

國立臺灣大學共同教育中心統計碩士學位學程



碩士論文

Master Program in Statistics

Center for General Education

National Taiwan University

Master Thesis

應用多模態機器學習於急性呼吸窘迫症候群之預測

Applying Multimodal Machine Learning for
Early Prediction of Acute Respiratory Distress Syndrome

郭庭嘉

Ting-Chia Kuo

指導教授: 周呈雲 博士

Advisor: Cheng-Ying Chou, Ph.D.

中華民國 111 年 7 月

July, 2022



摘要

急性呼吸窘迫症候群是病人進入加護病房的常見原因之一，並且有很高的死亡率，雖然現今已經有很多研究運用臨床資料和機器學習方法探討及時診斷與提前預測的模型，但幾乎沒有研究同時考慮了數值資料及影像資料。本研究使用了公開資料集 (MIMIC-IV 以及 MIMIC-CXR) 以獲取病人的臨床資料及胸部 X 光片影像資料，應用機器學習方法建立決策樹 (Decision Tree)、隨機森林 (Random Forest)、極限梯度提升 (XGBoost)、神經網路 (Neural Network) 等多種模型，並應用了多模態機器學習分析，比較單模態與多模態模型的表現。使用晚期融合的多模態模型在診斷及 12 小時、24 小時及 48 小時前的預測，接受者操作特徵曲線下面積 (AUROC) 約為 0.7951 至 0.8502，與單模態模型相比約可以提高 6.0% 至 9.3% 的模型表現，這個研究將可以協助急性呼吸窘迫症候群的診斷及早期預測。

關鍵字：急性呼吸窘迫症候群、多模態機器學習、胸部 X 光片、極限梯度提升、晚期融合





Abstract

Acute respiratory distress syndrome (ARDS) is one of the most common causes of admission to the intensive care unit and has a high mortality rate. Although there were several studies applying machine learning techniques to the issue of ARDS prediction, few studies combined numerical and image data. This study collected clinical data and chest radiograph images from publicly available databases (MIMIC-IV and MIMIC-CXR) and applied machine learning methods to establish models such as Decision Tree, Random Forest, XGBoost, and Neural Networks. Moreover, multimodal machine learning were applied and the performance of single- and multi-modality models were compared. The multi-modality models with late-level fusion demonstrated the AUROC of 0.7951~0.8502 for onset identification, 12-, 24-, and 48-hr prediction, which improved about 6.0%~9.3% compared with the single-modality models. This study can assist improved prediction and early recognition of ARDS.

Keywords: Acute Respiratory Distress Syndrome, Multimodal Machine Learning, Chest Radiograph, eXtreme Gradient Boosting, Late-level Fusion





Contents

	Page
摘要	i
Abstract	iii
Contents	v
List of Figures	ix
List of Tables	xi
Denotation	xiii
Chapter 1 Introduction	1
1.1 Background	1
1.2 Purpose of Research	3
1.3 Thesis Organization	4
Chapter 2 Literature Reviews	7
2.1 ARDS Prediction	7
2.2 Multimodal Deep Learning	11
Chapter 3 Materials	15
3.1 Data Source	15
3.1.1 MIMIC-IV	15
3.1.2 MIMIC-CXR	17



3.2	Data Processing	18
3.2.1	Data extraction	18
3.2.2	Feature selection	19
3.2.3	Patient selection	22
3.3	Data Analysis	24
Chapter 4	Methods	29
4.1	Data Standardization	29
4.2	Data Augmentation	30
4.3	Imbalanced Data	31
4.4	Models	33
4.4.1	Decision tree	33
4.4.2	Random forest	34
4.4.3	Extreme gradient boosting	35
4.4.4	Convolutional neural network	36
4.5	Fusion Strategies of Multi-modality	39
Chapter 5	Model Evaluation	43
5.1	Evaluation Metrics	43
5.2	Stratified Cross-Validation	45
Chapter 6	Results	49
6.1	Single-modality Models	49
6.2	Multi-modality Models	54
6.3	Model Comparison	55

Chapter 7 Discussion and Conclusion	59
7.1 Discussion	59
7.2 Limitations	60
7.3 Conclusion	61
References	63
Appendix A — Cross-validation Results	71



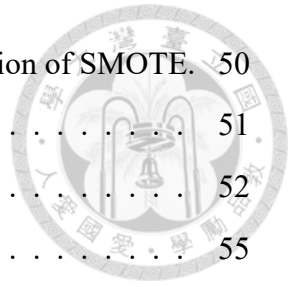




List of Figures

1.1	The framework of this study.	5
2.1	Fusion strategies using deep learning [1].	13
3.1	The structure of the MIMIC-IV database. The required data files for this study are marked in red.	16
3.2	The structure of the MIMIC-CXR database. The required data files for this study are marked in red.	17
3.3	The graphical description of data selection.	21
3.4	The flowchart of patient selection.	23
3.5	The histograms of 18 clinical features.	25
3.6	The histograms of 18 clinical features by different groups.	26
3.7	The KDE plots of 18 clinical features.	27
4.1	The examples of the CXR image augmentation.	31
4.2	The structure of a decision tree.	34
4.3	The main concept of the random forest algorithm.	35
4.4	The schematic diagram of ANN with 2 hidden layers	37
4.5	The architecture of a 5-layer dense block [2].	38
4.6	The architecture of DenseNet with different layers [2].	39
4.7	The structure of the multi-modality models.	40
5.1	The graph of the ROC curve with false positive rate on the x -axis and true positive rate on the y -axis.	44
5.2	The implementation of 5-fold cross-validation.	46

6.1	The patient number in the training set with the implementation of SMOTE.	50
6.2	ROC curves of numerical-only models.	51
6.3	ROC curves of image-only models.	52
6.4	ROC curves of single- and multi-modality models.	55
6.5	AUROC of the different models with cross-validation.	57





List of Tables

2.1	Overview of ARDS-related studies with machine learning techniques. . .	10
3.1	The introduction of the required data files in the MIMIC-IV database. . .	16
3.2	The introduction of the required data files in the MIMIC-CXR database. .	17
3.3	The corresponding identity numbers of the needed features for labeling in the MIMIC-IV database.	19
3.4	The corresponding identity numbers of the 19 selected features in the MIMIC-IV database.	20
3.5	Description of the rules for data selection.	21
3.6	The number of patients included in this study.	23
3.7	Demographic characteristics of subjects included in this study.	24
5.1	The confusion matrix for binary classification.	43
5.2	The proportion of the positive cases in different sets after the stratified 5-fold cross-validation.	47
6.1	Performance of the numerical-only and image-only models.	53
6.2	Performance of the multi-modality models.	54
A.1	Results of the repeated k -fold cross-validation for onset identification. . .	71
A.2	Results of the repeated k -fold cross-validation for 12-hr prediction. . . .	72
A.3	Results of the repeated k -fold cross-validation for 24-hr prediction. . . .	73
A.4	Results of the repeated k -fold cross-validation for 48-hr prediction. . . .	74





Denotation

ARDS	急性呼吸窘迫症候群 (Acute Respiratory Distress Syndrome)
CNN	卷積神經網絡 (Convolutional Neural Network)
CXR	胸部 X 光片 (Chest Radiograph)
DenseNet	密集連接卷積網絡 (Dense Convolutional Network)
DICOM	醫療數位影像傳輸協定 (Digital Imaging and Communications in Medicine)
DT	決策樹 (Decision Tree)
EHR	電子健康紀錄 (Electronic Health Record)
ICD	國際疾病分類 (International Classification of Diseases)
ICU	加護病房 (Intensive Care Unit)
ResNet	深度殘差網路 (Deep Residual Network)
RF	隨機森林 (Random Forest)

ROC 接受者操作特徵 (Receiver Operating Characteristic)

SMOTE 合成少數過採樣技術 (Synthesized Minority Oversampling Technique)

XGBoost 極限梯度提升 (eXtreme Gradient Boosting)





Chapter 1 Introduction

1.1 Background

Intensive care units (ICUs) are specialist wards of hospitals that can provide intensive treatment and close monitoring for critically ill patients [3, 4]. They are also called critical care units (CCUs) or intensive therapy units (ITUs). ICUs in today's healthcare system are extremely important since they can provide critical care and life support. However, patients in the ICU have higher mortality due to their precariousness.

This thesis concentrated on the issues related to acute respiratory distress syndrome (ARDS), which is one of the most common causes of admission to the ICU. A recent study showed that ARDS has approximately 10% prevalence in ICU and a high mortality rate of about 30~40% [5, 6]. ARDS is a life-threatening lung disease in which the lungs become severely inflamed from infection or injury. Breathing will become increasingly difficult once the lungs cannot provide enough oxygen. Several clinical disorders can cause ARDS, including pneumonia, sepsis, aspiration of gastric contents, and major trauma. Some scenarios are also associated with ARDS development, such as acute pancreatitis, near drowning, and smoke inhalation.

The most acceptable definition of ARDS is the Berlin definition, which was introduced in 2012. An international expert panel convened by the European Society of Intensive Care Medicine, the American Thoracic Society, and the Society of Critical Care Medicine revised the ARDS definition [7, 8]. Berlin definition mainly consists of 4 parts:

- **Timing:**

- Within 1 week of a known clinical insult or new or worsening respiratory symptoms

- **Chest imaging:**

- Bilateral opacities —not fully explained by effusions, lobar/lung collapse, or nodules

- **Origin of edema:**

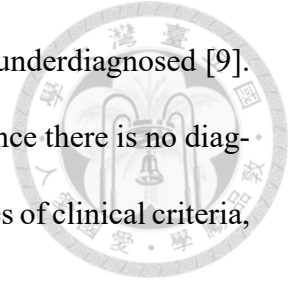
- Respiratory failure not fully explained by cardiac failure or fluid overload
- Need objective assessment (e.g., echocardiography) to exclude hydrostatic edema if no risk factors present

- **Oxygenation:**

- Mild: $200 \text{ mmHg} < \text{PaO}_2/\text{FiO}_2 \leq 300 \text{ mmHg}$ with PEEP or CPAP $\geq 5 \text{ cmH}_2\text{O}$
- Moderate: $100 \text{ mmHg} < \text{PaO}_2/\text{FiO}_2 \leq 200 \text{ mmHg}$ with PEEP $\geq 5 \text{ cmH}_2\text{O}$
- Severe: $\text{PaO}_2/\text{FiO}_2 \leq 100 \text{ mmHg}$ with PEEP $\geq 5 \text{ cmH}_2\text{O}$

According to the Berlin definition, the diagnosis of ARDS is based on clinical features and chest imaging. Early recognition of ARDS is important since effective therapies will result in different outcomes. An accurate diagnosis may improve treatment and reduce

mortality. However, a recent study showed that ARDS is frequently underdiagnosed [9]. It is a complex and challenging task for identifying ARDS patients since there is no diagnostic test for ARDS. Patients diagnosed with ARDS must meet a series of clinical criteria, which involve numerical and radiological features [10].



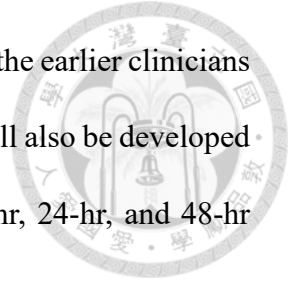
With the development of artificial intelligence (AI), AI-based technologies such as machine learning and deep learning have been widely applied to different fields such as finance, marketing, transportation, and national security. Moreover, AI techniques are also applied in the medical field and accepted by the public [11, 12]. Nowadays, people apply machine learning techniques and tools to solve medical diagnostic and prognostic problems, such as the prediction of disease progression and overall patient management.

1.2 Purpose of Research

Under the conception of AI, this study aimed to find an appropriate method to help the clinician identify the ARDS patients by using machine learning techniques. The goal of the research is to diagnose ARDS early and accurately. With an accurate diagnosis, the probability of missed or delayed diagnosis of ARDS can be reduced. Furthermore, an early prediction may improve treatment and reduce mortality.

In this thesis, ICU data were used for modeling and analysis. ICU data are large-scale and are essential for clinical data analysis as ICU generates thousands of data points per day. The ICU data were considered as two categories: image and numerical data. Image data are CXR images and numerical data include patient demographics, vital signs, and routinely collected measurements. This thesis applied machine learning techniques and developed models with different modality data and frameworks to find a proper method

for timely diagnosis. The earlier the ARDS patients can be identified, the earlier clinicians can respond and make the decision of treatment. Predictive models will also be developed for the early prediction of ARDS and the predict time includes 12-hr, 24-hr, and 48-hr prior to the onset.



This thesis is the first study to develop multi-modality models for identifying and predicting ARDS with numerical and image data. Patients included in this study were collected from the open databases and were required to have both image and numerical data since the main purpose of this study is to develop multi-modality models. The requirement of having both image and numerical data makes the patients in this study be a smaller subset. Moreover, patients with both image and numerical data are more likely to be potential ARDS patients with high suspicion by clinicians, and making the subset more difficult to be distinguished between diseased and non-diseased.

The performance of different models were compared. The algorithm developed in this study may improve the early prediction of ARDS and provide assistance for clinicians. The structure of multi-modality models developed in this study may also be applied to other issues.

1.3 Thesis Organization

The framework of this study is shown in Figure 1.1. This research can divide into 5 parts, which will be explained in the following chapters. First, Chapter 1 is the introduction to this study. Next, Chapter 2 reviews the present researches on the issue related to ARDS prediction and the current methods for analyzing multi-modality data in the medical field.

Chapter 3 is the introduction of the data sources of this thesis. The databases used

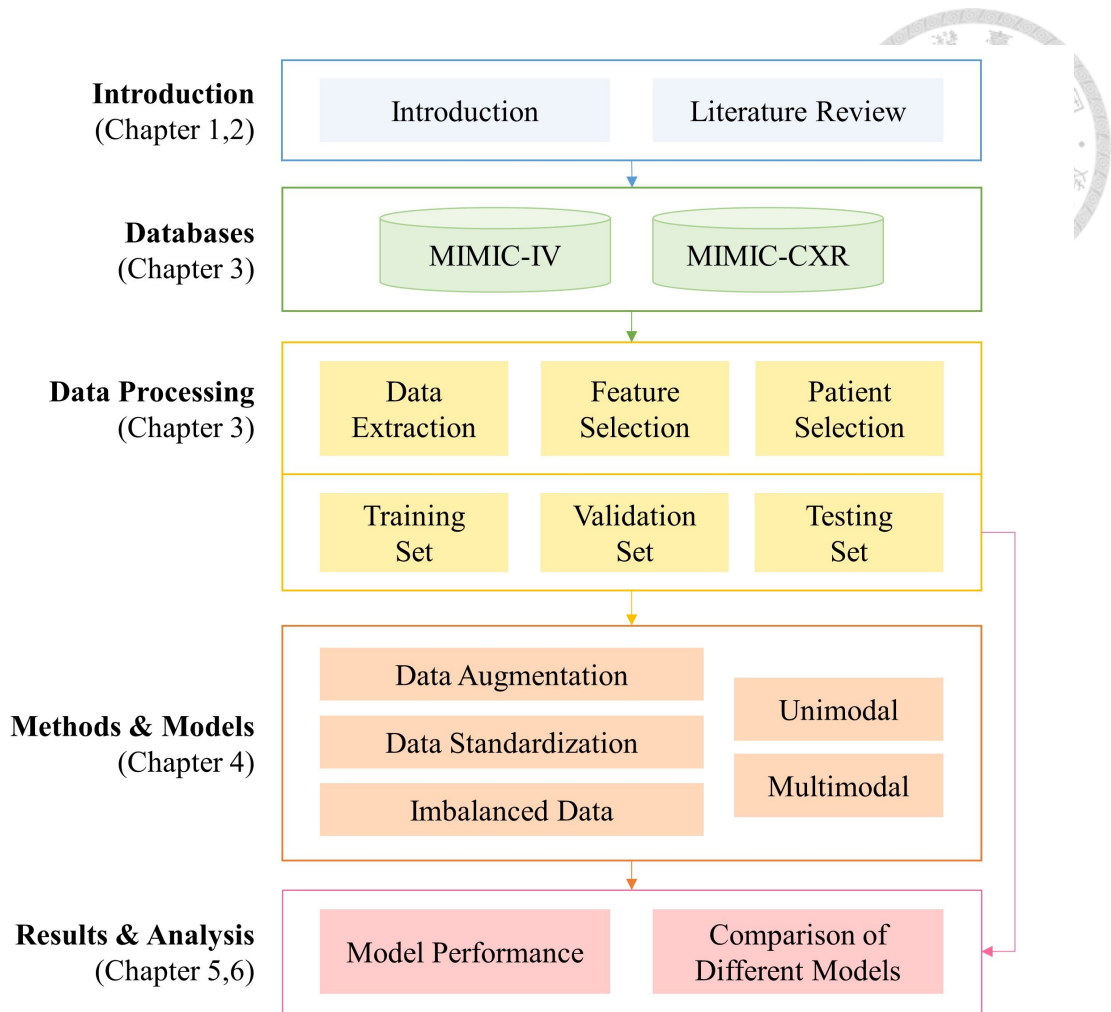


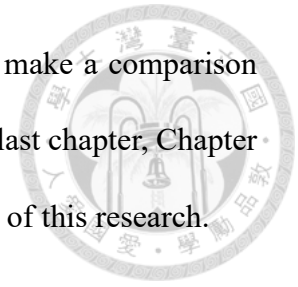
Figure 1.1: The framework of this study.

in this research contain extensive and complete information about patients admitted to the ICU; moreover, they are de-identified and freely available. The methods of data processing will also be explained in Chapter 3, which includes steps to extract data and criteria of patient selection.

The machine learning algorithm used in this thesis will be introduced in Chapter 4, including decision tree (DT), random forest (RF), eXtreme gradient boosting (XGBoost), and convolutional neural network (CNN). Different model structures designed in this thesis, including single-modality and multi-modality models, will also be explained.

Chapter 5 introduce the method of model evaluation. Chapter 6 presents the per-

formance of models and the results of this study. Moreover, I will make a comparison between the different models applied and select the best model. The last chapter, Chapter 7, will conclude this thesis and discuss the limitation and application of this research.





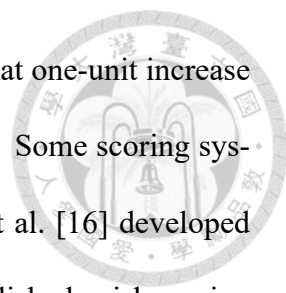
Chapter 2 Literature Reviews

This chapter will introduce previous studies on predicting ARDS and applying multimodal deep learning in medicine. It is helpful for the researcher to understand previous studies. The results of these studies could be taken as references and gave directions for continuing research on related topics.

2.1 ARDS Prediction

Studies related to ARDS have been researched for several years due to the high incidence and mortality. Since patients were commonly assumed to benefit from early diagnosis and intervention, there were many studies researched on the timely diagnosis or early prediction of ARDS.

The lung injury prediction score (LIPS) is a numerical index derived to accurately estimate the probability of developing acute lung injury (ALI) or ARDS, 14 predictors such as high-risk trauma, high-risk surgery, alcohol abuse, and smoking history were included in this index [13]. Soto et al. [14] performed a competing risk Cox regression analysis to analyze the association of LIPS and ARDS development. In the study, LIPS helped identify patients at risk of ARDS or dying during hospitalization. Bauman et al. [15] calculated LIPS for patients in the surgical critical care unit (SICU) and used logistic regression



model for predicting the development of ARDS. His study showed that one-unit increase in LIPS, the odds of developing ARDS increased by a factor of 1.50. Some scoring systems were also designed for the task of early identification. Pepe et al. [16] developed several equations for calculating ARDS scores. Liang et al. [17] established a risk scoring system for predicting ARDS among hospitalized patients with coronavirus disease, which used univariate logistic regression model and a multivariate logistic regression model.

Besides, there were some researches, which applied machine learning techniques in recent years [18–25]. Machine learning algorithms are successfully developed as a common and popular tool for classification issues. Nowadays, the technique of machine learning has been widely applied to many issues of different fields, such as creditworthiness of customers, speech recognition, product recommendation, and medical diagnosis.

Yang et al. [18] applied L2 regularized logistic regression, artificial neural network (ANN), adaptive boosting (AdaBoost), and XGBoost for estimating $\text{PaO}_2/\text{FiO}_2$ ratio, which can aid the diagnosis of ARDS. Features were noninvasive physiological parameters of patients and XGBoost algorithm had the best results.

Sidney et al. [19] applied an XGBoost gradient boosted tree model for early ARDS prediction. Features were extracted from the EHR and radiology reports and predicted the ARDS labels at 12-, 24-, and 48-hour windows prior to onset.

Ding et al. [20] applied a random forest model to identify ARDS with baseline characteristics, clinical features, and laboratory features on the first day of admission. The predictive model included the following 11 predictors: minimum and maximum respiratory rate, minimum and maximum heart rate, minimum systolic blood pressure, MAP, temperature, WBC count, glucose levels, haematocrit, and sodium.

Zeiberg et al. [21] developed a risk stratification model for ARDS through L2-regularized logistic regression and XGBoost on EHR data. The minimum, maximum, mean, median, standard deviation, and interquartile range were calculated for each feature and then mapped to binary variables.

Fei et al. [22] applied an ANN model with a total of 13 input variables to predict the risk and severity of ARDS following severe acute pancreatitis, the severity of ARDS for mild, moderate, and severe was according to the Berlin definition.

Singhal et al. [23] presented a machine learning algorithm called eARDS to predict COVID19 patients who developed ARDS by XGBoost algorithm and multi-center validation. In the study, they evaluated a number of machine learning methods including neural networks, support vector machines, random forests, logistic regression, and XGBoost. They selected the XGBoost model since its superior performance.

The 6 studies mentioned above only used EHR data for researching, which were all numerical data. There were also studies only used image data for researching. The following 2 studies used CXR images and performed machine learning or deep learning analysis.

Reamaroon et al. [24] applied multiple machine learning models including random forest, AdaBoost, random under-sampling boosting (RUSBoost), robust boost, and total boost to propose an automatic detection model. Materials in their study were CXR images, but the input features of models were numerical data. Features were extracted from CXR, such as statistics calculated from the CXR histogram and the gray-level co-occurrence matrix (GLCM).

Sjoding et al. [25] used CNN to detect ARDS. The model was firstly trained to de-

tect 14 common descriptive chest radiograph findings and then used transfer learning to detect ARDS. The internal test sets were designed to be reviewed by additional physicians. Their model showed similar performance compared with physicians and achieved an expert physician-level performance.

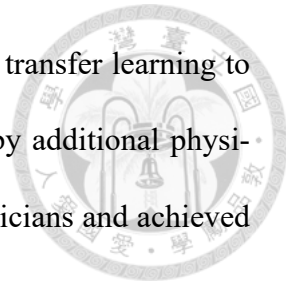
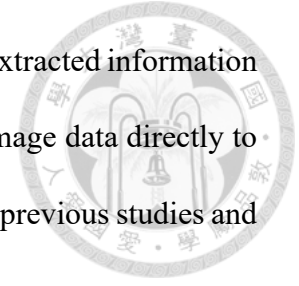


Table 2.1: Overview of ARDS-related studies with machine learning techniques.

Author	Timing	Data	Feature type	Method	AUC
Yang et al. (2020) [18]	0	MIMIC-III	Numerical	XGBoost	0.913 (predict P/F ratio)
Sidney et al. (2020) [19]	0	MIMIC-III	Numerical, radiology reports	XGBoost	0.905
	12-hr				0.827
	24-hr				0.810
	48-hr				0.790
Ding et al. (2019) [20]	0	Five ICUs in the Beijing metropolitan area	Numerical	Random forest	0.870
Zeiberg et al. (2019) [21]	0	large tertiary care center	Numerical	XGBoost	0.810
Fei et al. (2019) [22]	0	Surgical ICU of Nanjing Hospital	Numerical	ANN	0.859
Singhal et al. (2021) [23]	12-hr	the Cerner Health Facts Deidentified Database	Numerical	XGBoost	0.890
Reamaroon et al. (2021) [24]	0	ICUs at Michigan Medicine	Image (radiomics)	AdaBoost	0.830
Sjoding et al. (2021) [25]	0	CheXpert, MIMIC-CXR, hospital at University of Michigan, University of Pennsylvania	Image	DenseNet121, transfer learning	0.920

Almost all of the studies mentioned above used only numerical data or only image

data from patients. While few of them combined numerical data and extracted information from radiology reports, nearly no studies combined numerical and image data directly to help diagnose ARDS. This thesis considered the extracted features of previous studies and attempted to combine the advantages of numerical and image data.



2.2 Multimodal Deep Learning

Multimodal deep learning has become increasingly popular since deep learning has been reported significant success in many fields nowadays. The meaning of multimodal deep learning is to create models that can process and link information from multiple types of modalities, including image, video, text, audio, body gestures, facial expressions, and physiological signals. Multimodal deep learning has been successfully applied in many fields, such as autonomous driving, video classification. The application to medicine is also successful in many researches [26, 27].

Tiulpin et al. [28] proposed a novel method based on machine learning that used radiographic data, physical examination, patient' s medical history, and anthropometric data to predict the progression of knee osteoarthritis. They developed CNN models to directly leverage raw knee Digital Imaging and Communications in Medicine (DICOM) images and used a gradient boosting machine classifier with fused features.

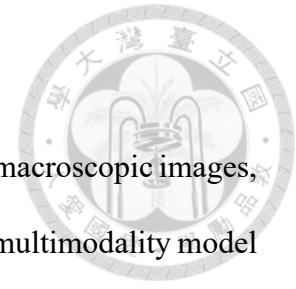
Huang et al. [29] built multimodal fusion models that combine information from both computerized tomography (CT) scans and EMR to automatically detect pulmonary embolism cases. They trained an imaging-only model, EMR-only neural network model, and 7 different fusion architecture models and compared performance. In the research, the late-fusion model was the best performing model and outperformed the single modality

model.

Yap et al. [30] combined multiple modalities together, including macroscopic images, dermatoscopic images, and metadata. In their work, they applied the multimodality model for the classification of skin lesions. Features of the image were separately extracted by deep residual network (ResNet) model, and then concatenated the extracted features and metadata, after that, sent them to the other neural network to solve the binary classification task.

Yala et al. [31] developed a breast cancer risk model by mammography-based deep learning model which outperformed the established clinical breast cancer risk models. The model was trained to predict whether the breast would develop breast cancer in 5 years by using full-field mammogram, the X-ray picture of the breast, and risk factor information such as age, weight, height, menarche age, menopausal status, etc.

Above are the applications of multimodal deep learning, all of them proved an improvement on the traditional single modality models. Additionally, Huang et al. [1] made a review of different techniques to combine medical imaging with EHR in 2020 and illustrated the three main different data fusion strategies, including early fusion, joint fusion, and late fusion (Figure 2.1). In this study, the multimodal analysis would be applied and the methods of early- and late-level fusion would be implemented.



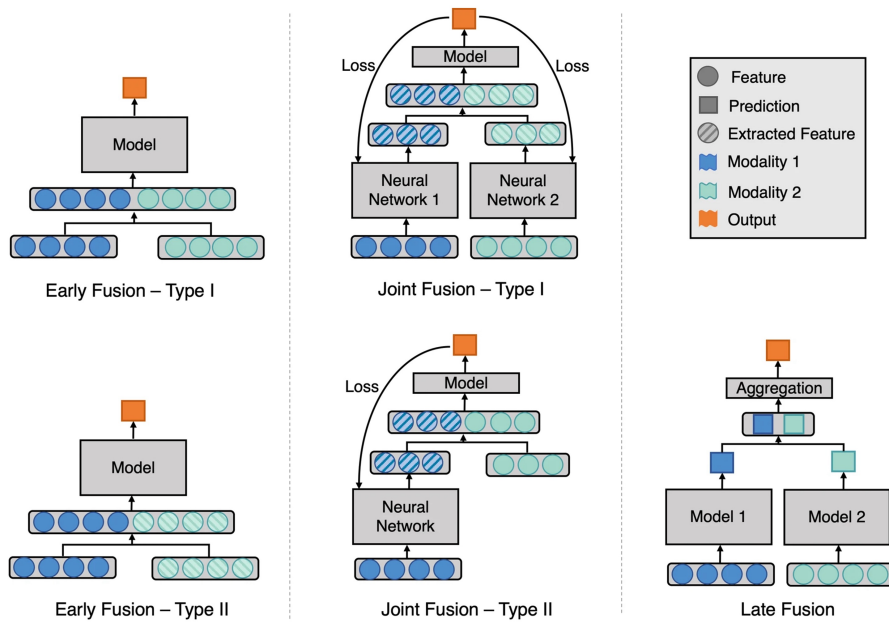


Figure 2.1: Fusion strategies using deep learning [1].





Chapter 3 Materials

In this Chapter, I will describe our data in detail. Section 3.1 is the introduction of the publicly available databases used in this research. Section 3.2 will describe the data processing methods; last, Section 3.3 is exploratory data analysis and visualization.

3.1 Data Source

Data considered in the study were obtained from The Medical Information Mart for Intensive Care (MIMIC)-IV [32] and MIMIC Chest X-ray (MIMIC-CXR) [33] databases. Both of them provide critical care data for patients admitted to the ICU at Beth Israel Deaconess Medical Center (BIDMC). These 2 databases are publicly available and de-identified according to the Health Insurance Portability and Accountability Act (HIPAA) Safe Harbor provision [34].

3.1.1 MIMIC-IV

The MIMIC-IV database contained patients' medical records between 2008 - 2019 and consisted of 27 tables in CSV format [32]. Data of the MIMIC-IV database include the information of patient's demographics, admission date, laboratory measurements, medication prescription, intravenous fluid inputs during the hospital stay, transfer, and discharge

information; besides, tables are grouped into 3 modules: core, hosp, and icu. These modules aim to highlight their intended use. The structure of the MIMIC-IV database is shown in Figure 3.1 and I mainly used about 10 tables of the database with the latest version (v1.0) for this work. Table 3.1 introduces the contents of 10 required data files.

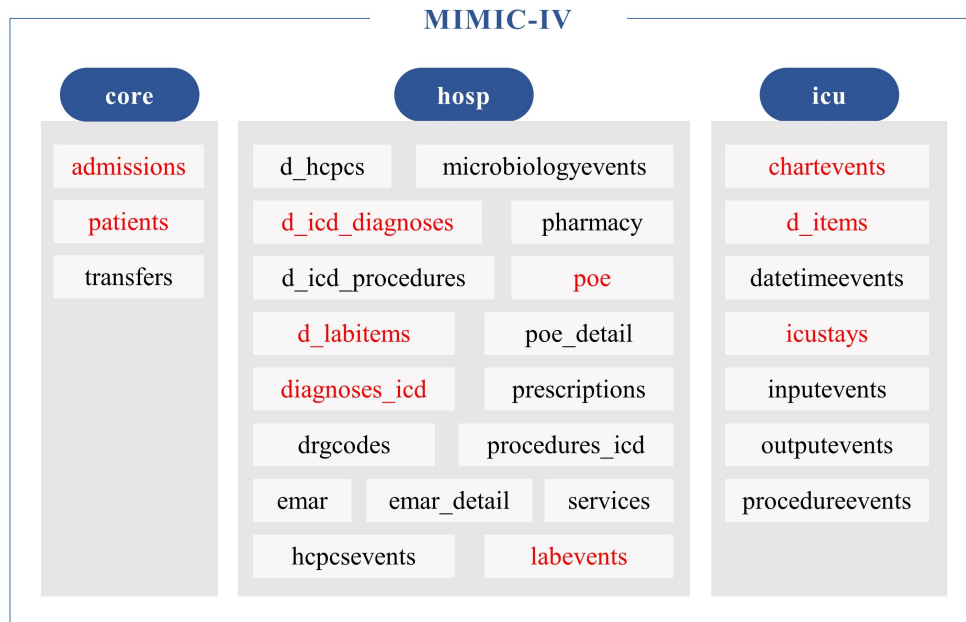
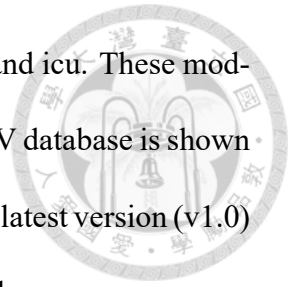


Figure 3.1: The structure of the MIMIC-IV database. The required data files for this study are marked in red.

Table 3.1: The introduction of the required data files in the MIMIC-IV database.

File	Information
admissions	About the admission, discharge, and transfer information of patients.
patients	About the patient's gender, age, and date of death. The information was all shifted and de-identified.
d_icd_diagnoses	The dictionary of the ICD-9 and ICD-10 codes.
d_labitems	The definition of lab measurements.
diagnoses_icd	About the ICD-9 and ICD-10 codes of patients.
poe	The information of provider order entry (POE), including Medications, Nutrition, Radiology etc.
labevents	About the laboratory measurements of patients.
chartevents	About the available charted data of patients.
d_items	The dictionary of the measurement item.
icustay	About the stay information of ICU, including the care unit and length.



3.1.2 MIMIC-CXR

The MIMIC-CXR database collected data between 2011 - 2016 and contained patients' CXR images in DICOM and JPEG format, radiology reports, and structured labels determined by natural language processing (NLP) tools [33]. CXR is an imaging test that is the most frequent radiological test performed in the ICU. It is also effective and can provide clinicians with additional information about the patient's organs like lungs and heart. In this study, radiology reports and images in DICOM format were used. The structure and required data files of MIMIC-CXR are shown in Figure 3.2.

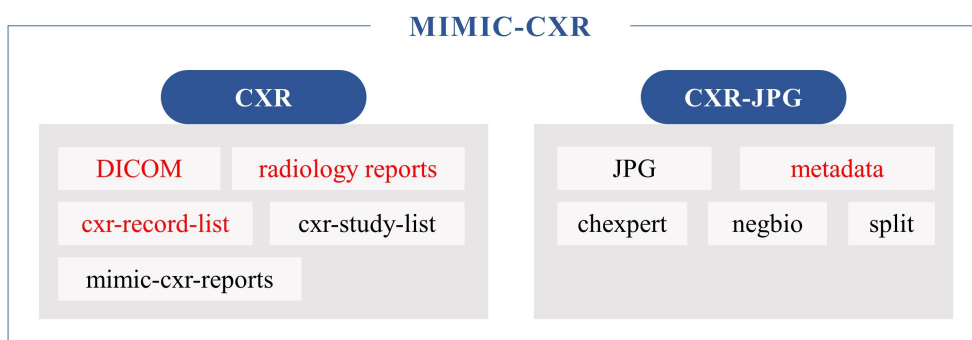


Figure 3.2: The structure of the MIMIC-CXR database. The required data files for this study are marked in red.

Table 3.2: The introduction of the required data files in the MIMIC-CXR database.

File	Information
DICOM	CXR images in DICOM format.
radiology reports	De-identified radiology reports in txt format.
cxr-record-list	Record the connection key value between the patients and images.
metadata	14 structured labels extracted from an NLP tool.



3.2 Data Processing

3.2.1 Data extraction

Clinical data in the MIMIC-IV and MIMIC-CXR databases were recorded by person, time, and category. For patients in the database, each person's data was organized into one-hour intervals, and thus I could readily obtain the data of each patient with a time series relationship. Besides, I only extracted patients' data for the first 7 days since the definition of ARDS includes that it must occur within 7 days of admission.

The filter work was performed by PostgreSQL, which is an object-relational database management system (DBMS). It is a powerful tool for data selecting, sorting, and filtering. Related instructions, such as 'Select', 'Where', 'Full Join', and 'Group By', were used to prepare the data. It is an efficient tool for data preprocessing and is also a recommended tool in the official documentation of MIMIC-IV. The data filtering steps are as follows:

- Filter the required column of the data files by instruction 'Select'.
- Filter the identity number of the required variables by instruction 'Where'.
- Calculate the measurement time for these variables.
- Use instruction 'group by' to attribute variables to the patient and time.
- Join patient's information of multiple tables by instruction 'Full Join'.

The information on feature selection and patient selection will be explained in the following sections. In addition to the features required by the models, the information on the partial pressure of oxygen (PaO_2), the fraction of inspired oxygen (FiO_2), the level of positive end-expiratory pressure (PEEP), the International Classification of Diseases (ICD)

code with versions 9 and 10, and chest radiograph reports were also extracted in order to label patients as positive and negative cases of ARDS accurately. The corresponding identity numbers of these features in the MIMIC-IV database are shown in Table 3.3.

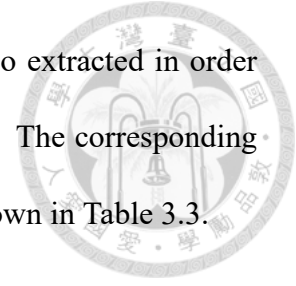


Table 3.3: The corresponding identity numbers of the needed features for labeling in the MIMIC-IV database.

Feature Name	Item ID
The partial pressure of oxygen (PaO_2)	50821
The fraction of inspired oxygen (FiO_2)	50816, 223835
The level of positive end-expiratory pressure (PEEP)	50819

I followed the Berlin definition as the standard for the labeling work [7]. Following the Berlin definition of ARDS, patients with cardiac failure should not be considered positive cases, and they are excluded by ICD codes. ICD codes can provide the medical classification of patients. Furthermore, I searched the text in chest radiograph reports to ensure that bilateral opacities were present for positive labels. The ratio of PaO_2 to FiO_2 (P/F ratio) is a powerful tool for identifying hypoxemia and determining the severity of ARDS. PaO_2 is a measurement of oxygen pressure in arterial blood and reflects how well oxygen can move from the lungs to the blood. FiO_2 is the percentage of oxygen that a person inhales. The onset time for ARDS was determined as the first time of co-occurrence for the $\text{PEEP} \geq 5 \text{ cmH}_2\text{O}$ and $\text{P/F ratio} \leq 300 \text{ mmHg}$. Moreover, due to the definition of acute, the onset should occur within 7 days of admission.

3.2.2 Feature selection

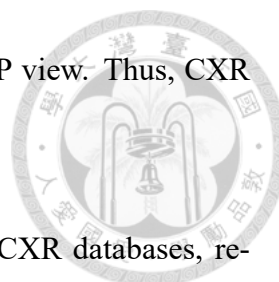
19 characteristics were extracted from the MIMIC-IV database and considered them as input features for the numerical model, including 2 demographics (age, gender), 12 clin-

ical measurements (systolic blood pressure, diastolic blood pressure, mean blood pressure, respiratory rate, SpO₂, heart rate, temperature, Glasgow Coma Scale, tidal volume, plateau pressure, minute volume), and 6 laboratory measurements (lactate, pH, creatinine, bilirubin, platelet, WBC). Except for age and gender, the corresponding identity numbers of other selected features in the MIMIC-IV database are shown in Table 3.4. Features were selected based on the review of the literature [19–21] and the clinician’s recommendations.

Table 3.4: The corresponding identity numbers of the 19 selected features in the MIMIC-IV database.

Feature Name	Item ID	Feature Name	Item ID
Age	-	Tidal Volume	224685, 224684, 224686
Gender	-	Plateau Pressure	224696
Systolic Blood Pressure	220050, 220179	Minute Volume	224687
Diastolic Blood Pressure	220051, 220180	Lactate	50813
Mean Blood Pressure	220052, 220181	pH	50820
Respiratory Rate	220210, 224690	Creatinine	50912
SpO ₂	220277	Bilirubin	50885
Heart rate	220045	Platelet	51265
Temperature	223761, 223762	WBC	51311, 51301
Glasgow Coma Scale	223900, 223901, 220739		

On the other hand, I selected CXR images from the MIMIC-CXR database with the frontal anterior-posterior (AP) view for the image model. Compared to the posterior-anterior (PA) view, the AP view would magnify the heart size. However, patients in the



ICU usually have difficulty standing and can only take CXR with AP view. Thus, CXR images with AP views were chosen in this study.

The features were extracted from the MIMIC-IV and MIMIC-CXR databases, respectively. Although patients in ICU were continuously monitored, some measurements were unavailable at all hours. To avoid the problem of missing values, measurements within six hours as a substitution were allowed. (Figure 3.3)

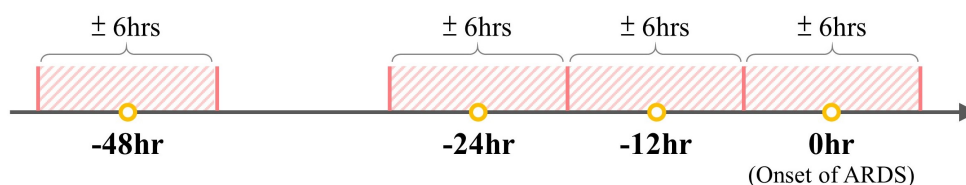


Figure 3.3: The graphical description of data selection.

Measurements within ± 6 hours were acceptable.

Table 3.5 is the description of the rules for data selection. While extracting data, the features had different acceptable times due to the different properties. For example, the measurements of platelet and WBC need to be obtained by a blood test. Due to the consideration of radiation dose, CXR is performed only when it is necessary for the examination. Therefore, measurements within 1 day as a substitution were allowed for features that were not measured frequently.

Table 3.5: Description of the rules for data selection.

Acceptable time	Feature
Within 6 hours	Systolic blood pressure, diastolic blood pressure, mean blood pressure, respiratory rate, SpO ₂ , heart rate, temperature, Glasgow Coma Scale, tidal volume, plateau pressure, minute volume
Within the day	Lactate, pH, creatinine, bilirubin, platelet, WBC, CXR



3.2.3 Patient selection

There were 382,278 patients in the MIMIC-IV database and 65,379 patients in the MIMIC-CXR database. Criteria for patient inclusion and exclusion were adopted during selection and patients who stayed in the ICU for more than 3 months were excluded. The inclusion and exclusion processes are shown in Figure 3.4, which are as follows:

- Patients aged 18 years old or older were included.
- Patients should have measurements of vital signs, PaO₂, and FiO₂.
- Patients should have CXR records.

Following the inclusion and exclusion process and the Berlin definition of ARDS, there were 356 patients labeled as ARDS-positive cases. Among them, only 324, 241, and 138 patients had numerical data and image data available at 12-hr, 24-hr, and 48-hr prior to onset, respectively. The imbalanced condition of positive and negative cases is common in clinical situations. For example, if a disease has a prevalence of about 10%, then the ratio of negative and positive cases will be approximately 9. However, the ratio of negative and positive cases in this study is smaller since only a few negative cases had CXR records in this study. This situation may be due to the implementation of clinical treatment, where clinicians may not perform chest imaging on patients without signs of related symptoms. The number of negative cases was approximately five times greater than positive ones. For consistency, I followed the same concept to select the negative cases at 12-hr, 24-hr, and 48-hr prior to onset. Overall, 1,861 patients were labeled as ARDS negative cases, and 1,620, 1,205, and 690 patients had data available at 12-hr, 24-hr, and 48-hr prior to onset, respectively. The number of patients eligible for this study of different time models is shown in Table 3.6.

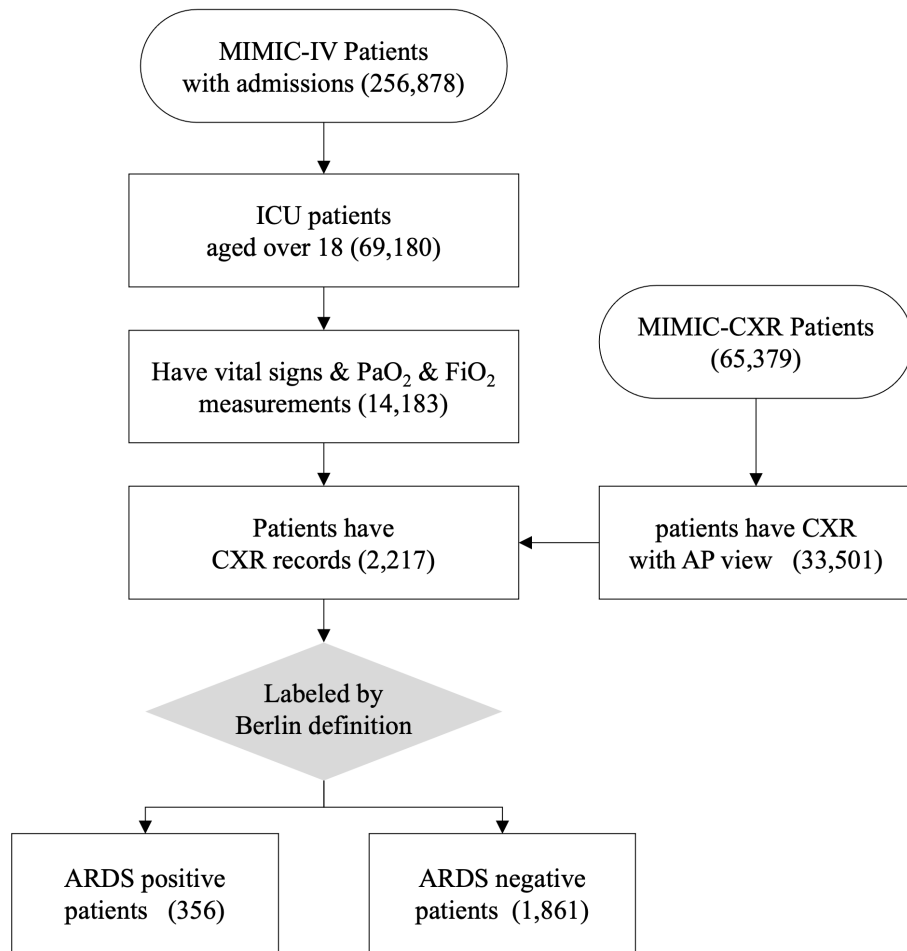


Figure 3.4: The flowchart of patient selection.

Table 3.6: The number of patients included in this study.

	With ARDS (Positive cases)	Without ARDS (Negative cases)	Total
Onset identification	356	1,861	2,217
12-hr prediction	324	1,620	1,944
24-hr prediction	241	1,205	1,446
48-hr prediction	138	690	828



3.3 Data Analysis

Demographics of patients included in this study are shown in Table 3.7. It can be observed from the table that the gender of the patients is about half male and half female. Meanwhile, the patients are mostly over 50 (79.6%) and mostly white or black (73.4%). As for the length of days staying in the ICU, it is relatively average. The distribution of demographics represents the characteristics of the databases.

Table 3.7: Demographic characteristics of subjects included in this study.

Characteristics		Count	%
Gender	Male	1239	55.9
	Female	978	44.1
Age (year)	18-29	142	6.4
	30-39	122	5.5
	40-49	188	8.5
	50-59	397	17.9
	60-69	523	23.6
	≥70	845	38.1
Ethnicity	White	1376	62.1
	Black	250	11.3
	Hispanic and Latino	85	3.8
	Asian	64	2.9
	Other	442	19.9
Length of stay (day)	<2	412	18.6
	2-5	889	40.1
	6-10	454	20.5
	>10	462	20.8

A histogram was used to observe the data distributions. Histograms of the extracted features are shown in Figure 3.5 and Figure 3.6, where gray represents all, orange represents the positive cases, and yellow represents the negative cases. However, the shape of the histogram is affected by the bin setting and the data size. Thus, I used the kernel den-

sity estimate (KDE) plot, which shows the estimation of the probability density function, to observe the difference between the positive and negative cases. (Figure 3.7)

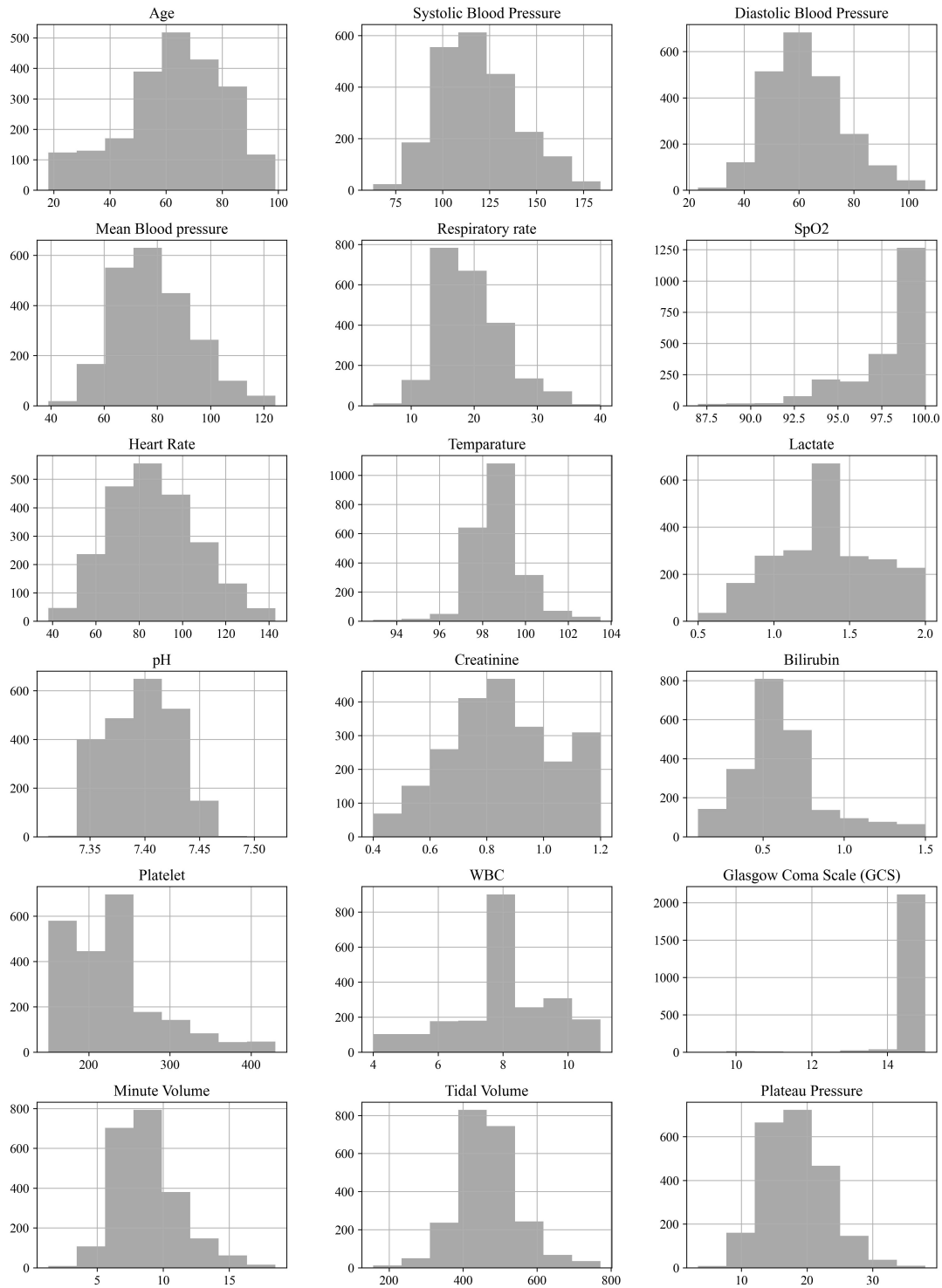
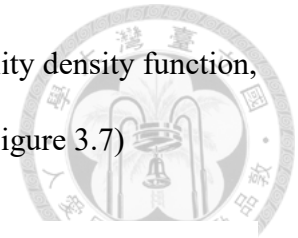


Figure 3.5: The histograms of 18 clinical features.

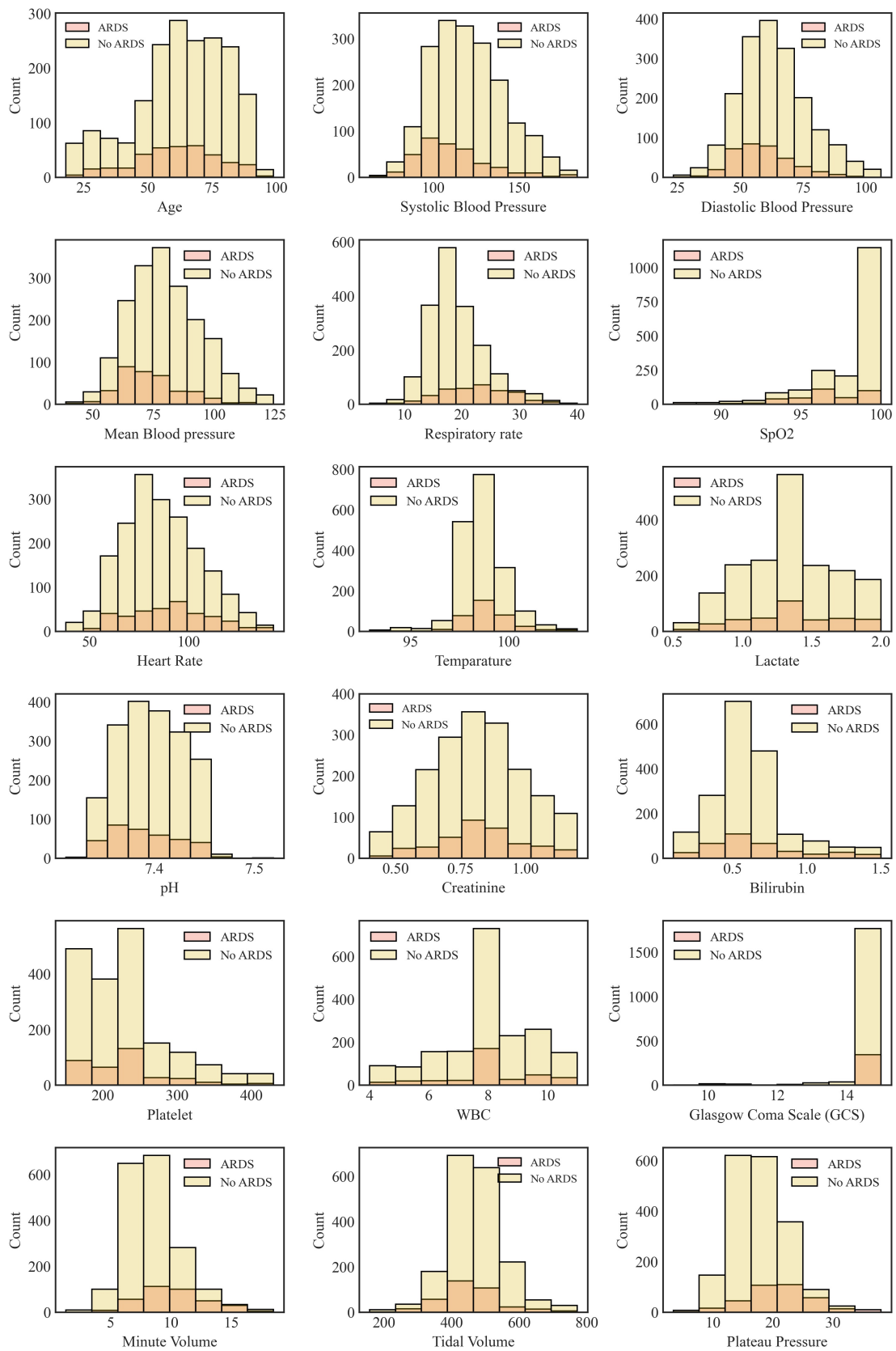


Figure 3.6: The histograms of 18 clinical features by different groups.

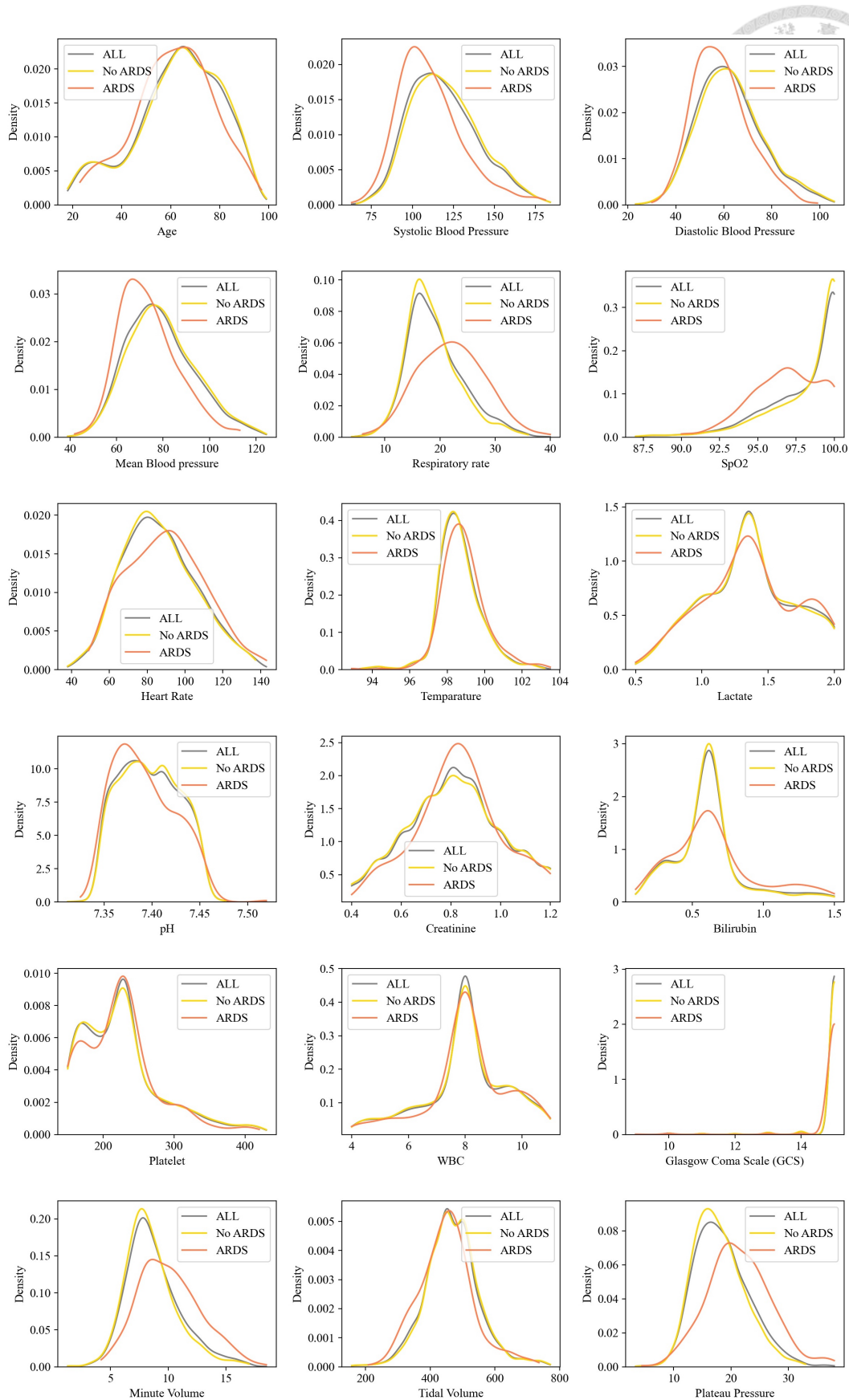


Figure 3.7: The KDE plots of 18 clinical features.





Chapter 4 Methods

This chapter is an introduction to machine learning and deep learning models; in addition, the techniques needed for the model training process will also be introduced. Section 4.1 to Section 4.3 present the details of the used methods. Section 4.4 and Section 4.5 describe the structure of single modality and multi-modality models adopted in this study.

4.1 Data Standardization

Data standardization is the process of rescaling the original data. For machine learning algorithms such as gradient descent-based and distance-based algorithms, feature scaling is crucial since the range of features in the original data differs. Without scaling, features with large magnitudes will probably dominate the models. In contrast, the tree-based model does not need to be standardized, since the tree-based algorithms are composed of multiple nodes and the scale of features does not affect the model.

In this study, the features were scaled through standardization (Z-score normalization) and the data standardization step was done by the *sklearn.preprocessing.StandardScaler* module in Python. The standard value of a sample x can be calculated by (4.1):

$$z = \frac{(x - \mu)}{s}, \quad (4.1)$$



where μ is the mean and s is the standard deviation of x . After the standardization, the values of each feature would all have a mean equal to zero and a standard deviation equal to one after the standardization.

4.2 Data Augmentation

Data augmentation is a technique used to increase data size for the training process. The main concept of data augmentation is to add slightly modified copies of already existing data and vary the training data. Data augmentation is commonly used for improving deep learning models in image classification problems. Classical image transformations include rotating, cropping, zooming, and histogram-based methods [35].

Whether the transformations are reasonable to the data should be noticed while applying data augmentation in the training process. For example, in this study, horizontal or vertical flips on CXR images should not be operated since if the organs' position was transposed, the different positions might confuse the outcome. The augmentations applied in this study were geometric and color transformations, including random rotation, random zooming, and random brightness (Figure 4.1). Data augmentation only needs to be performed on the training set. Transformations were operated on the training set; thus, the model can be trained with various data and improve its classifying ability. Geometric transformations were used to overcome positional biases, and color transformations were used to avoid inconsistencies in brightness. Transformations were performed by the *monai.transforms* module in Python.

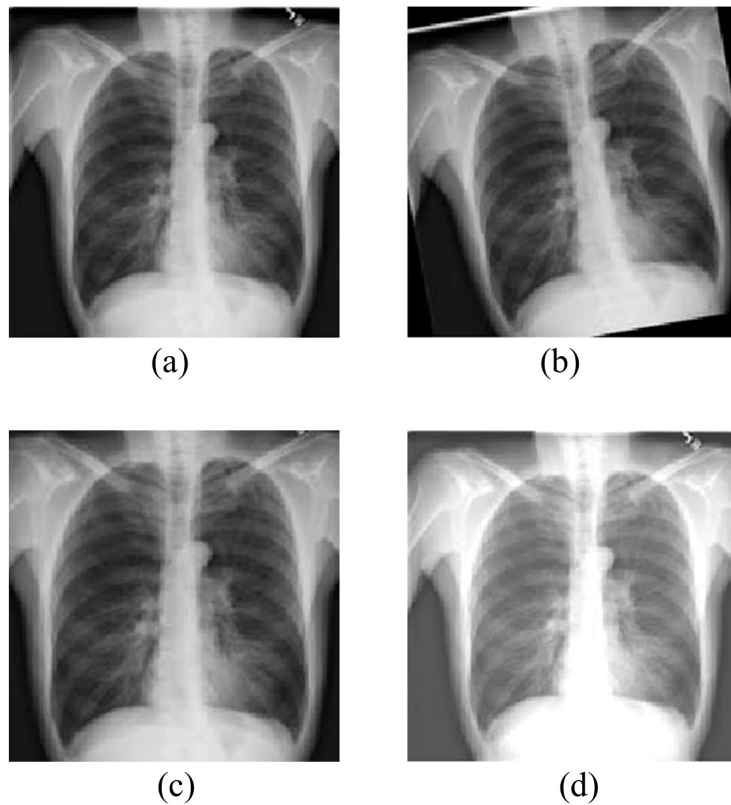


Figure 4.1: The examples of the CXR image augmentation.

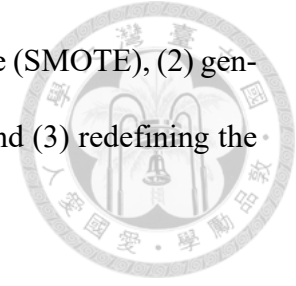
(a) Original (b) Random rotation (c) Random zooming (d) Random brightness.

4.3 Imbalanced Data

The class imbalance problem is common since real-world data usually have different numbers in each class. Data is mainly comprised of normal cases with only a small percentage of abnormal cases on topics such as medical diagnosis, machine fault detection, and fraud detection. In this study, the available patient data was imbalanced at each setting time and the ratio of patients with and without ARDS is about 1 to 5. If the class distribution of the data is imbalanced, the model may tend to predict the majority class rather than actually learning from the data.

There are 3 approaches for handling the imbalanced data problem in this thesis, in-

cluding (1) performing the synthetic minority oversampling technique (SMOTE), (2) generating more images of minority class through data augmentation, and (3) redefining the weight of loss functions.



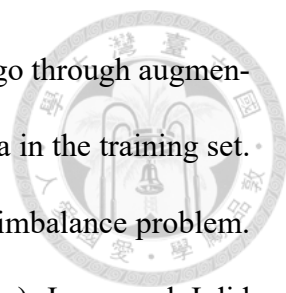
Some resampling methods were proposed for handling the imbalance problem, including oversampling and undersampling. These 2 methods aim to adjust the class distribution by decreasing the gap between the positive and negative cases. In this study, the method of oversampling was selected and performed through the synthetic minority oversampling technique (SMOTE). The central concept of SMOTE is to synthesize new data from the existing samples of the minority class. The procedure of SMOTE consists of the following steps: [36]

- Select pattern X_0 from the minority class.
- Pick one of the K nearest neighbors X of X_0 which also belongs to the minority class.
- Create a new pattern Z on a random point on the line segment connecting the pattern X_0 and the selected neighbor X . (The representation of Z is shown in Equation 4.2)

$$Z = X_0 + w(X - X_0), \quad (4.2)$$

where w is a uniform random variable in the range $[0,1]$.

For the task of CXR image classification, data augmentation was applied to solve the data imbalance problems. In addition to enhancing the quality of training data and avoiding overfitting, data augmentation can also be used to balance the data. The probability of the positive and negative cases to operate augmentation was adjusted, and the difference is



about five times. Thus, data with positive labels were more likely to go through augmentation and generate a slightly adjusted image, which balanced the data in the training set. Moreover, the weight of loss functions were redefined to handle the imbalance problem. More importance was placed on the loss of positive cases (the minority). In general, I did not ignore any data from the majority class since I did not operate the approach of under-sampling, which is the advantage of the methods in this thesis to deal with imbalanced data.

The mentioned approaches for solving the problem of imbalanced data were all done through the modules in python, including (1) *imblearn.over_sampling* for performing SMOTE, (2) *torch.utils.data.sampler.WeightedRandomSampler* was used to make the positive cases have 4 times probability be selected in every dataloader and operated augmentations, and (3) The parameter *scale_pos_weight* in *xgboost.XGBClassifier* was set to control the balance of positive and negative weights and was set to around 5~6. The parameter *pos_weight* in *torch.nn.BCEWithLogitsLoss* was set to around 1.2~1.5 for the training of neural networks.

4.4 Models

4.4.1 Decision tree

A decision tree is a well-known tool for classification and regression since it can be visualized and is easily understood [37]. Like the name, the decision tree is a tree-structured model that includes nodes and branches. The modeling was done by the module *sklearn.tree.DecisionTreeClassifier* in Python. The number of nodes and branches will

increase if the depth of the tree gets deeper. The top node is called the root node, and the node at the end of the tree is called the leaf node or the decision node (Figure 4.2).

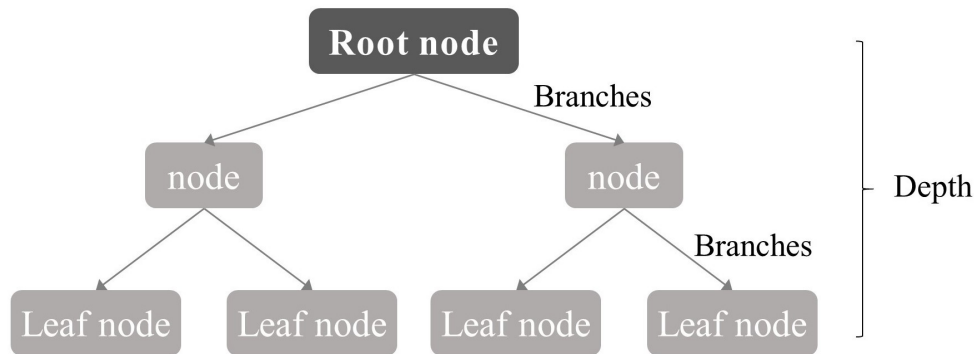


Figure 4.2: The structure of a decision tree.

In the tree-based models, data would split multiple times according to specific cutoff values in the features. After data in the root node pass through numerous branches and nodes, it will stop at the leaf node and get a class label.

4.4.2 Random forest

Random forest is an ensemble learning algorithm of decision trees and uses a bagging (bootstrap aggregation) framework [38]. The modeling was done by the module *sklearn.ensemble.RandomForestClassifier* in Python. The main concept of random forest is making class prediction through a large number of decision trees (Figure 4.3) and the steps of random forest is described as follows:

- Establish random subsets of data by the method of bagging.
- Construct decision trees over the subsets of data, and each tree will consider different features. If the total number of features is M , the suggested number of features in each tree is about \sqrt{M} .

- Make predictions from the outputs of the established decision trees. Common methods for ensemble output include majority voting and averaging.

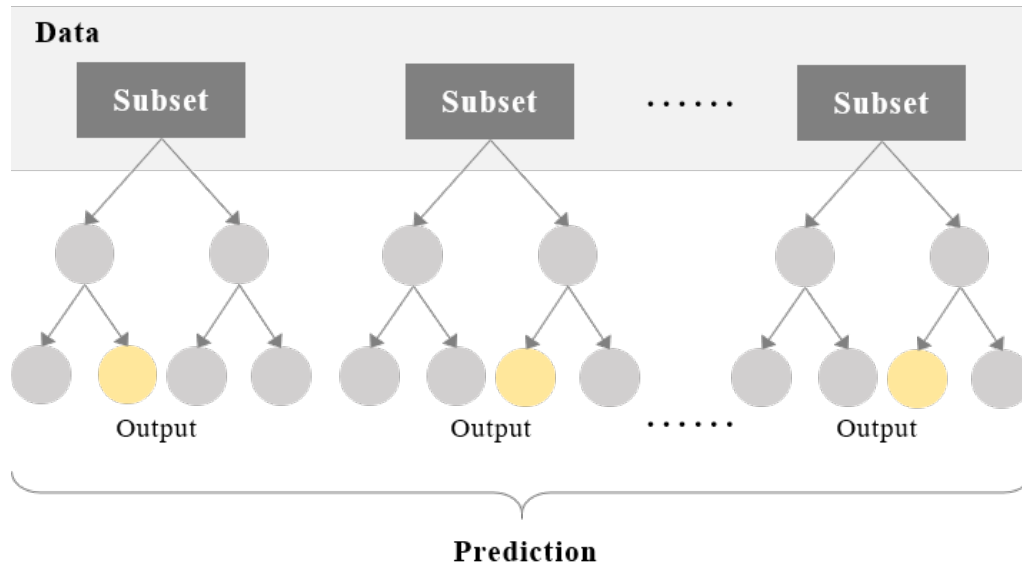
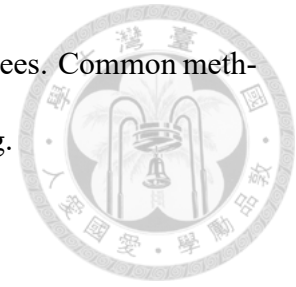


Figure 4.3: The main concept of the random forest algorithm.

4.4.3 Extreme gradient boosting

XGBoost is another ensemble learning algorithm of decision tree [39]. Modeling was done using the module *xgboost* in Python. Unlike random forest uses a bagging framework mentioned in the previous section, XGBoost uses a gradient boosting framework. The meaning of ensemble learning is to ensemble multiple weak classifiers to obtain a strong classifier and the main difference between bagging and boosting is that the classifiers relate to each other in boosting. The misclassified cases were passed to the next classifiers; thus, the classifiers could learn from the misclassified cases.

Moreover, the regularized objective function is improved, combining a differentiable convex loss function and a penalty term for model complexity.



$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \quad (4.3)$$

where

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2. \quad (4.4)$$

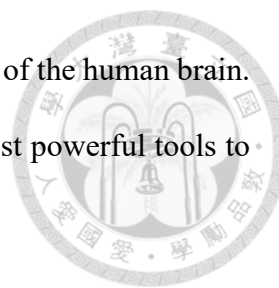
The objective function consists of the loss function l and the regularization function Ω . The former measures how well the model fits on the training data and the latter measures the complexity of the model. In the loss function l , i is the index of data, where y_i is the i -th data and \hat{y}_i is the prediction of i -th data. In the regularization function Ω , k is the index of trees, T is the number of leaves, ω is the leaf weight of the tree, γ and λ represent the complexity of the model.

XGBoost model is known for the advantage of high efficiency and accuracy; thus, it has become one of the most popular machine learning algorithms nowadays.

4.4.4 Convolutional neural network

Deep learning is a subset of machine learning with a structure similar to the human brain. The complex and deep structure that demands enormous amounts of computing power is considered a limitation of deep learning. However, deep learning has become publicly acceptable and increasingly popular since the successful development of cloud computing and graphics processing unit (GPU) in recent years.

The concept of neural networks (NNs) was first developed in the 1940s [40] before the idea of convolutional neural networks (CNNs). NNs mimicked the brain's perfor-



mance with biological neurons and were to realize a simplified model of the human brain. Over the past few decades, NNs have been considered one of the most powerful tools to handle large amounts of data and solve classification problems.

The structure of NN mainly consists of an input layer, an output layer, and multiple hidden layers with several neurons. Figure 4.4 is the schematic diagram of an artificial neural network (ANN) with every neuron connecting to another. Each connection was associated with weights and thresholds. The neuron can be thought as a mathematical function. The features in the input layer can go through different neurons, get the intermediate output, and continuously go to the next hidden layers. After the last hidden layer, the output layer will output the model prediction value.

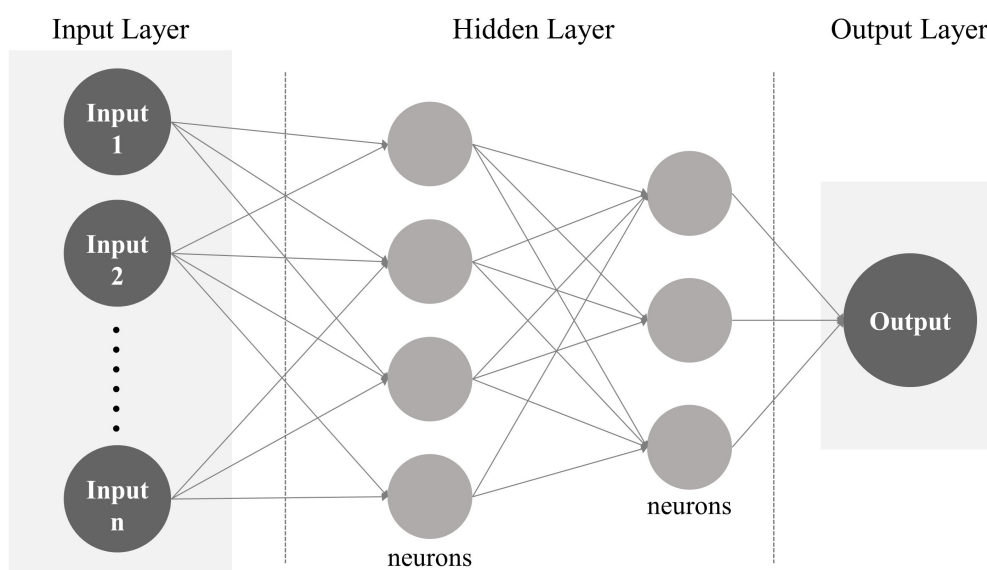


Figure 4.4: The schematic diagram of ANN with 2 hidden layers

CNNs were first developed in the 1980s [40] and are widely used today to recognize objects in images. CNN is a well-known algorithm for image processing since its remarkable accuracy. The classical architecture of CNN includes convolution, pooling, and fully connected layers. Convolution layers are used to extract features from the input image by

multiple kernels and preserve the nearby information of pixels. The pooling layers, such as max-pooling and mean-pooling, are used to reduce the spatial size of the feature maps. After the convolution and pooling layers, the extracted features will be flattened and fed to the fully connected layers.

For the image classification task in this study, the Densely Connected Convolutional Network (DenseNet) was applied [2], which is pre-trained on ImageNet [41]. Unlike the traditional CNNs, DenseNet connects each layer to every other layer; that is, there would be $\frac{L(L+1)}{2}$ direct connections for L layers. The architecture of a 5-layer dense block is shown in Figure 4.5. The advantages of DenseNets include that they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

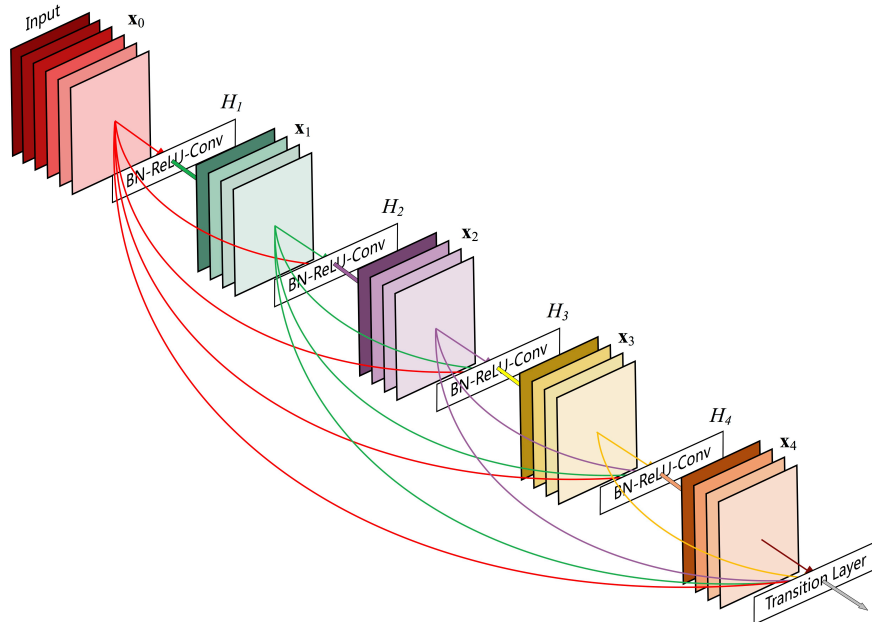


Figure 4.5: The architecture of a 5-layer dense block [2].
(Note that each layer takes all preceding feature-maps as input.)

DenseNet with 121-layer and 169-layer were applied in this thesis for the image-only model. The difference between them is the number of dense blocks. (Figure 4.6) In layers of Dense Block(3) and (4), 169-layer DenseNet has 8 and 16 blocks, respectively, more than 121-layer DenseNet.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56 28 × 28	1 × 1 conv 2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28 14 × 14	1 × 1 conv 2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14 7 × 7	1 × 1 conv 2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool 1000D fully-connected, softmax			

Figure 4.6: The architecture of DenseNet with different layers [2].

The module *pytorch* was used for neural network models, the package *torch.nn* in *pytorch* can construct the models. The image-only models in this study used the module *torchvision.models.densenet121* and *torchvision.models.densenet169*, which were both with *weights* equal to 'DEFAULT'.

4.5 Fusion Strategies of Multi-modality

Since the data sources include patient numerical and image data, I applied a multi-modal analysis. The concept of the multi-modality model is to exploit the features of multiple modalities. There are different fusion strategies that can be implemented, and this study used early-level and late-level fusion. The structure of the multi-modality mod-

els with varying fusion strategies is shown in Figure 4.7.

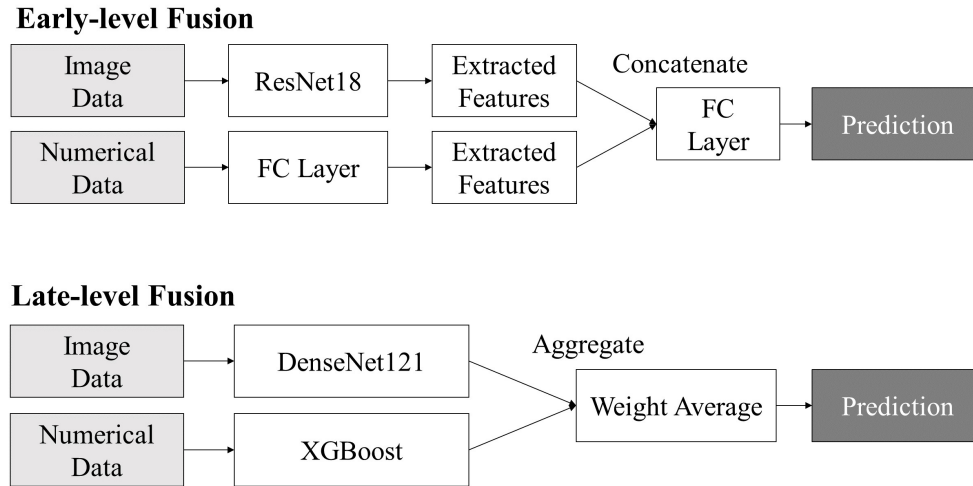


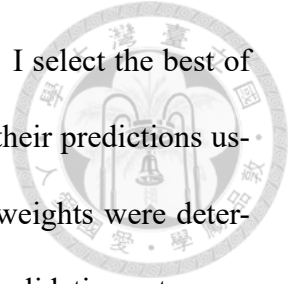
Figure 4.7: The structure of the multi-modality models.

2 different fusion strategies were applied: (a) Early-level Fusion (b) Late-level Fusion.

Early-level fusion extracts features of different modalities data and concatenates them before the classification model. Thus, it is also called feature-level fusion. Different neural network structures were used to extract features from the image and numerical data separately. For the image data, an 18-layer residual neural network (ResNet18) was used to extract features, which is a CNN model with residual blocks. On the other hand, a 3-layer fully-connected (FC) neural network was used to extract features from numerical data. Data standardization was performed on each column of numerical data separately before passing the neural network to avoid particular features dominating the objective function and impacting the model’s performance. The size of the extracted features of the image and numerical data were both 1×256 , and the size of the concatenation would be 1×512 . The extracted features would be fed into another 3-layer FC neural network after concatenation to make the final predictions.

Late-level fusion, or decision-level fusion, combines the predictions of multiple mod-

els. Late fusion is relatively common due to being easily computed. I select the best of numerical-only and image-only models, respectively, and aggregate their predictions using the weight averaging function to get the final predictions. The weights were determined by the performance of the two single-modality models in the validation set; more importance was given to the model that performed better.







Chapter 5 Model Evaluation

5.1 Evaluation Metrics

The model performance was evaluated based on the confusion matrix and the receiver operating characteristic (ROC) curves. A confusion matrix is often used to describe the performance of a classifier since it is easy to understand. It can also clearly show the number of cases correctly and incorrectly classified. The application of the confusion matrix is not limited to the binary classification and can also be used in multi-class classifiers. Since this work was a binary classification task (ARDS positive or negative), the confusion matrix is in the form of a 2×2 square matrix (Table 5.1), and some indicators such as accuracy ($\frac{TN+TP}{TN+FP+FN+TP}$), sensitivity ($\frac{TP}{TP+FN}$), and specificity ($\frac{TN}{TN+FP}$) can be calculated from it.

Table 5.1: The confusion matrix for binary classification.

		Prediction	
		0	1
Actual	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

ROC analysis has been widely applied to clinical areas, including diagnostic tests, laboratory testing, epidemiology, radiology, and bioinformatics. An ROC curve is plotted with false positive rate (1-specificity) on the x -axis and true positive rate (sensitivity) on the y -axis at different classification thresholds (Figure 5.1). It can show the trade-off between specificity and sensitivity, and every point on the ROC curve represents different thresholds. The area under the ROC (AUROC) is used to measure the ability of a classifier to discriminate between classes. AUROC ranges from 0 to 1, where 0 represents a model with 100% incorrectly predicted and 1 represents a model with 100% correctly predicted.

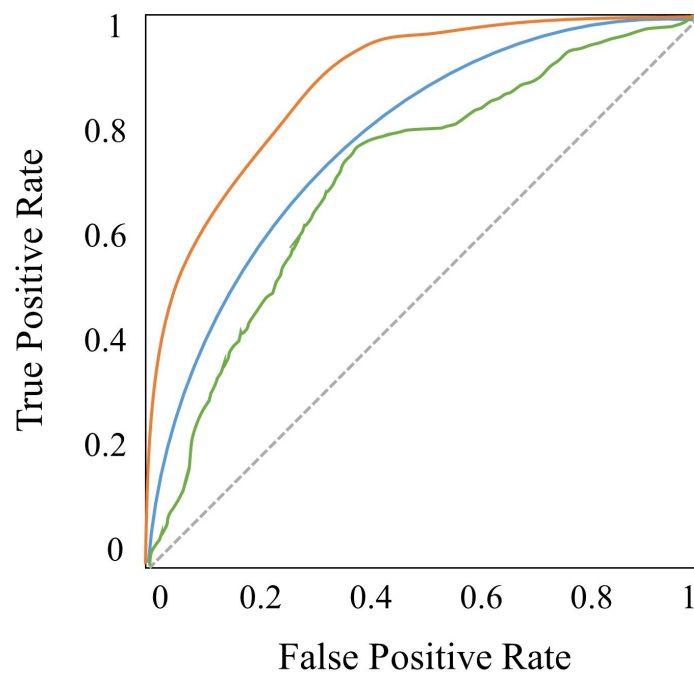
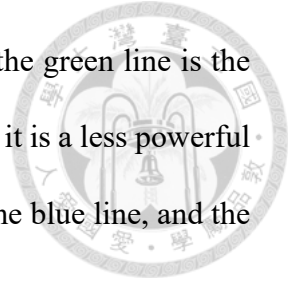


Figure 5.1: The graph of the ROC curve with false positive rate on the x -axis and true positive rate on the y -axis.

In Figure 5.1, the gray dotted line is the diagonal line and represents a random classifier that only has an AUROC of 0.5. The ROC curve is generally above the diagonal line. If a ROC curve presents under the diagonal line, it indicates that the performance of this classifier may be worse than a random copper toss. The orange, blue, and green lines in

Figure 5.1 represent 3 classifiers with different performance. Since the green line is the closest to the diagonal and has the lowest AUROC among the 3 lines, it is a less powerful classifier. Overall, the orange line represents a better classifier than the blue line, and the blue line represents a better classifier than the green line.



5.2 Stratified Cross-Validation

Cross-validation is a statistical technique for evaluating machine learning models and is used to ensure the ability of the model [42]. It is a common method for presenting the results of machine learning models. Moreover, it can be used to avoid overfitting. The cross-validation is performed through the following 3 steps:

- Divide the data into equal parts (k folds).
- Each part would be used as the validation set in turn.
- Evaluate the model performance and repeat the steps.

The patient data included in this study were randomly split into training and testing sets. The training set would be further divided into training and validation and operate k -fold stratified cross-validation, which would never be used for testing. The testing set was a hold-out group and was fixed for different models to compare the model's performance fairly. Besides, the data size of the testing set was the same as the validation set (Figure 5.2). In general, the training and validation sets are used for the training process, where the validation set is used to validate the model performance during training. In contrast, the testing set is used to evaluate the final model performance and will not be involved in the training process.

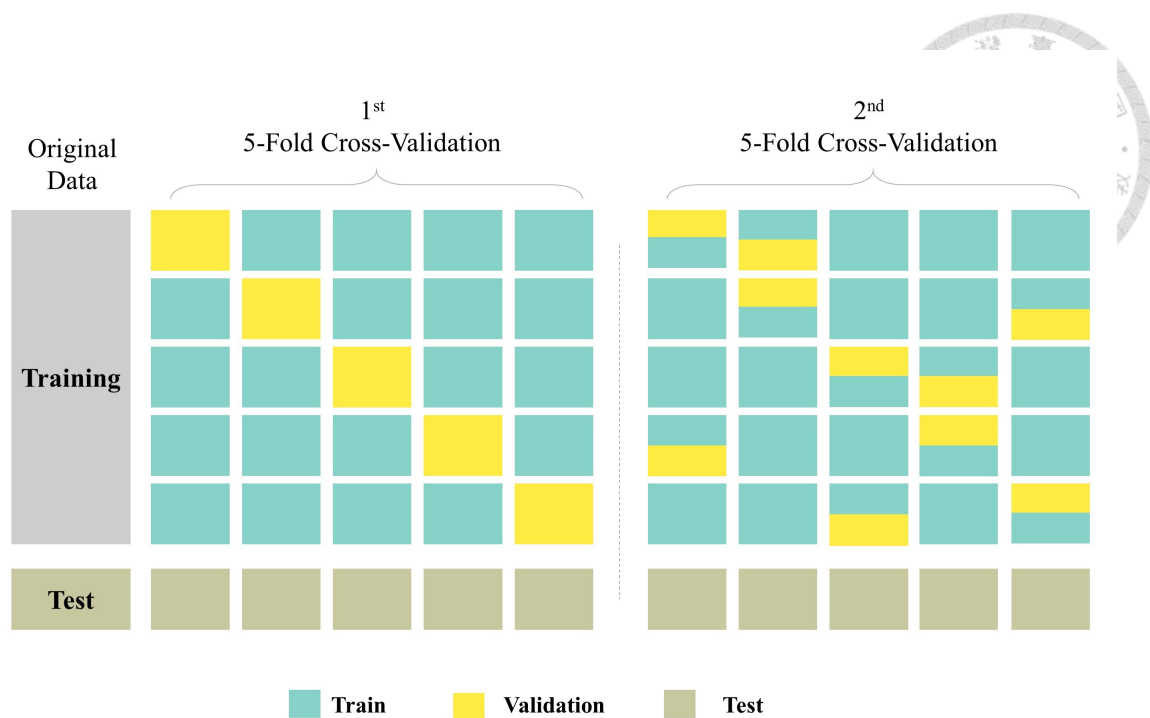


Figure 5.2: The implementation of 5-fold cross-validation.

The k -fold stratified cross-validation approach was used for splitting data and was done using the `sklearn.model_selection.StratifiedKFold` module in Python. The purpose of the cross-validation is to ensure the stability of the model performance during the training process. While the standard type of cross-validation is single run, the repeated cross-validation can also be used. The model performance obtained from the repeated cross-validation can be more accurate to the real situation.

While splitting the data, the stratified cross-validation approach allows data in different folds have almost the same ratio of positive and negative patients. The stratified method is based on the labels of patients with and without ARDS, which is the distinctive characteristic of this approach. The number of divisions, k , was set to 5 for onset identification and 12-hr prediction, and 10 for 24-hr and 48-hr prediction since the data were relatively smaller. In this study, the k -fold stratified cross-validation was operated twice. Thus, the model performance would be evaluated on $k \times 2$ iterations.

Take the data of 0-hr identification for example, Figure 5.2 and Table 5.2 illustrate the method of splitting the data and present the proportion of the positive cases in each set, respectively. After data splits, the ratio of positive and negative cases in each set and fold retains the same due to the implementation of k -fold stratified cross-validation. The proportion of the positive cases in different sets will be almost the same, about 15.95~16.21%.

Table 5.2: The proportion of the positive cases in different sets after the stratified 5-fold cross-validation.

	Positive cases	Negative cases	Proportion of the positive cases (%)
<i>1st</i>			
Fold 1	59	311	15.95
Fold 2	60	310	16.21
Fold 3	60	310	16.21
Fold 4	59	310	15.99
Fold 5	59	310	15.99
Test	59	310	15.99
<i>2nd</i>			
Fold 1	59	311	15.95
Fold 2	60	310	16.21
Fold 3	60	310	16.21
Fold 4	59	310	15.99
Fold 5	59	310	15.99
Test	59	310	15.99





Chapter 6 Results

This chapter summarizes the performance of all models developed in this thesis. Section 6.1 and 6.2 present the performance of single- and multi-modality models. Section 6.3 makes a comparison between the model performance of different models.

In the study, the final model evaluation was operated on the hold-out testing set and the performance was calculated from the results of k -fold stratified cross-validation. ($k=5$ for onset and 12-h prediction, and $k=10$ for 24-h and 48-h prediction.) Overall, in the testing set, there were 222, 195, 145, and 83 patients at onset, 12-hr, 24-hr, and 48-hr prior to the onset, respectively.

6.1 Single-modality Models

There were 2 types of single-modality models in this thesis, numerical-only and image-only. First, the numerical-only models include the DT, RF, and XGBoost algorithms, which are all tree-based algorithms. In general, RF and XGBoost are considered the advancement of DT. Both RF and XGBoost applied the ensemble technique, where RF with the bagging technique and XGBoost with the boosting technique.

During the training process, the SMOTE approach was operated on the minority class

to handle the imbalance problem and balance the data. After implementing SMOTE, the training sets would have the same number of patients with ARDS and without ARDS. Take the data of onset identification for example, Figure 6.1 shows the patient number of the training set changed by the implementation of SMOTE. In the beginning, the number of negative cases was about five times of the number of positive cases. After SMOTE, the data was balanced and the training set was enriched.

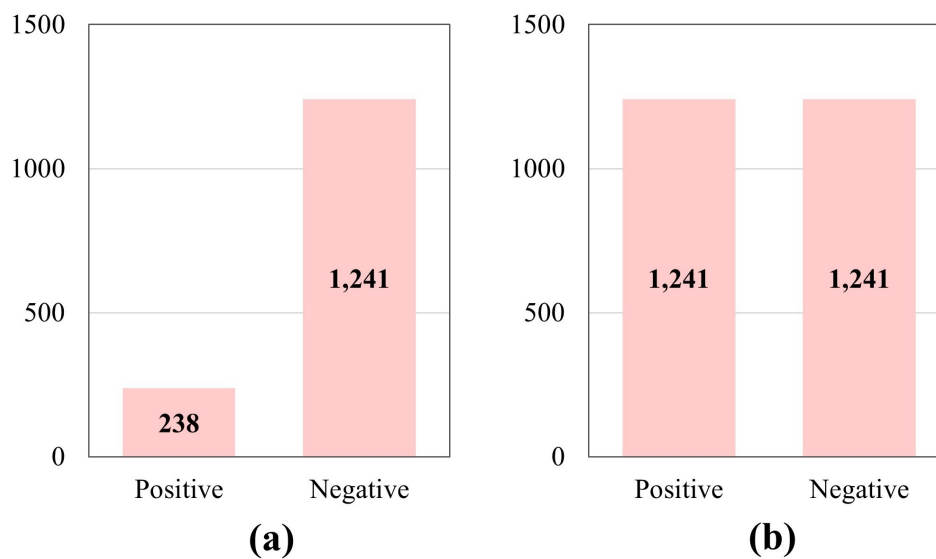


Figure 6.1: The patient number in the training set with the implementation of SMOTE.
(a) Before SMOTE (b) After SMOTE

The approach of SMOTE was used with the DT and RF algorithms. In addition, there were 2 types of XGBoost models with operating SMOTE and adjusting the class weight in the loss function, respectively. Thus, there were 4 different models for numerical-only data.

- Decision Tree with SMOTE.
- Random Forest with SMOTE.
- XGBoost with SMOTE.

- XGBoost with weight adjustment.



Figure 6.2 summarizes the ROC curves of the above 4 models. Observed from the figure, the purple (XGBoost with SMOTE) and blue (XGBoost with weight adjustment) curves have relatively large AUC, obviously.

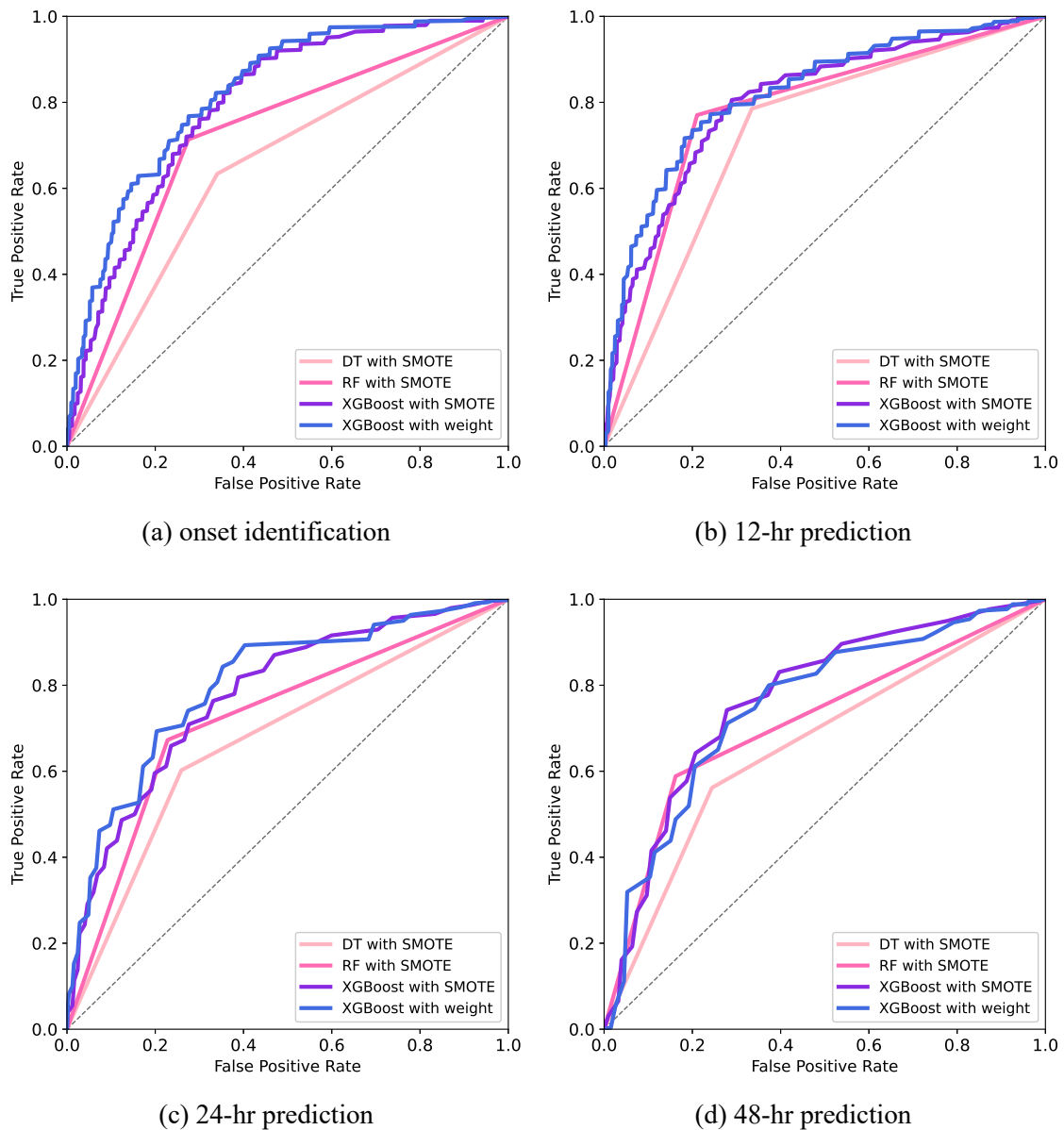


Figure 6.2: ROC curves of numerical-only models.

On the other hand, the image-only models include DenseNet121 and DenseNet169, which are both CNN structures with different layer numbers. Figure 6.3 summarizes the

ROC curves of the image-only models.

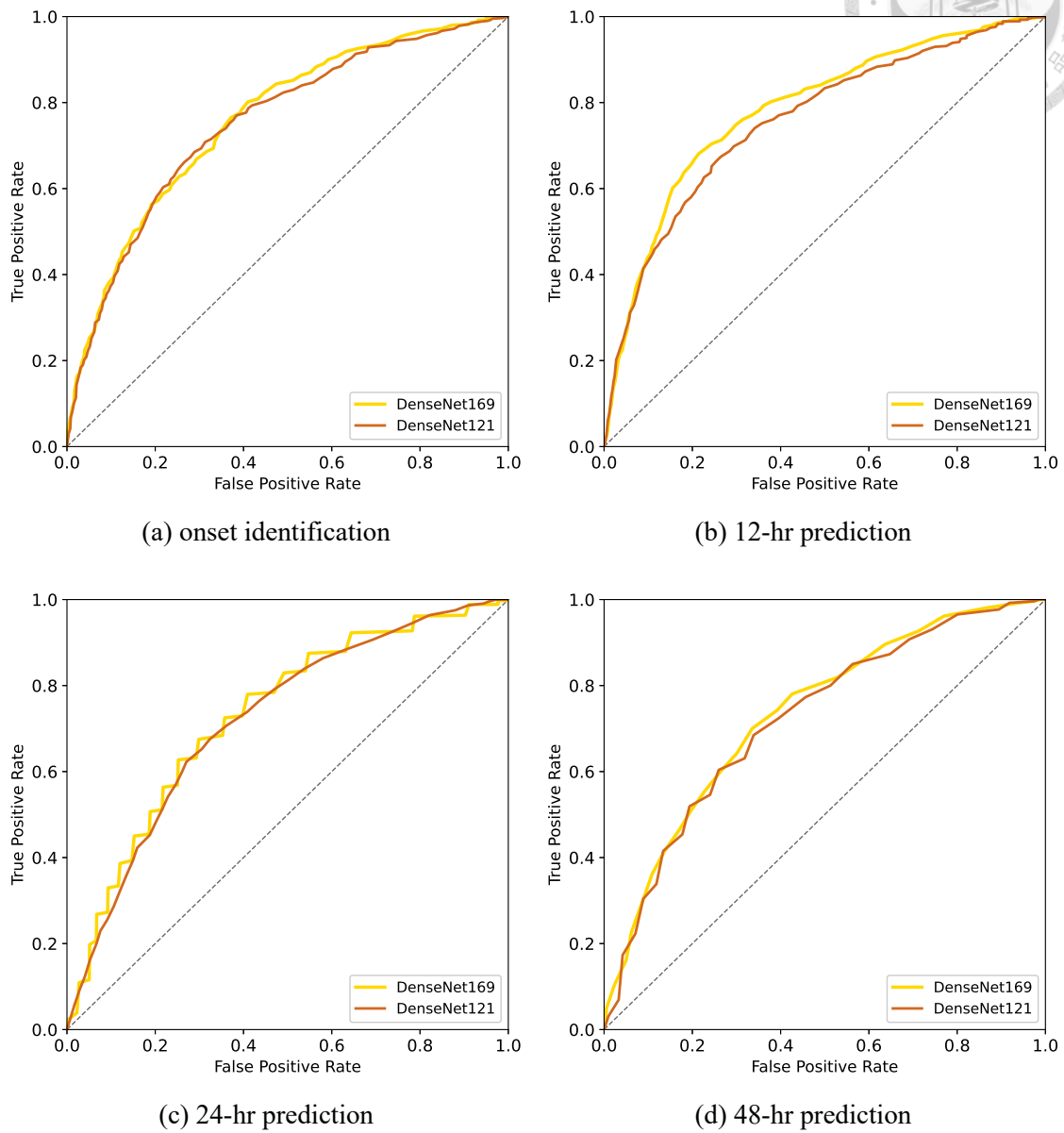


Figure 6.3: ROC curves of image-only models.

Table 6.1 summarizes the performance of the single-modality models developed in this study. Compared with the numerical-only models, the image-only models performed better than DT with SMOTE at all times and performed worse than XGBoost with SMOTE or weight adjustment. Sometimes, the image-only models performed better than RF with SMOTE (onset identification and 48-hr prediction), and others performed worse than RF with SMOTE (12-hr and 24-hr prediction).

Table 6.1: Performance of the numerical-only and image-only models.

Data are presented as mean (\pm standard deviation) and evaluated through k -fold stratified cross-validation. Abbreviations: AUC, area under the ROC curve; ACC, accuracy.

Model		Onset identification	12-hr prediction	24-hr prediction	48-hr prediction
Numerical-only model					
DT with SMOTE	AUC	0.6466 (± 0.02)	0.7252 (± 0.01)	0.6715 (± 0.05)	0.6586 (± 0.04)
	ACC	0.6553 (± 0.05)	0.7003 (± 0.05)	0.7252 (± 0.04)	0.7220 (± 0.04)
RF with SMOTE	AUC	0.7194 (± 0.01)	0.7798 (± 0.01)	0.7224 (± 0.02)	0.7132 (± 0.03)
	ACC	0.7233 (± 0.01)	0.7809 (± 0.01)	0.7550 (± 0.01)	0.7947 (± 0.02)
XGBoost with SMOTE	AUC	0.7934 (± 0.01)	0.8022 (± 0.01)	0.7781 (± 0.02)	0.7695 (± 0.02)
	ACC	0.7382 (± 0.01)	0.7750 (± 0.01)	0.7805 (± 0.02)	0.7987 (± 0.03)
XGBoost with weight adjustment	AUC	0.8360 (± 0.01)	0.8044 (± 0.01)	0.8034 (± 0.02)	0.7532 (± 0.02)
	ACC	0.7432 (± 0.01)	0.7660 (± 0.02)	0.7351 (± 0.02)	0.7120 (± 0.07)
Image-only model					
DenseNet121	AUC	0.7526 (± 0.02)	0.7588 (± 0.02)	0.7210 (± 0.02)	0.7218 (± 0.03)
	ACC	0.7246 (± 0.06)	0.7364 (± 0.04)	0.6893 (± 0.05)	0.7580 (± 0.02)
DenseNet169	AUC	0.7608 (± 0.01)	0.7839 (± 0.02)	0.7307 (± 0.02)	0.7338 (± 0.03)
	ACC	0.7176 (± 0.05)	0.7377 (± 0.04)	0.6821 (± 0.05)	0.6960 (± 0.06)



6.2 Multi-modality Models

The multi-modality models applied in this study include models with early-level and late-level fusion. For early-level fusion, the features of image and numerical data were extracted from ResNet18 and a 3-layer FC neural network, respectively. After the extraction work, the features would be concatenated and fed into another 3-layer FC neural network to make the final predictions.

During the operation of late-level fusion, the best models over the numerical-only and image-only models were selected and aggregated with weight average function; that is, XGBoost with weight adjustment and DenseNet169. Table 6.2 summarizes the model performance of the multi-modality models developed in this study and the mean and standard deviation of each indicator were calculated by $k \times 2$ iterations.

Table 6.2: Performance of the multi-modality models.

Data are presented as mean (\pm standard deviation) and evaluated through k -fold stratified cross-validation. Abbreviations: AUC, the area under the ROC curve; ACC, accuracy.

Model		Onset identification	12-hr prediction	24-hr prediction	48-hr prediction
Multi-modality model					
Early-level fusion	AUC	0.7916 (± 0.01)	0.7870 (± 0.02)	0.7934 (± 0.02)	0.7620 (± 0.02)
	ACC	0.7408 (± 0.02)	0.7608 (± 0.03)	0.7324 (± 0.04)	0.7587 (± 0.03)
Late-level fusion	AUC	0.8502 (± 0.01)	0.8442 (± 0.01)	0.8240 (± 0.02)	0.7951 (± 0.02)
	ACC	0.7941 (± 0.03)	0.7910 (± 0.03)	0.7786 (± 0.03)	0.7740 (± 0.04)



6.3 Model Comparison

The result of different models at the same time point is placed in Figure 6.4. It can be observed that the yellow lines are the closest to the diagonal, and the dark green lines are the outermost, where the yellow lines represent the image-only models and the dark green lines represent the late-level fusion models.

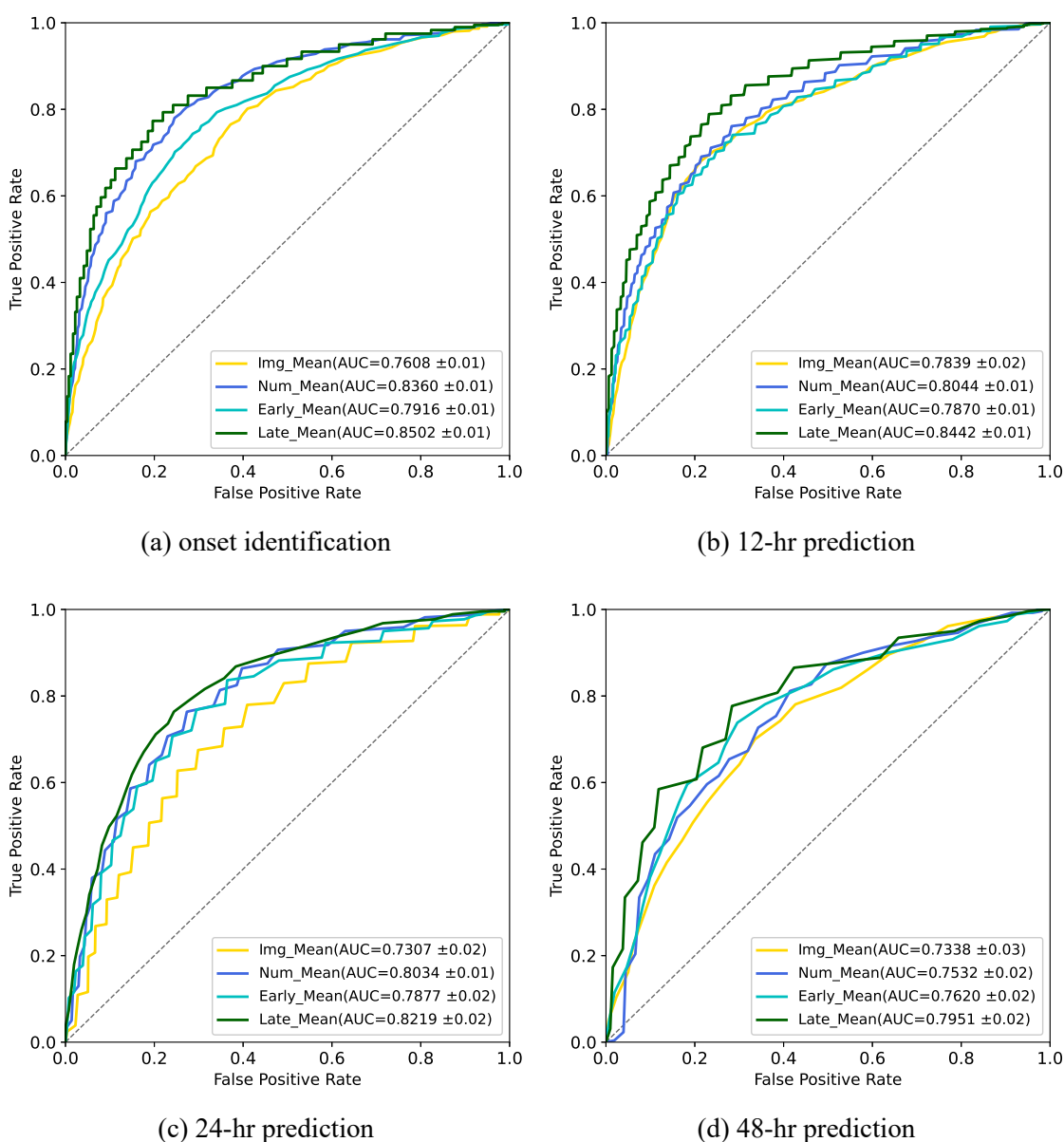
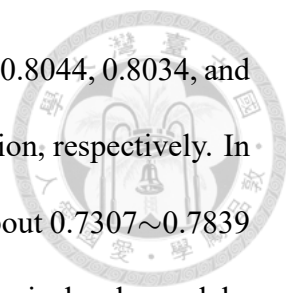


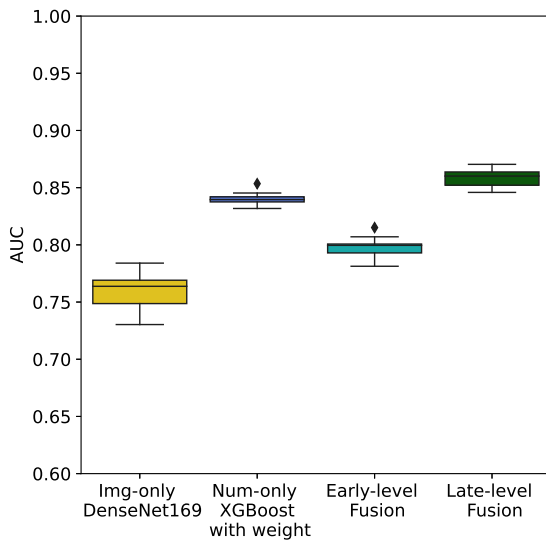
Figure 6.4: ROC curves of single- and multi-modality models.

AUC are presented as mean (\pm standard deviation) and evaluated through k -fold stratified cross-validation.

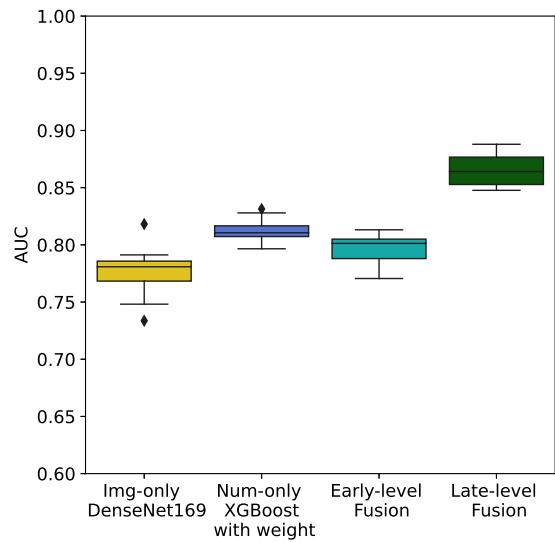


The numerical-only models demonstrated an AUROC of 0.8360, 0.8044, 0.8034, and 0.7532 for onset identification, 12-hour, 24-hour, and 48-hour prediction, respectively. In the meantime, the image-only models demonstrated an AUROC of about 0.7307~0.7839 for different times, which were relatively poor compared to the numerical-only models. For the multi-modality models, the model with early fusion demonstrated AUROC similar to the numerical-only models. The model with late fusion showed an AUROC of 0.8502, 0.8442, 0.8240, and 0.7951 for onset identification, 12-hr, 24-hr, and 48-hr prediction, respectively. Overall, the multi-modality model with late fusion performed a higher AUROC than others, and the image-only models had the lowest AUROC. Figure 6.5 shows the box plots of model performance with cross-validation.

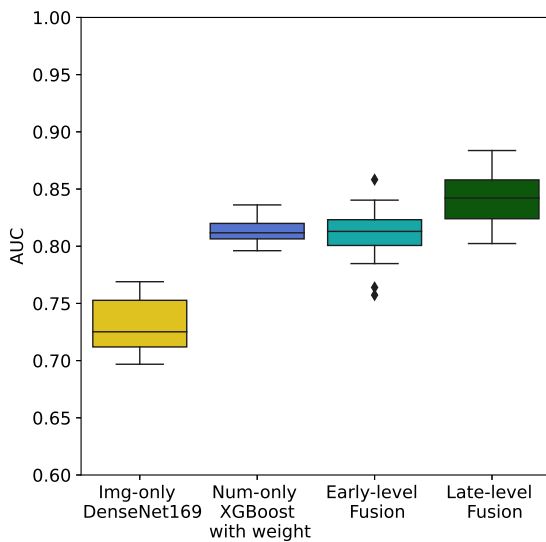
The performance of the above models decreased slightly when the prediction times got longer, where the best of the onset identification models demonstrated an AUROC of 0.8502, and the best of the 48-hour predictive model demonstrated an AUROC of 0.7951. Additionally, the multi-modality models improved the performance by about 6.0%~9.3% compared with the image-only models and about 1.4%~4.2% compared with the numerical-only models developed in this study. This research presented improvements over single-modality models, and the algorithm developed in this study can improve the identification and early prediction of ARDS.



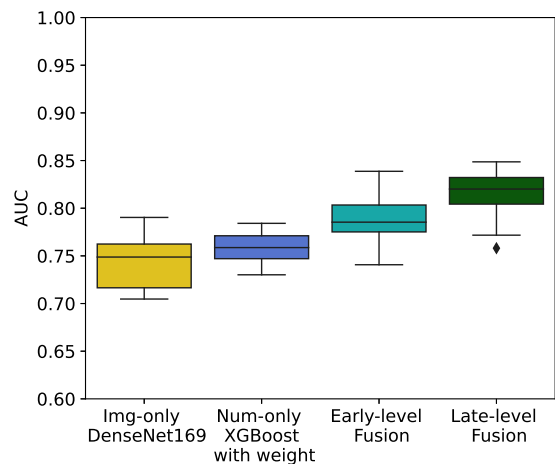
(a) onset identification



(b) 12-hr prediction



(c) 24-hr prediction



(d) 48-hr prediction

Figure 6.5: AUROCs of the different models with cross-validation.





Chapter 7 Discussion and Conclusion

7.1 Discussion

Since deep learning has been successfully applied to feature learning for single modalities, the application to multiple modalities has been proposed. Besides, the application of deep learning in the medical field has become increasingly extensive. This study developed single-modality and multi-modality models to predict ARDS, including onset identification, 12-hr, 24-hr, and 48-hr prediction.

Timely identification of ARDS patients is a challenging task, as there is no diagnostic test for ARDS. Patients diagnosed with ARDS must meet a series of clinical criteria that involve numerical and radiological features following the Berlin definition of ARDS in 2012 [7]. There are also many background factors that can affect the diagnosis of ARDS, such as low nurse-to-patient ratios or physician-to-patient ratios. Furthermore, the clinical data are often not available at the same time and CXR is usually not operated continuously within 24 hours, making the diagnosis of ARDS even more difficult. Thus, it is common to miss or delay the diagnosis of ARDS.

This study applies multi-modality models to diagnose and predict ARDS, and the results show that multi-modality models with late fusion performed a higher AUROC

than others. While most of the current studies consider only numerical or only image data and perform unimodal analysis, this research improves the performance for predicting ARDS through multimodal analysis. The application of multimodal analysis is also the most distinctive characteristic of this study.

In this study, the advantage of the multi-modality models is only shown on the model with late-level fusion, and the model with early-level fusion does not have outstanding performance. This may be because the features extracted from the neural networks were not well combined or because the architecture of the neural networks was not suitable. In order to improve the performance, different structures had been used in the early-level fusion to extract features from CXR images, including ResNet50, DenseNet121, and DenseNet169. However, the deeper structures did not show improvement on the trial of ARDS identification and eliminated the concerns about the lack of the layers. Fusing multimodal data may be a complex task, and the method of combining multi-modality data efficiently is a worth exploring issue and can be improved further.

7.2 Limitations

There are still some limitations of this research. First, only about 2,217 eligible patients met the inclusion criteria, which means that if I wanted to develop a prediction model 48 hours or more before the onset of ARDS, the number of patients would be less.

Second, MIMIC-IV and MIMIC-CXR, the databases used in this study, were both single-center databases. The clinical data includes numerical and image data, which were collected from BIDMC patients. That is, the validation and evaluation of the models were carried out at a single center. Single-center studies often lack data diversity. For

example, the patients' ethnicity in MIMIC-IV and MIMIC-CXR was mainly white, and other characteristics of these databases may also differ from different databases. Thus, external verification is needed to ensure that the models work well in other countries or hospitals. I expect to verify and evaluate this tool through other prospective validations in the future.

7.3 Conclusion

This study used the numerical and image data from ICU patients and applied single- and multi-modality models to identify ARDS. Moreover, I also developed predictive models, including 12-hr, 24-hr, and 48-hr predictions before the onset. In conclusion, XGBoost performed better than other single-modality models, and multi-modality models with late-level fusion performed better than other models. The algorithm developed in this study can improve the identification and early prediction of ARDS and assist clinicians.

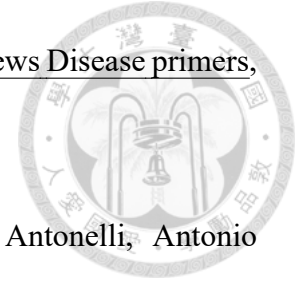




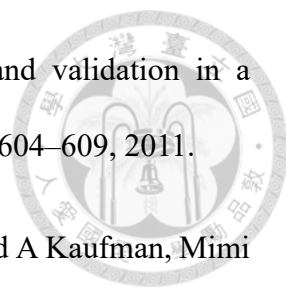
References

- [1] Shih-Cheng Huang, Anuj Pareek, Saeed Seyyedi, Imon Banerjee, and Matthew P Lungren. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. NPJ Digital Medicine, 3(1):1–9, 2020.
- [2] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4700–4708, 2017.
- [3] National Health Service (NHS). Intensive care. <https://www.nhs.uk/conditions/intensive-care/>. Accessed: 2021-11-08.
- [4] Tim Wenham and Alison Pittard. Intensive care unit environment. Continuing Education in Anaesthesia Critical Care & Pain, 9(6):178–183, 12 2009.
- [5] National Health Service (NHS). Acute respiratory distress syndrome. <https://www.nhs.uk/conditions/acute-respiratory-distress-syndrome/>. Accessed: 2021-11-08.
- [6] Michael A Matthay, Rachel L Zemans, Guy A Zimmerman, Yaseen M Arabi, Jeremy R Beitler, Alain Mercat, Margaret Herridge, Adrienne G Randolph, and Car-

olyn S Calfee. Acute respiratory distress syndrome. Nature reviews Disease primers, 5(1):1–22, 2019.



- [7] Niall D Ferguson, Eddy Fan, Luigi Camporota, Massimo Antonelli, Antonio Anzueto, Richard Beale, Laurent Brochard, Roy Brower, Andrés Esteban, Luciano Gattinoni, et al. The berlin definition of ARDS: an expanded rationale, justification, and supplementary material. Intensive Care Medicine, 38(10):1573–1582, 2012.
- [8] Vito Fanelli, Aikaterini Vlachou, Shirin Ghannadian, Umberto Simonetti, Arthur S Slutsky, and Haibo Zhang. Acute respiratory distress syndrome: new definition, current and future therapeutic options. Journal of Thoracic Disease, 5(3):326, 2013.
- [9] Giacomo Bellani, John G Laffey, Tàì Pham, Eddy Fan, Laurent Brochard, Andres Esteban, Luciano Gattinoni, Frank Van Haren, Anders Larsson, Daniel F McAuley, et al. Epidemiology, patterns of care, and mortality for patients with acute respiratory distress syndrome in intensive care units in 50 countries. JAMA, 315(8):788–800, 2016.
- [10] Giacomo Bellani, Tàì Pham, and John G Laffey. Missed or delayed diagnosis of ARDS: a common and serious problem. Intensive Care Medicine, 46(6):1180–1183, 2020.
- [11] George D Magoulas and Andriana Prentza. Machine learning in medical applications. In Advanced Course on Artificial Intelligence, pages 300–307. Springer, 1999.
- [12] Akbar K Waljee and Peter DR Higgins. Machine learning in medicine: a primer for physicians. American Journal of Gastroenterology, 105(6):1224–1226, 2010.
- [13] Cesar Trillo-Alvarez, Rodrigo Cartin-Ceba, Daryl J Kor, Marija Kojicic, Rahul Kashyap, Sweta Thakur, Lokendra Thakur, V Herasevich, M Malinchoc, and

- 
- O Gajic. Acute lung injury prediction score: derivation and validation in a population-based sample. European Respiratory Journal, 37(3):604–609, 2011.
- [14] Graciela J Soto, Daryl J Kor, Pauline K Park, Peter C Hou, David A Kaufman, Mimi Kim, Hemang Yadav, Nicholas Teman, Michael Hsu, Tatyana Shvilkina, et al. Lung injury prediction score in hospitalized patients at risk of acute respiratory distress syndrome. Critical Care Medicine, 44(12):2182, 2016.
- [15] Zachary M Bauman, Marika Y Gassner, Megan A Coughlin, Meredith Mahan, and Jill Watras. Lung injury prediction score is useful in predicting acute respiratory distress syndrome and mortality in surgical critical care patients. Critical Care Research and Practice, 2015, 2015.
- [16] Paul E Pepe, Ronald G Thomas, Marie Anne Stager, Leonard D Hudson, and C James Carrico. Early prediction of the adult respiratory distress syndrome by a simple scoring method. Annals of Emergency Medicine, 12(12):749–755, 1983.
- [17] Mengyuan Liang, Miao He, Jian Tang, Xinliang He, Zhijun Liu, Siwei Feng, Ping Chen, Hui Li, Yu'e Xue, Tao Bai, et al. Novel risk scoring system for predicting acute respiratory distress syndrome among hospitalized patients with coronavirus disease 2019 in wuhan, china. BMC infectious Diseases, 20(1):1–10, 2020.
- [18] Pengcheng Yang, Taihu Wu, Ming Yu, Feng Chen, Chunchen Wang, Jing Yuan, Jiameng Xu, and Guang Zhang. A new method for identifying the acute respiratory distress syndrome disease based on noninvasive physiological parameters. PLOS ONE, 15(2):e0226962, 2020.
- [19] Sidney Le, Emily Pellegrini, Abigail Green-Saxena, Charlotte Summers, Jana Hoffman, Jacob Calvert, and Ritankar Das. Supervised machine learning for the early

prediction of acute respiratory distress syndrome (ARDS). Journal of Critical Care, 60:96–102, 2020.



[20] Xian-Fei Ding, Jin-Bo Li, Huo-Yan Liang, Zong-Yu Wang, Ting-Ting Jiao, Zhuang Liu, Liang Yi, Wei-Shuai Bian, Shu-Peng Wang, Xi Zhu, et al. Predictive model for acute respiratory distress syndrome events in ICU patients in China using machine learning algorithms: a secondary analysis of a cohort study. Journal of Translational Medicine, 17(1):1–10, 2019.

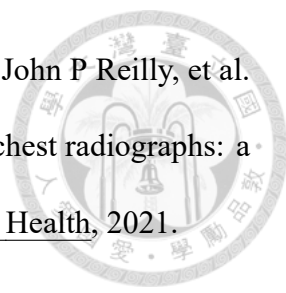
[21] Daniel Zeiberg, Tejas Prahlad, Brahmajee K Nallamothu, Theodore J Iwashyna, Jenna Wiens, and Michael W Sjoding. Machine learning for patient risk stratification for acute respiratory distress syndrome. PLOS ONE, 14(3):e0214465, 2019.

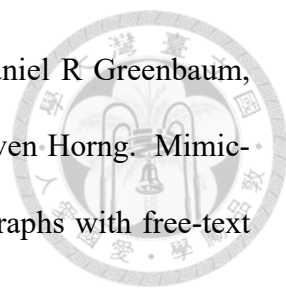
[22] Yang Fei, Kun Gao, and Wei-qin Li. Prediction and evaluation of the severity of acute respiratory distress syndrome following severe acute pancreatitis using an artificial neural network algorithm model. HPB, 21(7):891–897, 2019.

[23] Lakshya Singhal, Yash Garg, Philip Yang, Azade Tabaie, A Ian Wong, Akram Mohammed, Lokesh Chinthala, Dipen Kadaria, Amik Sodhi, Andre L Holder, et al. eARDS: A multi-center validation of an interpretable machine learning algorithm of early onset acute respiratory distress syndrome (ARDS) among critically ill adults with COVID-19. PLOS ONE, 16(9):e0257056, 2021.

[24] Narathip Reamaroon, Michael W Sjoding, Jonathan Gryak, Brian D Athey, Kayvan Najarian, and Harm Derksen. Automated detection of acute respiratory distress syndrome from chest X-rays using directionality measure and deep learning features. Computers in Biology and Medicine, 134:104463, 2021.

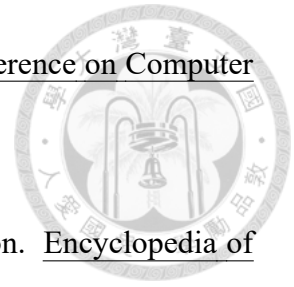
[25] Michael W Sjoding, Daniel Taylor, Jonathan Motyka, Elizabeth Lee, Ivan Co, Dru

- 
- Claar, Jakob I McSparron, Sardar Ansari, Meeta Prasad Kerlin, John P Reilly, et al. Deep learning to detect acute respiratory distress syndrome on chest radiographs: a retrospective study with external validation. The Lancet Digital Health, 2021.
- [26] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In ICML, 2011.
- [27] Jabeen Summaira, Xi Li, Amin Muhammad Shoib, Songyuan Li, and Jabbar Abdul. Recent advances and trends in multimodal deep learning: A review. arXiv preprint arXiv:2105.11087, 2021.
- [28] Aleksei Tiulpin, Stefan Klein, Sita MA Bierma-Zeinstra, Jérôme Thevenot, Esa Rahtu, Joyce van Meurs, Edwin HG Oei, and Simo Saarakkala. Multimodal machine learning-based knee osteoarthritis progression prediction from plain radiographs and clinical data. Scientific Reports, 9(1):1–11, 2019.
- [29] Shih-Cheng Huang, Anuj Pareek, Roham Zamanian, Imon Banerjee, and Matthew P Lungren. Multimodal fusion with deep neural networks for leveraging ct imaging and electronic health record: a case-study in pulmonary embolism detection. Scientific Reports, 10(1):1–9, 2020.
- [30] Jordan Yap, William Yolland, and Philipp Tschandl. Multimodal skin lesion classification using deep learning. Experimental Dermatology, 27(11):1261–1267, 2018.
- [31] Adam Yala, Constance Lehman, Tal Schuster, Tally Portnoi, and Regina Barzilay. A deep learning mammography-based model for improved breast cancer risk prediction. Radiology, 292(1):60–66, 2019.
- [32] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV, 2021.

- 
- [33] Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. Scientific Data, 6(1):1–8, 2019.
- [34] Accountability Act. Health insurance portability and accountability act of 1996. Public law, 104:191, 1996.
- [35] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. Journal of Big Data, 6(1):1–48, 2019.
- [36] Dina Elreedy and Amir F. Atiya. A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance. Information Sciences, 505:32–64, 2019.
- [37] Yan-Yan Song and LU Ying. Decision tree methods: applications for classification and prediction. Shanghai Archives of Psychiatry, 27(2):130, 2015.
- [38] Gérard Biau and Erwan Scornet. A random forest guided tour. TEST, 25(2):197–227, 2016.
- [39] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, pages 785–794, 2016.
- [40] Jürgen Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61:85–117, 2015.
- [41] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet:

A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255. IEEE, 2009.

- [42] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. Encyclopedia of Database Systems, 5:532–538, 2009.







Appendix A — Cross-validation Results

The appendix reports the full results of the repeated k -fold cross-validation. Table A.1 to Table A.4 shows the AUCs for different types of onset identification, 12-hr, 24-hr, and 48-hr predictive models. The following tables will only show the best ones for the numerical and image-only models in this research. That is, XGBoost with weight adjustment and DenseNet169.

Table A.1: Results of the repeated k -fold cross-validation for onset identification.

Model	(1 st)	Fold1	Fold2	Fold3	Fold4	Fold5
Single-modality model						
Numerical-only	AUC	0.8369	0.8395	0.8381	0.8425	0.8404
Image-only	AUC	0.7449	0.7659	0.7599	0.7369	0.7648
Multi-modality model						
Early-level fusion	AUC	0.8151	0.7960	0.7998	0.8010	0.8070
Late-level fusion	AUC	0.8537	0.8596	0.8649	0.8704	0.8459
	(2 nd)	Fold1	Fold2	Fold3	Fold4	Fold5
Single-modality model						
Numerical-only	AUC	0.8318	0.8327	0.8535	0.8373	0.8453
Image-only	AUC	0.7811	0.7303	0.7840	0.7627	0.7702
Multi-modality model						
Early-level fusion	AUC	0.7813	0.7995	0.7919	0.7996	0.7900
Late-level fusion	AUC	0.8647	0.8487	0.8517	0.8608	0.8607



Table A.2: Results of the repeated k -fold cross-validation for 12-hr prediction.

Model	(1 st)	Fold1	Fold2	Fold3	Fold4	Fold5
Single-modality model						
Numerical-only	AUC	0.8123	0.8088	0.8279	0.8175	0.8005
Image-only	AUC	0.7660	0.7481	0.7912	0.7860	0.7337
Multi-modality model						
Early-level fusion	AUC	0.8132	0.7706	0.7741	0.8017	0.8061
Late-level fusion	AUC	0.8771	0.8879	0.8837	0.8508	0.8508
	(2 nd)	Fold1	Fold2	Fold3	Fold4	Fold5
Single-modality model						
Numerical-only	AUC	0.8068	0.8141	0.8086	0.8313	0.7966
Image-only	AUC	0.7850	0.7848	0.8182	0.7751	0.7768
Multi-modality model						
Early-level fusion	AUC	0.7995	0.7842	0.8095	0.8009	0.8017
Late-level fusion	AUC	0.8586	0.8755	0.8670	0.8612	0.8477

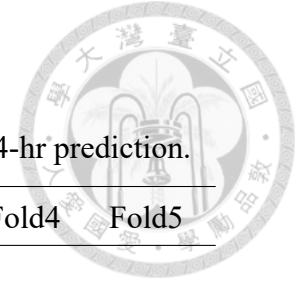


Table A.3: Results of the repeated k -fold cross-validation for 24-hr prediction.

Model	(1 st)	Fold1	Fold2	Fold3	Fold4	Fold5
Single-modality model						
Numerical-only	AUC	0.7994	0.8003	0.8215	0.8073	0.8265
Image-only	AUC	0.7127	0.7039	0.7385	0.7285	0.7656
Multi-modality model						
Early-level fusion	AUC	0.8378	0.8232	0.8213	0.8032	0.7994
Late-level fusion	AUC	0.8023	0.8165	0.8653	0.8265	0.8570
		Fold6	Fold7	Fold8	Fold9	Fold10
Single-modality model						
Numerical-only	AUC	0.8361	0.8061	0.8278	0.8173	0.8169
Image-only	AUC	0.7219	0.7656	0.7098	0.7690	0.7089
Multi-modality model						
Early-level fusion	AUC	0.8019	0.7640	0.8403	0.7952	0.8157
Late-level fusion	AUC	0.8837	0.8420	0.8611	0.8524	0.8119
Model	(2 nd)	Fold1	Fold2	Fold3	Fold4	Fold5
Single-modality model						
Numerical-only	AUC	0.7965	0.7961	0.8164	0.8098	0.8282
Image-only	AUC	0.7440	0.7581	0.7519	0.6968	0.7168
Multi-modality model						
Early-level fusion	AUC	0.7848	0.8232	0.7573	0.8103	0.8040
Late-level fusion	AUC	0.8111	0.8424	0.8611	0.8407	0.8369
		Fold6	Fold7	Fold8	Fold9	Fold10
Single-modality model						
Numerical-only	AUC	0.8144	0.8119	0.8065	0.8082	0.8115
Image-only	AUC	0.7139	0.7164	0.6993	0.7469	0.7552
Multi-modality model						
Early-level fusion	AUC	0.8261	0.8224	0.8582	0.8207	0.8011
Late-level fusion	AUC	0.8803	0.8536	0.8449	0.8119	0.8324

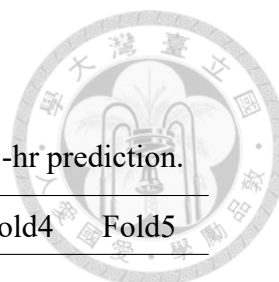


Table A.4: Results of the repeated k -fold cross-validation for 48-hr prediction.

Model	(1 st)	Fold1	Fold2	Fold3	Fold4	Fold5
Single-modality model						
Numerical-only	AUC	0.7301	0.7450	0.7419	0.7705	0.7605
Image-only	AUC	0.7618	0.7667	0.7643	0.7903	0.7258
Multi-modality model						
Early-level fusion	AUC	0.8027	0.7792	0.8151	0.7407	0.7940
Late-level fusion	AUC	0.8226	0.8151	0.8449	0.8350	0.8201
		Fold6	Fold7	Fold8	Fold9	Fold10
Single-modality model						
Numerical-only	AUC	0.7804	0.7804	0.7320	0.7829	0.7519
Image-only	AUC	0.7481	0.7047	0.7122	0.7047	0.7475
Multi-modality model						
Early-level fusion	AUC	0.7928	0.7742	0.8052	0.7754	0.7680
Late-level fusion	AUC	0.8635	0.8189	0.8375	0.8089	0.8300
Model	(2 nd)	Fold1	Fold2	Fold3	Fold4	Fold5
Single-modality model						
Numerical-only	AUC	0.7475	0.7630	0.7612	0.7494	0.7457
Image-only	AUC	0.7878	0.7345	0.7568	0.7593	0.7270
Multi-modality model						
Early-level fusion	AUC	0.7841	0.7742	0.8201	0.7866	0.8387
Late-level fusion	AUC	0.8226	0.8635	0.8337	0.8213	0.8027
		Fold6	Fold7	Fold8	Fold9	Fold10
Single-modality model						
Numerical-only	AUC	0.7568	0.7841	0.7562	0.7661	0.7730
Image-only	AUC	0.7134	0.7680	0.7593	0.7494	0.7146
Multi-modality model						
Early-level fusion	AUC	0.7792	0.7643	0.7891	0.7841	0.8313
Late-level fusion	AUC	0.8710	0.8275	0.7928	0.8697	0.8263