>
<tr>
<td>42822</td>
<td>Chronic systolic heart failure</td>
</tr>
<tr>
<td>42830</td>
<td>Diastolic heart failure (unspecified)</td>
</tr>
<tr>
<td>42821</td>
<td>Acute systolic heart failure</td>
</tr>
<tr>
<td>42831</td>
<td>Acute diastolic heart failure</td>
</tr>
<tr>
<td>42820</td>
<td>Systolic heart failure (unspecified)</td>
</tr>
<tr>
<td>42843</td>
<td>Acute on chronic combined systolic and diastolic heart failure</td>
</tr>
<tr>
<td>42842</td>
<td>Chronic combined systolic and diastolic heart failure</td>
</tr>
<tr>
<td>42840</td>
<td>Combined systolic and diastolic heart failure (unspecified)</td>
</tr>
<tr>
<td>42841</td>
<td>Acute combined systolic and diastolic heart failure</td>
</tr>
<tr>
<td>4281</td>
<td>Left heart failure</td>
</tr>
<tr>
<td>4289</td>
<td>Heart failure (unspecified)</td>
</tr>
</tbody>
</table>

There are around 10000 CHF patients and around 14000 visits. Of this, 3500 patients were readmitted at least once. We have considered only those patients who had at least two visits from the database. The age and gender distribution for CHF patients can be shown in the Figure 5.
As seen in Figure 5, age was grouped into categories. From Figure 5, it is evident that higher age group is more prone to CHF diseases. Apart from age and gender, there are also some more important features available in MIMIC-III for CHF patients which will be discussed in Chapter 4.
METHODOLOGY

4.1 SETTING UP MIMIC-III

In order to set up MIMIC-III on a local database, we followed tutorials available in Physionet 1. Once MIMIC-III is loaded in a local Postgres database, we can connect to it using psycopg2 in Python.

4.2 INPUT FEATURES

4.2.1 Hand-picked features (based on literature)

In [29], [30], [31], some of the important features for CHF patients are listed. Of these, common features that play significant role in CHF patients prediction are blood pressure, gender, age, heart rate and so we have considered those features for outcome prediction. It was mentioned in Chapter 3 that age of patients greater than 89 were changed with an offset to make their age greater than 300. Those patients who were aged more than 300 were set to 90.

As described in Chapter 3, MIMIC-III has information from two critical care information systems. More than two item ID corresponds to one measurement. There are multiple measurements corresponding to each patient for a single admission. Hence, out of those values we chose a minimum and a maximum value irrespective of the care information system. MIMIC-III team [32] has given a range of values for each measurement as seen in Table 3. The values in table represents a range in which values of features may lie. Note that, the above mentioned features were given as input features as such but were not used to calculate the severity condition of the patient.

<table>
<thead>
<tr>
<th>MEASUREMENT</th>
<th>RANGE (MINIMUM, MAXIMUM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Respiratory rate</td>
<td>0, 70</td>
</tr>
<tr>
<td>Heart rate</td>
<td>0, 300</td>
</tr>
<tr>
<td>Systolic blood pressure</td>
<td>0, 400</td>
</tr>
<tr>
<td>Diastolic blood pressure</td>
<td>0, 300</td>
</tr>
</tbody>
</table>

1 https://mimic.physionet.org/tutorials/install-mimic-locally-windows/
The correlation among hand picked features can be seen in Figure 6.

Most highly correlated features are minimum systolic blood pressure and minimum diastolic blood pressure, maximum systolic blood pressure and maximum diastolic blood pressure with a correlation coefficient of nearly 0.6. This is because of the linear relationship between systolic blood pressure and diastolic blood pressure in general [33]. Apart from the above mentioned features, it is evident from Figure 6 that there is less correlation among other features. So, no feature selection is done for hand picked features.

These features were given as input in different ways for different outcomes. In case of readmission, prediction is done as to if a patient will be readmitted after a particular visit, shown in Figure 7a. So, whole information from a particular visit is considered. However, in case of in-hospital mortality and length of stay prediction, predictions are happening at the start of a visit as to what will happen at the end of that particular visit as shown in Figure 7b. Hence, instead of giving whole information about a visit, only the measurements obtained during first 24 hours of visit are considered as input for these two predictions rather than whole visit information.
4.2 Input features

Figure 7: Prediction point for different predictions

4.2.2 Readmission prediction features

Hospital score [34] is also seen as an important predictor in case of readmission predictions. Hospital score is calculated from various variables from the medical dataset. These variables include haemoglobin level, discharge from oncology service, sodium level of a patient, procedure on a patient during hospital stay, admission type of a patient (urgent or emergent), number of hospital admissions during the previous year and length of stay. However, in our input we have not calculated hospital score. Instead, we have given these features as categorical input. Features considered for readmission prediction and their categorical split up can be seen in Table 4.

Table 4: Readmission features

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>CATEGORY (IF POSITIVE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haemoglobin level (&lt;12g/dL)</td>
<td>1</td>
</tr>
<tr>
<td>Sodium level (&lt;135mEq/L)</td>
<td>1</td>
</tr>
<tr>
<td>Procedure during hospital stay</td>
<td>1</td>
</tr>
<tr>
<td>Admission type (urgent or emergent)</td>
<td>1 (urgent)</td>
</tr>
<tr>
<td># of hospital admissions previous year</td>
<td>0-1 - 0, 2-5 -1, &gt;5 - 2</td>
</tr>
<tr>
<td>Length of stay</td>
<td>&lt;5 - 0, &gt;5 - 1</td>
</tr>
</tbody>
</table>
4.2.3 Diagnosis and medication codes

There are 4,894 unique diagnosis codes in MIMIC-III database. The codes in DIAGNOSES_ICD table can be mapped to corresponding disease by utilising D_ICD_DIAGNOSES dictionary table from the database. There are 8,130 number of medications in MIMIC-III. Medications prescribed within first 24 hours of admission only have been taken into consideration for in-hospital mortality and length of stay prediction outcomes. And all medications prescribed during a particular visit have been considered for readmission outcome prediction. In case of diagnoses, there is no timestamp to find when a patient was actually diagnosed after admission. So, all the diagnoses codes in a particular visit have been considered for all prediction outcomes.

4.2.4 Time gap

Time gap between each visit for a patient is calculated based on discharge time of previous visit and admit time of current visit. This time is calculated in number of days. Instead of giving time as continuous numbers, it has been converted into four categories as in [8] and can be seen in Table 5.

<table>
<thead>
<tr>
<th>NUMBER OF DAYS</th>
<th>CATEGORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-30</td>
<td>0</td>
</tr>
<tr>
<td>30-60</td>
<td>1</td>
</tr>
<tr>
<td>60-90</td>
<td>2</td>
</tr>
<tr>
<td>Greater than 90</td>
<td>3</td>
</tr>
</tbody>
</table>

The input extraction can be found in the github repository [2].

Different tables considered from MIMIC-III and information obtained from each table can be seen in Figure 8 followed by a brief description.

• From ADMISSIONS table, we consider information about a patient admission such as admission ID which is unique for each admission, admit time, discharge time for that particular admission, birth time of a patient, subject ID which is unique for each patient, time gap between visits is calculated using admit time and previous discharge time of a patient, admission type of a patient, and on

particular admission, number of visits made in previous year, length of stay for a particular visit.

- From DIAGNOSES_ICD table, ICD9 codes are mapped with dictionary table ‘D_ICD_DIAGNOSES’ to find out diagnoses made on a patient by matching them with subject ID and admission ID.

- From CHARTEVENTS table, some important features of CHF patients like systolic and diastolic blood pressure, respiratory rate, heart rate and sodium level are extracted. Each feature has an item id which is mapped with dictionary table ‘D_ITEMS’ to know which features are available for a patient. Required features are identified with item ID and extracted. Each patient has many number of readings recorded during the admission. The minimum and maximum values of features are considered.

- From PATIENTS table, gender information is extracted by mapping with admission ID.

- From PRESCRIPTIONS table, information about medications prescribed for a patient is extracted. This is given as NDC (National Drug Code) in the database. NDC 3 is a 10-digit number where first 4 numbers represents the labeler, 4 numbers represents the drug and last 2 numbers represent package size. So, second segment with 4 numbers is considered as input for medications.

- From PROCEDURES_ICD table, feature for readmission is taken by matching admission ID from ADMISSIONS table.

---

3 https://www.fda.gov/Drugs/DevelopmentApprovalProcess/UCM070829
4.3 DATA REPRESENTATION

A patient can have multiple diagnoses in a visit. Each visit of a patient is represented as multi-hot vector with diagnoses codes representing 0 or 1 indicating whether a patient had a particular diagnosis in a visit or not. Interpretability \cite{35} refers to knowledge of understanding how a particular prediction is obtained from predictive model. In case of healthcare, models that are developed for prediction is going to have an impact on patient’s health. Under such situations, physicians should be able to trust the model’s prediction which is possible with interpretable models. Representation used in \cite{15}, has been tested for interpretability by clinicians. The representation of medication codes mimics as that of the diagnoses code representation. All CHF features are normalised and given as input. In case of gender, values are given as 0 and 1 for male and female respectively. So, each visit has 4894 diagnoses codes, 8130 medication codes and 10 CHF features which sums to 13034 inputs in case of without time and 13038 inputs in case of inputs with time. Note that these numbers refer to number of input before feature selection. This input is given to all the models discussed in Section 4.5.

4.4 PREDICTION OUTCOMES

4.4.1 Hospital readmission

In case of readmission prediction outcome, if a patient has second visit after a long time, for instance, 300 days, it cannot be considered as readmission and it should be accounted as a new admission since reason behind current admission may not be same as previous admission. Hence, introducing a time window for readmission seems to be necessary. We have used time windows 30, 60 and 90 days and output labels are created accordingly. In case of readmission with time window of 60 days, patients readmitted within 0-60 days of previous discharge has been considered. Similarly, 90 day readmissions consists of visits made within 0-90 days from previous discharge time.

4.4.2 Length of stay

Length of stay is the duration between admit time and discharge time of a patient for a particular visit in number of days. The statistics of length of stay for CHF patients can be seen in Listing 1.

<table>
<thead>
<tr>
<th>Listing 1: Length of stay statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
</tr>
<tr>
<td>std</td>
</tr>
<tr>
<td>min</td>
</tr>
</tbody>
</table>
Based on the statistical values, we have converted length of stay to four categories for multi-class classification. The category split up can be seen in Table 6.

<table>
<thead>
<tr>
<th>NUMBER OF DAYS</th>
<th>CATEGORY</th>
<th>NUMBER OF VISITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-5</td>
<td>0</td>
<td>1689</td>
</tr>
<tr>
<td>6-8</td>
<td>1</td>
<td>1212</td>
</tr>
<tr>
<td>9-14</td>
<td>2</td>
<td>1300</td>
</tr>
<tr>
<td>15-126</td>
<td>3</td>
<td>1227</td>
</tr>
</tbody>
</table>

4.4.3 **In-hospital mortality**

In-hospital mortality is predicted to find out whether a patient expires in the hospital or not. These labels were taken from “HOSPITAL_EXPIRE_FLAG” from ADMISSIONS table.

4.5 **SINGLE VISIT MODELS**

The models that we have implemented for single visit predictions are random forest [36], XGBoost classifier [19], simple neural network and a neural network inspired from [15] and [37] which we will call as “Channel-wise neural network”.

4.5.1 **Random forest classifier**

All input features mentioned in Section 4.2 were concatenated together and given as input. Hyperparameter tuning for this model was performed using GridSearchCV in Python. Parameters that were tuned can be seen below in Listing 2.

```python
parameters = {
    'bootstrap': [True],
    'max_depth': [80, 90, 100, 110],
    'max_features': [2, 3, 10],
    'min_samples_split': [8, 10, 12],
    'n_estimators': [30, 50, 100]
}
```
4.5.2 *XGBoost classifier*

XGBoost classifier also has same concatenated inputs like random forest and was trained and tested with and without time. GridSearchCV was used for hyperparameter tuning. Parameters which were tuned for XGBoost classifier can be seen in Listing 3.

Listing 3: Parameter tuning for XGBoost

```python
params = {
    'min_child_weight': [1, 5, 10],
    'n_estimators': [100, 300, 500, 600],
    'max_depth': [3, 4, 5]
}
```

4.5.3 *Neural network*

Input to neural network were also given by concatenating all input features. To control overfitting, dropout has been used [38]. The model has been built using Keras Sequential model [39] with one hidden layer. Parameters tuned for neural network model are number of hidden neurons in a layer, optimizers, batch size and dropout ratio.

4.5.4 *Channel-wise neural network*

According to authors of [37], giving different inputs in separate channels enables a model to learn about each variable separately before combining all of them together. That is, the model will learn some vital information from each of the variables separately before they are concatenated together. Inputs to the model are of mixed data type, that is, diagnoses and medications are categorical and hand picked features are numeric. Keras has a functional API to handle these kind of mixed inputs [39]. So, we give diagnoses codes as a separate input in first input layer, then medication codes in a separate input layer and CHF features concatenated with time gap as categorical and demographic features are given in a separate input layer. Same parameters as mentioned 4.5.3 were tuned in this model as well. The model can be seen in Figure 9. Same architecture has been used for model without time. Only change will be reduction in number of inputs in the second layer since time information will not be concatenated with other features.
As seen from the explanation above for inputs, it is evident that number of features for a single visit of a patient is almost 13000. One common way of reducing input dimension is to eliminate rarely occurring diseases [16]. Same procedure has been done for both diagnoses codes and medication codes. In case of diagnoses codes, if a particular disease has not occurred even in 5% of the patient, that particular disease has not been considered. In this way, diagnoses codes were brought down to 243 features instead of 4894 and medication codes to 100 instead of 8130 in case of readmission prediction. With in-hospital mortality prediction and length of stay prediction, medications is reduced to 11 after feature selection as we have considered information only till 24 hours after admission. The number of inputs for different predictions after feature selection can be seen in Table 7.

<table>
<thead>
<tr>
<th>PREDICTION OUTCOME</th>
<th>DIAGNOSES FEATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Readmission</td>
<td>243</td>
</tr>
<tr>
<td>In-hospital mortality</td>
<td>243</td>
</tr>
<tr>
<td>Length of stay</td>
<td>243</td>
</tr>
</tbody>
</table>
Figure 10 is for readmission prediction. In case of in-hospital mortality prediction and length of stay, there is only one dense layer after concatenation of inputs. Note that this is common for all the models.

4.7 TWO VISIT MODELS

For two visit models, inputs considered are diagnoses, medications and features of current visit and previous visit with time gap between these two visits.

4.7.1 Channel-wise neural network

This model is same as the one explained in Section 4.5.4. Instead of giving diagnoses and medications as separate inputs in different channels, we have given first visit of a patient in one channel and second visit of a patient in another channel. Each visit of a patient will have information about the patient, diagnoses, medication and other features concatenated together.

4.7.1.1 Visit and time concatenated

This model will henceforth be referred as “MLP concat”. Number of inputs in each channel varies from that of Figure 10. In channel 1 (input_172) and channel 3 (input_174), number of inputs will change depending on the prediction outcome as in Table 7. And in channel 3 (input_173), number of inputs is 4 which corresponds to time gap between two visits of the patient converted to categorical as in Table 5.
### 4.7.1.2 Visit multiplied by temporal factor

This model will henceforth be referred as “MLP temporal”. The model structure will be same as the one from the previous Section 4.7.1.1. Here, instead of concatenating time with information of two visits, time is used as a factor to reduce the importance of previous visit information. This is kind of forced learning where we are setting factors to reduce the importance of current visit based on time gap which can be seen in Table 8.

<table>
<thead>
<tr>
<th>NUMBER OF DAYS</th>
<th>TEMPORAL FACTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-30</td>
<td>1</td>
</tr>
<tr>
<td>30-60</td>
<td>0.75</td>
</tr>
<tr>
<td>60-90</td>
<td>0.5</td>
</tr>
<tr>
<td>Greater than 90</td>
<td>0.25</td>
</tr>
</tbody>
</table>

![Figure 11: Channel-wise models for readmission prediction (forced temporal)](image)

Model can be seen in Figure 11. As seen from this figure, importance of previous visit is modified by multiplying the previous visit information with a temporal factor. Once this is done, previous visit and current visit information are concatenated together to make predictions. Note that the figure here represents readmission predic-
tion. The number of inputs and dense layer will vary accordingly for length of stay and in-hospital mortality prediction.

4.7.2 Highway models

In highway models, models were designed in such a way to learn the importance of previous visit information by using gating mechanisms.

4.7.2.1 Highway temporal

This model will be referred as “HW temporal”. In highway temporal model, as discussed in Section 2.2.5, two gates - transform and carry gate, are used to learn the importance of previous visit using time as input to these gates. Output from this gate is multiplied with previous input and predictions are done.

![Highway temporal model](image)

Figure 12: Highway temporal model

As seen in Figure 12, input_9 layer is the time input which is fed to transform gate and output from carry gate is multiplied with visit_1 layer which corresponds to previous visit input and this output is concatenated with visit_2 layer which represents current visit and then predictions are done.
### 4.7.2.2 Highway single gate

This model will be referred as "HW SG". In highway single gate model, time gap is given as input to transform gate (input_12 in Figure 13). Value obtained from carry gate is multiplied with previous visit information. Also, previous visit(visit_1 in Figure 13) information undergoes non-linear transformation(transformed_data in Figure 13) according to the Equation 7. These two previous visit information is then added together to obtain a final previous visit representation which is concatenated with current visit(input_11 in Figure 13) to make predictions.

![Figure 13: Highway single gate](image)

### 4.7.2.3 Highway MLP

This model will be referred as "HW MLP". In this model, in addition to bypassing information of previous visit based on time, this information is also bypassed separately irrespective of time. As shown in Figure 14, output from carry gate and transform gate is multiplied with previous visit input information. Similarly, learning temporal factor based on time is done by employing transform gate time and carry gate time. Output from these two gates are then concatenated with current visit which is fed as input for making prediction outcomes.
4.7.3 Long short term memory

LSTM models were built as baseline models to compare the performance of proposed highway models. In LSTM, input to the models were current and previous visit information along with time gap information. To see how models perform when full patient history is considered, LSTMs were also implemented with whole patient history. Number of samples in this case reduced to 1974. This model can be seen in Figure 15.
When output labels are created as mentioned in Section 4.4, there is a class imbalance. Class distribution for all prediction outcomes can be seen in Table 9.

<table>
<thead>
<tr>
<th>PREDICTION</th>
<th>CLASS 0</th>
<th>CLASS 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 day readmission</td>
<td>4512</td>
<td>916</td>
</tr>
<tr>
<td>60 day readmission</td>
<td>4105</td>
<td>1323</td>
</tr>
<tr>
<td>90 day readmission</td>
<td>3869</td>
<td>1559</td>
</tr>
<tr>
<td>In-hospital mortality</td>
<td>4970</td>
<td>458</td>
</tr>
</tbody>
</table>

In Table 9, class 0 represents that a patient has not readmitted for readmission outcome prediction and that a patient has not expired in case of in-hospital mortality prediction. Class 1 represents that a patient has been readmitted in the context of readmission prediction outcome and that a patient has expired in case of in-hospital mortality prediction. Class imbalance problem is handled by cost-sensitive learning [24] in which class weights are given for minority class based on class ratio for each prediction outcome.
4.9 NEURAL NETWORK PARAMETERS AND LEARNING

For all neural network models, adam optimizer is used. For binary classification, binary_crossentropy was used and for multi-class classification, categorical_crossentropy is used. In order to avoid overfitting, early stopping was done. Parameter tuning was done for all neural network models to find out some of the hyperparameters like number of hidden nodes, number of hidden layers, dropout ratio, learning rate. In order to find right parameters for each models, hyperparameter tuning is done by splitting the dataset into training, validation and test set. Splitting is done as 80%, 10% and 10% for training, validation and test set respectively. Once hyperparameters were found, results were obtained using 15 fold cross validation.

4.10 EVALUATION CRITERIA

The models are evaluated using Area Under Curve Region Operating Characteristic Curve (AUC ROC) score \[^{40}\] which tells how good a model can distinguish between different classes. Higher the value, the model is able to distinguish class 0 from class 1. These values are based on sensitivity or True Positive Rate (TPR) and specificity. Equations describing sensitivity and specificity can be seen in Equations \(^{11}\) and \(^{12}\) respectively. ROC curve is plotted with FPR in Equation \(^{13}\) on x-axis and sensitivity on y-axis at different cut-off points.

\[
\text{Sensitivity} = \frac{TP}{TP + FN} \tag{11}
\]
\[
\text{Specificity} = 1 - \text{FPR} \tag{12}
\]
\[
\text{FPR} = \frac{TN}{TN + FP} \tag{13}
\]

The reason why ROC AUC scores is used as evaluation criteria is because accuracy is the ratio of number of correct classifications to total number of samples which gives good accuracy even when a model predicts only one class. But in this case with an imbalanced dataset, accuracy is not a good evaluation criteria. In medical data, predicting sick patient is more important. So evaluating the model with ROC AUC, which considers FPR as well, is one of the better choices than accuracy.

ROC AUC scores are reported for binary classification in hospital readmission and in in-hospital mortality predictions and multi class classification for length of stay prediction outcome. Confidence interval tells how precise the results are \[^{41}\]. By precise, we mean that, narrow confidence interval yields a better estimate than a wider confidence interval. All the results are reported with 95% confidence interval for all the models discussed in Section 4.5. Confidence intervals
were calculated from the results obtained using 15 fold cross validation.

The fundamental point of this thesis being influence of time in predicting clinical outcomes, significance of time has been proved by conducting statistical t-test evaluation for the results obtained from models with and without time. The results for models were obtained by bootstrapping technique for 30 iterations and 15 fold cross-validation.
RESULTS AND DISCUSSION

5.1 MODEL PERFORMANCE

5.1.1 Single visit model

In this section, results for all the models are reported using only current visit information.

5.1.1.1 Before feature selection

From Figure 16, it can be seen that Channel-wise neural network performs better than other classifiers in case of with and without time in most of the predictions. This proves an argument in the paper [37] that giving multiple inputs in separate channel improves the performance of a model. However, in case of model with time, XGBoost performs almost same as channel-wise neural network.

From Figure 16, it can also be seen that time plays a significant role in the prediction outcome of patients. To signify the importance of time, t-test was conducted for all models with and without time. P values obtained from t-test was less than 0.01 for all models before feature selection thus proving the importance of time in predicting patient outcomes.
5.1.1.2 After feature selection

From Figure 16, it can be seen that performance of the models has been improved by feature selection mentioned in Section 4.6. Results obtained after feature selection can be seen in Figure 17.

![Figure 17: Single visit models ROC AUC after feature selection](image)

In Figure 17, it can be seen that both neural network and channel-wise network perform similarly after reducing the input dimension by feature selection. Hence, it can be seen that after dimension reduction there is no need for using separate channels for diagnoses and medications.

5.1.1.3 After changing the inputs for different predictions

As mentioned in Section 4.6, inputs were changed for different predictions and results are shown in Figure 18.

![Figure 18: Single visit models ROC AUC after input change](image)

From Figure 18, it can be seen that performance of the models have increased for hospital readmission prediction outcome after adding the input features for hospital score and performance of the models for prediction outcomes in-hospital mortality and length of stay.
decreased after considering the inputs only within 24 hours of admission. This is an expected behaviour because patient information has been restricted to first 24 hours after admission rather than considering the whole information during the visit.

5.1.2 Two visit models

How much of a role does patient history play in clinical outcome predictions? In order to find the significance of patient history information for different prediction outcomes, information about previous visit of a patient has been included along with current visit and the model performance has been discussed.

5.1.2.1 After feature selection

The models evaluated for the inputs after feature selection are MLP concatenated models and highway models. In Figure 19, it can be seen that in case of without time, performance is same due to the fact that in case of highway models, comparison for without time is MLP concat with two visits as inputs without time input.

![Two visit models after feature selection](image)

Figure 19: Two visit models after feature selection

Also from Figure 19, it can be seen that performance of highway models is better but it is not statistically significant. In order to decide the reliability of a model, results are further discussed in Section 5.2.

5.1.2.2 After changing the inputs for different predictions

After changing the input as mentioned in Section 4.6, models evaluated are MLP concatenated, MLP temporal, highway models and LSTM with only two visits and LSTM with full patient history.
From Figure 20, it can be seen that, performance of the models have increased for readmission prediction outcomes and decreased for in-hospital mortality and length of stay predictions. It can also be seen that performance of LSTM has improved when whole patient history is considered in case of readmission predictions whereas in case of length of stay and in-hospital mortality predictions, performance was almost same as LSTM with two visits as input. LSTMs are baseline models.

On the whole, performance of highway models are better than concatenated and forced learning models in case of readmission. But in case of length of stay and in-hospital mortality predictions, simple models like neural network with only current visit as input has performed almost same as the models with two visits as input.

One interesting fact after adding readmission score features in readmission prediction is that in spite of time having significance in the prediction, ROC AUC scores has no much difference for models with time and without time. This can be due to addition of length of stay in the input features for readmission which is also a time related factor indicating how long a patient stays in the hospital during a particular visit. There was no correlation between time and LOS (time related feature) which we thought improved the model performance. The correlation value between timegap and LOS feature is $0.0428$. So, on the whole, the readmission score improved the performance of the model. [42].

5.1.3 Time significance

To prove that time plays significant role in prediction outcomes, statistical evaluation is done by conducting t-test. T-test was conducted for the results obtained from the models with and without time input. When p-value is less than 0.05, the results are significantly different.
It is noted that in all models for input with and without time, p-value was less than 0.01 thus proving that time has significant role in prediction outcomes. ROC AUC scores for 15 fold cross validation in highway MLP for 30 day readmission prediction outcome can be seen in Figure 21.

5.1.4 **Time input in different number of categories**

Does the category split up for time play a role in prediction outcomes? To find this out, we tried different category split up for time in two models - two visit models such as MLP concatenated and highway MLP model.

We have tested by giving number of categories such as smaller or larger time gap between visits and also including more categories. The tested category split up were as follows:

- Two categories: 0 - 30 day time gap and greater than 30 days.
- Two categories: 0 - 60 day time gap and greater than 60 days.
- Two categories: 0 - 90 day time gap and greater than 90 days.
- Six categories: 0-30, 30-100, 100-300, 300-600, 600-2000 and greater than 2000 days.
Figure 22: Different time categories for MLP concatenated and Highway MLP models

From Figure 22, it can be seen that there is no significant difference in performance of the models when categories of time is increased or decreased. So, choice of number of categories of time has no effect on the outcome predictions.

5.2 Model Interpretability

As discussed in Chapters 1 and 2, an important thing to consider when we work with medical data is reliability and interpretability of a model. From the results discussed above, it is evident that time plays an important role in the predictions. But what information does it give? What does the model learn based on time? To investigate this, we have done experiments by changing category of time input. Output from softmax or sigmoid layer has been investigated.

5.2.1 Highway model learning

When time input is given to carry gate, we expect that when time gap between two visits is more, carry gate output should be a smaller value than when time gap between two visits is less. This value is multiplied with previous visit and this information is bypassed. This is what can be seen in Figures 23a and 23b. Hence, learning of gates in highway model based on time input is similar to how we want it to be.
5.2 Model Interpretability

5.2.2 Role of time input in prediction outcomes

5.2.2.1 Learning of the models for time input in four categories

Expected behaviour, according to medical guidelines [43], [44], from the models in case of different predictions are as follows:

- In readmission prediction, as time gap between two visits increases, the risk that a patient might be readmitted within a specific time window is expected to decrease.

- In case of in-hospital mortality and length of stay predictions, increase in time gap between two visits is expected to increase the risk that a patient might either expire in the hospital during a particular visit or that length of stay for a particular visit might be longer.

Now, we will discuss about different models behaviour as we change timegap between visits to different categories. This kind of interpreting the behaviour of model has been discussed in [45]. In order to find out what a model learns from time, we have changed the categorical time input to different categories and obtained output from sigmoid or softmax layer depending upon the prediction outcome. Results are tabulated below for the following models - neural network with single visit, MLP concatenated model with two visits and highway MLP with two visits in Tables 10 and 11. In these tables, all the numbers reported are for whole data, 5428 samples.

Table 10: Behaviour of different models based on change in categorical time input

<table>
<thead>
<tr>
<th>Model</th>
<th>30 Day</th>
<th>60 Day</th>
<th>90 Day</th>
<th>Mortality</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN-one visit</td>
<td>0.4419</td>
<td>0.3318</td>
<td>0.9876</td>
<td>0</td>
</tr>
<tr>
<td>MLP concat-two visits</td>
<td>0.0545</td>
<td>0.1052</td>
<td>0.6728</td>
<td>0</td>
</tr>
<tr>
<td>HW MLP-two visits</td>
<td>0.894</td>
<td>0.9773</td>
<td>0.9740</td>
<td>0.5735</td>
</tr>
</tbody>
</table>
In Table 10, number of samples obeying the medical guidelines above are shown for hospital readmission and in-hospital mortality outcomes. It can be seen that the number of samples following medical guidelines mentioned above are higher in highway MLP model than in MLP concatenated model and single visit models. In case of in-hospital mortality outcome, it can be seen that there are no samples that follows the medical guidelines when time category is changed in neural network and MLP concatenated models. But, the number of samples obeying the medical guidelines for in-hospital mortality in Highway MLP model is considerably high. However, it should also be noted that not all the samples have followed the behaviour mentioned above as there are many other reasons for a patient visiting a hospital like accessibility to the hospital, insurance coverage, diagnosis, to name a few.

Table 11: Behaviour of different models based on change in categorical time input for length of stay

<table>
<thead>
<tr>
<th>MODEL</th>
<th>CLASS 0</th>
<th>CLASS 1</th>
<th>CLASS 2</th>
<th>CLASS 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN-one visit</td>
<td>0.0584</td>
<td>0.2044</td>
<td>0.0155</td>
<td>0</td>
</tr>
<tr>
<td>MLP concat-two visits</td>
<td>0.0923</td>
<td>0.3217</td>
<td>0.0022</td>
<td>0</td>
</tr>
<tr>
<td>HW MLP-two visits</td>
<td>0.9832</td>
<td>0.5934</td>
<td>0.1896</td>
<td>0.0101</td>
</tr>
</tbody>
</table>

In case of length of stay, as the predictions are multi class, we have tried to capture models behaviour for all four classes from the output of softmax layer which is shown in Table 11. Class 3 which is related to longer length of stay has not made predictions in accordance with medical guidelines for any of these models. Even then, by comparing the results from other three classes, highway MLP model has made predictions according to the medical guidelines compared to neural network and MLP concatenated models. Few examples of output from sigmoid and softmax layer, when categorical time input was changed can be seen in Figures 24, 25, 26 and 27.
Figure 24: Categorical risk change in simple neural network model

Figure 25: Categorical risk change in MLP concatenated model
It can be seen that models have made predictions according to medical guidelines to some extent. One interesting thing to note in case of in-hospital mortality prediction results is that, models behaviour was according to expected behaviour until time gap between visits was within 90 days in neural network one visit model and in MLP concatenated two visit model. When time gap between two visits was more than 90 days, instead of risk getting increased, risk got decreased. This is an unexpected behaviour. Although both the models achieve good performance for in-hospital mortality, decision making rule of model does not fall within the medical guidelines thus making it less trustworthy.
5.2 MODEL INTERPRETABILITY

Figure 27: Categorical risk change for length of stay(multiclass) in different models

One another thing to note from Figures 24, 25 and 26 is the values obtained from sigmoid and softmax layer. It can be seen that in neural network and in highway MLP model, confidence of the model in making predictions is more. That is, output of sigmoid or softmax layer is around 0.7 when true class is 1 and the output is around 0.01 or 0.2 when true label is class 0. But in MLP concatenated model, output from sigmoid or softmax layer is always between 0.35-0.55 irrespective of the true label.

5.2.3 Model reliability

In order to check the reliability of a model, reliability curves are drawn for MLP concatenated models and highway MLP models. This helps us to check if the predicted probability distribution is same as the true probability distribution [46]. The predicted probabilities are expected to be close to the actual distribution in each class. That is, the predicted probability distribution should be as close to a diagonal line. The predicted probabilities are distributed into equal number of bins. The number of samples with class 1 is calculated which is plotted against the original distribution. Reliability curve is obtained by plotting predicted probability frequency in x-axis and actual proba-
bility (empirical probability) in y-axis [47]. The curves for models are shown in Figure 28, 29, 30 and 31.

Figure 28: Reliability curve for 30 days readmission

Figure 29: Reliability curve for 60 days readmission

Figure 30: Reliability curve for 90 days readmission
As seen from Figures 28, 29, 30 and 31 for 30 days readmission, 60 days readmission, 90 days readmission and in-hospital mortality respectively, it is evident that reliability curves of highway MLP and MLP concatenated models are close to the diagonal. This shows that predicted probability distribution is similar to actual probability distributions.

5.2.4 Mean of features to learn about model’s decision

Mean of handpicked features for all the predictions in case of TP, FP, TN and FN were calculated. Apart from mean of handpicked features, mean of diagnoses codes and medication codes were also calculated. The actual mean of features were calculated to compare with the actual distribution. In case of readmission difference of features was in time. Time gap was less if a patient had readmission and greater if a patient had no readmission which can be seen in Figures 32a, 32b and 32c. In case of in-hospital mortality, there was difference in age feature. The mean of age was higher for patients who expires in the hospital than for patients who does not expire in the hospital in that particular visit which is shown in Figure 32d. The trend was same in models with both single and two visits. The mean was taken for neural network single visit model, MLP concatenated with two visits and highway MLP model. In all these cases, trend was the same.
Mean of features for 30 day readmission

Mean of features for 60 day readmission

Mean of features for 90 day readmission

Mean of features for in-hospital mortality

Figure 32: Mean of features
5.3 COMPUTATIONAL COMPLEXITY

Apart from reliability curve, computational complexity of MLP concatenated and highway MLP models were also calculated. The average computation time of each epoch and average number of epochs were calculated for highway MLP and MLP concatenated model which can be seen in Tables 12 and 13.

Table 12: Computation time and epochs of models for MLP concatenated model

<table>
<thead>
<tr>
<th>PREDICTION</th>
<th>COMPUTATION TIME PER EPOCH</th>
<th>AVERAGE EPOCHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 day readmission</td>
<td>0.544298 s</td>
<td>36.0</td>
</tr>
<tr>
<td>60 day readmission</td>
<td>0.445986 s</td>
<td>36.0</td>
</tr>
<tr>
<td>90 day readmission</td>
<td>0.443212 s</td>
<td>36.0</td>
</tr>
<tr>
<td>In-hospital mortality</td>
<td>0.508277 s</td>
<td>37.0</td>
</tr>
</tbody>
</table>

Table 13: Computation time and epochs of models for HW MLP model

<table>
<thead>
<tr>
<th>PREDICTION</th>
<th>COMPUTATION TIME PER EPOCH</th>
<th>AVERAGE EPOCHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 day readmission</td>
<td>0.597779 s</td>
<td>31.0</td>
</tr>
<tr>
<td>60 day readmission</td>
<td>0.455066 s</td>
<td>31.0</td>
</tr>
<tr>
<td>90 day readmission</td>
<td>0.512756 s</td>
<td>31.0</td>
</tr>
<tr>
<td>In-hospital mortality</td>
<td>0.543104 s</td>
<td>35.0</td>
</tr>
</tbody>
</table>

From Tables 12 and 13, it can be seen that computation time for each epoch in MLP concatenated model is lesser than highway MLP model. The total computation time was calculated using Equation 14.

Computation time = Time for each epoch * Average number of epochs * Number of folds

(14)

Table 14: Total computation time for MLP concat and HW MLP models

<table>
<thead>
<tr>
<th>PREDICTION</th>
<th>MLP CONCAT</th>
<th>HW MLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 day readmission</td>
<td>293.9213 s</td>
<td>277.9672 s</td>
</tr>
<tr>
<td>60 day readmission</td>
<td>240.8329 s</td>
<td>211.6061 s</td>
</tr>
<tr>
<td>90 day readmission</td>
<td>239.3347 s</td>
<td>238.4319 s</td>
</tr>
<tr>
<td>In-hospital mortality</td>
<td>282.0939 s</td>
<td>285.1296 s</td>
</tr>
</tbody>
</table>

When computation time is calculated for 15 fold cross-validation for all epochs as seen in Table 14, total computation time of highway
MLP model is lesser than MLP concatenated model for readmission prediction. But, total computation time for MLP concatenated model is lesser than highway MLP model for in-hospital mortality prediction.

5.4 SUMMARY

To summarise which model (MLP concatenated or Highway MLP) performed better, Tables 15 and 16 illustrates performance of MLP concatenated and highway MLP models based on ROC AUC scores, interpretability (behaviour of the model when time category is changed), reliability and computation complexity for different predictions. The significance of time for different prediction outcomes can also be seen in Table 15.

<table>
<thead>
<tr>
<th>OUTCOMES</th>
<th>TIME SIGNIFICANCE</th>
<th>ROC</th>
<th>INTERPRETABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 days</td>
<td>Yes</td>
<td>Highway MLP</td>
<td>Highway MLP</td>
</tr>
<tr>
<td>60 days</td>
<td>Yes</td>
<td>Highway MLP</td>
<td>Highway MLP</td>
</tr>
<tr>
<td>90 days</td>
<td>Yes</td>
<td>Highway MLP</td>
<td>Highway MLP</td>
</tr>
<tr>
<td>In-hospital mortality</td>
<td>Yes</td>
<td>Highway MLP</td>
<td>Highway MLP</td>
</tr>
<tr>
<td>LOS</td>
<td>Yes</td>
<td>MLP concatenated</td>
<td>Highway MLP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OUTCOMES</th>
<th>RELIABILITY</th>
<th>COMPUTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 days</td>
<td>Highway MLP</td>
<td>Highway MLP</td>
</tr>
<tr>
<td>60 days</td>
<td>Highway MLP</td>
<td>Highway MLP</td>
</tr>
<tr>
<td>90 days</td>
<td>Highway MLP</td>
<td>Highway MLP</td>
</tr>
<tr>
<td>In-hospital mortality</td>
<td>Highway MLP</td>
<td>Highway MLP</td>
</tr>
</tbody>
</table>

From Tables 15 and 16, it can be seen that Highway MLP model has performed better than MLP concatenated model in terms of ROC AUC, interpretability, reliability and computational complexity.

5.5 RESEARCH ANSWER

From the experiments conducted and the results obtained, it is evident that visit level irregularity is one of the important factors to be considered when making clinical outcome prediction. Although ROC AUC scores did not differ much when time was given as a feature or
when time was used to calculate the importance of previous visit, from the aspect of model interpretability it can be seen that using time input to calculate the importance of previous visit was better than using time gap as feature. If this time gap should be used as categorical or continuous input or number of categories for this input did not add much difference to the results.
6 CONCLUSION

6.1 CONCLUSION

From the results discussed in Chapter 5, it can be seen that visit level irregularity in EHR data is an important feature to consider in prediction outcomes. The role of time is significant in all prediction outcomes in all models.

Also, from results it can be seen that for prediction outcomes like in-hospital mortality and length of stay, have similar performance in models with single and two visits. In case of readmission predictions, model performance has improved with two visit input. So, patient history plays an important role in readmission predictions than in-hospital mortality and length of stay prediction outcomes.

In order to see importance of time as an input, time was given as a feature along with other inputs and also was used as a factor to measure the importance of previous visit of the patient. From results obtained for this setup, it can be seen that both the models (MLP concatenated and Highway MLP) performed almost similar for all predictions. It can also be seen that in reliability curve, highway MLP model is reliable. So, using these kind of gating mechanisms with MLP is feasible and proves to be simpler model than LSTM and MLP concatenated models in terms of complexity.

6.2 LIMITATIONS

There are certain constraints with the study done with MLP models (highway networks and concatenated models). Full patient history has not been considered in any predictions for the above mentioned models. Obviously, number of channels can be increased to consider previous visit’s information. However, this comes with a cost. More number of channels leads to more number of parameters. Hence, more amount of memory is required to consider whole patient history if the patient has more number of visits and also input dimension will increase accordingly.

Patients with only one admission has been neglected from the point of readmission prediction as it will lead to more class imbalance. All the predictions are also restricted to one specific cohort. However, this can be extended for whole MIMIC-III database without cohort restric-
As mentioned in Section 4.9, no separate test set was used for testing these models. Instead cross validation was used in which, testing set may have been used for training, which might have led to peeking in the data.

6.3 Discussion

There are few keypoints to note in this thesis. In general, performance of the models for readmission predictions is around 0.65. Achieving good performance in readmission predictions was quite difficult considering the fact that readmission of a patient may also depend on other factors like planned readmission, insurance type, accessibility to the hospital for a patient, medical insurance coverage and so on. Some of these information are unavailable in this dataset.

Med2vec representation has not been tested with these sort of predictions before. It is evident that readmission prediction can be worked on to improve the results. However, with this model, predictions like in-hospital mortality or length of stay can be made.

In this thesis, the main focus was on visit level irregularity in EHR data. Easy solution was by just adding this irregularity as a feature. Then, a different approach of handling the visit level irregularity using MLP instead of LSTMs was proposed. Simple solution of using irregularity as a feature and MLP with gating mechanisms gave almost similar results in terms of ROC AUC scores. To make conclusions about model performance, steps like interpreting a model by playing with categorical time feature, reliability of models and computational complexity were carried out. A significant difference between these models was seen when categorical time input was changed which can be seen in Tables 10 and 11.

Also, in this thesis, visit level irregularity has been handled using two visits. If prediction outcomes can be done with less patient history, then necessity to store large amount of data can be restricted to some extent. One more important point to emphasize here will be that privacy breach of patients will be reduced to a greater extent.

6.4 Future Work

This study can be extended to consider all the visits of a patient but dimension will increase when considering both diagnoses and medications. So, instead of diagnoses and medication codes, different features like hospital score for readmission, comorbidity index,
scores indicating the severity of illness in a patient (for example SAPS, APACHE, OASIS) and similar other scores can be used as features [48]. By doing so, problem of sparsity in data and high dimensionality of data can be addressed. Also, features suggested by clinical experts can be used as input based on the availability of information in dataset and models can be tested.

This model can also be used to address feature level irregularity, that is, if a particular diagnosis has occurred in many visits of a patient, importance of particular diagnosis can be calculated using this model.

Highway layers have been used as additional memory with LSTMs in speech recognition. In similar way, highway layers can also be used in LSTMs as an additional memory to handle visit level irregularity with EHR data.


[39] François Chollet et al. Keras. [https://keras.io](https://keras.io), 2015.


Ranjani Subramanyan
Master's in Embedded and Intelligent Systems
Halmstad University

Nandhini Subramanyan
Master's in Embedded and Intelligent Systems
Halmstad University