Department of Electrical and Electronic Engineering

Final Year Project Report 2023

Imperial College London

Project Title:	**Machine Learning Based Clinical Decision Support for Individualised Antibiotic Readmission Prediction**
Student:	**Boon Liang Wong**
CID:	**01700213**
Course:	**MEng EEE Technical Stream**
Project Supervisor:	**Professor Pantelis Georgiou & Mr William Bolton**
Second Marker:	**Professor Esther Rodriguez Villegas**

Plagiarism Statement

I affirm that I have submitted, or will submit, an electronic copy of my final year project report to the provided EEE link.

I affirm that I have submitted, or will submit, an identical electronic copy of my final year project to the provided Blackboard module for Plagiarism checking.

I affirm that I have provided explicit references for all the material in my Final Report that is not authored by me, but is represented as my own work.

I have used ChatGPT v4 as an aid in the preparation of my report. I have used it minimally to improve the quality of my English, however all technical content and references comes from my original text.

Acknowledgements

Firstly, I would like to express my deepest gratitude to Mr. William Bolton for all his support. He has been very helpful in giving me constructive feedback and advice on different ways to approach this project. His knowledge and experience in the field of machine learning for healthcare have assisted me in tackling the challenges of the project. Despite facing a lot of obstacles, his encouragement has been driving me forward.

I would also like to thank Professor Georgiou Pantelis for giving me the opportunity to work on this project. Despite being very busy, he has made a lot of effort to set up meetings to keep track of my progress as well as address my concerns about this project.

I would like to thank my partner, Vicky for providing me with unconditional support during my studies at Imperial College London. She has been giving me a lot of emotional and mental support.

Furthermore, I would also like to thank my sister, Mandy Wong, a practicing NHS clinician for her advice in selecting relevant clinical features in the data preprocessing steps. She has also helped me in understanding the general antibiotics treatment guideline and different medical terms as well as how each of them relates to ICU patients.

Last but not least, I would like to thank my parents for their unconditional love and support throughout my life.

Abstract

Antimicrobial resistance (AMR) is one of the leading causes of death worldwide. In 2019, bacterial AMR was found to be associated with approximately 4.95 million of deaths across 204 countries with 1.2 million of them directly attributed to it. Misuse and overuse of antibiotics are significant drivers in the development of AMR. To deal with AMR, a multi-modal approach is needed including antimicrobial stewardship to preserve the effectiveness of currently available agents. Recent research has focused on using electronic health records (EHRs) for infection diagnoses and antibiotic therapy selection through machine learning (ML), but little work has focused on antimicrobial stewardship. Inadequate, ineffective or incomplete antibiotic treatment often leads to antibiotic retreatment (antibiotic readmission), causing unnecessary and excessive use of antibiotics.

This project applied electronic health records of 2,189 intensive care unit (ICU) patients from the MIMIC-IV database to develop ML-based decision support models using deep learning approach **to reliably predict whether ICU patients will be retreated with antibiotic if current antibiotic treatment is discontinued**. This project is formulated and addressed as both classification and regression tasks. Accurately predicting antibiotic readmission (antibiotic retreatment) for ICU patients could be extremely helpful in optimising antibiotic treatment to combat AMR by providing individualised predictions to support the decision on continuation or cessation of antibiotic treatment.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

This chapter details the overview of antimicrobial resistance (AMR), its threats toward ICU patients, the rising cost of ICU beds and machine learning in healthcare as well as the objective and structure of this report.

1.1.1 Overview of Antimicrobial Resistance (AMR)

In 2019, World Health Organization (WHO) declared Antimicrobial Resistance (AMR) as one of the top 10 most alarming health concerns because it causes antimicrobials specifically antibiotics to have reduced effectiveness and efficacy in preventing and treating infections [14, 15]. More specifically, AMR happens when microorganisms like bacteria that cause infections become increasingly difficult to treat as antimicrobials such as antibiotics no longer kill or inhibit their growth [16, 17]. Inappropriate use of antibiotics is one of the major drivers of AMR as bacteria that survive antibiotic treatment can adapt mechanisms to develop resistance towards previously effective antibiotics [17, 18, 19]. According to one of the studies supported by the UK Government in 2014, AMR is predicted to result in 10 million of deaths annually and a 2-3.5% drop in the national Gross Domestics Product (GDP) by 2050, potentially leading to a loss of $100 trillion worldwide [20, 21]. This impact is primarily due to the lack of new antimicrobials discovery, rise in failure of antibiotic treatments, infections during medical procedures and decreased quality of life [22]. During the initial phase of the COVID-19 pandemic, there were many uncertainties about whether antimicrobials are effective in treating COVID-19 [23]. As a result, there was a sudden

change in antibiotic prescribing in the hospital where antibiotics are often overused and this intensified the issues of AMR across the globe [23, 21]. Therefore, drastic measures at all levels have to be taken to prevent the spread of AMR and to avoid it from causing higher mortality rate, prolonged hospital stay and more expensive treatment in the future [24].

1.1.2 AMR Threats Towards ICU Patients and Rising Cost of ICU Beds

Patients in the intensive care unit (ICU) are generally most impacted by AMR. Extensive research shows that antibiotic treatment is given to 70% of patients in ICU [25, 26]. Usually, a combination of broad-spectrum drugs (antibiotics that are efficient against common disease-causing bacteria) are administered to ICU patients with infections [25, 22]. However, this has contributed to the rapid development of multidrug-resistant pathogens (MDR) among ICU patients [27, 28, 21]. In a recent study, it is revealed that a significant proportion (30-60%) of antibiotic treatment for ICU patients is poorly prescribed as the treatment is either unnecessary or inadequate [29, 30, 31, 32, 33]. Therefore, the problems of AMR and MDR are further exacerbated. In ICU, there is also an increased risk of bacterial infection due to hospital ecology, frailty as well as the prevalence of invasive devices such as intubation [34, 27]. In fact, a study focusing on ICU patients in Europe indicates that around 20% of ICU stays were identified to acquire infection during the period of medical treatment and this is known as nosocomial infection [35, 27]. Hence, AMR is a growing and persistent threat to ICU patients.

Furthermore, medical care in ICU is extremely expensive. The daily cost of an ICU bed is found to be 3 times the cost of a normal hospital bed in Canada [36]. As the population around the world ages, the costs associated with ICU care are expected to increase [36]. Therefore, optimal use of ICU beds will be crucial to minimise cost and ensure efficient allocation of medical resources.

1.1.3 Machine Learning in Healthcare

As the world enters the age of Big Data, machine learning has been applied extensively across a lot of industries including healthcare. The availability of Electronic Health Records (EHRs) of patients further enables academic research to be conducted on different fields of medical study using ML. Research on ML in healthcare generally includes the data

preprocessing pipeline that consists of data cleaning and feature extraction as shown in part A of figure 1.1 [1]. The machine learning algorithms can be generally grouped into 3 main areas, supervised learning, unsupervised learning and reinforcement learning (Part B of figure 1.1) [1]. The main difference between the 3 areas is that supervised learning requires training the model with the actual ground truth known as label while unsupervised learning and reinforcement learning are algorithms that do not require any label [1]. Supervised machine learning is used as the main approach in this project for antibiotic readmission prediction.

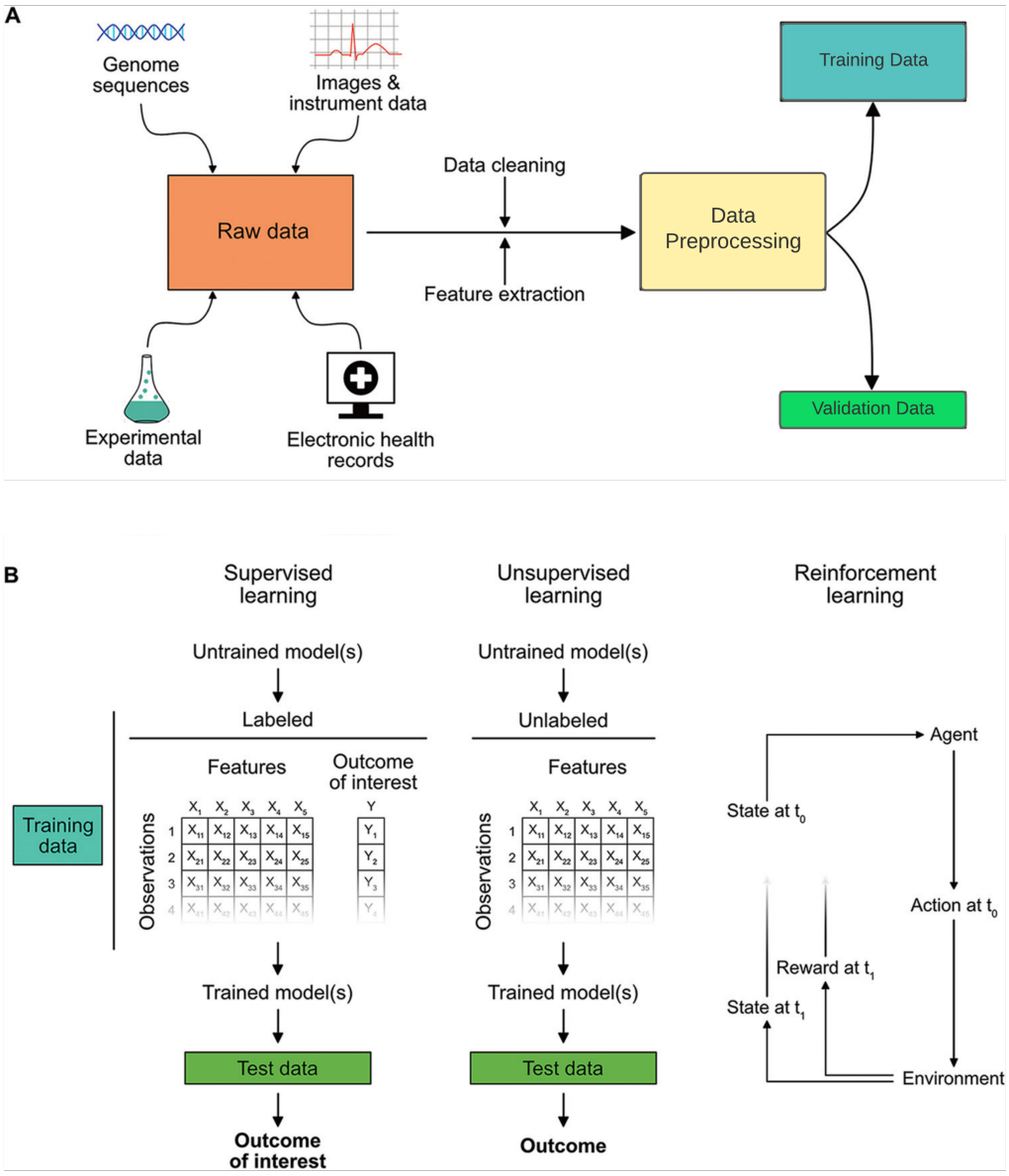

Figure 1.1: Overview of Machine Learning Approaches in Healthcare - A refers to data preprocessing and B refers to machine learning algorithms (Taken from [1])

1.1.4 Machine Learning for Antimicrobial Stewardship

Antimicrobial stewardship programmes (ASPs) are initiatives that aim to provide support so that antibiotics are used optimally and to determine treatment plans that are the most targeted and shortest in duration without sacrificing effectiveness [37]. Many nations have acknowledged AMR as a serious threat and committed to taking global action to address it as a top priority [21]. While ASPs have shown to be successful on a small scale, they often require a large amount of resources for manual data mining, data management and clinician feedback which makes it difficult to expand them to a larger scale. One such example is the statistical models-based clinical decision support systems (CDSSs) designed to assist clinicians in antimicrobial prescribing. These models used causal probabilistic network and they act as prediction models [38, 39].

However, most ASPs to date are not based on powerful techniques such as ML which allow us to make use of a large amount of health data available and provide individualised recommendations which until now was not really possible. To summarise, ML-based CDSS could be a very practical way to tackle AMR and act as useful tool for accelerating ASPs intervention while providing reliable predictions to clinicians.

1.2 Research Objective and Structure of Report

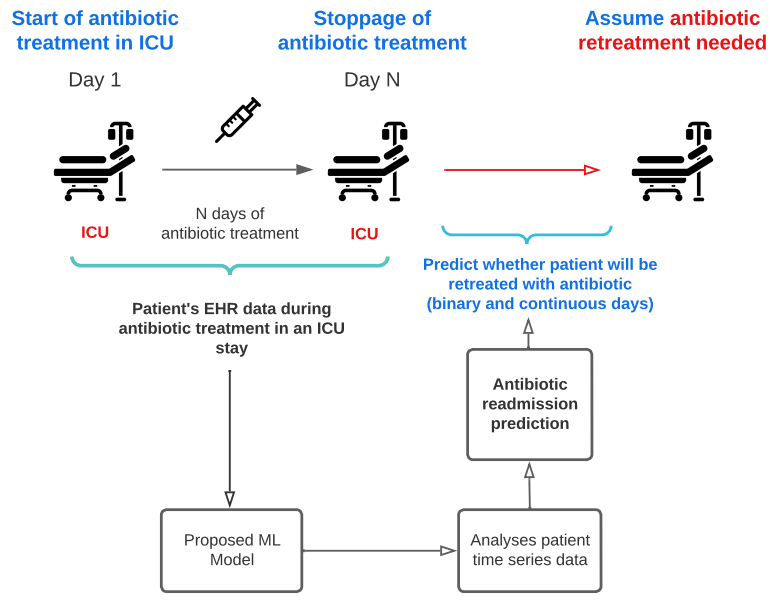

Figure 1.2: Proposed ML-based decision support model

Objective:

This project designs and develops ML-based decision support models using deep learning approach to reliably predict antibiotic readmission for patients in the ICU. Antibiotic readmission can be interpreted as antibiotic retreatment or the need for additional antibiotic treatment after initial course of antibiotic treatment has been completed but deemed failed or ineffective. Note that the terms "antibiotic retreatment" and "antibiotic readmission" are used interchangeably in this report. An example is illustrated in figure 1.2 to explain how the model works. Antibiotic readmission is highly undesirable as it leads to unnecessary overuse of antibiotics which contributes to the development of AMR. It is often linked with failures in the initial course of antibiotic treatment, encompassing misuse of antibiotics, undertreatment or ineffective treatment which can necessitate antibiotic retreatment.

The core concept here is to alleviate the possibility of antibiotic readmission in the ICU population by providing decision support to the clinicians to make an informed decision regarding the continuation or cessation of antibiotic therapy. If the likelihood of antibiotic readmission (prediction) for a patient is high, clinicians can choose to either prolong the antibiotic treatment or explore alternative treatment plan for timely intervention. If the likelihood is low, clinicians can choose to rely on the prediction to cease the treatment to prevent excessive use of antibiotics. This helps in promoting optimal antibiotic usage by optimising antibiotic treatment duration to tackle the issues of AMR.

This project is tackled as both binary classification and regression problems. Hence, multiple deep learning models are discussed in this report. To be specific, deep learning model trained on binary label (yes/no) will provide clinicians with a binary prediction on antibiotic readmission whereas model trained on continuous label (days) will provide a quantitative estimate of how many days apart the patients will likely be retreated with antibiotic again if current antibiotic treatment is stopped. Joint learning method is also explored where the model will provide binary and regression predictions simultaneously.

Aims of this project are to address the following topics:

- *In the task of predicting antibiotic readmission for ICU patients, is the deep learning approach using multivariate time series data a probable way that can provide reliable prediction in supporting the clinical decision on antibiotic treatment cessation?*

- *Compare the performance of deep learning models such as LSTM and CNN with baseline model like multilayer layer perceptron (MLP)*
- *Investigate and tackle this prediction task as both binary classification and regression problems and produce reliable prediction results*

Main contributions from this report are:

- *Constructed data preprocessing pipeline to generate dense time-series representation as input for deep learning models and identified a range of routinely collected EHR clinical variables that are meaningful for antibiotic readmission prediction*
- *Explored different advanced deep learning architectures like LSTM and CNN for antibiotic readmission prediction as well as their performance in both classification and regression tasks*
- *Proposed a deep learning architecture with promising performance in antibiotic readmission prediction - both binary and regression tasks*
- *Explored joint learning method to improve the prediction performance by training the model with both binary and continuous labels*
- *Provided critical evaluation of the results and interpreted the logic behind the prediction*

Challenges in dataset creation:

In terms of training and development of the ML model, this project will be utilising Electronic Health Records (EHRs) of ICU patients from MIMIC-IV. However, EHR or time series data is very challenging to process. For instance, during an ICU stay, patients generate a lot of biomarkers over time (heart rate, respiratory rate, SpO2, etc). Generally, a substantial number of biomarkers are required for accurate prediction. This means that the data has high dimensionality which easily leads to curse of dimensionality where analysing or visualising the data to identify patterns becomes extremely difficult. EHRs are also sensitive data that require patients' consent before it is used in any way. Hence, the number of training samples is significantly less than the number of dimensions. In addition to that, some biomarkers are recorded at different timestamps and frequencies together with a high level of missingness in some categories of biomarkers, this presents an extra

challenge for data preprocessing before feeding into any machine learning model. Hence, a data preprocessing pipeline has to be designed to resolve all the issues and this is discussed in the "data preprocessing" section.

In summary, this project starts by addressing challenges in EHR data, identifying and extracting important biomarkers. After that, raw data is converted into a clean dataset in order to explore the temporal and spatial features of the data. Finally, supervised deep learning models are developed to accurately predict antibiotic readmission of patients in the ICU. The performance of models is benchmarked against relevant baseline model using selected metrics.

This report is structured as follows:

- **Background** -
  - **Key Approach and Importance of Antibiotic Readmission Prediction** - This section discusses about key strategy to tackle AMR
  - **Related Works** - This section discusses about related works and their limitations
  - **Tackling Antibiotic Readmission Prediction** - This section discusses about formulation of machine learning problem as well as technical approach, challenges and techniques to tackle antibiotic readmission prediction
  - **Machine Learning** - This section discusses about mathematical background related to machine learning
- **Methodology**
  - **Dataset Preprocessing** - This section explains the steps that are involved in the data preprocessing pipeline
  - **Machine Learning Methods** - This section explains about the techniques used to address imbalanced data and the details related to training and validation of models
  - **Machine Learning Model Structure** - This section explains about the rationale behind the design of each deep learning model
  - **Model Evaluation** - This section talks about performance metrics that are selected to evaluate the models in both binary classification and regression tasks

- **Results**
  - **Data Analysis** - This section discusses about the characteristics of the processed dataset including the training set and test set
  - **Binary Classification Results** - This section discusses about results obtained from binary classifiers
  - **Regression Results** - This section discusses about results obtained from regression models
  - **Joint Learning Results** - This section discusses about results obtained from joint learning model
- **Discussion**
  - **Features Importance Analysis** - This section explains about the results from feature importance analysis
  - **Further Discussion** - This section discusses about the challenges and achievements in this project
  - **Limitations** - This section provides the limitations of this project
  - **Further Work** - This section provides the future work of this project
  - **Conclusion** - This section provides the conclusion of this project

Chapter 2

Background

This chapter details the importance of the prediction of antibiotic readmission. Apart from that, this chapter discusses related works on readmission prediction and proposes an approach in tackling the antibiotic readmission prediction task.

2.1 Key Approach and Importance of Antibiotic Readmission Prediction

2.1.1 Key Approach

Optimising antibiotic usage is the key approach for this project to combat AMR. Determining the ideal antibiotic treatment length is difficult as stopping treatment too early or too late will threaten the patient's condition and accelerate the development of antibiotic resistance [40, 41, 42, 43, 44]. Clinicians often need years of experience and expertise to decide the optimal treatment duration for a patient and the suitable timing to stop antibiotic treatment is particularly difficult to determine.

Since there is a lack of experience and expertise in this area, antibiotic treatment is often prescribed using general guidelines which are not customised to each patient. Therefore, the treatment decision seldom takes characteristics specific to the patient into careful consideration and hence is unable to determine the optimal antibiotic treatment, leading to underuse, misuse and overuse of antibiotics which are the major causes of AMR [40, 45]. Moreover, these contribute to a rise in the failure of antibiotic treatment which could lead to multiple consequences. For example, undertreatment or ineffective treatment

causes incomplete eradication of bacteria where surviving bacteria continue to multiply after treatment is completed, resulting in relapse of infection (symptoms of infection re-occur). Bacteria could also generate antibiotic resistance which can cause infection to persist. In short, failures in antibiotic treatment increase the risk of requiring antibiotic readmission / antibiotic retreatment.

Hence, I propose using deep learning models trained using routinely collected EHRs to support antibiotic optimisation decisions. More precisely, the models will be trained with multivariate time series data and the respective antibiotic readmission labels (binary or continuous value) in a way that the models are able to discover underlying temporal and spatial patterns to identify ICU patients who are likely to receive antibiotic readmission (binary or days) accurately. The prediction outcome can be used subsequently to assist the decision-making of antibiotic treatment continuation or cessation. As stated previously, based on the risk of antibiotic readmission, clinicians can choose to prolong or cease the antibiotic treatment as well as explore alternative treatments such as choosing different antibiotics. Thus, this optimises antibiotic treatment in terms of duration to ensure effective, useful antibiotic treatment can be delivered and to avoid misuse, underuse and overuse of antibiotics. With the ultimate aim of mitigating the development of AMR. The following part will elaborate on the importance of this prediction.

2.1.2 Relevance of Antibiotic Readmission Prediction

Figure 2.1 describes how antibiotic readmission prediction can serve as an important tool to support physicians in different ways; combat antibiotic resistance, improve the quality of treatment in ICU, enhance medical decision making and optimise costs for the hospital.

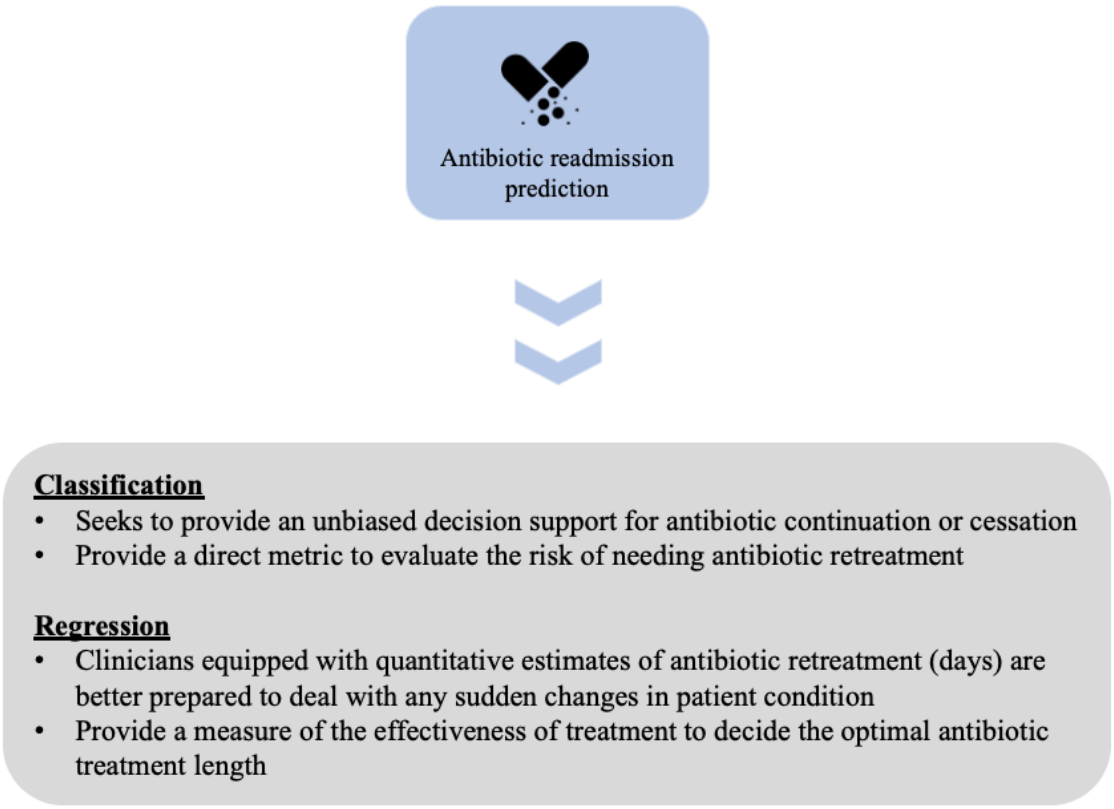

Figure 2.1: Relevance of Antibiotic Readmission Prediction

In addition, accurately predicting antibiotic readmission for ICU patients can significantly reduce the risk of patients being readmitted to the ICU and maximise the quality of healthcare for the patients. Furthermore, it could help reduce patient's ICU length of stay which could yield considerable cost savings and hospital resources can be allocated in the most optimal way. Through early invention, relapse of infection can also be prevented and improve patient outcomes as well as mitigate the risk of complications due to worsening of infection. Last but not least, side effects from prolonged antibiotic treatment can be minimsed by optimising the antibiotic treatment duration.

2.2 Related Work

Despite the importance of machine learning for antimicrobial stewardship, very few focused on mid-late stages of decision making such as treatment duration and cessation which are very important. Our key contribution in this project is to optimise the antibiotic treatment duration by providing individualised antibiotic readmission prediction through deep learning approach to improve the accuracy of prediction.

By analysing the underlying trends and exploiting the time series nature of EHR data, this project hopes to demonstrate how machine learning model trained on EHR data can contribute substantial improvements to enhance the patient's state representation, yielding state-of-the-art results on antibiotic readmission prediction.

2.2.1 Related Work on AMR, ICU and Hospital Readmission Prediction

In recent years, numerous ML-based clinical decision support models trained on patient data have been developed to tackle a range of problems in healthcare. For instance, ICU and hospital readmission prediction have received significant interest where a lot of machine learning models were proposed to guide the decision on the timing of discharge from ICU and hospital [46, 47, 48]. There are also a substantial amount of ML models implemented to understand, treat, and prevent AMR [40, 49] which focused more on infection risk prediction [50, 51], infection diagnosis [49], antimicrobial surveillance [52] and antimicrobial prescribing [53]. Nonetheless, a paper from Imperial provided a good foundation for this project to move ahead. In this paper, the model predicts the mortality outcomes and length of stay of patients under antibiotic temporality using long short-term memory (LSTM) autoencoder [40]. Inspired by this paper, I believe that antibiotic readmission prediction could provide another perspective to clinicians in optimising antibiotic treatment duration.

Since research in antibiotic readmission prediction is very limited or almost non-existent in the academic world, this chapter highlights related works on ICU and hospital readmission predictions as these papers are very relevant to this project in terms of technicality [46, 54, 55, 56, 57]. Although they are not in the context of AMR, it still provides valuable insights in tackling this project. The following table summarises the machine learning problems in these papers, the model(s) used in each paper and the possible shortcomings or limitations.

ML Problems	ML Methods
• Predicting Intensive Care Unit Readmission with Machine Learning Using Electronic Health Record Data [46] ☐ Binary classification task	• **Model:** Decision-tree-based ML algorithm • **Time series / temporal relationship: ✗** • **Possible shortcoming(s):** Temporal relationship is not explored, limited utility
• Interpretable Deep Learning Framework for Predicting all-cause 30-day ICU Readmissions [54] ☐ Binary classification task	• **Model:** LSTM (Temporal features), DNN (Static features) and XGD (Student layer) • **Time series / temporal relationship:** ✓ • **Possible shortcoming(s):** Difficulty in handling missing data, limited utility, time series data was not processed into regular sequences
• Early prediction of ICU readmissions using classification algorithms [55] ☐ Binary classification task	• **Model:** A total of 8 classifiers were used: Naive Bayes, J48, Random Forest, Sequential Minimal Optimisation, JRip, AdaBoost, Logit Boost, Iterative Classifier • **Time series / temporal relationship: ✗** • **Possible shortcoming(s):** Limited generalisability mentioned in paper
• Improving hospital readmission prediction using individualized utility analysis [56] ☐ Binary classification and regression tasks	• **Model:** Gradient boosted decision trees • **Time series / temporal relationship: ✗** • **Possible shortcoming(s):** Temporal relationship is not explored
• Forecasting Hospital Readmissions with Machine Learning [57] ☐ Binary classification task	• **Model:** SVM and Random Forest • **Time series / temporal relationship: ✗** • **Possible shortcoming(s):** Temporal relationship is not explored

Table 2.1: Current methods in predicting readmissions

Summary of possible limitations in current methods of readmission prediction:-

1. Most readmission predictions are formulated as binary classification machine learning problem, this might causes low interpretability of prediction and limited utility to clinicians. In this project, antibiotic readmission prediction is tackled as both binary classification and regression task, so multiple models could be used to provide more clinical utilities.
2. Detailed data preprocessing steps were not well explained in these papers
3. Did not explore both spatial and temporal relationships that exist in EHRs data
4. Deep learning approach was not well explored in readmission prediction

2.3 Tackling Antibiotic Readmission Prediction

2.3.1 Formulation of Machine Learning Problem and Technical Approach

As mentioned previously, the project formulates antibiotic readmission prediction as both binary classification task and more challenging regression task to provide more utilities to clinicians as well as joint learning task where the model outputs binary and regression predictions simultaneously.

General technical approach:-

1. Systematic data processing pipeline is designed to mitigate challenges in EHR data
2. Relatively large number of relevant features (>20) are extracted and analysed
3. Time series data is converted to interpretable data structure for analysis
4. In order to explore both inter-features and temporal relationships of data, various deep learning architectures will be tested to find out the most suitable model
5. On top of that, different machine learning techniques such as cross validation, stratified sampling, oversampling and feature scaling are used to properly train and validate the deep learning model
6. Testing will be conducted on separate held-out test set to evaluate the performance of model on unseen data and confirm findings

2.3.2 Challenges in Analysis of EHR

In the era of technology, Electronic Health Records (EHRs) of patients are becoming readily available and this provides valuable data for training machine learning models. However, EHRs data stored in different databases commonly exist a range of problems that makes data extraction and cleaning extremely challenging. Generally, the issues resolve around large number of missing observations, errors in data entries, units not being standardised and extreme outliers. In addition, the data is usually stored within complex relational database and sometimes with incomplete documentation about the database, leading to an extra level of difficulty in navigating the database to find relevant information.

Apart from that, the data from certain databases can be very complex without clinical expertise as some clinical data is stored as free-text like doctor's notes or comments and some other clinical variables are recorded using different units of measurement, for example, temperature readings can be recorded in either Celsius or Fahrenheit. In order to extract and create a dataset to study the relevant population in the ICU, a data preprocessing pipeline has to be carefully designed to address all these problems.

2.3.3 Technique to Retain Temporal Feature of Data

As mentioned in the introduction chapter, EHR data is rich time series data and biomarkers of a patient are sampled at different frequencies with a high level of missingness in certain categories of biomarkers. 2 common ways to tackle these issues are either to transform the time series data into a standard static data or regular temporal data, known as static transformation and temporal abstraction [58, 59, 60]. To credit the original author, the findings of retaining temporal features of data through static transformation and temporal abstraction are classified and summarised by this paper [60] which are used for the following elaboration.

Using static transformation, a pre-defined set of variables/biomarkers will be represented using a pre-specified timestamp [58, 59, 60]. For instance, if the red blood cell count of an ICU patient is measured every day and blood pressure is measured every hour, under static transformation, these biomarkers can be represented by values like most recent recorded red blood cell count and most recent measured blood pressure. These data are then used by standard ML classifiers like the random forest, SVM and decision trees to make the

prediction. Although this approach facilitates the prediction process with dimensionality reduction, the temporal relationship of data is ignored and underlying patterns are not explored, resulting in information loss [58, 59, 60].

Whereas temporal abstraction converts time series data into regular temporal data (uniformly represented data) by dividing each time series data into windows of equal length and values in each window is represented by statistical measures (mean, median, variance and etc) [58, 59, 60]. Abstraction and aggregation can reduce noise in data and avoid problems like different frequencies of sampling and missingness [61, 60]. Taking the previous example about red blood cell count and blood pressure, under temporal abstraction, these biomarkers are first aggregated by day and each value can be represented as the mean of red blood cell count and mean of blood pressure each day. This approach retains the temporal relationship of data and reduces information loss [60]. However, the performance of temporal abstraction is highly dependent on the size of windows (daily, hourly) and choices of statistical measures [60]. This means proper assumptions have to be made accordingly by understanding the characteristics of each feature which requires healthcare expertise / knowledge [60].

2.3.4 Possibility of CNN as Feature Extraction Layers

In the recent decade, Convolutional Neural Network (CNN) has found a lot of success across different areas of machine learning applications especially in image classification and computer vision tasks. CNN utilises convolutional operation with a pre-specified "grid" that slides across the data to extract the spatial relationships of the data. Other than spatial data like images, this architecture has also achieved different level of success using time series data as its ability to analyse and extract hidden patterns from longitudinal type data is exceptional [48] as shown in figure 2.2. Usually, multiple CNN layers are used as the feature extraction layers to create rich and meaningful latent representation which is fed into other structures like fully connected neural networks (FCNN) for further computation to produce the final output.

Hence, in this project, it would be worthwhile to discover and test the limit of CNN as feature extraction layers for multivariate time series data. However, the nature of CNN computation only allows it to focus on regional spatial data as the kernel slides[60]. Due to this, the ordering of features column in the data can be particularly significant in making

sure that CNN is able to generate useful features map [60]. In this paper [60], it is discussed that different configurations for CNN could be used to bypass this issue and the models have managed to achieve a relatively good performance for time series data.

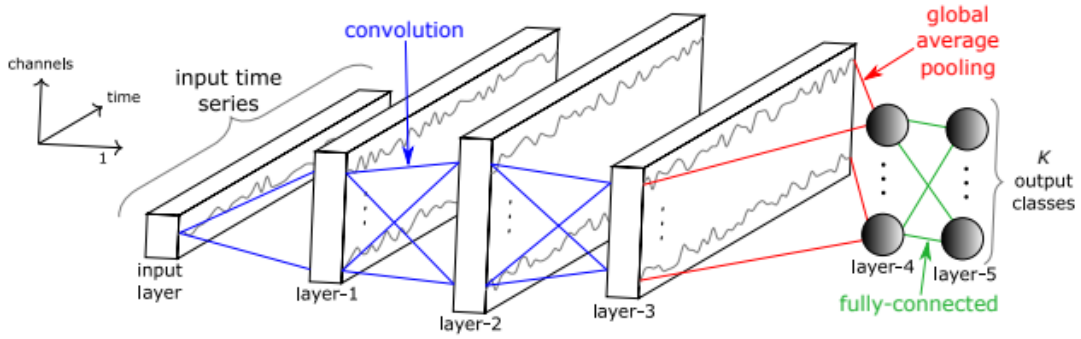

Figure 2.2: CNN for Time Series Data (Taken from [2])

2.4 Machine Learning

This section provides an overview of background theory and technical concepts that are used in the deep learning models of this project.

2.4.1 Fully Connected Neural Network (FCNN)

Neural network is one of the most important concepts in machine learning. As the name suggests, it is made up of neurons which are the processing units of the network. The operation behind a single neuron is shown in figure 2.3.

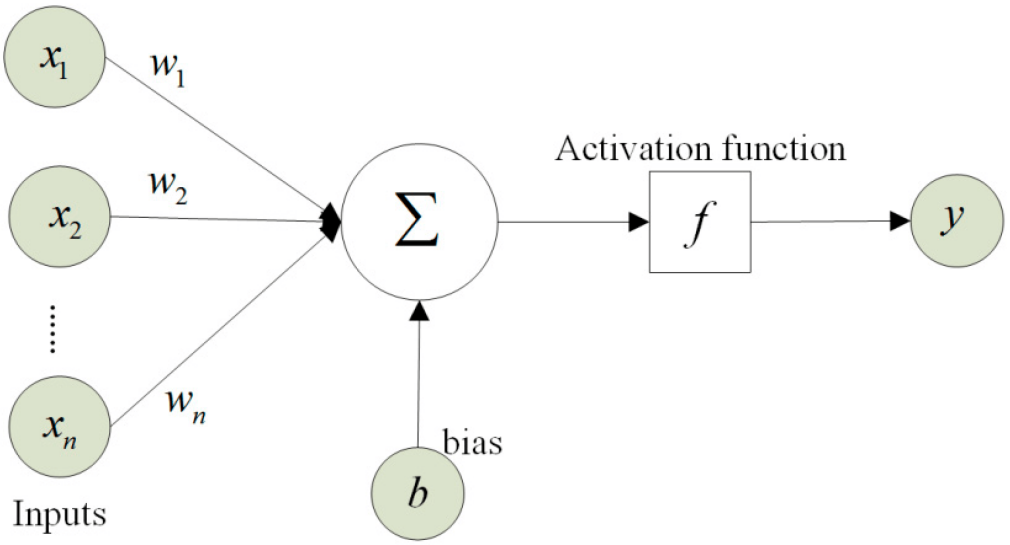

Figure 2.3: Artificial Neuron (Taken from [3])

Input, x and weight, w are multiplied together in the neuron and bias, b is added to their product. It is then passed into an activation function like ReLU to produce an output. The activation function is used to help capture complex patterns in the data.

A fully connected neural network (FCNN) is shown in figure 2.4. It is made up of layers of neurons; input layer, hidden layers and output layer. During forward propagation, input data is passed into the neural networks to generate prediction output. The error between output and ground truth is calculated using loss function to update the weights in the network through backpropagation. This process is iterative until the training process is finished. FCNN is commonly used as the final prediction layers in the ML architecture.

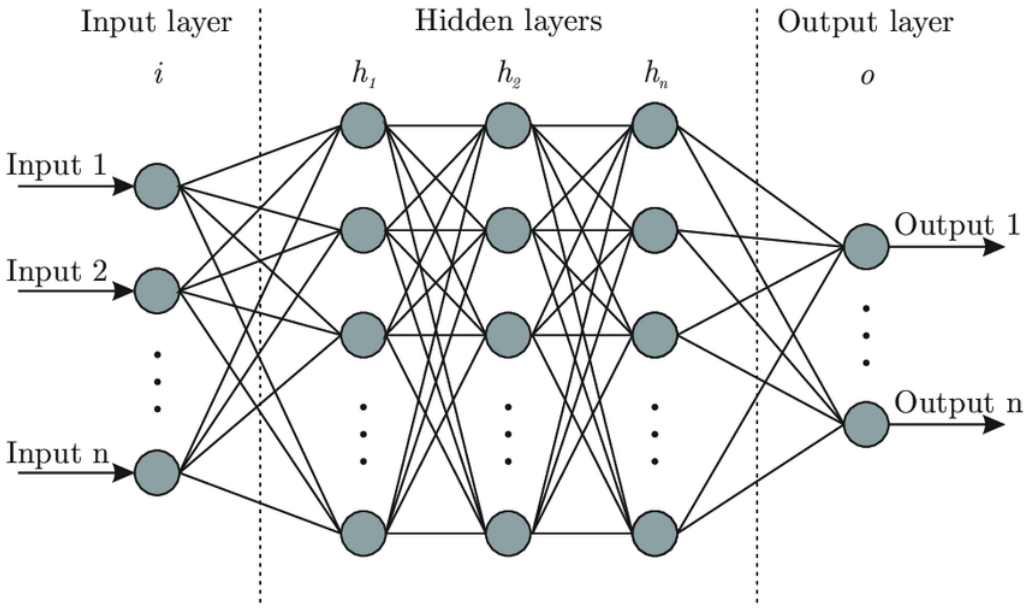

Figure 2.4: Fully Connected Neural Network (Taken from [4])

2.4.2 CNN

Convolutional neural network (CNN) is a special type of neural network that utilises convolutional layer to extract patterns from spatial data like images. It is widely used in image classification and computer vision tasks.

For ease of explaining, Convolution 2D is used in the following elaboration. Conv2D uses a convolution filter (multiple kernels form a filter) to convolve through the data to compute the feature maps by analysing the spatial relationships of data. Kernel size parameter is one of the parameters in the convolutional layer that set the size of the kernel and weights on the kernel are updated during training. 2D convolutional operation is illustrated in figure 2.5.

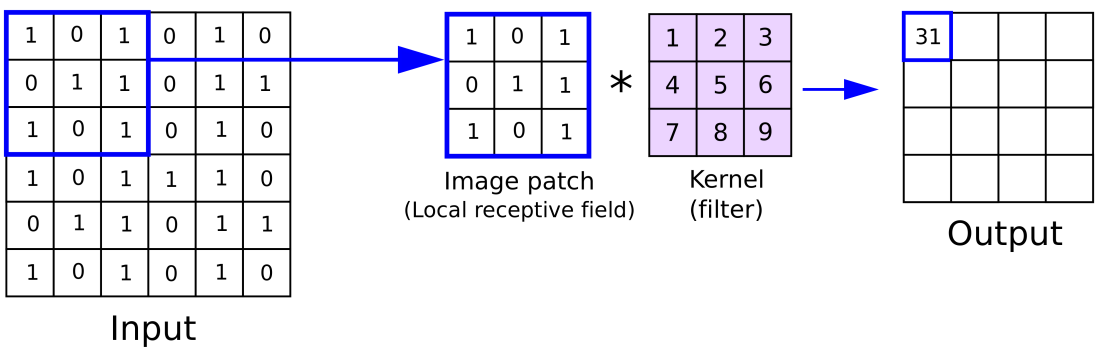

Figure 2.5: Convolution and Kernel (Taken from [5])

2.4.3 LSTM

Long Short Term Memory (LSTM) is generally seen as a more advanced version of Recurrent Neural Network (RNN) that is capable of processing long sequential data. Traditional RNN suffers from vanishing gradient problems as it is unable to remember or process long sequences of data. During backpropagation, the gradient becomes too small which affects the process of updating weights. To resolve this, LSTM utilises gates to retain or throw information in the data so that only key information is kept for final prediction. It has shown great success in the areas of time series forecasting, speech recognition and text translation.

Essentially, there are 3 main gates in a unit of LSTM, namely forget gate, input gate and output gate. Figure 2.6 shows the forget gate and input gate in LSTM.

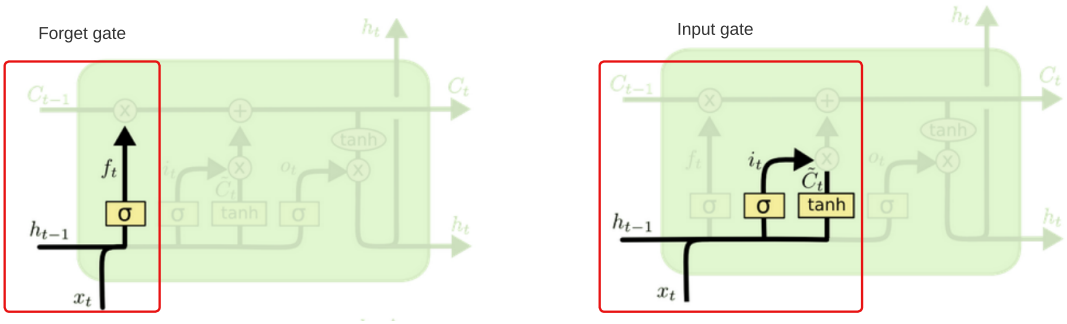

Figure 2.6: Forget gate and input gate (Adapted from [6])

In forget gate, previous internal cell state, h_{t-1} and input, x_t are passed into the sigmoid function to decide which information should be kept or down-weight as sigmoid function limits the output to the range of [0,1]. Output of forget gate is denoted as f_t. While for input gate, h_{t-1} and x_t are passed into both sigmoid and tanh functions. Tanh function helps to "scale" the magnitude between [-1,1] and the output from tanh function is denoted as C_t whereas sigmoid output is denoted as i_t. Mathematically,

$$f_t = \theta(W_f[h_{t-1}, x_t] + b_f)$$

$$i_t = \theta(W_i[h_{t-1}, x_t] + b_i)$$

$$\hat{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$$

where W refers to weight, b refer to bias and θ is the sigmoid function.

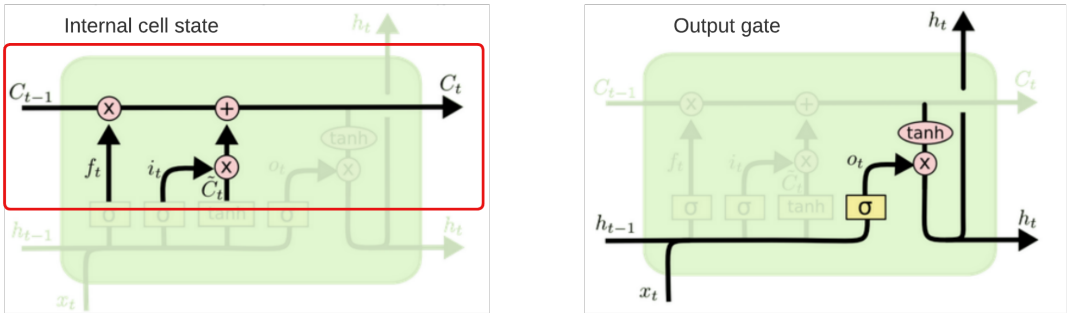

Figure 2.7: Updated internal cell state and output gate (Adapted from [6])

Internal cell state is updated using previous cell state, C_{t-1} with outputs from forget gate and input gate as shown in figure 2.7. h_{t-1} and x_t are also passed into output gate to produce o_t. o_t and c_t are multiplied and used as hidden state for subsequent computation. Mathematically, it is shown below,

$$C_t = f_t \otimes C^{t-1} \oplus i_t \otimes \hat{C}_t$$

$$o_t = \theta(W_o[h_{t-1}, x_t] + b_o)$$

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

An advanced architecture based on LSTM was also widely used by ML community where it allows information to flow in both directions and this structure is known as Bidirectional LSTM (BiLSTM) illustrated in figure 2.8. BiLSTM is capable of handling past and future information which is ideal for analysing the patient's condition during the antibiotic treatment.

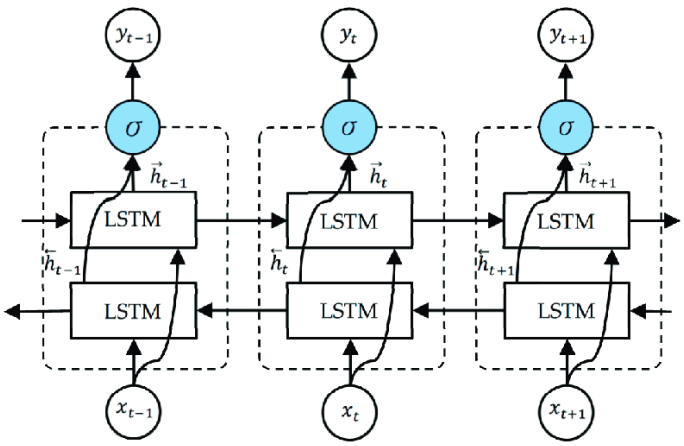

Figure 2.8: Bidirectional LSTM architecture (Taken from [7])

Chapter 3

Methodology

This chapter describes the data preprocessing pipeline for the antibiotic readmission prediction task with detailed elaboration on each step of the pipeline. This chapter also explains about MIMIC-IV which is the database used in this project.

3.1 Data Preprocessing

3.1.1 MIMIC-IV

MIMIC-IV is a medical database that has collected Electronic Health Records (EHRs) from over 40,000 patients who were admitted to Beth Israel Deaconess Medical Center (BIDMC) [62, 63]. This database is made accessible to the public and is widely used for healthcare academic and clinical research but access to the database is subject to completion of CITI Program's "Data or Specimens Only Research" course. All the data are deidentified to protect patients' identities following Health Insurance Portability and Accountability Act (HIPAA) Safe Harbor provision [62, 63].

Most of the patients' data are gathered and compiled from BIDMC's ICU department spanning 2008 to 2019 [62, 63]. In ICU, high-quality data of patients are often collected as these patients are constantly monitored. In general, clinical measurements for ICU patients are recorded at a higher frequency than the patients in regular wards. This data usually includes information such as patient admission, length of hospital stay, laboratory measurements and vital signs which has gained significant interest from researchers across various fields.

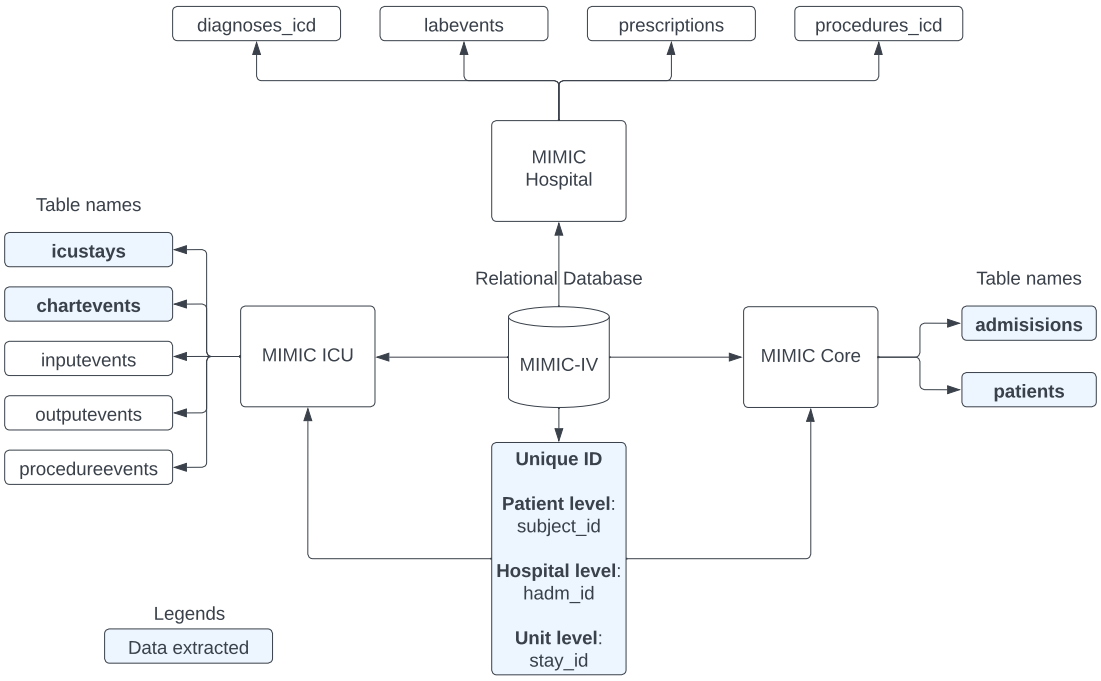

Figure 3.1: Simplified MIMIC-IV Relational Database Structure v1.0

MIMIC-IV adopted a relational database structure and categorised data into modules and tables to support human interpretation of data structure. For this project, the data preprocessing pipeline uses 2 modules, MIMIC Core and MIMIC ICU. MIMIC Core contains the patient's general information such as admission date and time, gender, ethnicity and age. While MIMIC ICU contains ICU-level data such as vital signs, lab results and medication-related information.

Diagram 3.1 shows how relevant information can be extracted from different modules and tables in the MIMIC cloud database, BigQuery through Custom Structured Query Language (SQL) using unique IDs of the patient, such as *subject_id*, *hadm_id* and *icustay_id* to navigate through the complex relational database. *subject_id* is the Unique ID related to the patient itself, *hadm_id* is the ID related to hospital admission and *icustay_id* is the ID related to ICU admission. This means that each patient *subject_id* can have multiple *hadm_id* because the patient can be admitted to the hospital multiple times. Similarly, every *hadm_id* can be associated with multiple *icustay_id* as the patient can be admitted to ICU multiple times in 1 hospital stay.

3.1.2 Overview of Data Preprocessing Processes and Project Framework

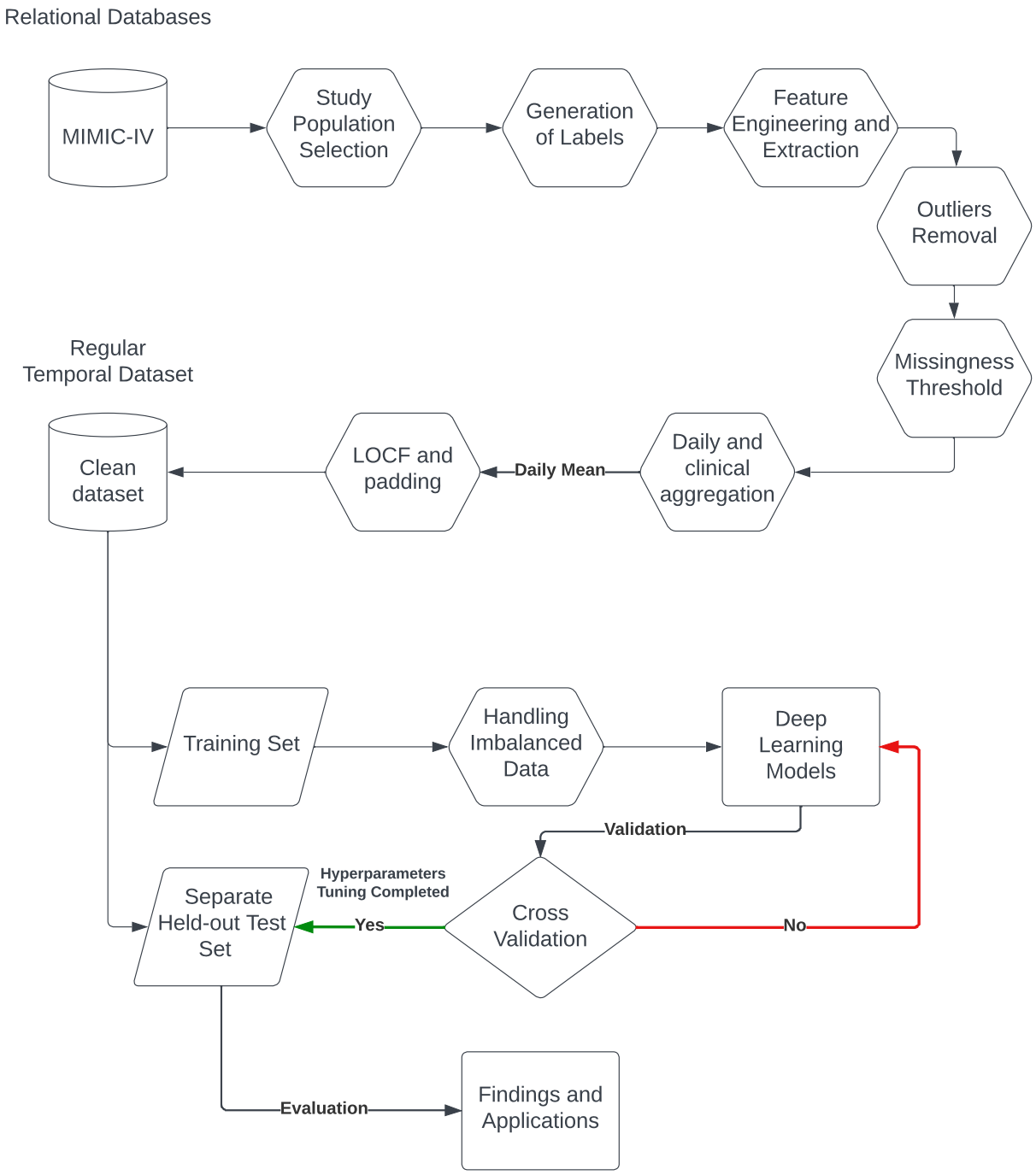

Figure 3.2: Data Preprocessing and Project Framework

The data preprocessing pipeline is splitted into 3 main stages; identifying study population + labels generation, extracting features/clinical variables and time series data construction which consists of daily aggregation, data imputation (LOCF) and padding to create fixed-size time series data for training the deep learning models.

3.1.3 Setting and Study Population

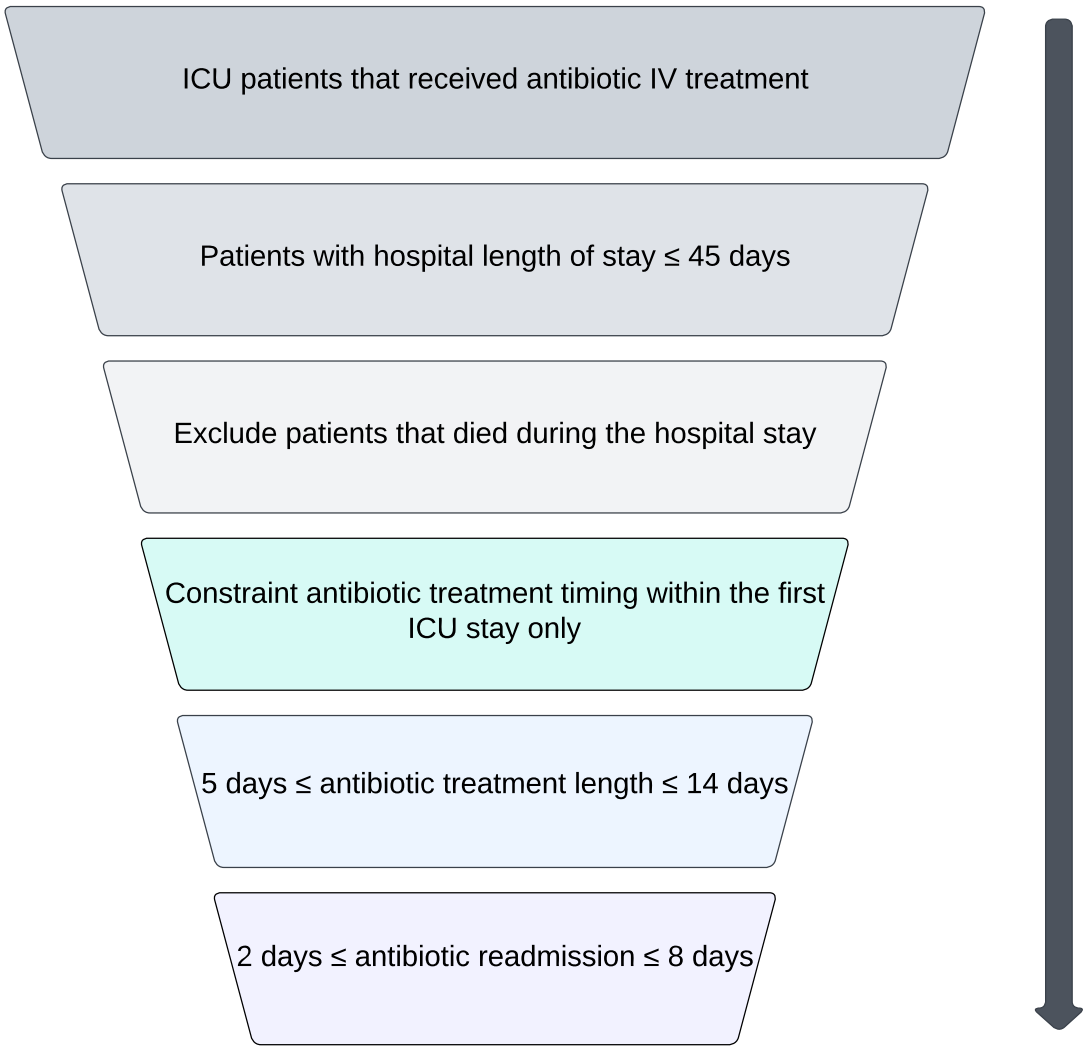

Figure 3.3: Filtering Process

The study population is selected by filtering ICU patients that received intravenous (IV) antibiotic treatment with a hospital length of stay (HLOS) of less than or equal to 45 days. ICU patients are chosen as these patients have comprehensive clinical data and antibiotic cessation is a pertinent question. The filter choice of limiting the patient population to those who received IV antibiotic treatment is designed to narrow down the cohort size specifically to those who received intensive antibiotic intervention for infectious diseases. It also removes outliers from the data given HLOS basically decreases exponentially after 45 days for most patients and those with HLOS longer than 45 days will likely be in a very critical condition, making it not suitable to train the model. After that, only patients that survived during the hospital stay will be retained and patients that died during or after

the antibiotic treatment will be excluded. For instance, the patient can die due to infection before the planned antibiotic treatment is fully executed, therefore, stoppage of antibiotic treatment is due to death rather than the doctor's decision. Hence, by excluding patients that died, negative cases (patients that did not receive antibiotic retreatment) will only consist of patients that discharged from the hospital after complete and effective antibiotic treatment while it is much safer to assume that positive cases consist of patients that received retreatment due to failures in the initial course of antibiotic treatment, leading to worsening of condition due to relapse of infection or other reasons. This clear separation between negative and positive cases can reduce noise in data and allow efficient training of the model.

Cases where start or stoppage of antibiotic treatment happened in the normal ward or 2nd ICU stay are removed to constraint the timing of antibiotic treatment in the first ICU stay only (i.e. start and stoppage of treatment must happen in a single ICU stay). This is important as the primary aim of the model is to study the patient's physiological state across the antibiotic treatment duration especially during the start and stoppage of treatment. For example, antibiotic treatment can last from first ICU stay until normal ward or even 2nd ICU stay. However, in normal ward, patients are not closely monitored so clinical data such as vital signs are often not measured which causes extra difficulties in understanding patient's conditions because of significant missing data. Hence, these cases are removed. Generally, patients will be staying in the normal ward before admitting to 2nd ICU stay so it is hard to track the patient's condition for the whole treatment duration, therefore, cases that are readmitted to 2nd ICU stay are also removed.

According to a clinical colleague, a duration of 5-14 days of antibiotic treatment is typically given to ICU patients. Doctors face greater difficulty in deciding when to discontinue antibiotic treatment for ICU patients within the 5-14 day range. Not only that, patients that received prolonged antibiotic treatment are at higher risk of developing AMR [64]. By limiting patients to those with antibiotic treatment larger than or equal to 5 days but smaller or equal to 14 days, antibiotic readmission prediction through a trained ML model can provide more utility to physicians in these scenarios. This step is also extremely crucial in producing denser time series representation where there are more clinical data per patient to work with. In the exploration phase, it is found that sparsity in time series data especially in antibiotic readmission prediction leads to detrimental model performance

while denser time series data improve the quality of prediction significantly. This is mainly due to the fact that there is no single clinical variable that can reliably indicate or suggest the completion of antibiotic treatment for an ICU patient. Therefore, more data per patient will be required to train the model.

Finally, for all these positive cases, only antibiotic retreatment that happened within 8 days after the stoppage of initial course of antibiotic treatment is included. This ensures that positive cases where ICU patients are retreated with antibiotics are mainly due to failures in the initial course of antibiotic treatment including incomplete or ineffective treatment rather than new infections outside of ICU or any other unpredictable reasons. In simple terms, 8 days after the first antibiotic treatment is completed, it is hard to identify whether patients are retreated due to new infection or antibiotic treatment failures. Therefore, all positive cases with antibiotic retreatment of larger than 8 days must be removed. 1 day antibiotic retreatment is also removed as there is a risk of entry error by nurses or doctors, leading to false positive cases. This filtering process is a result of a significant amount of research and clinical inputs as well as trial and error for generating high-quality dataset.

3.1.4 Labels Generation

To train supervised deep learning models, the following labels/outcomes of interest are extracted from MIMIC-IV data.

1. **Antibiotic readmission (antibiotic retreatment)**:

    - **Definition**: Antibiotic readmission is defined as the duration (days) between the start of new antibiotic treatment and the stoppage of the previous antibiotic treatment
    - **Continuous label**: The number of days between 2 consecutive antibiotic treatment (calculated by subtracting start date of new antibiotic treatment against the stoppage date of previous antibiotic treatment, the dates are rounded to nearest day for each patient). This label ranges from 2 days to 8 days
    - **Binary label**: If continuous label is positive, the binary label is recorded as "True" otherwise "False"

Note that all antibiotic retreatment that happened in either the normal ward or 2nd ICU stay are captured in this step and all the retreatment refer to intravenous antibiotic treatment. The processed labels were obtained from a cohort of 2,189 unique ICU stays. Out of 2,189 ICU stays, it is found that 17.5% of them or 383 cases are retreated with antibiotics with a mean treatment gap (antibiotic readmission) of 0.71 days and a standard deviation of 1.73 days. This is an imbalanced dataset as there are less than 20% of ICU stays with antibiotic readmission and specific machine learning technique is used to address these issues which will be explained in the machine learning method section.

3.1.5 Feature Extraction, Outliers Removal and Missingness Threshold

3 main categories of features are extracted for developing the antibiotic readmission prediction model; namely chart events, demographic information and computed features based on other clinical variables. Chart events are clinical data that are collected regularly which are related to patient's vital signs. These vital signs are measured using medical equipment or blood test such as heart rate, SpO2, blood pressure, white blood cells, etc. These data are used to train the model to interpret and understand patients' physiological conditions. Apart from that, demographic information of each patient is also extracted; age, gender and ethnicity as these data proved to increase the quality of prediction. 2 additional features that are computed using other clinical variables are also included. Overall, 39 most relevant features (34 chart events features, 3 demographic features and 2 computed features) are selected to train the model.

Chart events

A total of 914 features/biomarkers were extracted from MIMIC-IV's chart events and analysed based on correlation, relevance and input from clinical experts. Note that the biomarkers of each ICU stay are extracted during the antibiotic treatment period only. Pearson Correlation Coefficient, r is used to analyse the relationship between different features.

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}}$$

The correlation coefficient measures how closely any two sequences of features in data are correlated. This project also consulted a practicing clinician, Dr Mandy to identify important and frequently used biomarkers that are most helpful in understanding the overall condition of an ICU patient. A total of 34 clinical variables or features are extracted from MIMIC-IV chartevents tables. These features are mostly numerical data and routinely collected data through medical equipment or blood test. As min max normalisation will be applied at the later stage, extreme outliers will squeeze normal range data close to 0. Therefore, outliers have to be removed in advance. For instance, heart rate of larger than 500 should not be included as this is an obvious measurement error or entry error. Table 3.1 summarises the included range for each chart events feature. Entries outside of these ranges are removed and treated as missing entries.

Variables	Unit	Included Range
GCS - Eye Opening	n.a	1-4
GCS - Motor Response	n.a	1-6
GCS - Verbal Response	n.a	1-5
Heart Rate	BPM	33-175
Respiratory Rate	insp/min	5-51
O2 saturation pulseoxymetry	%	52-100
Temperature Fahrenheit	°F	35-105
Arterial Base Excess	mEq/L	-73.25-25
Lactic Acid	mmol/L	0-15.7
PH (Arterial)	pH	6.96-7.79
Arterial CO2 Pressure	mmHg	35-45
Arterial O2 pressure	mmHg	14-118
ALT	IU/L	0-8445
AST	IU/L	5-9285
Alkaline Phosphate	IU/L	9-2999
Anion gap	mEq/L	-7-47
HCO3 (serum)	mEq/L	7-49
Total Bilirubin	mg/dL	0-49.4
Calcium non-ionized	mg/dL	3-20
Chloride (serum)	mEq/L	61-142
Creatinine (serum)	mg/dL	0-14
Glucose (serum)	mg/dL	25-869
Magnesium	mg/dL	0.85-1.1
Phosphorous	mg/dL	1.12-1.45
Potassium (serum)	mEq/L	3.5-5.5
Sodium (serum)	mEq/L	0.9-8.8
Hemoglobin	g/dl	4.7-18.3
Platelet Count	K/uL	10-1208
Prothrombin time	Secs	9-120
PTT	Secs	17-150
WBC	K/uL	0-206
Hematocrit (serum)	%	12.7-56.5
SpO2 Desat Limit	%	35-100
Richmond-RAS Scale	n.a	-5.0-4.0

Table 3.1: Chart events features

Demographic Information

Apart from clinical variables, demographic information is also found to be very important in predicting antibiotic readmission. The demographic data is obtained from the MIMIC-IV's admissions table. 3 key information, namely age, ethnicity and gender are extracted and included as key features. Since gender and ethnicity are text-based data (static features), label encoding is applied to these features for ease of processing. Table 3.2 summarises the details of demographic features.

Variables	Valid Range
Age	18-97
Gender	Male, Female
Ethnicity	White, Asian, Black, Hispanic, Indian, Unknown, Other, Unable to Obtain

Table 3.2: Demographic Data

Computed features

Another 2 extra features are computed based on other variables. First, the "total duration" is computed and it is defined as the duration from admission into ICU until stoppage of antibiotic treatment. Note that antibiotic treatment can start a few days after the patient is admitted into ICU or at the same time when patient is admitted into ICU. Last but not least, antibiotic treatment duration is also calculated by subtracting the stoppage date against the start date of treatment. Table 3.3 summarises the details of computed features.

Variables	Valid Range
Total duration	5-24 days
Antibiotic treatment duration	5-14 days

Table 3.3: Computed features

Missingness Threshold

Selection of features also involved calculating the level of missingness for all available features and applying a missingness threshold of 70%. This means that features with larger than 70% level of missingness are discarded as insufficient data is detrimental to model performance. For instance, procalcitonin (PCT) and C-reactive protein (CRP) with more than 90% missingness level are removed despite being some of the primary biomarkers in determining stoppage of antibiotic treatment [65].

Dense vs Sparse - Time Series Data

Figure 3.4 below shows the correlation heatmap for the chosen features (dense time series representation). Whereas figure 3.5 shows the correlation heatmap for similar features but using data (sparse) that did not go through the filtering process mentioned in figure 3.3 obtained during exploration phase of project. Table 3.4 is also included to summarise the level of missingness for selected features before data imputation. Since correlation analysis assumes continuous and complete data, it is found that heatmap is a great method in visualising sparsity of the data and understanding inter-feature relationships. If the data is sparse, the correlation between closely related clinical variables/features tends to

be very small due to missing data and the results of correlation analysis will be very misleading. This can be seen in the heatmap in figure 3.5 (Most of them in deep blue color). Meanwhile, the correlation heatmap in figure 3.4 shows higher correlation (light blue) between closely related features. Denser data allows deep learning models to exploit the temporal relationships of patient data to give more reliable predictions.

Input Features	Level of Missingness (%)
Gender	0.00
Age	0.00
Ethnicity	0.00
Total Duration	0.00
Antibiotic Treatment Duration	0.00
Heart Rate	0.61
O2 saturation pulseoxymetry	0.68
Respiratory Rate	0.80
GCS - Eye Opening	0.93
GCS - Motor Response	0.98
GCS - Verbal Response	1.01
Sodium (serum)	2.28
Potassium (serum)	2.30
Chloride (serum)	2.31
Creatinine (serum)	2.33
HCO3 (serum)	2.37
Anion gap	2.45
Glucose (serum)	2.56
Hematocrit (serum)	2.82
Hemoglobin	2.94
WBC	2.94
Platelet Count	2.97
Magnesium	3.30
SpO2 Desat Limit	3.57
Phosphorous	4.14
Calcium non-ionized	4.50
Temperature Fahrenheit	5.31
Richmond-RAS Scale	16.61
Prothrombin time	34.02
PTT	34.66
PH (Arterial)	52.78
Arterial O2 pressure	53.14
Arterial CO2 Pressure	53.15
Arterial Base Excess	53.16
Lactic Acid	61.08
ALT	65.70
AST	65.75
Total Bilirubin	65.84
Alkaline Phosphate	66.28

Table 3.4: Summary of Missingness Level of Input Features Before Data Imputation (Pre-Outliers Removal)

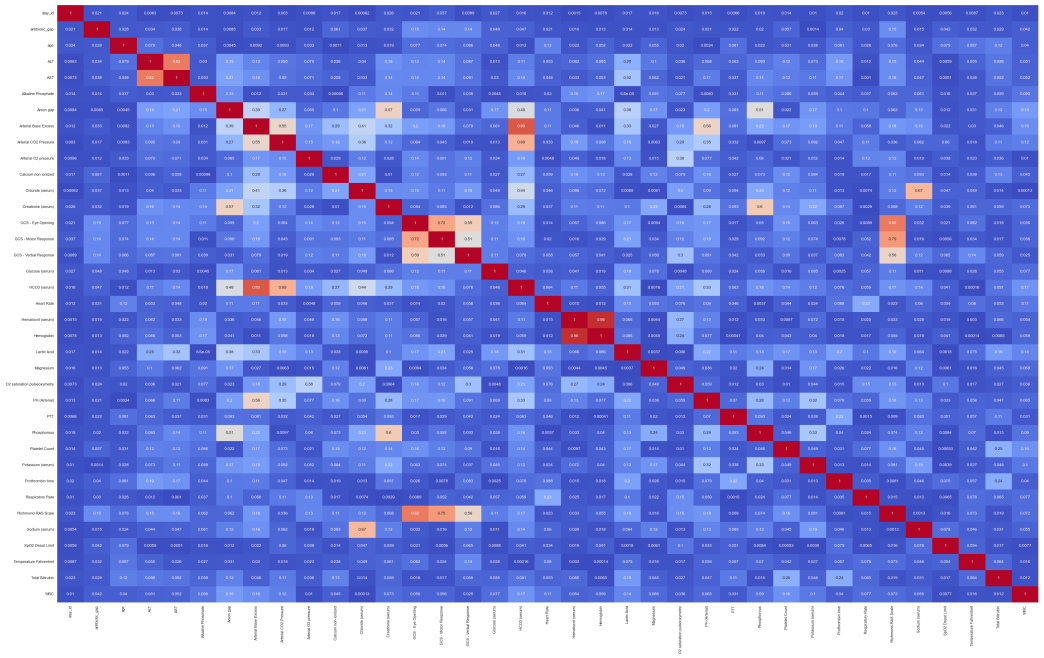

Figure 3.4: Correlation Heatmap for Dense Data

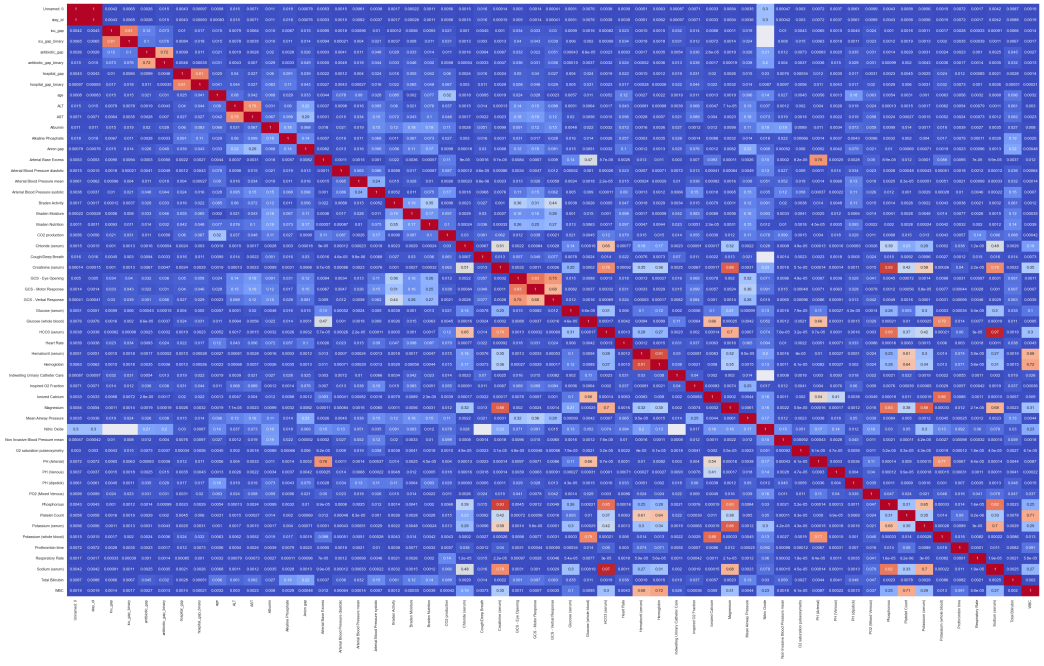

Figure 3.5: Correlation Heatmap for Sparse Data Including Patients with Short Treatment Length (obtained during exploration phase where more features are included)

3.1.6 Daily Aggregation and Data Imputation

The MIMIC-IV database provides precise timestamps for laboratory measurements and vital signs with a resolution in units of seconds. Despite this level of detail, many of the measurements are sampled at different frequencies, resulting in sparse time series data for some input features. For example, heart rate is measured multiple times per day while lactic acid is only measured when deemed required by a doctor. To make the data easier to work with and suitable for modern machine learning models that require denser time representations, these clinical observations or measurements throughout the antibiotic treatment in the ICU stay are grouped into daily intervals (i.e. the chart time is rounded up to day). This technique is aligned with the previously introduced temporal abstraction method in "technique to retain temporal feature of data" subsection and each daily window is represented by the mean of the measurements of clinical variables on that day. For instance, if there are 10 body temperature reading carried out in 1 day, all these temperature readings are averaged and used as the temperature reading data for that particular day. This process is known as daily aggregation and helps to balance out different variables that are measured at different frequencies.

However, some input features might still have a high level of missingness or entry errors. Hence, appropriate data imputation will be needed. Data imputation refers to the process of replacing missing or incorrect values in a dataset with plausible estimates in a way that does not significantly affect the overall distribution of the data. One useful machine learning method for data imputation is Last-Observation-Carried-Forward (LOCF). This method is widely used in clinically experimental studies. After the daily aggregation process is carried out, the mean value for each input feature is forward-filled (carried-forward) to adjacent time intervals as shown in figure 3.6. This is done to improve the quality and completeness of the data for analysis and modeling purposes. That being said, missing measurement could be a very distinctive feature. Taking an example, continuous measurement of Richmond-RAS Scale might suggest that patient is suffering from delirium due to infection or sepsis. Therefore, if Richmond-RAS Scale is not measured after continuous measurements on multiple days during the antibiotic treatment, this likely indicates the patient has recovered from this condition through antibiotic treatment. Note that this just is a theoretical inference about how missing measurements could be beneficial for analysis. Table 3.5 summarises the missingness level of input features after data imputation.

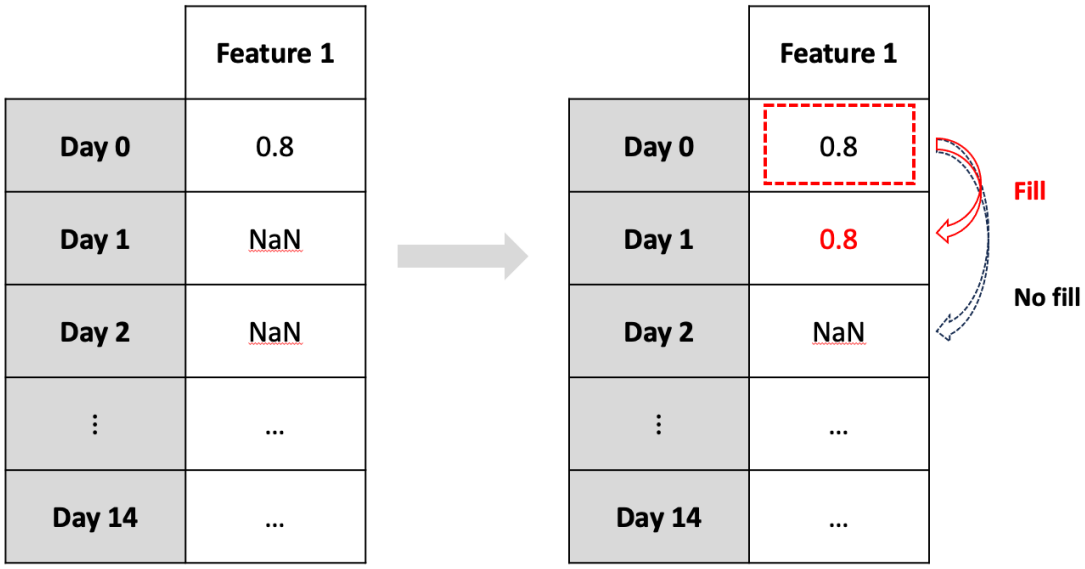

Figure 3.6: Forward fill method (LOCF)

3.1.7 Padding for Extended Rows and Missing Entries

As previously introduced, antibiotic treatment length is constrained to more than or equal to 5 days but less than or equal to 14 days. This means that for patients with antibiotic treatment of N days, there are N+1 days of data which includes the starting day and the subsequent N days of the treatment period. To explain further, if a patient started an antibiotic treatment on the 13rd and ended on 20th, this patient has received a total of 7 days of treatment (20-13), the extracted time series data for this patient will start from 13rd to 20th, hence there are 8 days of data (13rd,14th,15th,16th,17th,18th,19th,20th) including the starting day (13rd). For ease of processing, patients' data are converted into tabular format where row refers to day and column refers to feature. In this data, the maximum treatment length is 14 days. Hence, the maximum rows needed are 15. In order to create regular sequences for all ICU stays, each data sample must in the shape of 15 rows x 39 columns (15 days x 39 features). For patient data with less than 15 days, data is extended to a fixed size of 15 rows. For instance, patient with 7 days of treatment has 8 rows as starting day needs to be included, in this case, extra 7 empty rows with NaN values are added to extend it to 15 rows. In my data, I chose to set certain columns (features), namely gender, age, ethnicity, antibiotic treatment length and total duration to be constant in the extended rows to maintain consistency of these attributes and reduce missingness as these features are constant throughout the entire ICU stay. If different values are used for

these columns (features) in the extended rows, it may introduce distortions or mislead the model into thinking that these features might not be constant. Since other columns in the extended rows are treated as missing entries, the model should be able to capture this and understand that these are the extended rows for patients with shorter treatment duration. Figure 3.7 shows the data structure for a patient with 7 days of antibiotic treatment in ICU (8 rows of data) and 7 extended rows (before filling with '-1' value and min max normalisation).

After that, certain columns of training set and test set are normalised using min-max scaler (more details in the feature scaling section), all the extended rows and entries with NaN (missing entries) are then filled with '-1' values. '-1' is chosen as the padded value because it allows deep learning model to distinctly differentiate the extended rows and missing entries from valid entries as most of the features have been normalised to the range of [0,1]. The remaining features that are not normalised only consist of positive values, therefore models will treat them as valid entries. Most importantly, NaN entries are not easily processable by neural networks or standard architecture in Keras (Tensorflow) as they can lead to "dead neurons" during backpropagation. Therefore, '-1' is a good choice for all these reasons.

	age	ethnicity	gender	ALT	AST	Alkaline Phosphate	Anion gap	Arterial Base Excess	Arterial CO2 Pressure	Arterial O2 pressure	...	Prothrombin time	Respiratory Rate
32670	77	7	0	24.0	24.0	54.0	19.500000	-6.666667	26.333333	125.666667	...	18.4	21.458333
32671	77	7	0	32.0	52.0	57.0	14.666667	-5.000000	22.500000	123.500000	...	22.7	21.875000
32672	77	7	0	39.0	62.0	60.0	12.500000	-4.000000	21.000000	131.000000	...	24.4	19.750000
32673	77	7	0	48.0	69.0	71.0	12.000000	-1.000000	28.000000	139.000000	...	20.4	14.360000
32674	77	7	0	72.0	73.0	90.0	11.000000	-1.000000	28.000000	139.000000	...	14.7	11.125000
32675	77	7	0	86.0	61.0	87.0	12.500000	NaN	NaN	NaN	...	14.2	11.583333
32676	77	7	0	212.0	246.0	104.0	11.000000	NaN	NaN	NaN	...	25.8	14.500000
32677	77	7	0	246.0	185.0	109.0	10.500000	NaN	NaN	NaN	...	32.7	14.259259
32678	77	7	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
32679	77	7	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
32680	77	7	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
32681	77	7	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
32682	77	7	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
32683	77	7	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
32684	77	7	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

15 rows × 39 columns

Figure 3.7: Data structure for a patient with 7 days of treatment in ICU (8 rows of data) and 7 extended rows starting from index 32678 (before filling with '-1' value and min max normalisation)

Input variables	Level of missingness (%)
Gender	0.00
Age	0.00
Ethnicity	0.00
Total Duration	0.00
Antibiotic Treatment Duration	0.00
Heart Rate	0.54
O2 saturation pulseoxymetry	0.57
Respiratory Rate	0.62
GCS - Eye Opening	0.62
GCS - Motor Response	0.67
GCS - Verbal Response	0.68
SpO2 Desat Limit	1.20
Chloride (serum)	1.31
Creatinine (serum)	1.33
Potassium (serum)	1.33
HCO3 (serum)	1.34
Sodium (serum)	1.36
Anion gap	1.38
Glucose (serum)	1.52
Hematocrit (serum)	1.62
Hemoglobin	1.68
WBC	1.71
Platelet Count	1.73
Magnesium	1.78
Phosphorous	2.16
Calcium non-ionized	2.32
Temperature Fahrenheit	4.11
Richmond-RAS Scale	15.68
Prothrombin time	21.80
PTT	22.37
PH (Arterial)	41.35
Arterial O2 pressure	41.69
Arterial CO2 Pressure	41.69
Arterial Base Excess	41.71
Lactic Acid	46.12
Total Bilirubin	54.51
ALT	54.65
AST	54.68
Alkaline Phosphate	55.20

Table 3.5: Summary of Missingness Level of Input Features After Data Imputation (Pre-Outliers Removal)

3.1.8 Time Series Data Representation

Figure 3.8 illustrates the final data structure which is used as input for deep learning models like LSTM and CNN. Both LSTM and CNN models are capable of analysing 2D tensor data and for baseline model like Multi-layer perceptron (MLP), the following data structure is flattened before feeding into the model.

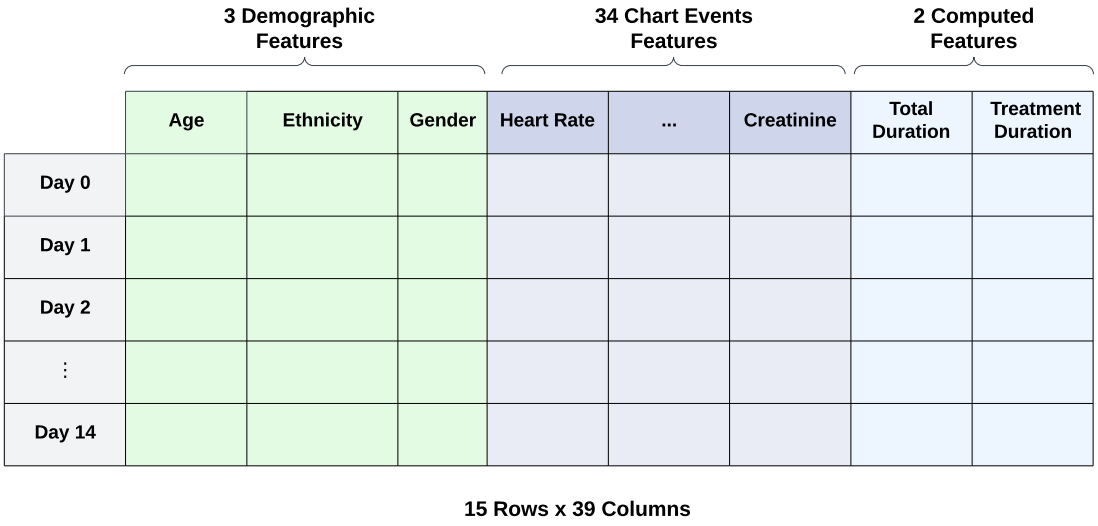

Figure 3.8: Input data structure per patient for deep learning models

3.2 Machine Learning Methods

This section explains machine learning techniques used in this project such as stratified sampling, 5-fold cross-validation, oversampling and feature scaling.

3.2.1 Stratified Sampling

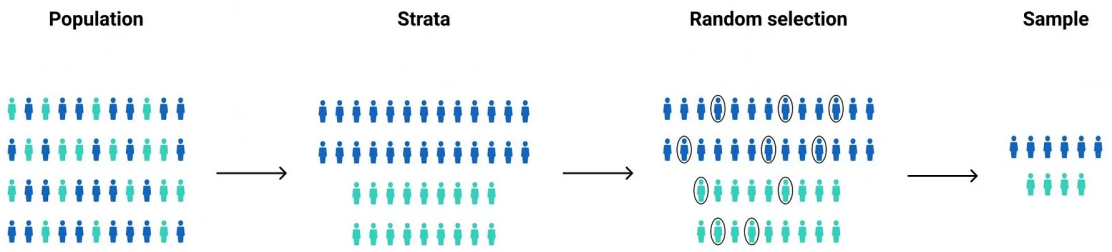

Figure 3.9: Stratified Sampling Visual Explanation (Taken from [8])

Stratified sampling is a machine learning technique that is widely used in dataset splitting to tackle the issues of minority class (imbalanced dataset). In general, this sampling method divides the population into subpopulations called strata based on pre-specified characteristic of the population like positive label, age or gender [8]. Next, random selection will be carried out on the strata to form a sample [8].

After the clean dataset is obtained, the binary labels and continuous labels are used as the stratified parameters in the train-test-split function from sklearn library to split the main dataset into training set (85%) and separate held out test set (15%) to ensure the distribution of positive cases is preserved in both sets. To recap, the processed dataset is imbalanced with only 17.5% of positive cases out of all the ICU stays. Therefore, it is crucial for the test set to have a positive case distribution of roughly 17.5% that mirrors the actual class distribution through stratified sampling so that the evaluation results through the test set correctly reflect the actual performance of the model. To summarise, this technique is preferable to normal random sampling as it helps maintain the positive case distribution in both training and test set. In table 4.1, we can observe that training set and test set have similar antibiotic readmission, age distribution and antibiotic treatment length statistics after stratified sampling. This method also helps reducing bias in the sampling process

and minimising differences between the training set and test set, ensuring the validity of this study [8].

3.2.2 Balancing Data Through Oversampling

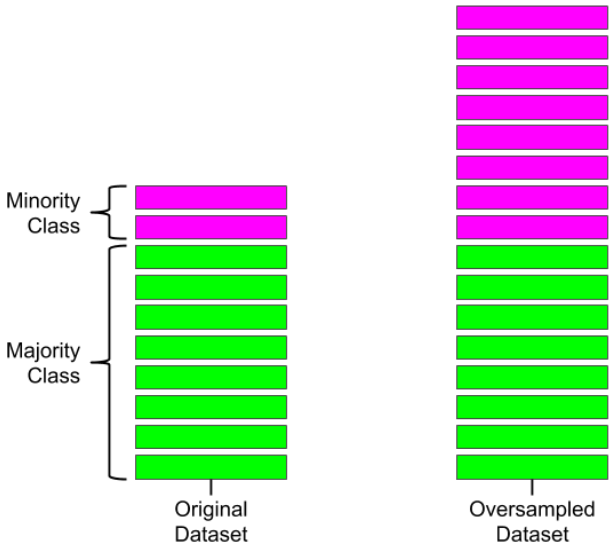

Figure 3.10: Oversampling (Taken from [9])

In clinical study, positive cases such as tumour detection, ICU readmission and antibiotic readmission tends to be a low probability event among ICU patients. This presents a challenge in training the classification and regression model as imbalanced data is generally known to cause poor prediction performance. The machine learning model has a bias to predict the majority class of the data when trained with imbalanced data. An example of this would be high accuracy in classification task but weak performance in other metrics like AUROC. This indicates that the model is unable to distinguish positive and negative cases correctly. Hence, it is very important to balance the data with an approximately equal number of positive cases and negative cases to train the model so that the model does not have a bias in its prediction.

For this reason, oversampling is used in this project to balance the training set. Note that the final models are trained and validated using cross-validation so **oversampling only happened within each training fold**. In the binary classification task, the positive cases are duplicated 3 times to achieve a more balanced antibiotic readmission rate of 45.9%. The prediction performance in the classification task was boosted significantly after oversampling. While in regression task and joint learning task, the positive cases are replicated

2 times to reach a rate of 38.8%. The reason why positive cases are replicated 3 times in classification task is that the classification model always attempts to predict the majority class of antibiotic readmission in the data to minimise overall classification loss instead of predicting based on the clinical variables of the patients. By pushing the percentage of positive cases in the data closer to 50% for classification task, the model is forced to learn the underlying patterns of positive cases, leading to more accurate prediction.

3.2.3 Stratified K-fold Cross Validation

Stratified k-fold cross-validation is a special type of cross-validation that can gauge the performance of the prediction model for an imbalanced dataset. This ML technique involved stratified sampling the data into k equal-sized folds while making sure that the percentage of positive cases is roughly the same for every fold. The deep learning model is first trained on k-1 folds and validated on the left-out fold. This process is repeated until the model has been trained on all data and validated on every fold as shown in figure 3.11 [10]. The evaluation metrics of the trained model are then averaged across every fold to assess the overall performance.

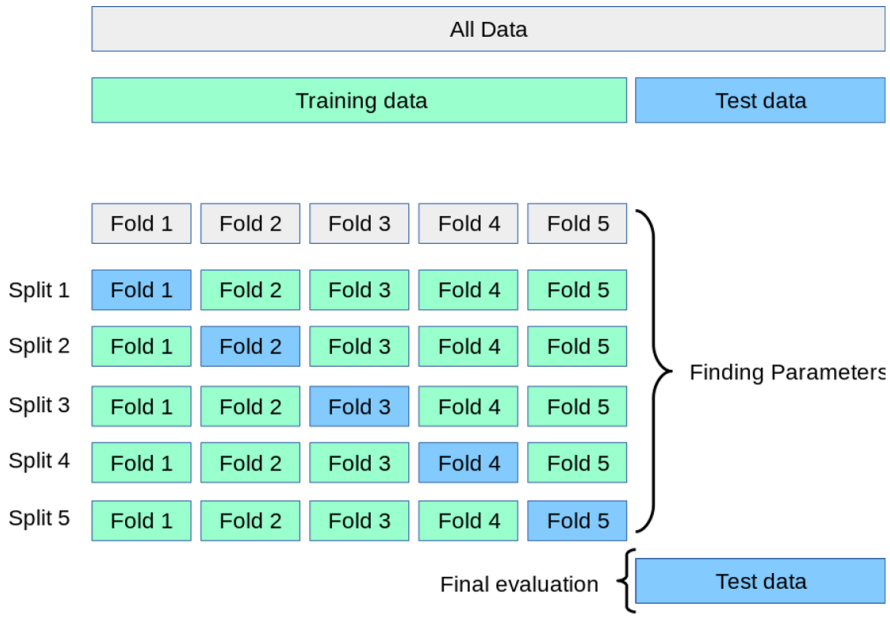

Figure 3.11: K-fold Cross Validation where k=5 (Adapted from [10])

Since the training set only consists of 1,860 ICU stays, to utilise every bit of the data, stratified 5-fold cross-validation is applied to validate the model performance more precisely. As the dataset is relatively small, 5-fold cross-validation allows the model to be

trained on 5 different subsets of data and validating the model on 5 different folds provides a great estimation of model performance. Each fold also represents roughly 17% of the overall dataset and this proportion is close to the size of the test set (15% of the overall dataset) which ensured validation to be conducted on a similar amount of data. Not only that, it helps assess the robustness of the model on different sets of data. From the training process, it is observed that the deep learning models are relatively sensitive to the variation of training data due to small dataset, models generally performed better in some folds and worse in some folds. If there is only a single split of training data into a validation set, it is hard to assess the model performance accurately since it would be significantly affected by the initial splitting of data. In cross-validation, this impact is reduced because the training data are splitted into multiple folds and performance is averaged on all the validation folds. Further details on cross-validation are given in the results section.

3.2.4 Features Scaling

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Min-max normalisation (feature scaling) is a linear transformation that is applied to the features of the data to scale them into the range of [0,1]. In order to analyse the clinical variables/features on the same scale, min-max normalisation is applied to the following list of features: Heart Rate, Respiratory Rate, O2 saturation pulseoxymetry, Temperature Fahrenheit, Arterial Base Excess, Lactic Acid, PH (Arterial), Arterial CO2 Pressure, Arterial O2 pressure, ALT, AST, Alkaline Phosphate, Anion gap, HCO3 (serum), Total Bilirubin, Calcium non-ionized, Chloride (serum), Creatinine (serum), Glucose (serum), Magnesium, Phosphorous, Potassium (serum), Sodium (serum), Hemoglobin, Platelet Count, Prothrombin time, PTT, WBC, Hematocrit, SpO2 Desat Limit and Richmond-RAS Scale.

Most clinical variables have very different scales, min-max normalisation ensures that the scaled features are comparable in terms of magnitude. From the model training perspective, normalisation improves the model convergence, thus shortening the training epochs. Interestingly, by applying min-max normalisation to a subset of features mentioned above instead of all the input features, the prediction performance is improved. The reason why age, treatment length, total duration and 3 GCS features are not scaled in the first place is to retain the original range of the features for the model to analyse in a direct way as inputs from clinical expert, Dr Mandy implies that these features can play an important

role in prediction. Label encoding is applied to gender and ethnicity so there is no need to normalise them. Since the min-max normalisation is only applied to most chart events features excluding 3 GCS features, demographic features and computed features, it is highly likely that the model is able to distinguish these 3 categories of features, leading to better overall performance.

3.2.5 Data Leakage Prevention

In order to prevent any potential data leakage, min-max normalisation is applied separately on the training set and test set after splitting. If min-max normalisation were to be applied to the overall dataset, test set distribution will be "leaked" to the training data, causing overly optimistic or pessimistic results. Therefore, to ensure the integrity of results, min-max scaler from the sklearn library is first used to fit against the training data to find out the minimum and maximum values of each feature. The training set and test set features are then scaled using the min and max obtained from the training set.

3.2.6 Summary of Methods

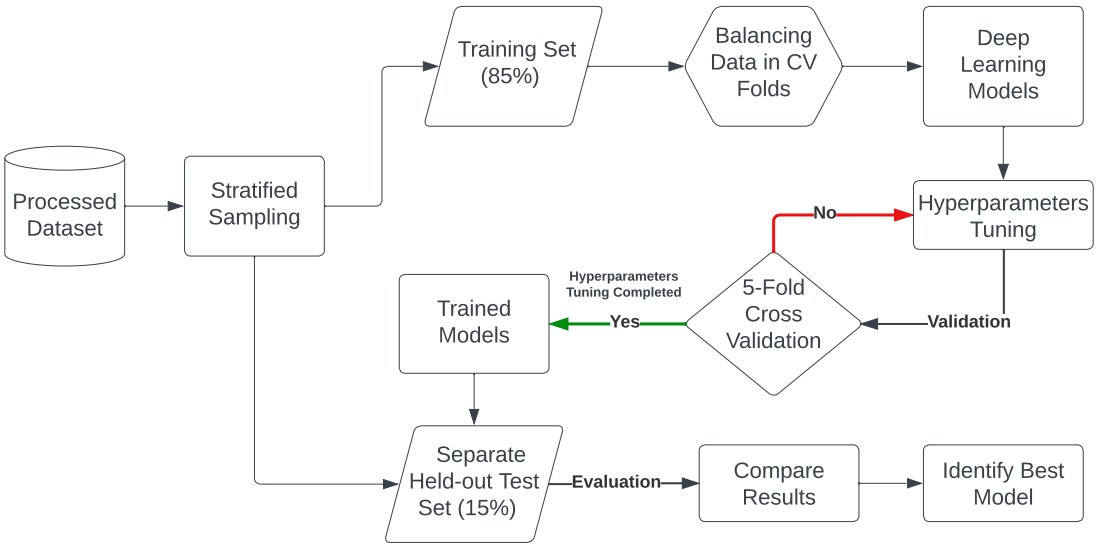

Figure 3.12: Machine Learning Framework (CV = Cross Validation)

The discussion in the previous section is illustrated in figure 3.12. To further elaborate on figure 3.12, a summary of the discussion is given below.

1. First, the processed dataset is stratified sampled into training set and test set (85% / 15% split)
2. Min max normalisation is applied to the training set and test set separately using normalisation parameters obtained from the training set
3. Training set is then used for 5-fold stratified cross-validation where the set is splitted into 5 different folds
4. Model is first trained on 4 folds and validated on the left out fold
5. In these 4 folds, the positive cases are oversampled to train the model
6. This process is repeated until the model has been trained on different combinations of folds and validated on every fold
7. The performance metrics on every fold are averaged to produce an estimation of validation results
8. Once the hyperparameters tuning is completed by maximising the validation results, the model is then evaluated on the separate held out test set. (Details about hyperparameters tuning can be found in Appendix A)

3.3 Machine Learning Model Structure

This section investigates the machine learning architectures used for the project. Illustrations of the architectures are provided to explain the rationales behind the design choices as well as how these models are trained with the time series data.

3.3.1 Prediction Models

As previously introduced, this project aims to explore different advanced deep learning models like LSTM and CNN architectures trained on time series data for antibiotic readmission prediction and to assess their performance as compared to baseline model like Multi-layer Perceptron. Accurate prediction of antibiotic readmission can provide guidance in determining optimal antibiotic treatment for the patient to tackle the issues of AMR and prevent treatment failure. This prediction task is addressed as both binary classification and regression tasks to assess the potential and limitations of prediction models under these 2 settings. A more challenging regression task could yield meaningful results for clinicians as there are more clinical utilities and information that clinicians can extract from it. In this study, the general architectures of the models in classification task are similar to the models in regression task.

This project explores 6 different models for antibiotic readmission prediction. Table 3.6 summarises the models that are trained and tested.

Architectures	**Classification**	**Regression**	**Joint Learning**
Masking + BiLSTM + CNN + FCNN (Proposed Model)	✓	✓	✓
Masking + BiLSTM + FCNN (1 BiLSTM Layer)	✓	X	X
Masking + BiLSTM + FCNN (2 BiLSTM Layers)	X	✓	X
CNN + BiLSTM + FCNN	✓	✓	X
CNN + FCNN	✓	✓	X
Multi-layer Perceptron (MLP)	✓	✓	X

Table 3.6: Summary of Architectures

General process for designing and training the models is shown below:

- Research of baseline model and advanced deep learning architectures that can process multivariate time series data in the areas of healthcare prediction
- Implementation of selected deep learning architectures
- Identifies key hyperparameters that affect the prediction results significantly and performs hyperparameters tuning manually

- Evaluation of model performance through 5-fold stratified cross validation

The binary labels and continuous labels introduced in section 3.1.4 are used to train the deep learning architectures for classification and regression tasks respectively. The training data are shuffled before splitting into folds for cross validation. The data is ordered based on every patient's earliest antibiotic administrated date, shuffling randomly changes the ordering of data to eliminate any potential bias between the ordering of data and the target variable (antibiotic readmission).

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1-y_i) \cdot log(1-p(y_i))$$

Figure 3.13: Binary Cross Entropy Loss Function (Taken from [11])

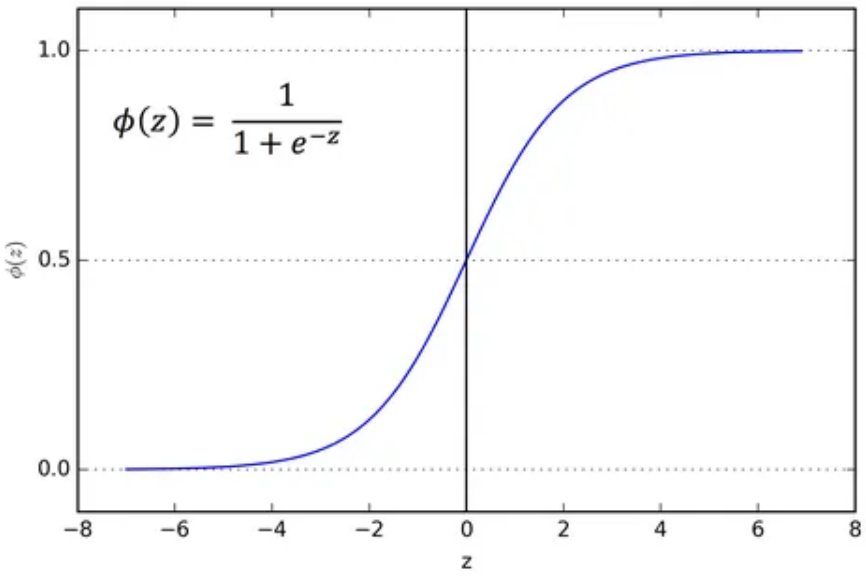

Figure 3.14: Sigmoid Activation Function (Taken from [12])

From the machine learning perspective, the main differences between classification task and regression task lie in the loss function and the final activation function. For classification task, the models are trained with binary cross-entropy loss function (see figure 3.13) and the prediction output is obtained through the sigmoid activation function (see figure 3.14 in the final dense layer. Binary cross-entropy is extremely useful for imbalanced dataset as it penalises the model heavily when a prediction made with very high certainty turns out to be wrong [11]. Sigmoid activation is used because the output of this activation function

lies between the range of (0,1), allowing intuitive analysis of the prediction output as probability [12]. In this task, a prediction threshold of 0.5 is applied where an output larger than 0.5 means that the patient is likely to be retreated with antibiotic and vice versa for output smaller than 0.5.

For regression task, the models are trained with Mean Squared Error (MSE) loss function and regression output is obtained directly from the final dense layer. Different loss functions such as MAE and Huber Loss are considered for regression task but MSE is chosen to penalise any large errors in the prediction. It is also found that training with MSE loss functions allows faster model convergence.

There are also some differences in the hyperparameters of the models for both tasks. All the models are trained using Adam optimizer with a default learning rate of 0.001. Google Colab Pro+, a cloud platform is utilised to train the deep learning models using T4 or A100 GPU with RAM up to 128GB to accelerate the training process. All deep learning models are implemented using Keras library with Tensorflow backend.

3.3.2 Baseline Model

Multi-layer perceptron (MLP) is chosen to serve as the baseline model for antibiotic readmission classification and regression tasks. The idea behind choosing MLP instead of commonly used logistics regression model is because logistics regression model is a simple model whereas MLP is a more complex model that consists of multiple layers of neural network with non-linear activation functions such as ReLu or LeakyReLu. It is capable of being applied in classification and regression tasks which makes it very suitable to be the baseline model. MLP also provides a much superior benchmark/baseline antibiotic readmission prediction for comparison with other advanced deep learning architectures. Other classical machine learning models such as support vector machine and random forest are not considered because it is very challenging to transform the data structure described in section 3.1.8 into interpretable structure as input for these classical models.

The schematic of the architecture of MLP is shown in figure 3.15. First, the input data is flattened into 1-dimensional tensor as MLP is not capable of processing 2D data. Flattened data is then fed into 4 layers of neural networks consisting 32, 8, 8 and 1 number of neurons in order. Between every layer of the dense network, a dropout layer is added

to prevent overfitting through regularisation. The dataset contains padded values of '-1', ReLu activation is not suitable to process negative values as it can cause issues during backpropagation, leading to "dead neurons". To mitigate the issues of "dead neurons", LeakyReLu activation function is used in all the dense layers as it is able to process the negative padded values. Optimised parameters for dropout layers differ for classification and regression tasks.

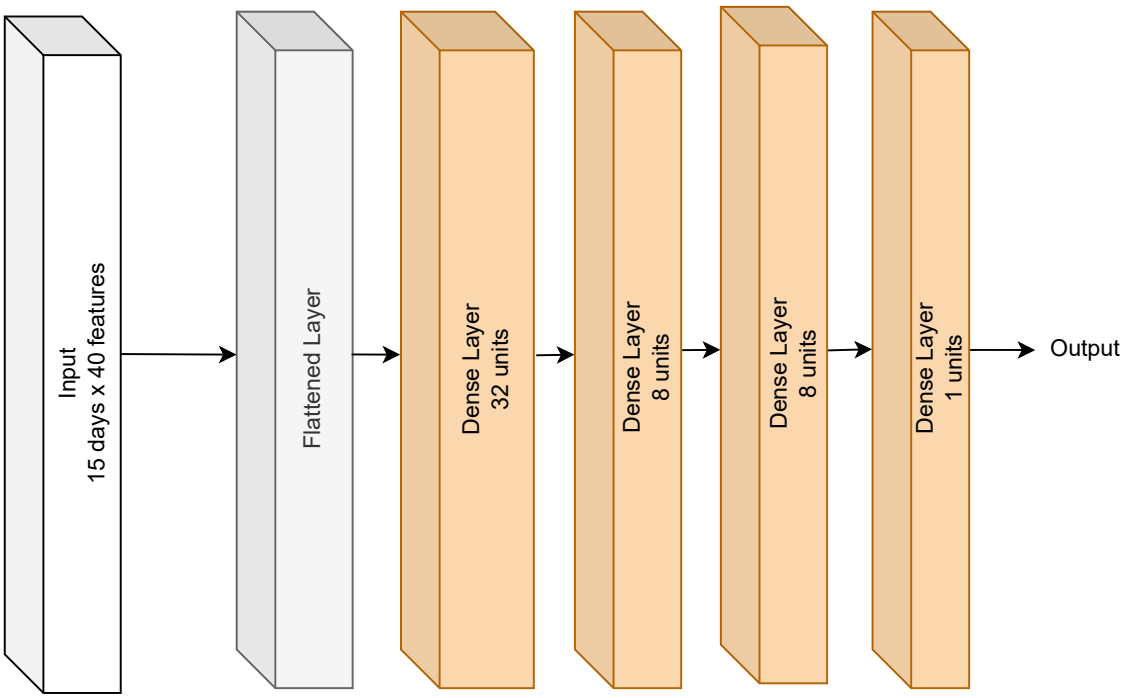

Figure 3.15: MLP - Baseline Model

3.3.3 Advanced Deep Learning Models

This subsection explains the advanced deep learning models explored in this project. For full disclosure, the architecture of combining LSTM with CNN is inspired by the paper [48] for ICU readmission prediction, there are some slight differences in the final architectures. The presentation of the architecture is inspired by [66].

CNN + FCNN

The combination of convolutional neural networks (CNN) and fully connected neural networks (FCNN) has always been a widely used deep learning architecture for computer vision and image classification tasks [67]. CNN has also been applied to multivariate time series as CNN's ability to learn inter-feature relationships is extremely powerful but it is

restricted in learning regional spatial features due to its convolutional nature [60].

Despite that, it is worth exploring how CNN can analyse the spatial relationship from the input features in the data for antibiotic readmission prediction. Since the time series representation is constructed as 2D data (tabular format) similar to image where row refers to day and column refers to feature, it will not present any challenge for CNN to process the data. Figure 3.16 illustrates the CNN + FCNN architecture implemented in this project.

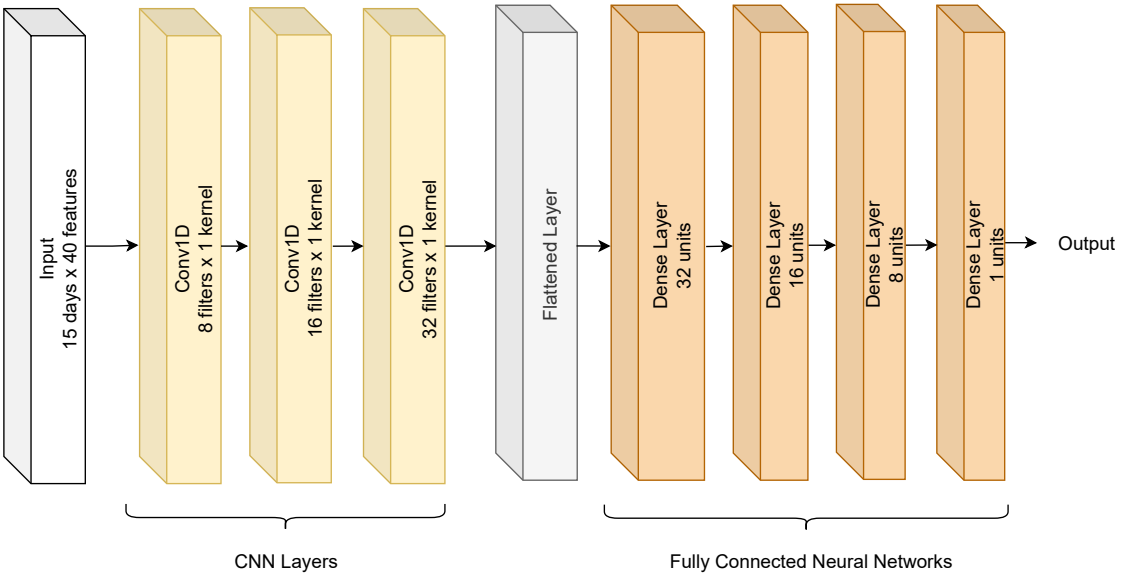

Figure 3.16: CNN + FCNN Architecture

In figure 3.16, the model is constructed with 3 Convolutional 1D layers (Conv1D) in a way with more filters per layer from 8, 16 to 32 filters with kernel size set as 1. Different from Conv2D layer, Conv1D layer only moves in 1 direction. In this task, the filter associated with Conv1D layers slides across the patient data day by day as shown in figure 3.17 and its output is used in the next layer.

After data has been processed by CNN layers, it should have explored the underlying patterns of the features and produced a meaningful features map / latent representation of the data. Then, this representation will be flattened and fed into 3 layers of fully connected neural networks with 32, 16 and 1 units of neurons accordingly for analysis to produce a final prediction. Dropout layers with varying rates are also added between every dense layer to mitigate issues of overfitting. In the early stage of validating the model, different combinations of filter size (64, 128 and 324 filters) were tested but the model quickly overfitted to the training data. As a result, the dense layers are restricted to the

combination of 32,16 and 1 units of neurons instead of the combination of 128, 64, 1 units. Similar to the baseline model, LeakyReLu activation function is used in all the dense layers to process negative padded values.

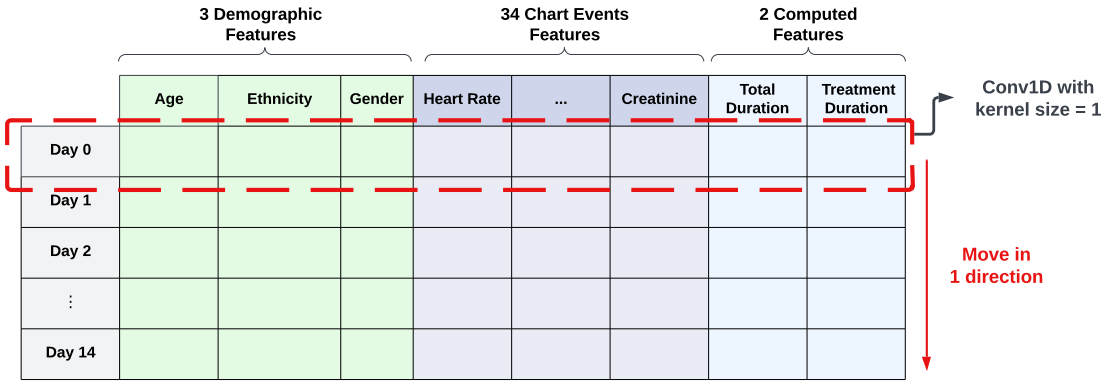

Figure 3.17: Conv1D Layer with Kernel = 1 (circled in red)

Masking + BiLSTM + FCNN

Although CNN has found success in solving different ML tasks, CNN can only exploit spatial relationships instead of temporal relationships. LSTM which is a specialise form of RNN is more commonly used to process sequential data such as multivariate time series. 2 variations of LSTM + FCNN architecture are designed for classification and regression tasks. Figure 3.18 shows the Masking + BiLSTM (1 layer) + FCNN architecture used in classification task while figure 3.19 shows the Masking + BiLSTM (2 layers) + FCNN architecture used in regression task.

The reason why 2 different models are designed is because regression task is much more challenging and it requires a deeper architecture to capture complex relationships in predicting antibiotic readmission. Hence, 2 layers of BiLSTM are used for regression task. Meanwhile, in classification, single layer of BiLSTM is very sufficient in processing the data, simpler model can also avoid overfitting to training data. Apart from that, hyperparameters tuning process can be shortened as there are fewer hyperparameters to be tuned.

In general, this architecture is made up of 3 different structures. First, the masking layer is used to mask the padded values of '-1'. Next, BiLSTM layer is used where each layer consists of (num_units=20) 40 hidden dimensions and the final part of the model has

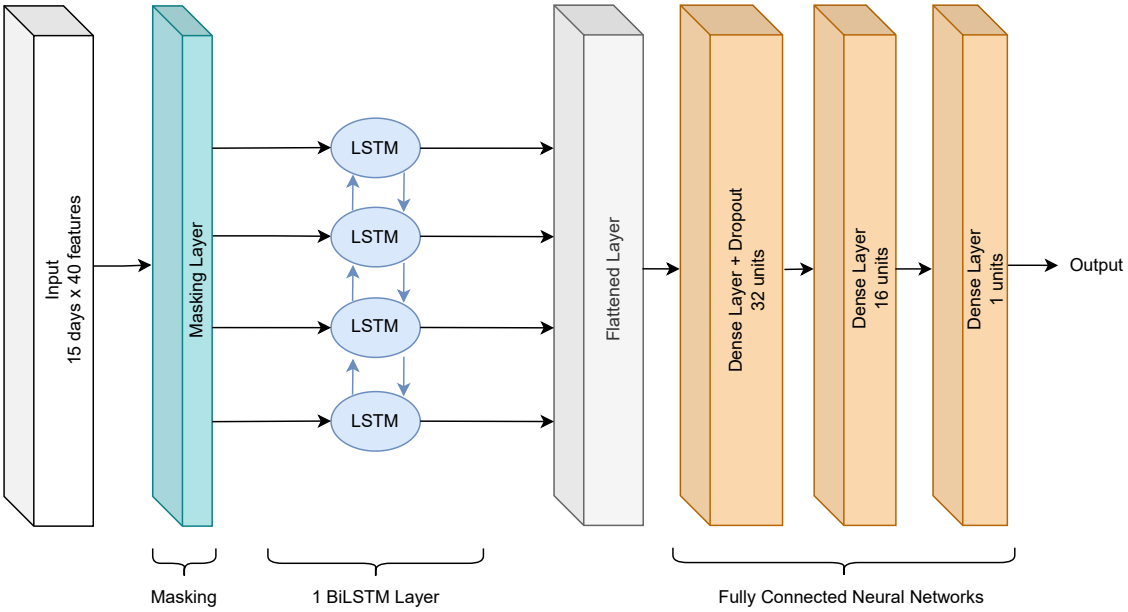

Figure 3.18: Masking + 1 Layer of BiLSTM + FCNN Model (Classification)

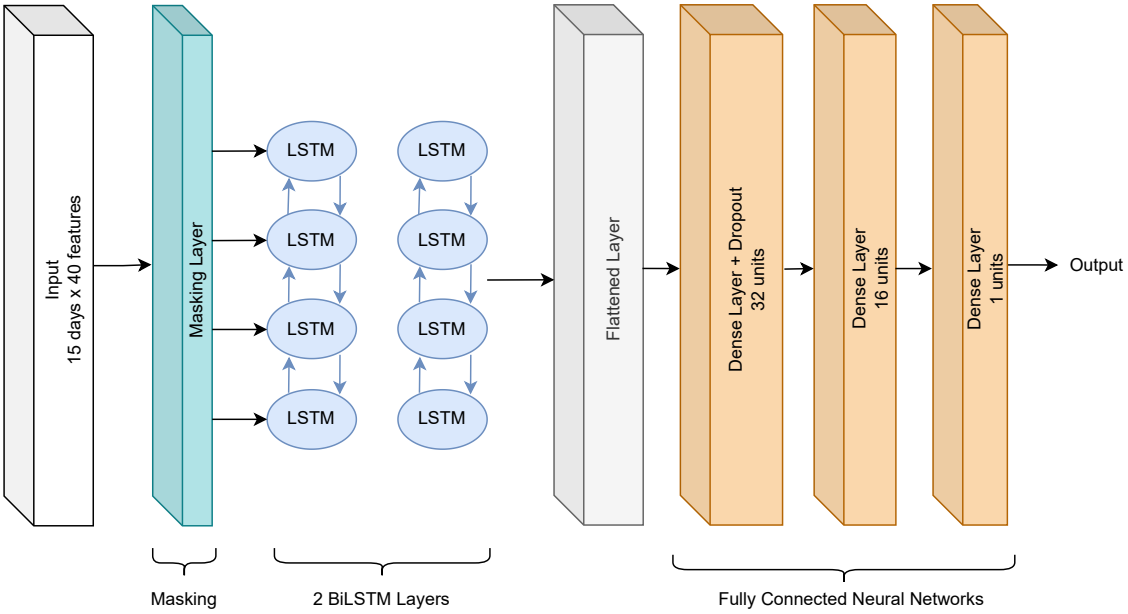

Figure 3.19: Masking + 2 Layers of BiLSTM + FCNN Model (Regression)

3 dense layers similar to those in CNN + FCNN model except that ReLu activation is used instead of LeakyReLu. As described in the data preprocessing section, the missing entries and extended rows in the data are padded with '-1'. By masking '-1' values in the inputs, padded values will be skipped, allowing BiLSTM layers to focus on valid entries of the data. The intuition behind selecting BiLSTM instead of the normal LSTM layer is that BiLSTM is more efficient in learning the temporal dependencies between patient's condition at the start of antibiotic treatment and at the end of the treatment. BiLSTM is extremely powerful in processing this kind of information by allowing data to be passed forward and backward. When the time series data is passed to BiLSTM layers, BiLSTM automatically processes the data day by day (row by row) and learns the temporal patterns across the timesteps internally. In the code, the first layer of BiLSTM is implemented in a way to pass all the hidden states of the LSTM to the next layer to capture sequential information. While for regression task, in the second layer, only the last hidden state will be sent to the subsequent layer. After that, it will be flattened for dense layers to process. Only 1 dropout layer is added after the dense layer with 32 units of neurons as the number of parameters in these 2 architectures is relatively lower than in the other models.

Combination of CNN and BiLSTM

The combination of CNN and BiLSTM is explored as it found success in ICU readmission prediction [48]. The idea behind this combination is to leverage CNN's capability of learning inter-features relationships and LSTM's ability to exploit temporal dependencies. 2 variations of this architecture are attempted, CNN + BiLSTM + FCNN model and Masking + BiLSTM + CNN + FCNN model. Figure 3.20 shows the architecture for CNN + BiLSTM + FCNN model, while figure 3.21 shows the architecture for Masking + BiLSTM + CNN + FCNN model.

CNN + BiLSTM + FCNN model

CNN + BiLSTM + FCNN model is constructed with CNN layers as the primary layers followed by BiLSTM and dense layers. The CNN layers are made up of Conv1D layers similar to the CNN+FCNN model to process the patient data day by day. CNN is placed before BiLSTM so that spatial relationships between features can be extracted to produce features map/latent representation of the data. BiLSTM is then used to process the features map sequentially for temporal pattern extraction. After that, the output of BiLSTM

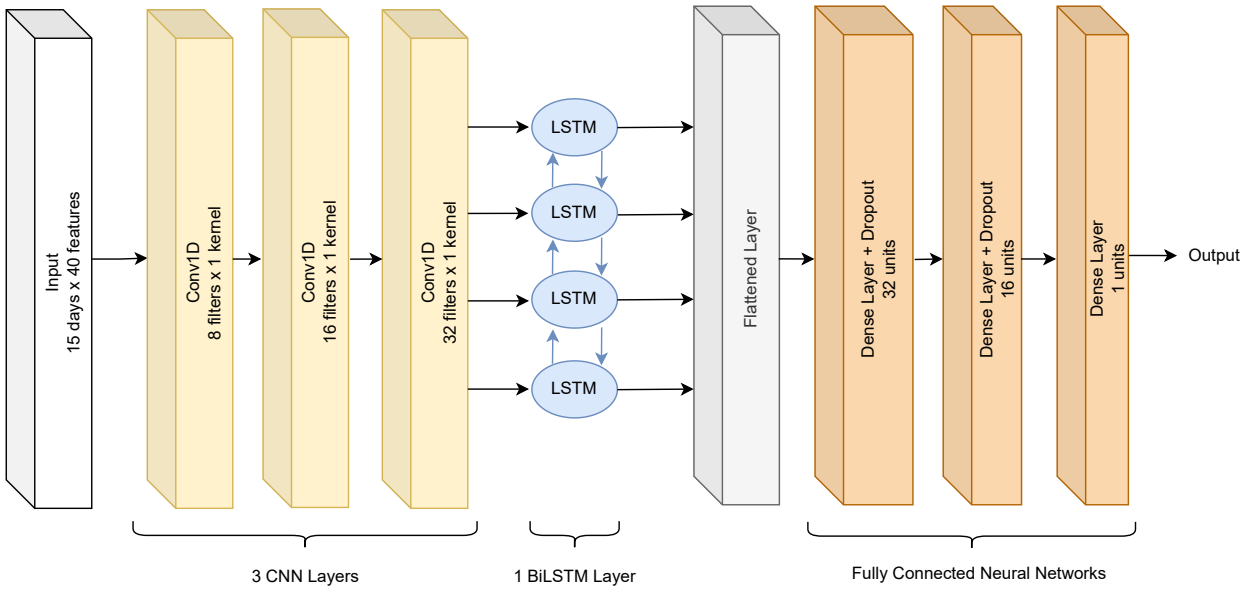

Figure 3.20: CNN + BiLSTM + FCNN

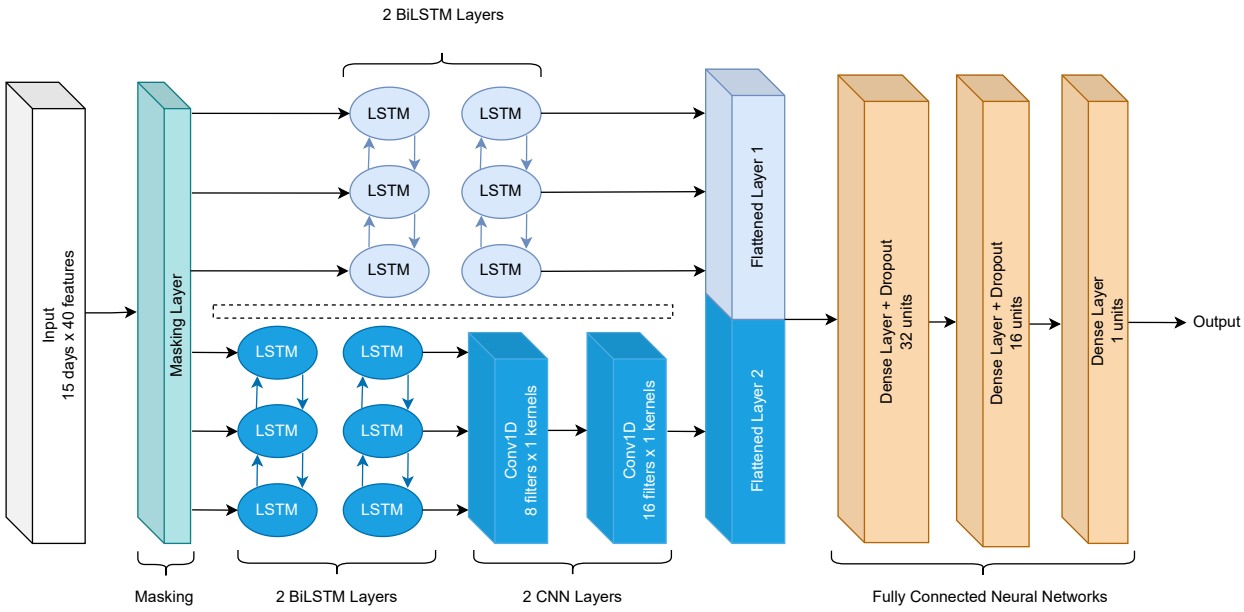

Figure 3.21: Masking + BiLSTM + CNN + FCNN

will be processed by dense layers for final prediction.

Masking + BiLSTM + CNN + FCNN (Proposed Model)

Different from CNN + BiLSTM + FCNN model, in the proposed model, masking layer is leveraged to disregard the padded values for efficient computation and to improve model performance as it helps the model to focus on meaningful data. The masked input is then sent to 2 different branches consisting of BiLSTM and BiLSTM+CNN layers for feature maps computation. The main difference between 2 different branches lies in the CNN layers. In the lower branch, kernel size is set to 1 for CNN. Whereas in the upper branch, only BiLSTM layers are used. This is of vital importance as CNN layers can produce different but meaningful features map based on inter-features relationships extracted by the kernels. The upper branch is mainly used to capture all temporal relationships of the patient's condition. The outputs from the 2 branches are flattened and concatenated to take advantage of different features map produced by each branch which could improve feature reuse and representation capacity. Another distinct difference between this model and CNN + BiLSTM + FCNN model is that BiLSTM is used as the primary layer for initial processing. In the BiLSTM layers, all hidden states of the LSTM are passed to the next layer to retain sequential information. This means that the features map produced by BiLSTM is in 2D form (timesteps x hidden dimensions) which is similar to the input data structure (15 rows x 39 features) and this allows CNN layers in the lower branch to be able to extract the spatial patterns within the learned temporal representation.

The hyperparameters for all models and the code are included in Appendix A

3.4 Model Evaluation

3.4.1 Performance Evaluation Through Cross Validation

As there are 2 different tasks to tackle, different sets of performance metrics are used to evaluate the models for binary classification and regression tasks. All deep learning models are trained and validated using 5-fold stratified cross validation. In binary classification task, AUROC metric is selected as the main metric to be maximised, accuracy and AUPRC metrics are also taken into account to balance out the tradeoff between all the metrics. Whereas for regression task, MAE and MSE are selected as the main metrics to be minimised. The idea is here to tune the hyperparameters of the model until the best average validation results across the five folds is achieved to prevent overfitting. At this point, the model has been trained with 5 different combinations of folds of the training data. Model from each fold is then evaluated on separate held-out test set. The test results are averaged to produce a reliable estimation of the model performance on unseen data. Standard deviation of the test results is also computed to calculate the prediction range as it is a strong indicator of the stability of the model.

3.4.2 Evaluation for Binary Classification Models

This subsection discusses about the performance metrics selected to evaluate the prediction performance of binary classifiers.

AUROC (Area Under the Receiver Operating Characteristic Curve)

The first metric used in performance evaluation is AUROC. This metric measures the model's ability to differentiate between positive cases and negative cases by measuring the area under the receiver operating characteristics curve (ROC) shown in figure 3.22 [13]. The ROC curve is constructed by plotting the true positive rate as y-axis and the false positive rate as x-axis at different decision thresholds [13]. This metric is more useful in analysing the model performance than accuracy especially for imbalanced dataset [13]. For example, if a dataset has 10 positive cases and 90 negative cases, the model can predict all cases as negative and still get 90% accuracy but it will perform badly in terms of AUROC.

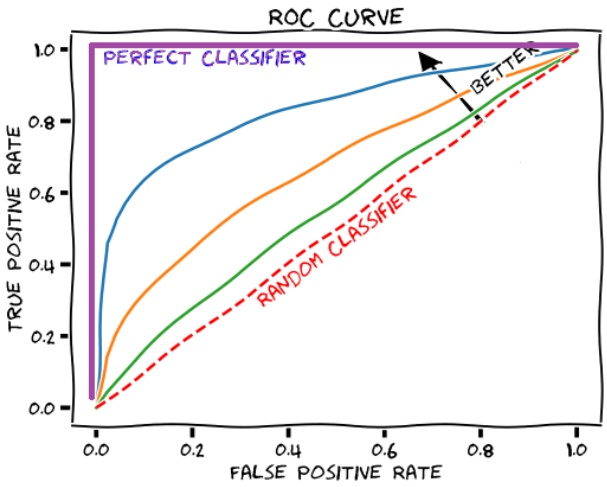

Figure 3.22: ROC Curve (Taken from [13])

As a rule of thumb [13],

- AUROC of 0.5 indicates that the model is not learning and it is a random classifier
- AUROC of 0.5-0.7 indicates that the model distinguishes positive cases and negative cases poorly
- AUROC of 0.7-0.8 indicates the model is learning and has at least 70-80% chances of correctly classifying positive cases and negative cases
- AUROC of 0.8-1.0 indicates the model can make an accurate prediction with high confidence

Accuracy

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total}}$$

Accuracy is also used as one of the evaluation metrics for binary classification task which is defined mathematically as above.

Precision

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Precision is a metric to assess the quality of positive case prediction which shows the ratio of correctly predicted positive cases out of all predicted positive cases.

Recall

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Recall is used to assess how well is the model's prediction in terms of marking all positive cases correctly which is measured using the ratio of correctly predicted positive cases out of all the actual positive cases.

AUPRC (Area under Precision-Recall Curve)

There is always a tradeoff between precision and recall where improving precision can impact recall or vice versa. Therefore, AUPRC is used which is another metric to evaluate the model's ability to detect and handle positive cases correctly in imbalanced data [68]. "For instance, if a model achieves perfect AUPRC, it means the model correctly predicted all positive cases (perfect recall) without marking any negative cases as positives (perfect precision)" [68]. The PR curve is constructed by plotting the precision rate as y-axis and recall rate as x-axis at different decision thresholds [68]. The area under the PR curve is known as AUPRC. Different from AUROC with a baseline of 0.5, the baseline of AUPRC is dependent on the percentage of positive cases in the dataset [68]. In this task, the baseline for AUPRC is 0.176 as there are 17.6% of positive cases in the test set.

F1 Score

$$\text{F1 Score} = \frac{\text{True Positives}}{\text{True Positives} + \frac{1}{2}(\text{False Positives} + \text{False Negatives})}$$

The F1-score is calculated as the "mean" of precision and recall values [69]. It is particularly useful for imbalanced dataset as this metric provides a balanced measure for both precision and recall [69]. F1 is defined mathematically as shown above [69].

3.4.3 Evaluation for Regression Models

This subsection discusses about the performance metrics selected to evaluate the prediction performance of regression models, namely MAE, MSE and RMSE.

MAE

Statistically, mean absolute error (MAE) is a measure of errors between two continuous variables. In machine learning field, MAE is an evaluation metric and a fundamental tool

that is frequently used for regression models. MAE formula is shown below:-

$$MAE = \frac{\sum |y_i - x_i|}{N}$$

y_i refers to prediction by model, x_i refers to actual result and N is the total number of training sample. A model with great prediction performance will have a low MAE. Since our prediction for antibiotic readmission is continuous values, this metric is extremely important in understanding how far the prediction deviates from ground truth in L1 sense.

MSE

In statistics, mean squared error (MSE) measures the average squared difference between the predicted value and ground truth. MSE is the most common loss function for training regression models. MSE formula is shown below:-

$$MSE = \frac{\sum (y_i - x_i)^2}{N}$$

Different from MAE, MSE is used as both training loss function and evaluation metric. In general, the lower the MSE, the better the performance of the model. This metric measures the difference between predicted results and actual results in L2 sense, allowing us to assess the model from a different viewpoint.

RMSE

RMSE is a good indicator to estimate the standard deviation of a typically observed output from model prediction. Formally, RMSE is defined as below:-

$$RMSE = \sqrt{\frac{\sum (y_i - x_i)^2}{N}}$$

As a rule of thumb, low RMSE is desirable for this model. Units of RMSE are the same as MAE (i.e. days in this case), hence, it is directly comparable with MAE. In fact, RMSE can be used together with MAE, the difference between the results for MAE and RMSE is positively correlated to the variance of errors in the prediction [70]. Note that in RMSE, errors are squared and squared rooted, therefore, large errors will push RMSE to a large value as well. This also implies that RMSE should always be larger than MAE [70].

Chapter 4

Results

This chapter provides an analysis of the processed dataset including the statistics and distributions of the study population. Evaluation results of different deep learning models are discussed in the chapter.

4.1 Data Analysis

The dataset is split into training set and test set, each set holds 85% and 15% of data respectively. Table 4.1 shows the statistics related to the data.

Statistics	**Overall**	**Train (85%)**	**Test (15%)**
Number of ICU Stays	2189	1860	329
Mean age	63.61	63.56	63.92
Mean of antibiotic treatment length	7.93 days	7.95 days	7.86 days
Mean of antibiotic readmission (continuous label)	0.71 days	0.71 days	0.72 days
Standard deviation of antibiotic readmission	1.73 days	1.73 days	1.76 days
Percentage of positive cases	17.50%	17.47%	17.63 %
Positive cases statistics			
Mean of antibiotic readmission	4.05 days	4.04 days	4.07 days
Standard deviation of antibiotic readmission	1.91 days	1.90 days	1.96 days

Table 4.1: Dataset Statistics

Out of the 2,189 ICU stays, it is found that most patients have a mean age of 63.61 with a mean antibiotic treatment duration of 7.93 days. All these ICU stays have a mean antibiotic retreatment of 0.71 days with a standard deviation of 1.73 days. As mentioned previously, 17.50% of the total cases or 383 cases received antibiotic retreatment with a mean of 4.05 days and a standard deviation of 1.91 days. As patients that died during the hospital stay have been removed, this indicates that 82.5% of patients have recovered and

were discharged from the hospital without any antibiotic retreatment.

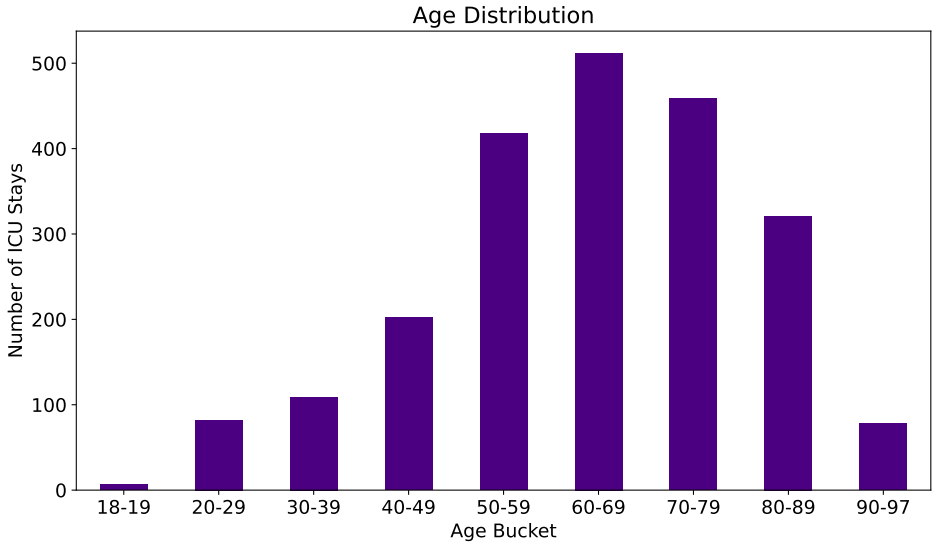

Figure 4.1: Age Distribution of Dataset

Figure 4.1 shows the age distribution of the processed data. It is observed that age group of 60-69 dominated the study population. The age distribution also exhibits normal distribution characteristics. Approximately 80% of the patients are in the age range of 50-97 as the older population tends to be more susceptible to infectious diseases due to weakened immune systems. Other chronic conditions such as cardiovascular diseases, diabetes and high blood pressure are also more prevalent in the older population which explains why age group of 50-97 made up most of the population in ICU.

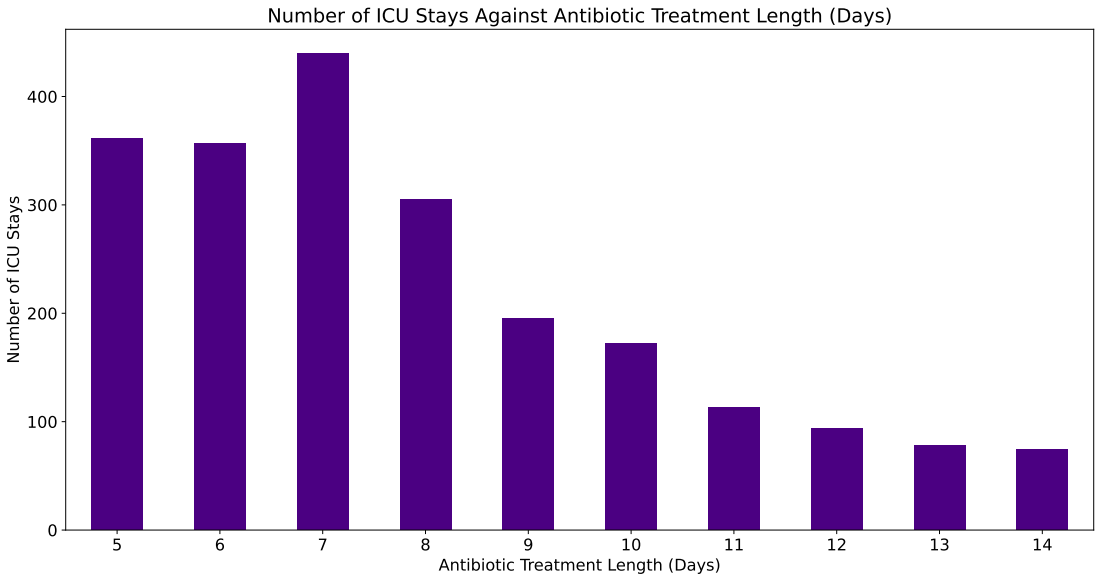

Figure 4.2: Antibiotic Treatment Length Distribution

Figure 4.2 shows the antibiotic treatment length distribution for the overall dataset. It is observed that most patients have a treatment length of 5-7 days. This observation is inline with the antibiotic treatment guideline Infectious Diseases Society of America (IDSA) that recommends a minimum of 5 to 7 days of antibiotics for adults with a re-evaluation at 3 to 5 days to ensure no relapse of infection [71]. Since these antibiotic treatment is carried out in ICU, it is likely that the study population tends to have a more serious infection. Hence, there is a substantial number of cases with antibiotic treatment length larger than 7 days as certain infections such as pneumonia and urinary tract infections can lead to severe infections like sepsis among ICU patients and such conditions require a longer antibiotic treatment.

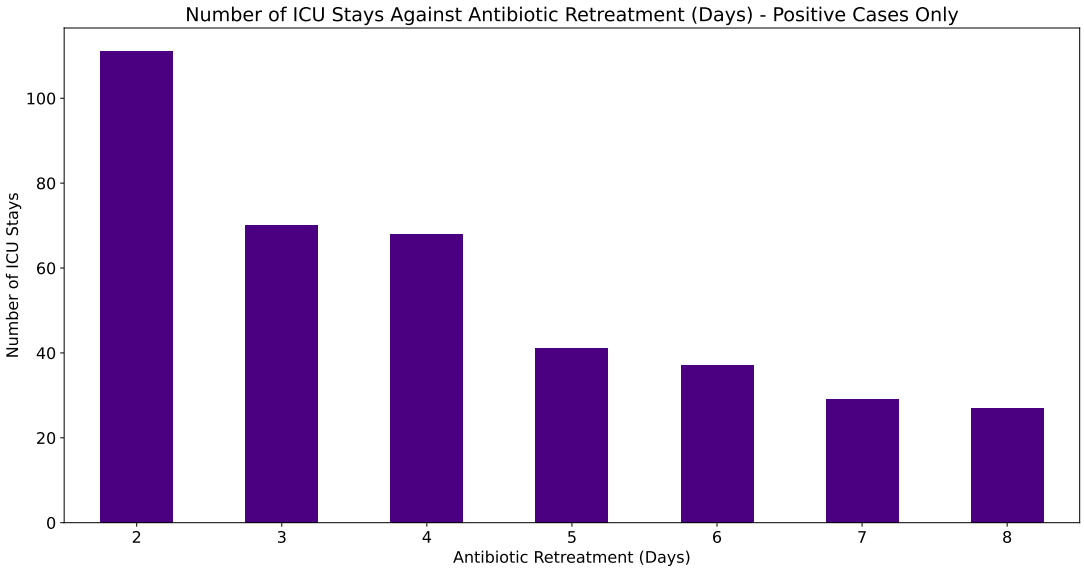

Figure 4.3: Antibiotic Readmission Distribution for Positive Cases

Figure 4.3 shows the antibiotic readmission distribution for positive cases. In these positive cases, it is found that most ICU stays in the dataset are retreated with antibiotics in 2 days followed by 3 and 4 days. The distribution shows an exponentially decreasing trend. Antibiotic readmission timing depends entirely on the patient's condition and past medical history. Although patients' condition is generally stable when treatment is stopped, there is a possibility of an ongoing infection as symptoms of infection may only temporarily subside but later reoccur. For some patients that have received constant antibiotic treatment in the past, they may have developed AMR towards multiple antibiotics and infection is likely to persist if they are treated with past administrated antibiotics. Therefore, antibiotic retreatment might be required for them. To conclude, in this study population, if antibiotic retreatment were to occur, it tends to happen early after antibiotic treatment completion.

4.1.1 Training Set and Test Set

To make sure that the training set and test set have similar distribution and characteristics, further data analysis is conducted. Throughout the project, multiple splitting of the dataset has been tested to ensure that the test set correctly reflects the distribution of the training set. In the initial phase of the study, large discrepancies between validation results and test results are observed. The main reason for this is that the overall dataset is relatively small and the data is imbalanced. Although stratified splitting is applied to ensure class distribution is preserved, it does not guarantee that the test set has similar characteristics to the training set. For instance, age distribution or ethnicity distribution of the test set can differ significantly from training set, leading to an incorrect evaluation of the model. Hence, splitting into training set and test set is carefully selected to avoid any dataset mismatch.

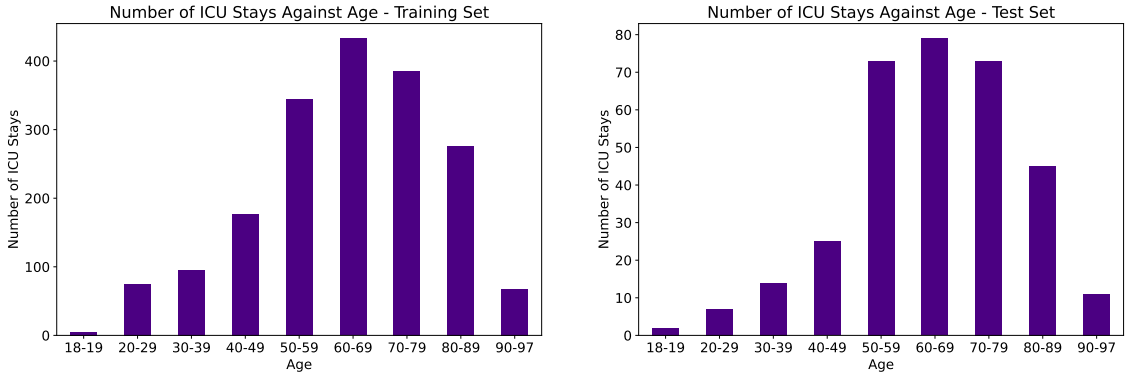

Figure 4.4: Age Distribution for Training and Test Set

Figure 4.4 shows the age distribution of training set and test set. From the graph, we can observe that both the training set and test set have very similar age distributions.

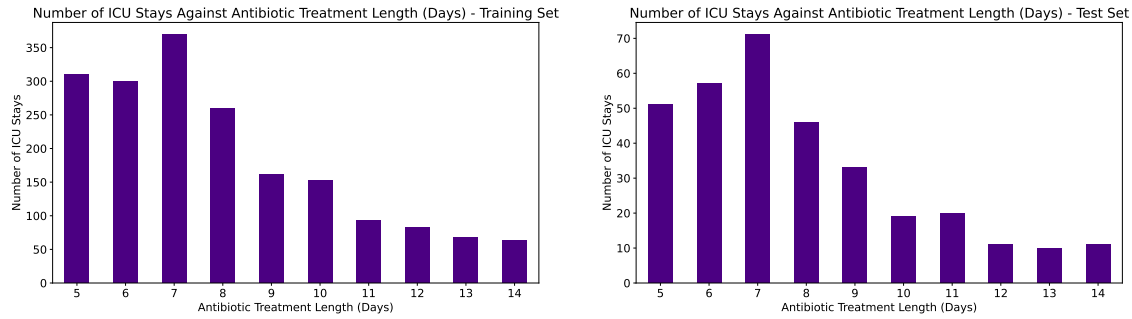

Figure 4.5: Antibiotic Treatment Length Distribution for Training and Test Set

Figure 4.4 shows the antibiotic treatment length distribution of training set and test set.

Apart from age distribution, another key distribution is the treatment length. In figure 4.4, it is clear that most ICU stays in both sets have 7 days of antibiotic treatment and the number of ICU stays decreases exponentially after 7 days. Therefore, we can conclude that the selected split results in a test set with similar characteristics to the training set.

4.2 Binary Classification Results

The following section discusses about binary classification results obtained from models described in the machine learning model structure section.

4.2.1 Performance of Different Deep Learning Models

To reiterate, the models are trained and validated using 5-fold cross validation. AUROC is selected as the primary metric to be maximised during hyperparameters tuning as this metric is a good indicator for the screening capability of the model in identifying patients that require antibiotic readmission. Whilst it is important to have high AUROC, hyperparameters are also selected based on the best overall validation performance. Hence, metrics like accuracy and AUPRC are also considered during validation process to strike a balance between AUROC, accuracy and AUPRC metrics. During cross validation, the model is trained with different combinations of folds and it is then evaluated on 5 different validation folds. These results are averaged to produce a single validation result. Extensive but not exhaustive hyperparameters search is conducted for every model to maximise this validation result. Once the validation results are maximised, the model in each fold is evaluated on the test set. The mean and standard deviation of test results are computed where the standard deviation is used to establish the "prediction range". Table 4.2 shows the validation AUROC and test AUROC of different models.

	Validation AUROC	**Test AUROC**
Proposed Model	**0.71**	**0.76**
Masking + BiLSTM + FCNN (1 layer of BiLSTM)	0.70	0.75
CNN + BiLSTM + FCNN	0.69	0.75
CNN + FCNN	0.69	0.73
Multi-layer Perceptron (MLP)	0.69	0.73

Table 4.2: Validation AUROC and Test AUROC of Different Models

First of all, we can observe that increase in validation AUROC leads to an increase in test AUROC, indicating that there is no significant mismatch between validation folds and test set. From the results, proposed model achieved the highest AUROC in both validation and test set. The discrepancy between validation and test results is within a reasonable range, signifying that the proposed model did not overfit to training data. This shows that the proposed model has a strong generalisability when dealing with unseen data. Similar observation is found in other models but with weaker performance in AUROC as compared to the proposed model.

	AUROC	AUPRC	Accuracy
Proposed Model	**0.76 (0.74 - 0.78)**	**0.54 (0.52 - 0.57)**	**0.74 (0.72 - 0.75)**
Masking + BiLSTM + FCNN (1 layer of BiLSTM)	0.75 (0.75 - 0.76)	0.54 (0.52 - 0.55)	0.73 (0.71 - 0.74)
CNN + BiLSTM + FCNN	0.75 (0.73 - 0.77)	0.53 (0.52 - 0.54)	0.73 (0.71 - 0.74)
CNN + FCNN	0.73 (0.72 - 0.75)	0.52 (0.50 - 0.54)	0.69 (0.64 - 0.75)
Multi-layer Perceptron (MLP)	0.73 (0.70 - 0.75)	0.49 (0.45 - 0.52)	0.70 (0.61 - 0.79)

	Recall	Precision	F1-score
Proposed Model	**0.71 (0.69 - 0.73)**	**0.64 (0.63 - 0.65)**	**0.65 (0.63 - 0.66)**
Masking + BiLSTM + FCNN (1 layer of BiLSTM)	0.70 (0.69 - 0.72)	0.63 (0.62 - 0.65)	0.64 (0.63 - 0.66)
CNN + BiLSTM + FCNN	0.70 (0.69 - 0.71)	0.63 (0.62 - 0.64)	0.64 (0.63 - 0.65)
CNN + FCNN	0.68 (0.66 - 0.69)	0.62 (0.60 - 0.63)	0.61 (0.58 - 0.64)
Multi-layer Perceptron (MLP)	0.65 (0.64 - 0.66)	0.61 (0.60 - 0.62)	0.60 (0.55 - 0.65)

Table 4.3: Performance of Different Models Evaluated on Test Set

The test results of different models including their standard deviation/prediction range are summarised in table 4.3. It is found that proposed model demonstrated the strongest performance across all evaluation metrics. It achieved an AUROC of 0.76 and AUPRC of 0.54 which outperformed the baseline model (MLP) by 3% and 5% respectively. AUROC of 0.76 suggests that the proposed model has a 76% chance of clearly differentiating between

positive cases and negative cases. On top of that, with an AUPRC of 0.54 that beats the baseline of 0.176 (as there are 17.6% positive cases in test set), the proposed model is very efficient and powerful in identifying positive cases correctly (while avoiding in marking negative cases as positives) than all other models. This is also evident in the recall metric with a score as high as 0.71. Apart from that, the proposed model also has an accuracy of 0.74, precision of 0.64 and F1 score of 0.65. This clearly shows that proposed model strikes a good balance between different metrics especially in AUPRC and accuracy. For all models, in this imbalanced dataset, at a certain point, increasing accuracy would result in drop in AUPRC because the model tends to predict more cases as negatives to boost the accuracy since majority cases are negatives. At the same time, false negatives would also go up, causing drop in recall and affecting AUPRC score consequently.

Masking + BiLSTM + FCNN model underperformed the proposed model in all metrics by 1% except for AUPRC but its performance is on par with CNN + BiLSTM + FCNN model which is a more complex model. The superior performance of both the proposed model and the Masking + BiLSTM + FCNN model is likely to be driven by the masking layer as the masking layer is the key differentiation of these 2 models from the others. The masking layer reduces noises in the data and helps the model to focus on valid entries instead of padded values in data. Another observation is that models with BiLSTM tend to perform better, showing that BiLSTM is capable of relating the sequential relationship between the patient condition at the start and at the end of antibiotic treatment. It is also worth mentioning that there is a substantial difference between MLP and other more advanced deep learning architecture in terms of AUPRC and recall metrics. From the results, advanced models are generally more competent in identifying positive cases.

Model Stability Analysis

The prediction range of each model provides valuable information about the stability of the model. The proposed model shows relatively stable performance across the 5 folds. In comparison, Masking + BiLSTM + FCNN model has slightly better stability in terms of AUROC, AUPRC and Recall metrics. On the other hand, MLP is the least stable across all metrics which is within expectation.

Findings from Hyperparameters Tuning Process

Based on the hyperparameters tuning process and observations on the validation results, it is found that different models have different characteristics. Note that hyperparameters tuning is carried out entirely based on the validation results only. The first finding is that the proposed model requires fine-tuning in its dropout layers to prevent overfitting on the training data. It also exhibits a strong ability in detecting positive cases by having a high AUPRC score but tradeoff between AUPRC and accuracy is observed all the time. For example, during the tuning process, accuracy, precision and F1-score of the proposed model were observed to be relatively low most of the time. As a tradeoff, it has high AUROC and AURPC scores. This is because the proposed model places a lot of weight on finding positive cases which is a desirable trait as the cost of false positive and false negative are very different and this is explained in the next part. Overall, careful fine tuning on the hyperparameters of proposed model is required to reach a good balance between different metrics as shown in the reported results.

The performance for Masking + BiLSTM + FCNN is relatively stable but its top-line performance could be limited as the model is comparatively simpler so it is not as sensitive to hyperparameters tuning. Another observation is that CNN + BiLSTM + FCNN model struggled in achieving high validation AUROC (>0.68-0.69). It is hard to identify which key hyperparameter in this model can push the validation AUROC higher. For the rest of the models, issue with backpropagation or "dead neurons" happened frequently if ReLu activation function is used for dense layers. This issue is resolved by using LeakyReLu activation function.

AUROC Curve

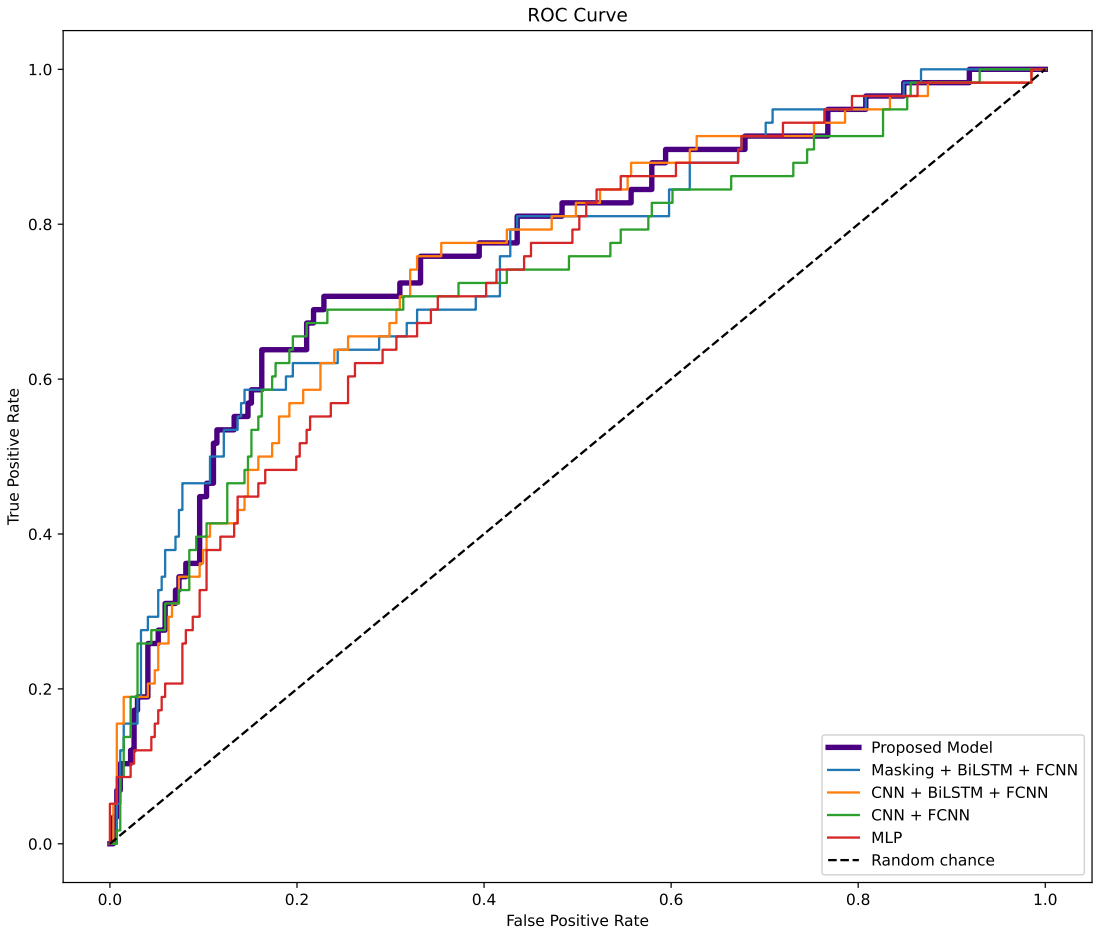

Figure 4.6: AUROC Plot for Different Models

To plot the AUROC curve in figure 3.22, for each model, one of the folds that roughly represents the mean AUROC test result is selected. We can see that the proposed model has the highest AUROC score followed by Masking + BiLSTM + FCNN and the rest. These models are trained to classify patients that require antibiotic retreatment from those who do not require antibiotic treatment by learning how clinicians determine antibiotic retreatment. However, there is an inherent variance in the decision of antibiotic administration by clinicians so it is unlikely that AUROC can go as high as 0.8 or 0.9.

Confusion Matrix of Proposed Model

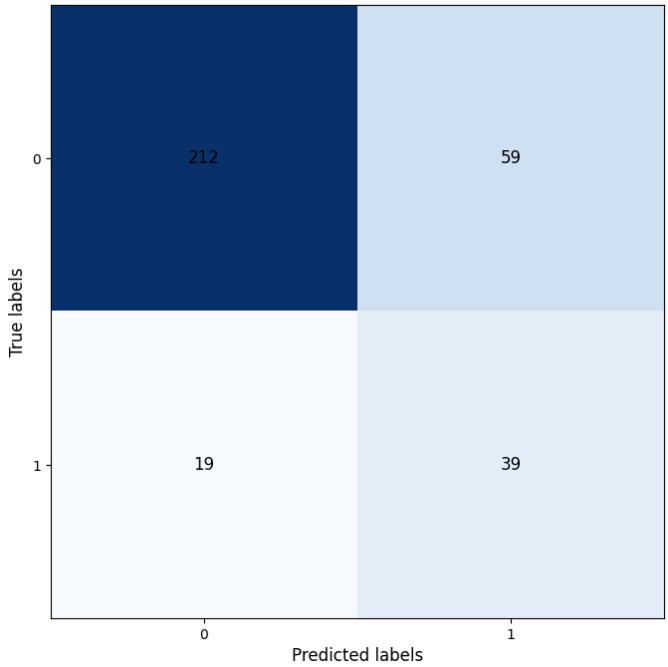

Figure 4.7: Confusion Matrix for selected fold of proposed model

Using the model from the same fold selected to plot the AUROC curve, confusion matrix is computed to visualise the true positive (TP) , false positive (FP), true negative (TN) and false negative (FN) of the prediction of proposed model on the test set. Out of the 329 cases in the test set, the proposed model is able to predict 251 (TP+TN) of them correctly, reaching an accuracy of 76.3%. 59 cases are false positives while 19 cases are false negatives. From the antibiotic treatment and patient perspective, the cost of false negative is actually much higher than false positive. Assuming the clinician follows the recommendation from this model, in the case of false positive, clinician will not stop the antibiotic treatment of a patient and this would cause antibiotic treatment to be longer than it should be. Whereas the consequences of false negative are much more severe, false negative essentially means that antibiotic treatment is stopped when longer treatment or more effective treatment (use of more potent antibiotics) is required. As a result, infection will likely relapse as patient is undertreated or treated ineffectively, causing a higher risk of death. Therefore, another set of antibiotic treatment will be required for false negative's patient. Hence, the risk of developing AMR is much higher for false negative patient as false positive only causes extra 2-3 days of extended treatment or alternative treatment but false negative patient needs at least 5-7 days of new antibiotic treatment. False negative case also likely leads to worse patient outcomes.

AUPRC Curve of Proposed Model

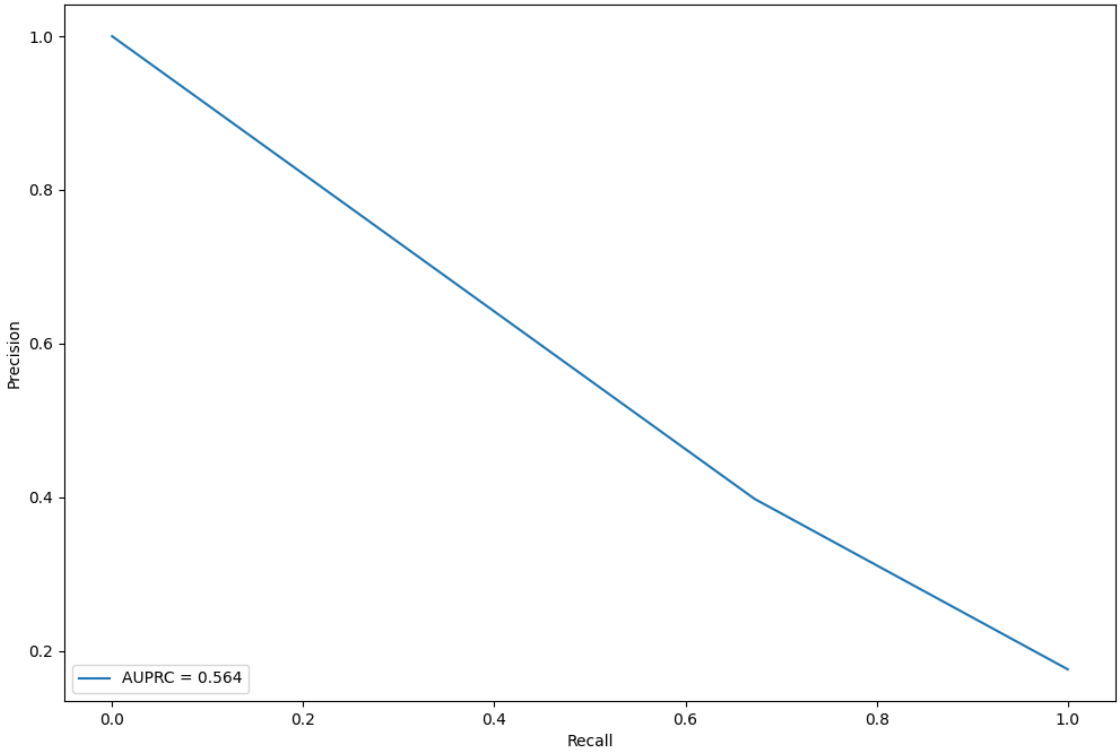

Figure 4.8: AUPRC Plot for selected fold of Proposed Model

Following the argument of false positives and false negatives, high AUPRC score is desirable as the model is more effective in identifying positive cases while minimising the error in predicting false positive. For this selected fold, proposed model is able to achieve an AUPRC of 0.564 (baseline is 0.176).

4.2.2 SHAP Analysis

In order to interpret the logic behind the proposed model, SHAP (SHapley Additive exPlanations) analysis is conducted to measure the importance of features on the prediction. Figure 4.9 shows the top 20 most important features from SHAP analysis.

This result is obtained by averaging all the predictions for the test case that come from all 5 folds of the proposed model. Note that the SHAP package does not support Tensorflow Version 2.4+, therefore a lower version of Tensorflow is used to train the model. Despite using the reported hyperparameters, there is still some slight variation in the backend computation. Therefore, the trained model used for SHAP Analysis is not the exact model reported in table 4.3. Nonetheless, the SHAP analysis result should not deviate too much

from the optimal model. Another important aspect is that only a subset of features is normalised in the data, this leads to a difference in scale between features which could affect the relative importance ranking. However, the prediction performance is built on top of having only a subset of features being normalised. Hence, the features importance analysis will be based on these 2 main assumptions; model trained with a lower version of Tensorflow and difference in scale between features.

In figure 4.9, it is found that GCS - Verbal Response is the most important feature followed by Age, GCS - Motor Response, GCS - Eye Opening, Arterial Base Excess and Total Duration. A detailed interpretation of the result is included in the discussion section.

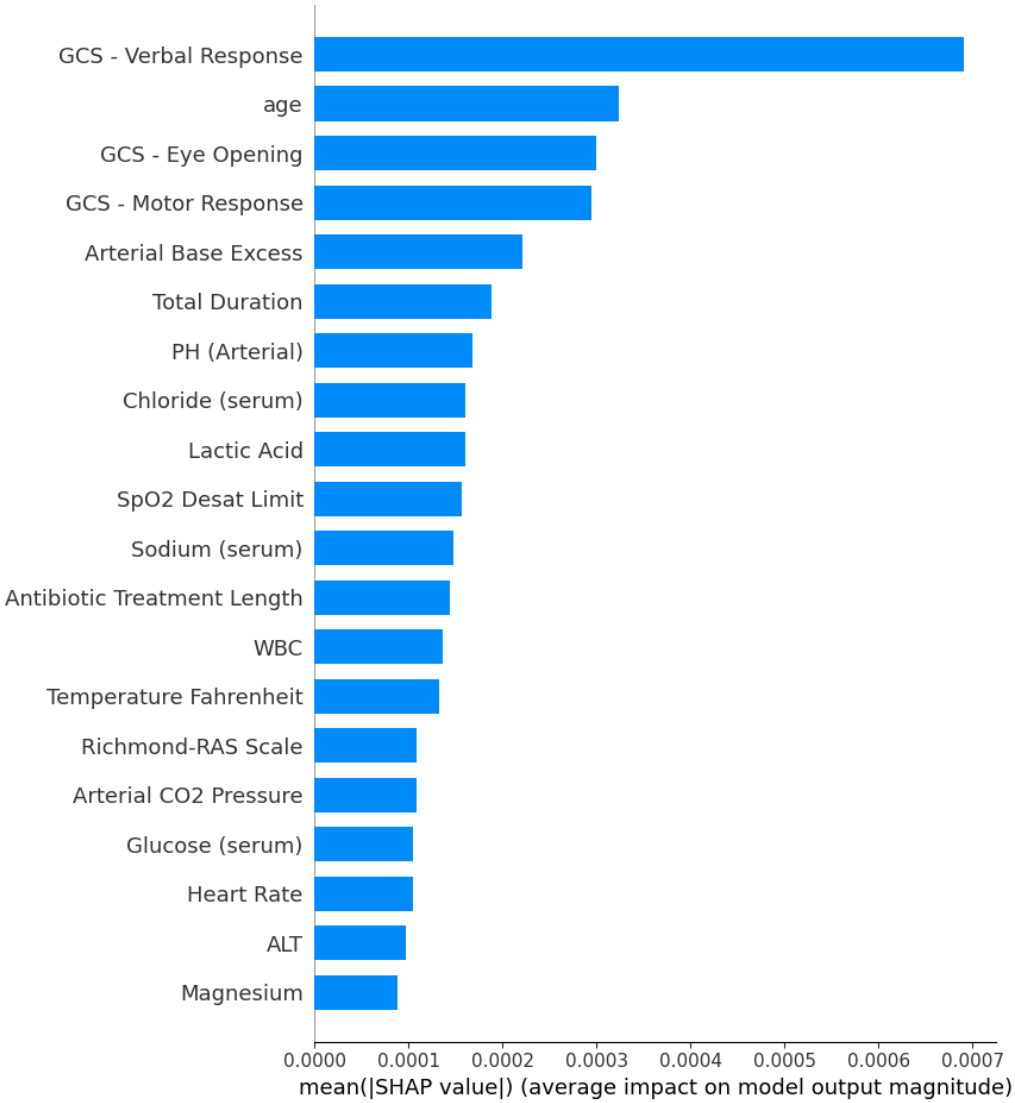

Figure 4.9: Top 20 Most Important Features - SHAP Analysis

4.3 Regression Results

4.3.1 Performance of Different Deep Learning Models

The process of tuning hyperparameters and obtaining test results is similar to binary classification task. The key difference here is that MSE and MAE are the primary metrics to optimise during the training process. Table 4.4 summarises the performance of different models in the regression task.

	MSE	**MAE**	**RMSE**
Proposed model	**2.94 (2.90 - 2.98)**	**1.26 (1.20 - 1.31)**	**1.71 (1.70 - 1.73)**
Masking + BiLSTM + FCNN (2 Layers of BiLSTM)	3.06 (2.98 - 3.14)	1.31 (1.22 - 1.39)	1.75 (1.73 - 1.77)
CNN + BiLSTM + FCNN	3.13 (2.81 - 3.44)	1.34 (1.21 - 1.46)	1.77 (1.68 - 1.85)
CNN + FCNN	3.27 (3.06 - 3.48)	1.46 (1.38 - 1.53)	1.81 (1.75 - 1.86)
Multi-layer Perceptron (MLP)	3.47 (3.17 - 3.76)	1.52 (1.43 - 1.61)	1.86 (1.78 - 1.94)

Table 4.4: Performance of Different Models in Regression Task

Similar to binary classification task, the proposed model remains the model with the best results across all the evaluation metrics. The proposed model achieved MSE of 2.94, MAE of 1.26 and 1.71 in RMSE. It outperformed the Masking + BiLSTM + FCNN model by a respectable margin.

A slight difference from the classification task is that Masking + BiLSTM + FCNN model has an extra layer of BiLSTM being added to increase the complexity of the model in order to explore more sophisticated relationships in the data. This additional complexity could have contributed to making Masking + BiLSTM + FCNN model the second best model. The common feature between the proposed model and Masking + BiLSTM + FCNN model is that Masking + BiLSTM is used as the primary layers in both models. This indicates that the combination of Masking + BiLSTM structure is very effective in analysing the underlying patterns of time series data to predict antibiotic retreatment. It also reinforces the findings from the classification task whereby the masking layer helps keep the model performance stable and robust. In summary, the performance of both models is largely influenced by this factor. The regression task is more challenging to

tackle, therefore well designed complex models will perform better. To summarise, all the advanced deep learning models surpassed the performance of the baseline model, MLP. From the stability perspective, proposed model also significantly outperformed all other models which is shown in its small prediction variation across all the folds.

Prediction Analysis

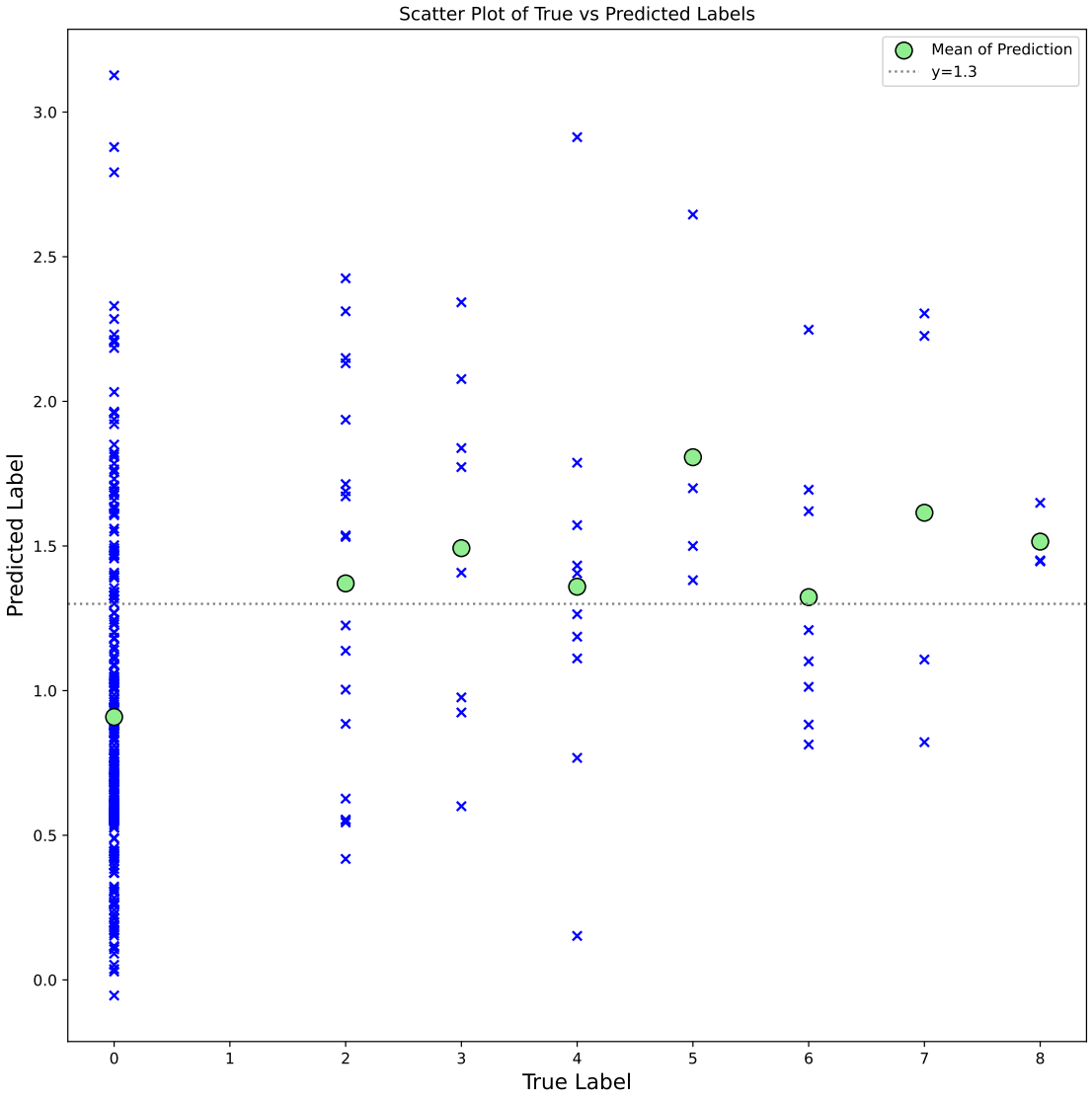

Figure 4.10: Prediction of Proposed Model

Proposed model with the lowest MSE score across the 5 folds is selected to visualise its prediction output and it is illustrated in figure 4.10. Note that antibiotic retreatment/label of 1 day is removed as described in the data preprocessing pipeline. It is observed that most of the predictions lie in the range of 0-3 days. Ideally, the prediction should scatter around the line y=x.

The difficulty of the regression task lies in the fact that

- First, the model has to differentiate between positive cases and negative cases implicitly
- Then, the model can proceed with predicting the antibiotic retreatment
- However, the data is imbalanced, there are not enough positive cases to train the model to reliably predict the antibiotic retreatment with day-level accuracy

Another additional challenge is that the larger the antibiotic retreatment days, the less obvious the "symptoms" of any potential ongoing infection for a patient and the less risk of treatment failure, making the model more likely to predict some negative cases (that exhibits slight risk of antibiotic retreatment) in the larger range of antibiotic retreatment. For instance, if a patient is retreated 8 days after the initial antibiotic treatment, the characteristics of this patient will be very similar to the patient that never receive any retreatment as the "symptoms" of potential ongoing infection is not very obvious and infection only relapses after 8 days. From a clinical perspective, it is also very challenging to distinguish negative cases and positive cases with large antibiotic retreatment days accurately, so it is expected that the model will struggle to produce accurate prediction for these cases.

Despite that, there is still a clear separation between negative cases (True Label=0) and positive cases (True Label>0) in terms of the mean of prediction. We can observe that for all the positive cases, prediction means are well above the prediction value of 1.3 days. While the mean prediction for negative cases is around 0.9 days. This difference indicates that model is still able to distinguish positive cases from negative cases in a broad sense. Although the prediction results are far from satisfactory, this result should be able to serve as a benchmark or foundation for future work in this area. To my knowledge, very few or none has attempted in predicting antibiotic retreatment especially as regression task.

4.4 Joint Learning Results

4.4.1 Performance of Joint Learning Model in Classification and Regression Task

Following the discussion of model performance in regression task, the joint learning task is attempted by training the joint learning model (modified version of proposed model) as shown in figure 4.11 with both binary and continuous labels to boost the prediction performance by exploiting the commonality between the 2 different labels.

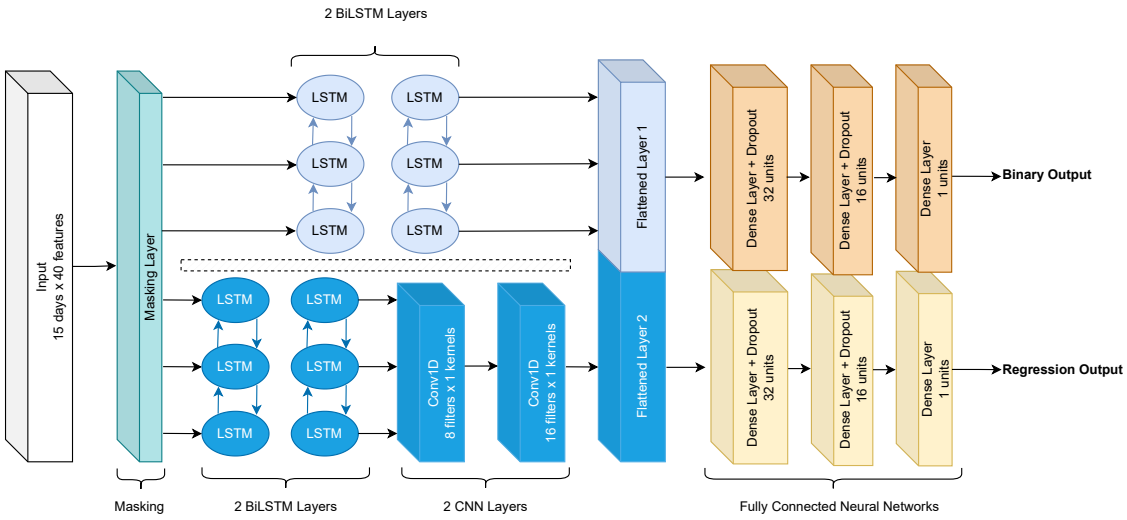

Figure 4.11: Architecture of Joint Learning Model (The flattened layer is common layer for both outputs, hyperparameters setting is included in Appendix A)

	AUROC	AUPRC	Accuracy
Joint Learning Model	**0.76 (0.74 - 0.77)**	**0.50 (0.46 - 0.54)**	**0.80 (0.79 - 0.80)**
Proposed Model	**0.76 (0.74 - 0.78)**	**0.54 (0.52 - 0.57)**	**0.74 (0.72 - 0.75)**
Masking + BiLSTM + FCNN (1 layer of BiLSTM)	0.75 (0.75 - 0.76)	0.54 (0.52 - 0.55)	0.73 (0.71 - 0.74)
CNN + BiLSTM + FCNN	0.75 (0.73 - 0.77)	0.53 (0.52 - 0.54)	0.73 (0.71 - 0.74)
CNN + FCNN	0.73 (0.72 - 0.75)	0.52 (0.50 - 0.54)	0.69 (0.64 - 0.75)
Multi-layer Perceptron (MLP)	0.73 (0.70 - 0.75)	0.49 (0.45 - 0.52)	0.70 (0.61 - 0.79)

In the classification task, joint learning model's accuracy outperformed all the other models by a large margin but high accuracy comes with a price where this model underperformed

	Recall	Precision	F1-score
Joint Learning Model	**0.67 (0.64 - 0.70)**	**0.66 (0.64 - 0.67)**	**0.66 (0.64 - 0.68)**
Proposed Model	**0.71 (0.69 - 0.73)**	**0.64 (0.63 - 0.65)**	**0.65 (0.63 - 0.66)**
Masking + BiLSTM + FCNN (1 layer of BiLSTM)	0.70 (0.69 - 0.72)	0.63 (0.62 - 0.65)	0.64 (0.63 - 0.66)
CNN + BiLSTM + FCNN	0.70 (0.69 - 0.71)	0.63 (0.62 - 0.64)	0.64 (0.63 - 0.65)
CNN + FCNN	0.68 (0.66 - 0.69)	0.62 (0.60 - 0.63)	0.61 (0.58 - 0.64)
Multi-layer Perceptron (MLP)	0.65 (0.64 - 0.66)	0.61 (0.60 - 0.62)	0.60 (0.55 - 0.65)

Table 4.5: Joint Learning Model Performance in Classification Task

the proposed model in some other metrics. Although its AUROC is still comparable with the others model, the AUPRC score is lower at 0.50. This implies that the ability of joint learning model to detect positive cases is weaker and this is also evident in the recall metric. Conversely, the model's precision and F1 score are higher. We can observe that there is some tradeoff between different metrics.

	MSE	**MAE**	**RMSE**
Joint Learning Model	**2.88 (2.77 - 2.99)**	**1.23 (1.10 - 1.35)**	**1.70 (1.66 - 1.73)**
Proposed model	**2.94 (2.90 - 2.98)**	**1.26 (1.20 - 1.31)**	**1.71 (1.70 - 1.73)**
Masking + BiLSTM + FCNN (2 Layers of BiLSTM)	3.06 (2.98 - 3.14)	1.31 (1.22 - 1.39)	1.75 (1.73 - 1.77)
CNN + BiLSTM + FCNN	3.13 (2.81 - 3.44)	1.34 (1.21 - 1.46)	1.77 (1.68 - 1.85)
CNN + FCNN	3.27 (3.06 - 3.48)	1.46 (1.38 - 1.53)	1.81 (1.75 - 1.86)
Multi-layer Perceptron (MLP)	3.47 (3.17 - 3.76)	1.52 (1.43 - 1.61)	1.86 (1.78 - 1.94)

Table 4.6: Joint Learning Model Performance in Regression Task

While in the regression task, joint learning model performance is improved as compared to the proposed model (regression) with a lower value in MSE, MAE and RMSE as shown in 4.6. This means that the joint learning method is useful in boosting the prediction performance on regression task.

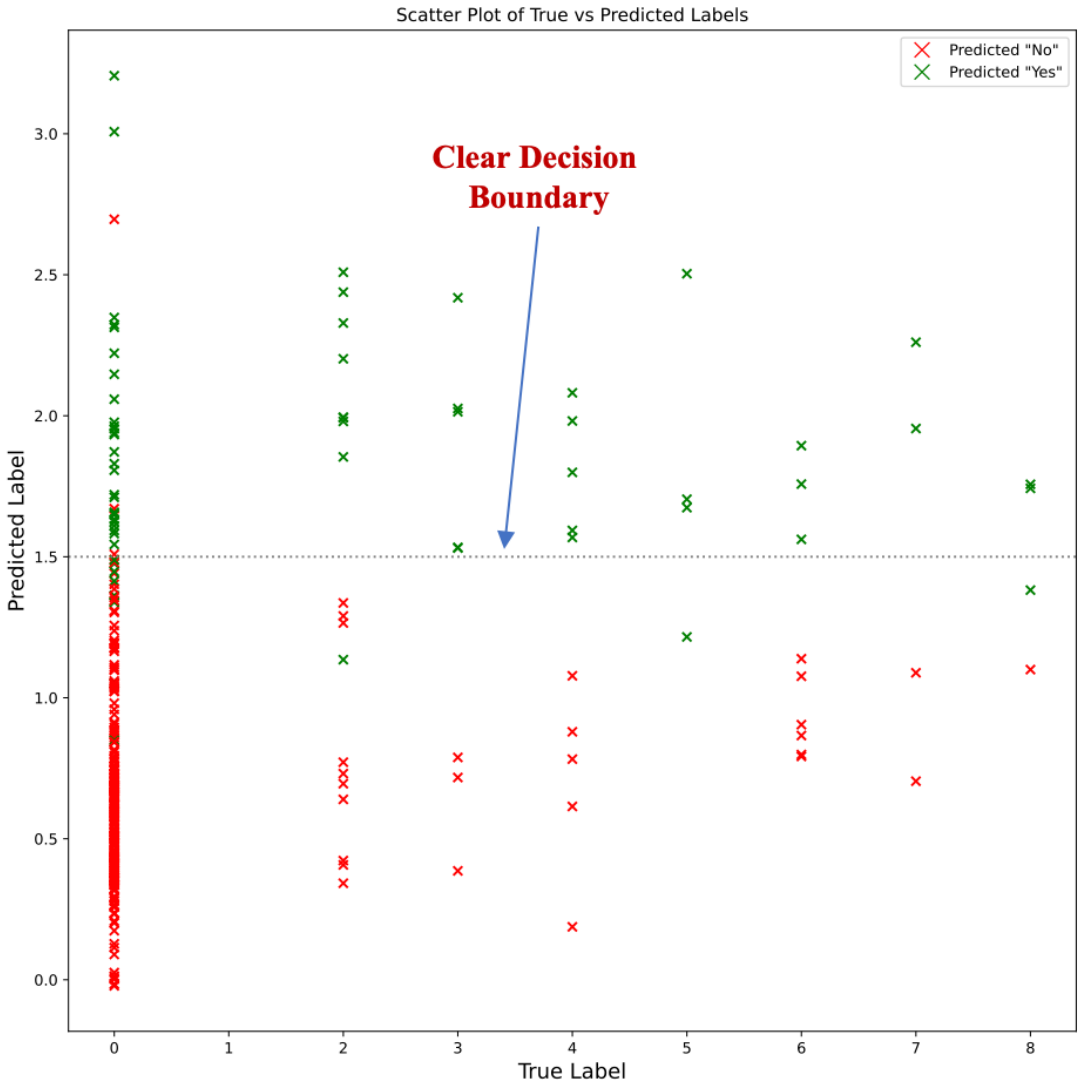

Figure 4.12: Prediction of Joint Learning Model

In order to analyse the prediction results of the joint learning model in the test set, the fold with the best overall performance is chosen to plot figure 4.12 above. The green markers in the graph refer to cases where the model predicted "retreatment required" and red markers refer to cases where the model predicted "no retreatment required". From the plot, it is observed that there is a clear decision boundary learned by the model at 1.5 days, meaning that cases with predicted retreatment days of more than 1.5 days are generally predicted "retreatment required" by the binary output of the model. Thus, the binary and regression prediction output agreed with each other. This decision boundary is desirable as the model is much more confident in predicting "retreatment required" when regression output is >1.5 days. For this reason, unless the corresponding regression outputs of cases are >1.5 days, the model is more constrained to predict any cases as positive cases, resulting in a drop in

AUPRC score. For positive cases that were predicted "retreatment required" (true label>0 and green marker), the regression predictions are mostly at the range of 1.5-3 days which are very reasonable. However, the predictions are still not scattered around the line y=x. Despite this, the prediction could still provide some utilities to clinicians if both binary and quantitative estimation of antibiotic retreatment are needed.

Chapter 5

Discussion

This chapter provides further analysis on the prediction of proposed model in binary classification task. Limitations, further work and conclusion of this study are also included in this chapter.

5.1 Features Importance Analysis

This subsection is dedicated to explaining the SHAP analysis conducted on proposed model in the binary classification task. For ease of referencing, the SHAP analysis result is included on the next page as figure 5.1.

GCS - Verbal Response

GCS - Verbal Response score is a measure of the patient's level of consciousness on whether the patient is responsive or able to engage in conversation fluently. This score ranges from 1-5 where a score of 1 means that the patient is unconscious and a score of 5 means the patient is oriented. The higher the score, the better the overall condition of the patient. By visually inspecting the dataset, most patients have a GCS verbal score of 1 when they start their antibiotic treatment and a score of 4-5 when the treatment ended. This suggests that the deep learning model relies heavily on GCS - Verbal Response response to assess the physiological condition of patients particularly the severity of their conditions or infection to make reliable predictions. Apart from that, the time series nature of the data allows the model to capture the evolution of the patient's condition using the GCS - verbal response score throughout the entire antibiotic treatment. Another point is that the GCS - Verbal Response is one of the most common clinical variables in the ICU setting as patients are

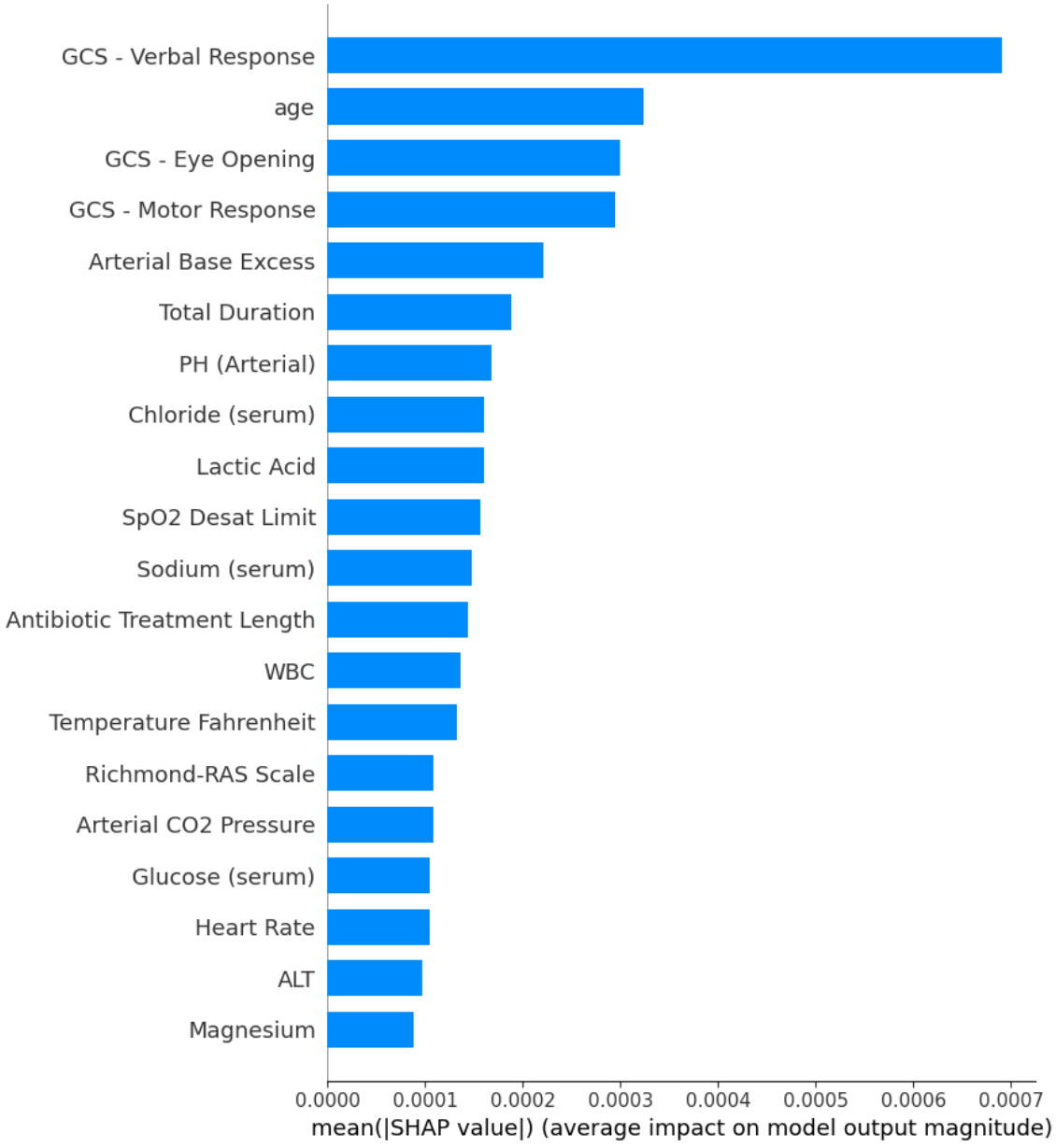

Figure 5.1: Top 20 Most Important Features - SHAP Analysis (2)

closely monitored by the doctors. Therefore, the level of missingness for this feature is very low so the data for this feature is generally complete and continuous. In simple terms, the GCS - Verbal Response allows the model to analyse the recovery progress of a patient during the treatment period in a more direct way. Although detailed analysis is not carried out specifically on the dataset for this feature, there is a strong possibility that if a patient shows great improvement in this score from 1 to 5 throughout the treatment, the risk of requiring antibiotic retreatment is much lower.

Age

Following GCS - Verbal Response, age is the next most important feature. It is widely agreed by the medical community that the larger the age, the weaker the immune system. Due to aging body cells, the metabolism rate is much lower in older individuals, leading to a lower rate of antibiotic production by the body. Therefore, when facing infection, younger individual has a tendency to recover at a faster rate and low risk of needing antibiotic retreatment. Further literature review on the relation between age and infection shows that aging causes immunosenescence which gives rise to a higher risk of infection [72]. The physiologic changes due to old age also make an individual more susceptible and vulnerable to infection [72]. Thus, age is a crucial factor that can significantly influence the risk of relapse of infection which could lead to antibiotic retreatment.

GCS - Motor Response and Eye Opening

Similar argument in GSC - Verbal Response applies to GCS - Motor Response and Eye Opening. The reason why GCS - Verbal Response is more important is that Verbal Response score is much more intuitive to the model. Infection generally causes delirium or sepsis among ICU patients and these conditions affect patient's consciousness and ability to converse. Most patients that experience delirium or sepsis are either confused or in a coma. Hence, this results in an obvious drop in the GCS - Verbal Response among infected patients as compared to the other GCS scores. It is no surprise that the model places more emphasis on the GCS - Verbal response as compared to the other GCS score. Nevertheless, GCS - Motor Response and Eye Opening give valuable information on the patient's recovery progress from another perspective.

Arterial Blood Gas (ABG) - Arterial Base Excess and Lactic Acid

Arterial blood gas (ABG) is a rapid test that usually takes only a few minutes to complete, it measures the level of oxygen and carbon dioxide as well as pH level in the blood [73]. This test includes measuring arterial base excess and pH (arterial) in the blood and it is commonly carried out on patients that are in mild-to-severe conditions, particularly ICU patients. According to Dr. Mandy, it enables medical doctors to identify the possible causes of acute deterioration (e.g. low level of oxygen in the body, electrolyte imbalance, acid-base imbalance) to ensure the right treatment can be delivered in time. She further explained that almost all infections can cause alteration to the blood dynamics given that infection induces toxicity to body organs. One of the most prevalent infections in ICU that can result in abnormal ABG test is related to lung infections such as pneumonia, this condition could have added to the importance of this feature. This is because lung infection reduces oxygen intake into the body, disrupting the level of oxygen in the blood and causing the accumulation of carbon dioxide. This subsequently leads to acidosis which is easily captured by ABG test. Therefore, the machine learning model places considerate weights on the ABG test specifically arterial base excess and pH (arterial) variables to make reliable predictions as ABG test is an important indicator that can gauge the severity of the infection as well as a way to identify the causes of deterioration. This test is also readily available in the ICU, providing enough set of data points for analysis. For full disclosure, the arguments for ABG test incorporated clinical guidance and information from Dr. Mandy.

Total Duration = Antibiotic Treatment Length + (If Any) Days in ICU Before Treatment Starts

Unexpectedly, antibiotic treatment length is not the most important feature for predicting antibiotic retreatment. It is commonly assumed that extended antibiotic treatment reduces the risk of antibiotic retreatment. From a wide perspective, there is a much more complex relationship between antibiotic treatment length and antibiotic retreatment. Antibiotic treatment length for every individual varies a lot. It is heavily dependent on the patient's underlying disease and condition. Therefore, antibiotic treatment length for every individual is determined by a number of factors and it is challenging to determine the most optimal treatment length. Thus, it might not be straightforward to say that if a patient has received more than 7 days of treatment, the risk of antibiotic retreatment is minimal

because the condition that this patient experiences might require 14 days of treatment for a full recovery. There is a non-linear relationship between treatment length and antibiotic retreatment. In addition to this, all patients in this dataset have received at least 5 days of antibiotic treatment which is a relatively extensive treatment length. In view of this, other factors might play more significant roles in determining the risk of antibiotic retreatment. Hence, the model might place a stronger emphasis on analysing total duration (length of ICU stay until treatment stops) which is inclusive of the antibiotic treatment length as this feature might provide a more comprehensive view of the overall treatment received by patients in ICU.

There might be question raised on "Why diagnosis information was not included as a feature?". To answer this, we need to understand how diagnosis info is stored. Generally, diagnosis info can be found in doctor's notes or under (International Classification of Diseases - ICD) ICD-9/ICD-10 as numeric codes. To include diagnosis information, it is necessary to convert doctor's notes that are in free text or extract ICD-9/10 directly from the database. In order to let the model understand the diagnosis info, we have to either train free-text as embeddings or include ICD as one-hot encoding in the features because a patient can have multiple conditions/complications at the same time so an extra column in the input feature is not sufficient as shown in this paper [48]. The former is very time-consuming and not feasible within the project timeframe, while the latter would introduce a lot more dimensions to the size of input features, further increasing training difficulty to identify patterns in the data. In MIMIC-IV, most diagnosis information is also not associated with timestamp, resulting in difficulty to incorporate it into the time series data. To simplify, the model cannot be trained with the assumption that diagnosis information is always available at the start of antibiotic treatment. On top of that, the size of the dataset is very limited for this to be achievable.

5.2 Further Discussion

One of the most challenging parts of this project is dataset creation. Navigating the complex MIMIC-IV's relational database requires extensive reading on documentation as well as expertise in SQL to extract information. Following that, specifying the setting and study population that makes this project feasible while making the most impact for this area of study is extremely rigorous as a high level of understanding about antibiotic treatment in ICU is needed. Throughout this project, numerous trials have been attempted to clean and create a dataset from MIMIC-IV but all sorts of issues have been encountered especially the timing of antibiotic treatment and retreatment. This is mainly due to the complexity of the time series nature of EHR data. It is shown that clever design choice of limiting the study population to patients that have received IV antibiotic treatment in their first ICU stay on a single hospital stay enables the project to focus on patients that are at the highest risk of developing AMR. On top of that, by constraining the start and stoppage timing of treatment within 1 ICU stay, noises in data are further reduced which makes analysis of patient's condition and predicting antibiotic readmission through deep learning achievable. While the issue of the sparsity of data is resolved by setting a minimum treatment length of 5 days. Overall, the filtering process and techniques described in the data preprocessing pipeline is a major milestone that signifies partial success of this project and could be very beneficial for researchers in the ML healthcare field.

A number of standard machine learning practices are also taken in place to ensure validity of this study, for instance stratified sampling , oversampling in cross validation, data leakage prevention and etc. To elaborate on this topic, initially, min max normalisation was applied to training set and test set with their respective maximum and minimum parameters (the correct way is to use parameters in the training set only). It is observed that this caused serious data leakage and distorted the evaluation results of the model particularly models' accuracy are underestimated by 3% to 4%. Therefore, a lot of efforts are dedicated to making sure that all aspects of this study are conducted based on best practices. During the training and evaluation process, a lot of measures are also in place to ensure the reproducibility of results. First, random seeds are set for the operating system environment, Python, Tensorflow, Numpy as well as Tensorflow's inter and intra-parallelism. On top of that, the platform used, GPU, Tensorflow version, and any relevant information are recorded. The history logs of the model training and test evaluation are also included

in the code as evidence to support the reported results. However, reproducibility of results can still be affected by the non-determinism of GPU computation (floating-point calculation) and changes in Google Colab's GPU acceleration so some variations in results are inevitable.

The first observation from the results section is that all models show more desirable outcomes in the classification task than in the regression task. This means that the regression task is much more challenging as expected. Looking at the joint learning model, it is observed that regression loss accounts for roughly 85% of the overall loss, while classification loss only contributes approximately 15%. This indicates a lot of the losses are largely driven by regression task, further affirming that regression is very difficult to be tackled. Nonetheless, the joint learning model still shows highly comparable results to the proposed model in classification task and outperformed all the other models in regression task. This is a piece of strong evidence that signifies regression results are further improved through the joint learning method. In fact, prediction results from the joint learning model with an MAE of 1.23 days are potentially highly useful clinically as doctors have been shown to be poor in predicting regression-based tasks historically (e.g. MAE of 3.82 days for predicting hospital length of stay from emergency department - [74]).

On the other hand, it is also quite probable that the performance of the joint learning model is limited due to the size of the dataset and the number of positive cases. This statement applies to the proposed model in the regression task. Whilst the proposed model in regression task is not able to achieve day-level accuracy, it could still classify the negative cases and positive cases implicitly as observed in the mean of prediction (negative cases predictions have a mean of around 0.9 days and positive cases predictions have means of around 1.3 days). Whereas for the classification task, the proposed model clearly shows excellent performance with an AUC of 0.76. Overall, the proposed architecture is able to achieve great performance in different tasks.

To sum up, the findings from this project are very novel, it provides a good research precursor or foundation to predict antibiotic readmission. By combining the models created with other antibiotic stewardship tools, holistic decision support can be provided to assist with understanding patient recovery trajectory and supporting antibiotic treatment optimisation decisions.

5.3 Limitations

Classification Task

The proposed model has some limitations in classification task despite outperforming all other models in the test set. As discussed previously, throughout the hyperparameters tuning process and observations on the validation results, it is found that the model suffers from low accuracy in exchange for high AUPRC. This results in a drop across the precision and F1 score metrics with the recall score being affected slightly as all these metrics are dependent/related to accuracy. This is because the proposed model is very sensitive to changes in hyperparameters than other models with simpler architecture. In general, the proposed model places a lot of weight on identifying positive cases which is the minority class in the dataset, leading to lower accuracy (+ precision and F1 scores) because the majority of cases in the data are negative cases. Expect an approximately 1-4% drop in accuracy and a 1-3% drop in recall, precision and F1 scores if the model is not properly tuned. To achieve the reported results, careful and fine hyperparameters tuning is essential for balancing AUPRC and accuracy.

That being said, the performance of the proposed model in the classification task could be limited due to the number of positive cases and the small training set. Apart from that, the proposed model has more capacity to learn as compared to other models. Therefore, it can easily overfit to data in certain training folds, leading to suboptimal performance in the test set. However, it is worth noting that high AUROC score is still observed all the time, suggesting that the proposed model has good capability in discriminating between positive cases and negative cases. Another observation in the classification task is that it was particularly difficult to tune the hyperparameters of CNN + BiLSTM + FCNN model as its performance is also very sensitive to hyperparameters changes.

Common limitations

The common limitation for all the tasks is that it is impossible to explore all the possible sets of hyperparameters for each model because hyperparameters tuning is a very lengthy and computationally expensive process. Therefore, this study is about evaluating different models as objectively as possible to provide a reliable estimation of their performance. Another limitation is that there might be some positive cases extracted from the dataset happened due to newly acquired infection instead of failures in initial antibiotic treatment.

Although we limited the positive cases to those who received antibiotic retreatment within a short timeframe (2-8 days) in order to focus on patients that received retreatment due to failures in initial antibiotic treatment but it is almost imposssible to fully eradicate positive cases caused by newly acquired infection or any unpredictable reasons.

5.4 Further Work

There are still several areas that can be extended to further verify and refine the findings of this study.

First, we can reinforce the findings from this project through external validation by using a different database like eICU. It allows us to test the generalisability of models using data collected in a different setting or health system but this could pose several challenges from the data preprocessing perspective. For instance, some clinical variables might be extensively collected in MIMIC-IV but do not exist in other databases. Other possible differences include units of measurement and measurement frequencies. Hence, a thorough investigation has to be done to identify the characteristics of the database and the potential limitations of the external validation results.

Another approach could be assimilating and expanding the size of dataset by combining data from multiple medical databases. Dataset expansion is a very timing consuming task due to similar reasons mentioned above. With a larger dataset, trained model should be able to analyse and make more accurate predictions theoretically. Another benefit of assimilating data from multiple sources is that features that were previously excluded such as diagnosis information, PCT and CRP can be included to examine their impacts on antibiotic retreatment prediction. Apart from that, the number of cases in the test set will also be increased, enabling accurate evaluation of the model. In this project, it is found that data splitting could result in significant variance between datasets. Therefore, this should also be mitigated to a certain extent after dataset expansion.

In this study, only a handful of deep learning architectures have been tested and evaluated. Other newly invented architecture in the ML field like Transformer or Autoencoder can also be explored to find out their limits and possibilities of being applied in the antibiotic readmission prediction. Due to the strict timeline and lack of computational resources, hyperparameters search was conducted manually for all the models. In the future, exhaustive and extensive hyperparameters search can be carried out using method like grid search to train and evaluate the performance of models systematically.

5.5 Conclusion

In this project, antibiotic readmission prediction is addressed as both binary and regression task. Joint learning method is also explored as a way to boost the prediction performance. Multiple key milestones are achieved. First, a carefully designed data preprocessing pipeline has been created to solve a number of issues related to EHRs data. Specific study population in the ICU were identified and focused on to maximise the impact of this study. The data preprocessing pipeline involved identification and selection of routinely collected clinical variables/features that are useful to analyse the patient's condition from the start until the end of antibiotic treatment in the ICU. Following that, patients' data are extracted and converted into interpretable time series representation using techniques such as outliers removal, daily aggregation, last observation carried forward and padding.

Different machine learning methods are applied to train and evaluate the performance of models holistically. In this study, a range of deep learning models have been explored and investigated in both classification task and regression task. A deep learning architecture, Masking + BiLSTM + CNN + FCNN is also proposed and benchmarked against other advanced deep learning architectures. The proposed model achieved excellent clinical-level performance in classification and regression tasks with AUC of 0.76 and MAE of 1.26 days respectively. Improved performance in regression task is observed with MAE of 1.23 days and MSE of 2.88 days when the proposed architecture is trained using joint learning method. Further analysis of the prediction of proposed model in the classification task indicated that the features used in predicting antibiotic readmission by the model is highly inline with inputs from clinician that emphasise on clinical representation and recovery progress of patient. The proposed architecture developed in this project offers reliable prediction to assist clinicians on antibiotic treatment continuation or cessation decisions and it demonstrated different capabilities across 3 different tasks, classification, regression and joint learning. For example, proposed model (classification) is strong in detecting positive cases, proposed model (regression) provides accurate quantitative estimations while joint learning model has an interpretable decision boundary for predictions and provides classification + regression predictions simultaneously. Actual application could be based on single or multiple models. Ultimately, it should be dependent on which task or which specific capability is more helpful in the ICU setting by obtaining clinicians' feedback. Since this area of study is not well explored, findings included in this report could bring

significant contribution to antibiotic optimisation through machine learning. In summary, antibiotic readmission prediction is viable through deep learning approach, this could help in improving stewardship and mitigating the development of AMR.

Appendix A

Hyperparameters Setting for All Models and Code

In binary classification task, the fine tuning process generally involves 2 key hyperparameters namely, training epochs and dropout rates. Note that other hyperparameters are tuned but these 2 parameters are used specifically for fine tuning. The general approach is to identify the upper limit of validation AUROC as close as possible for each model by adjusting dropout rates and training epochs as well as prevent overfitting to data as the training dataset is relatively small. Since the tradeoff between accuracy and AUPRC is commonly observed in this task, fine-tuning dropout rates with increment or decrement of 0.01 is performed to find a great balance between AUROC, accuracy and AUPRC metrics. The idea is to identify the hyperparameters that give the best validation results across all metrics while having great performance in AUROC. Similar tuning approach is used in regression task (minimising MSE and MAE) and joint learning task (balancing all relevant metrics - AUROC, AUPRC, MSE and MAE).

A.1 Binary Classification Models

Common Settings

Platform	Google Colab
Tensorflow	2.12.0 Version
GPU Setting in Colab	T4, High RAM
Loss Function	Binary Crossentropy
Optimizer	Adam, default_value = 0.001

Table A.1: Common Settings

Hyperparameter	Setting
Lower Branch	
BiLSTM - Layer 1	25 units , return_sequences = True
BiLSTM - Layer 2	25 units , return_sequences = True
Conv1D - Layer 1	8 filters, 1 kernel, ReLU
Conv1D - Layer 2	16 filters, 1 kernel, ReLU
Upper Branch	
BiLSTM - Layer 1	25 units , return_sequences = True
BiLSTM - Layer 2	25 units , return_sequences = True
Concatenation	
Dense - Layer 1	32 units, ReLU
Dropout Layer	0.29
Dense - Layer 2	16 units, ReLU
Dropout Layer	0.04
Dense - Layer 3	1, Sigmoid

Table A.2: Proposed Model

Hyperparameter	Setting
BiLSTM - Layer 1	20 units , return_sequences = True
Dense - Layer 1	32 units, ReLU
Dropout Layer	0.39
Dense - Layer 2	16 units, ReLU
Dense - Layer 3	1, Sigmoid

Table A.3: Masking + BiLSTM + FCNN (1 layers of BiLSTM)

Hyperparameter	Setting
Conv1D - Layer 1	8 filters, 1 kernel, ReLU
Conv1D - Layer 2	16 filters, 1 kernel, ReLU
Conv1D - Layer 3	32 filters, 1 kernel, ReLU
BiLSTM	30 units, return_sequences = True
Dense - Layer 1	32 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.1
Dense - Layer 2	16 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.1
Dense - Layer 3	1, Sigmoid

Table A.4: CNN + BiLSTM + FCNN

Hyperparameter	Setting
Conv1D - Layer 1	8 filters, 1 kernel, ReLU
Conv1D - Layer 2	16 filters, 1 kernel, ReLU
Conv1D - Layer 3	32 filters, 1 kernel, ReLU
Dense - Layer 1	32 units, LeakyReLU (alpha =0.01)
Dropout Layer	0.1
Dense - Layer 2	16 units, LeakyReLU (alpha =0.01)
Dropout Layer	0.1
Dense - Layer 3	8 units, LeakyReLU (alpha =0.01)
Dropout Layer	0.01
Dense - Layer 4	1, Sigmoid

Table A.5: CNN + FCNN

Hyperparameter	Setting
Dense - Layer 1	32 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.1
Dense - Layer 2	8 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.1
Dense - Layer 3	8 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.05
Dense - Layer 4	1, Sigmoid

Table A.6: MLP

A.2 Regression Models

Common Settings

Platform	Google Colab
Tensorflow	2.12.0 Version
GPU Setting in Colab	A100, High RAM (Compute Units Needed)
Loss Function	MSE
Optimizer	Adam, default_value = 0.001

Table A.7: Common Settings

Hyperparameter	**Setting**
Lower Branch	
BiLSTM - Layer 1	35 units , return_sequences = True
BiLSTM - Layer 2	35 units , return_sequences = True
Conv1D - Layer 1	8 filters, 1 kernel, ReLU
Conv1D - Layer 2	32 filters, 1 kernel, ReLU
Upper Branch	
BiLSTM - Layer 1	35 units , return_sequences = True
BiLSTM - Layer 2	35 units , return_sequences = True
Concatenation	
Dense - Layer 1	32 units, ReLU
Dropout Layer	0.22
Dense - Layer 2	16 units, ReLU
Dropout Layer	0.1
Dense - Layer 3	1, Linear

Table A.8: Proposed Model

Hyperparameter	**Setting**
BiLSTM - Layer 1	20 units , return_sequences = True
BiLSTM - Layer 2	20 units , return_sequences = False
Dense - Layer 1	32 units, ReLU
Dropout Layer	0.25
Dense - Layer 2	16 units, ReLU
Dense - Layer 3	1, Linear

Table A.9: Masking + BiLSTM + FCNN (2 layers of BiLSTM)

Hyperparameter	Setting
Conv1D - Layer 1	8 filters, 1 kernel, ReLU
Conv1D - Layer 2	16 filters, 1 kernel, ReLU
Conv1D - Layer 3	32 filters, 1 kernel, ReLU
BiLSTM	35 units, return_sequences = True
Dense - Layer 1	32 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.25
Dense - Layer 2	16 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.05
Dense - Layer 3	1, Linear

Table A.10: CNN + BiLSTM + FCNN

Hyperparameter	Setting
Conv1D - Layer 1	8 filters, 1 kernel, ReLU
Conv1D - Layer 2	16 filters, 1 kernel, ReLU
Conv1D - Layer 3	32 filters, 1 kernel, ReLU
Dense - Layer 1	32 units, LeakyReLU (alpha =0.01)
Dropout Layer	0.25
Dense - Layer 2	16 units, LeakyReLU (alpha =0.01)
Dropout Layer	0.1
Dense - Layer 3	8 units, LeakyReLU (alpha =0.01)
Dropout Layer	0.05
Dense - Layer 4	1, Linear

Table A.11: CNN + FCNN

Hyperparameter	Setting
Dense - Layer 1	32 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.1
Dense - Layer 2	8 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.01
Dense - Layer 3	8 units, LeakyReLU (alpha =0.05)
Dropout Layer	0.01
Dense - Layer 4	1, Linear

Table A.12: MLP

A.3 Joint Learning Model

Common Settings

Platform	Google Colab
Tensorflow	2.12.0 Version
GPU Setting in Colab	T4, High RAM (Compute Units Needed)
Loss Function for Binary Classification	Binary Crossentropy
Loss Function for Regression	MSE
Optimizer	Adam, default_value = 0.001

Table A.13: Common Settings

Hyperparameter	**Setting**
Lower Branch	
BiLSTM - Layer 1	40 units , return_sequences = True
BiLSTM - Layer 2	40 units , return_sequences = True
Conv1D - Layer 1	8 filters, 1 kernel, ReLU
Conv1D - Layer 2	32 filters, 1 kernel, ReLU
Upper Branch	
BiLSTM - Layer 1	40 units , return_sequences = True
BiLSTM - Layer 2	40 units , return_sequences = True
Concatenation	
Binary Output Branch	
Dense - Layer 1	32 units, ReLU
Dropout Layer	0.2
Dense - Layer 2	16 units, ReLU
Dropout Layer	0.05
Dense - Layer 3	1, Linear
Regression Output Branch	
Dense - Layer 1	32 units, ReLU
Dropout Layer	0.2
Dense - Layer 2	16 units, ReLU
Dropout Layer	0.05
Dense - Layer 3	1, Linear

Table A.14: Joint Learning Model

A.4 Code

The code repository can be found in `https://github.com/blw219/FYP-Antibiotic` (Access Required)

Bibliography

[1] Melis N Anahtar, Jason H Yang, and Sanjat Kanjilal. Applications of machine learning to the problem of antimicrobial resistance: an emerging model for translational research. *Journal of clinical microbiology*, 59(7):e01260–20, 2021.

[2] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[3] Trang Thi Kieu Tran, Sayed M Bateni, Seo Jin Ki, and Hamidreza Vosoughifar. A review of neural networks for air temperature forecasting. *Water*, 13(9):1294, 2021.

[4] Facundo Bre, Juan M Gimenez, and Víctor D Fachinotti. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158:1429–1441, 2018.

[5] Siddharth Sankhe. Convolutional neural network. `https://siddharthsankhe.medium.com/convolutional-neural-network-dc942931bff8` [Accessed: (June 14, 2023)].

[6] Christopher Olah. Understanding lstm networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/` [Accessed: (June 14, 2023)].

[7] Yung-Hui Li, Latifa Nabila Harfiya, Kartika Purwandari, and Yue-Der Lin. Real-time cuffless continuous blood pressure estimation using deep learning model. *Sensors*, 20(19):5606, 2020.

[8] L. Thomas. Stratified sampling: Definition, guide amp; examples. `https://www.scribbr.com/methodology/stratified-sampling/` [Accessed: (May 17, 2023)].

[9] Dr. S. Rostami. Class imbalance and oversampling. https://datacrayon.com/machine-learning/class-imbalance-and-oversampling/ [Accessed: (May 17, 2023)].

[10] 3.1. cross-validation: Evaluating estimator performance. https://scikit-learn.org/stable/modules/cross_validation.html [Accessed: (May 17, 2023)].

[11] Arize AI. Binary cross entropy: Where to use log loss in model monitoring. https://arize.com/blog-course/binary-cross-entropy-log-loss/ [Accessed: (May 18, 2023)].

[12] S. Sharma. Activation functions in neural networks. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 [Accessed: (May 18, 2023)].

[13] R. Draelos. Measuring performance: Auc (auroc). https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/ [Accessed: (May 20, 2023)].

[14] Francesca Prestinaci, Patrizio Pezzotti, and Annalisa Pantosti. Antimicrobial resistance: a global multifaceted phenomenon. *Pathogens and global health*, 109(7):309–318, 2015.

[15] World Health Organization. Antimicrobial resistance. https://www.who.int/news-room/fact-sheets/detail/antimicrobial-resistance [Accessed: (May 31, 2023)].

[16] World Health Organization. Antimicrobial resistance. https://www.who.int/health-topics/antimicrobial-resistance [Accessed: (June 21, 2023)].

[17] Imperial College London. What is antimicrobial resistance? https://www.imperial.ac.uk/arc/about-us/what-is-amr/ [Accessed: (June 21, 2023)].

[18] Joy Marie Lim. Can antibiotic misuse put your health at risk? https://www.mountelizabeth.com.sg/health-plus/article/dangers-of-antibiotic-misuse [Accessed: (June 21, 2023)].

[19] European Medicines Agency. Antimicrobial resistance. https://www.ema.europa.eu/en/human-regulatory/overview/public-healththreats/antimicrobial-resistance [Accessed: (June 21, 2023)].

[20] J. o'neill, antimicrobial resistance: Tackling a crisis for the health and wealth of nations, review on antimicrobial resistance, 2014.

[21] Leonid Chindelevitch, Elita Jauneikaitea, Nicole E Wheeler, Kasim Allel, Bede Yaw Ansiri-Asafoakaa, Wireko A Awuah, Denis C Bauer, Stephan Beisken, Kara Fan, Gary Grant, et al. Applying data technologies to combat amr: current status, challenges, and opportunities on the way forward. *arXiv preprint arXiv:2208.04683*, 2022.

[22] Jan J De Waele, Jerina Boelens, and Isabel Leroux-Roels. Multidrug-resistant bacteria in icu: fact or myth. *Current Opinion in Anesthesiology*, 33(2):156–161, 2020.

[23] J. hsu, how covid-19 is accelerating the threat of antimicrobial resistance, bmj 369 (2020) m1983.

[24] Institute for health metrics and evaluation (ihme), university of oxford. global bacterial antimicrobial resistance burden estimates 2019. seattle, united states of america: Institute for health metrics and evaluation (ihme), 2022.

[25] Intensive care units as epicentres for antimicrobial resistance development report 2017, http://resistancecontrol.info/2017/intensive-care-units-as-epicentres-for-antimicrobial-resistance-development/.

[26] Jean-Louis Vincent, Yasser Sakr, Mervyn Singer, Ignacio Martin-Loeches, Flavia R Machado, John C Marshall, Simon Finfer, Paolo Pelosi, Luca Brazzi, Dita Aditianingsih, et al. Prevalence and outcomes of infection among patients in intensive care units in 2017. *Jama*, 323(15):1478–1487, 2020.

[27] Nele Brusselaers, Dirk Vogelaers, and Stijn Blot. The rising problem of antimicrobial resistance in the intensive care unit. *Annals of intensive care*, 1:1–7, 2011.

[28] Steven L Solomon and Kristen B Oliver. Antibiotic resistance threats in the united states: stepping back from the brink. *American family physician*, 89(12):938–941, 2014.

[29] Jason A Roberts, Sanjoy K Paul, Murat Akova, Matteo Bassetti, Jan J De Waele, George Dimopoulos, Kirsi-Maija Kaukonen, Despoina Koulenti, Claude Martin, Philippe Montravers, et al. Dali: defining antibiotic levels in intensive care unit patients: are current β-lactam antibiotic doses sufficient for critically ill patients? *Clinical infectious diseases*, 58(8):1072–1083, 2014.

[30] DC Bergmans, MJ Bonten, CA Gaillard, FH Van Tiel, S Van Der Geest, PW De Leeuw, and EE Stobberingh. Indications for antibiotic use in icu patients: a one-year prospective surveillance. *The Journal of antimicrobial chemotherapy*, 39(4):527–535, 1997.

[31] Charles-Edouard Luyt, Nicolas Bréchot, Jean-Louis Trouillet, and Jean Chastre. Antibiotic stewardship in the intensive care unit. *Critical care*, 18(5):1–12, 2014.

[32] Marin H Kollef and Victoria J Fraser. Antibiotic resistance in the intensive care unit. *Annals of internal medicine*, 134(4):298–314, 2001.

[33] Marin H Kollef. Optimizing antibiotic therapy in the intensive care unit setting. *Critical care*, 5:1–7, 2001.

[34] Lakhbar Ines, Duclos Gary, and Leone Marc. Does antimicrobial resistance affect clinical outcomes in the icu? *ICU Management Practice*, 22(4), 2022.

[35] Jean-Louis Vincent, David J Bihari, Peter M Suter, Hajo A Bruining, Jane White, Marie-Helene Nicolas-Chanoin, Michel Wolff, Robert C Spencer, and Margaret Hemmer. The prevalence of nosocomial infection in intensive care units in europe: results of the european prevalence of infection in intensive care (epic) study. *Jama*, 274(8):639–644, 1995.

[36] Canadian Institute for Health Information. Care in canadian icus, 2016.

[37] OJ Dyar, B Huttner, J Schouten, C Pulcini, et al. What is antimicrobial stewardship? *Clinical microbiology and infection*, 23(11):793–798, 2017.

[38] Mical Paul, Steen Andreassen, Evelina Tacconelli, Anders D Nielsen, Nadja Almanasreh, Uwe Frank, Roberto Cauda, Leonard Leibovici, and TREAT Study Group. Improving empirical antibiotic treatment using treat, a computerized decision support system: cluster randomized trial. *Journal of Antimicrobial Chemotherapy*, 58(6):1238–1245, 2006.

[39] Leonard Leibovici, Galia Kariv, and Mical Paul. Long-term survival in patients included in a randomized controlled trial of treat, a decision support system for antibiotic treatment. *Journal of Antimicrobial Chemotherapy*, 68(11):2664–2666, 2013.

[40] William J. Bolton, Timothy M. Rawson, Bernard Hernandez, Richard Wilson, David Antcliffe, Pantelis Georgiou, and Alison H. Holmes. Machine learning and synthetic

outcome estimation for individualised antimicrobial cessation. *Frontiers in Digital Health*, 4, 2022.

[41] Pranita D Tamma, Melissa A Miller, and Sara E Cosgrove. Rethinking how antibiotics are prescribed: incorporating the 4 moments of antibiotic decision making into clinical practice. *Jama*, 321(2):139–140, 2019.

[42] Bradley J Langford and Andrew M Morris. Is it time to stop counselling patients to "finish the course of antibiotics"? *Canadian Pharmacists Journal: CPJ*, 150(6):349, 2017.

[43] Alison H Holmes, Luke SP Moore, Arnfinn Sundsfjord, Martin Steinbakk, Sadie Regmi, Abhilasha Karkey, Philippe J Guerin, and Laura JV Piddock. Understanding the mechanisms and drivers of antimicrobial resistance. *The Lancet*, 387(10014):176–187, 2016.

[44] Brad Spellberg. The new antibiotic mantra—"shorter is better". *JAMA internal medicine*, 176(9):1254–1255, 2016.

[45] Robin ME Janssen, Anke JM Oerlemans, Johannes G Van Der Hoeven, Jaap Ten Oever, Jeroen A Schouten, and Marlies EJL Hulscher. Why we prescribe antibiotics for too long in the hospital setting: a systematic scoping review. *Journal of Antimicrobial Chemotherapy*, 2022.

[46] Juan C Rojas, Kyle A Carey, Dana P Edelson, Laura R Venable, Michael D Howell, and Matthew M Churpek. Predicting intensive care unit readmission with machine learning using electronic health record data. *Annals of the American Thoracic Society*, 15(7):846–853, 2018.

[47] Yinan Huang, Ashna Talwar, Satabdi Chatterjee, and Rajender R Aparasu. Application of machine learning in predicting hospital readmissions: a scoping review of the literature. *BMC medical research methodology*, 21(1):1–14, 2021.

[48] Yu-Wei Lin, Yuqian Zhou, Faraz Faghri, Michael J Shaw, and Roy H Campbell. Analysis and prediction of unplanned intensive care unit readmission using recurrent neural networks with long short-term memory. *PloS one*, 14(7):e0218942, 2019.

[49] Timothy M Rawson, Bernard Hernandez, Richard C Wilson, Damien Ming, Pau Herrero, Nisha Ranganathan, Keira Skolimowska, Mark Gilchrist, Giovanni Satta, Pan-

telis Georgiou, et al. Supervised machine learning to support the diagnosis of bacterial infection in the context of covid-19. *JAC-antimicrobial resistance*, 3(1):dlab002, 2021.

[50] Bernard Hernandez, Pau Herrero, Timothy Miles Rawson, Luke SP Moore, Benjamin Evans, Christofer Toumazou, Alison H Holmes, and Pantelis Georgiou. Supervised learning for infection risk inference using pathology data. *BMC medical informatics and decision making*, 17(1):1–12, 2017.

[51] TM Rawson, B Hernandez, LSP Moore, O Blandy, P Herrero, M Gilchrist, A Gordon, C Toumazou, S Sriskandan, P Georgiou, et al. Supervised machine learning for the prediction of infection on admission to hospital: a prospective observational cohort study. *Journal of Antimicrobial Chemotherapy*, 74(4):1108–1115, 2019.

[52] Bernard Hernandez, Pau Herrero-Viñas, Timothy M Rawson, Luke SP Moore, Alison H Holmes, and Pantelis Georgiou. Resistance trend estimation using regression analysis to enhance antimicrobial surveillance: A multi-centre study in london 2009–2016. *Antibiotics*, 10(10):1267, 2021.

[53] Bernard Hernandez, Pau Herrero, Timothy M Rawson, Luke SP Moore, Esmita Charani, Alison H Holmes, and Pantelis Georgiou. Data-driven web-based intelligent decision support system for infection management at point-of-care: Case-based reasoning benefits and limitations. In *HEALTHINF*, pages 119–127, 2017.

[54] Parvez Rafi, Arash Pakbin, and Shiva Kumar Pentyala. Interpretable deep learning framework for predicting all-cause 30-day icu readmissions. *Texas A&M University*, 2018.

[55] Melina Loreto, Thiago Lisboa, and Viviane P. Moreira. Early prediction of icu readmissions using classification algorithms. *Computers in Biology and Medicine*, 118:103636, 2020.

[56] Michael Ko, Emma Chen, Ashwin Agrawal, Pranav Rajpurkar, Anand Avati, Andrew Ng, Sanjay Basu, and Nigam H. Shah. Improving hospital readmission prediction using individualized utility analysis. *Journal of Biomedical Informatics*, 119:103826, 2021.

[57] Panagiotis Michailidis, Athanasia Dimitriadou, Theophilos Papadimitriou, and Periklis Gogas. Forecasting hospital readmissions with machine learning. In *Healthcare*, volume 10, page 981. MDPI, 2022.

[58] Iyad Batal, Dmitriy Fradkin, James Harrison, Fabian Moerchen, and Milos Hauskrecht. Mining recent temporal patterns for event detection in multivariate time series data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–288, 2012.

[59] Alexander Shknevsky, Yuval Shahar, and Robert Moskovitch. Consistent discovery of frequent interval-based temporal patterns in chronic patients' data. *Journal of biomedical informatics*, 75:83–95, 2017.

[60] Mohammad Amin Morid, Olivia R Liu Sheng, Kensaku Kawamoto, and Samir Abdelrahman. Learning hidden patterns from patient multivariate time series data using convolutional neural networks: A case study of healthcare cost prediction. *Journal of Biomedical Informatics*, 111:103565, 2020.

[61] Robert Moskovitch and Yuval Shahar. Fast time intervals mining using the transitivity of temporal relations. *Knowledge and Information Systems*, 42(1):21–48, 2015.

[62] Bulgarelli L. Pollard T. Horng S. Celi L. A. Mark R. (2021) Johnson, A. Mimic-iv (version 1.0). *PhysioNet. https://doi.org/10.13026/s6n6-xd98.*

[63] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

[64] Erida Gjini, Francisco FS Paupério, and Vitaly V Ganusov. Treatment timing shifts the benefits of short and long antibiotic treatment over infection. *Evolution, Medicine, and Public Health*, 2020(1):249–263, 2020.

[65] Luís Coelho, Pedro Póvoa, Eduardo Almeida, Antero Fernandes, Rui Mealha, Pedro Moreira, and Henrique Sabino. Usefulness of c-reactive protein in monitoring the severe community-acquired pneumonia clinical course. *Critical care*, 11(4):1–9, 2007.

[66] Bas Theodoor Straathof. A deep learning approach to predicting the length of stay of newborns in the neonatal intensive care unit, 2020.

[67] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[68] R. Draelos. Measuring performance: Auprc and average precision. https://glassboxmedicine.com/2019/03/02/measuring-performance-auprc/ [Accessed: (May 31, 2023)].

[69] Thomas Wood. F-score, deepai. https://deepai.org/machine-learning-glossary-and-terms/f-score [Accessed: (June 11, 2023)].

[70] Mean absolute error (mae) and root mean squared error (rmse). https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm [Accessed: (June 11, 2023)].

[71] Anthony W Chow, Michael S Benninger, Itzhak Brook, Jan L Brozek, Ellie JC Goldstein, Lauri A Hicks, George A Pankey, Mitchel Seleznick, Gregory Volturo, Ellen R Wald, et al. Idsa clinical practice guideline for acute bacterial rhinosinusitis in children and adults. *Clinical infectious diseases*, 54(8):e72–e112, 2012.

[72] Nadim G El Chakhtoura, Robert A Bonomo, and Robin LP Jump. Influence of aging and environment on presentation of infection in older adults. *Infectious Disease Clinics*, 31(4):593–608, 2017.

[73] Cleveland Clinic. Arterial blood gas (abg): What it is, purpose, procedure levels. https://my.clevelandclinic.org/health/diagnostics/22409-arterial-blood-gas-abg [Accessed: (June 11, 2023)].

[74] Gregory Mak, William D Grant, James C McKenzie, and John B McCabe. Physicians' ability to predict hospital length of stay for patients admitted to the hospital from the emergency department. *Emergency medicine international*, 2012, 2012.