Leveraging Advanced Image Analysis and Learning Using Privileged Information for Clinical Decision Support

by

Zijun Gao

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Bioinformatics)
in The University of Michigan
2023

Doctoral Committee:

Professor Kayvan Najarian, Chair
Associate Professor Alan Boyle
Assistant Professor Jonathan Gryak, City University of New York
Assistant Professor Jie Liu
Associate Professor Michael W. Sjoding
Associate Professor Qiong Yang

Zijun Gao

zijung@umich.edu

ORCID iD: 0000-0003-4651-0156

To my mother, Rongqing Zhou,

and my sister, Chencen Wang

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and appreciation to the following individuals who have played a significant role in my graduate study and the completion of this thesis:

First and foremost, I am incredibly grateful to my advisor, Dr. Kayvan Najarian, for his invaluable guidance, support, and encouragement throughout this journey. His expertise, patience, and commitment to excellence have been instrumental in shaping the work presented in this thesis. I am truly fortunate to have had such an exceptional mentor, and his profound impact on my academic and personal growth cannot be overstated.

I would also like to extend my sincere thanks to my thesis committee. Dr. Michael Sjoding has provided invaluable clinical perspectives, and his guidance and domain knowledge were indispensable in shaping the direction of my research. Dr. Jie Liu and Dr. Qiong Yang supervised my rotations and provided immense support in helping me navigate the early stages of graduate study. I am grateful to Dr. Jonathan Gryak for his countless support and valuable advice regarding my research. His professionalism and dedication to excellence have made a significant impact on my growth as a researcher. I also want to thank Dr. Alan Boyle for his insightful feedback and constant presence throughout my graduate journey, starting from the preliminary exam. The expertise and scholarly contributions of my thesis committee have greatly enhanced the quality of this work.

I would like to express my gratitude to the members of the Biomedical and Clinical

Informatics Lab and the Department of Computational Medicine and Bioinformatics for their constant support. In particular, I am deeply appreciative of Emily Wittrup for her help and the collaborative and enriching experiences we have shared. Working with her has been an absolute pleasure. I would also like to thank Julia Eussen and Dr. Margit Burmeister for their guidance and support. Additionally, I want to express my gratitude to Hojae Lee for her support as a peer, as our discussions have been intellectually stimulating and have enriched my perspective in numerous ways.

To my dear friends Shuze Wang, Heming Yao, Lu Lu, Zhi Li, and Nanxiang Zhao, who have been by my side throughout my time at the University of Michigan, I want to express my deepest gratitude. The friendship we have shared and the countless memories we have created together have made this journey all the more meaningful. The support, laughter, and camaraderie we have experienced have been invaluable, and I am grateful for their unwavering presence in my life. I would also like to extend my appreciation to Zhongyu Yang for the time we have spent together, growing and thriving. Additionally, I want to express my gratitude to Keijing Li and Yan Tang, my remote friends who share the same love for literature. Our conversations and shared passion have brought immense joy to my life.

Finally, I would like to express my deepest thanks and love to my family. Especially, I want to thank my mom, Rongqing Zhou, for the warmth and trust she always provides, for her courage, wisdom, and sincerity, and for the humble and deep understanding she gives me. I am always encouraged and comforted by her unwavering support.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

LIST OF ABBREVIATIONS

AI Artificial Intelligence

ARDS Acute Respiratory Distress Syndrome

AST Aspartate Aminotransferase

AUROC Area Under the Receiver Operating Characteristic Curve

AUPRC Area Under the Precision-Recall Curve

AVI Audio Video Interleave

BYOL Boost Your Own Latent

CAD Coronary Artery Disease

CNN Convolutional Neural Networks

CXR Chest X-ray

DICOM Digital Imaging and Communications in Medicine

DINO distillation with no labels

DDP Denoising Diffusion Probabilistic

DL Deep Learning

DISR Double Input Symmetrical Relevance

EHR Electronic Health Records

GBDT Gradient Boost Decision Tree

GD Generalized Dice loss

GCS Glasgow Coma Scale

GPU Graphics Processing Unit

IoU Intersection over Union

LASSO Least Absolute Shrinkage and Selection Operator

LCA Left Coronary Arteries

LUPI Learning Using Privileged Information

LUPAPI Learning Using Partially Available Privileged Information

LR Logistic Regression

ML Machine Learning

MRMR Minimum Redundancy Maximum Relevance

MV Mechanical Ventilation

NN Neural Networks

nPC Number of Principle Components

nFea Number of Features

PCA Principle Component Analysis

PCIs Percutaneous Coronary Interventions

PEEP Positive End-Expiratory Pressure

PLR Privileged Logistic Regression

QCA Quantitative Coronary Angiography

RCA Right Coronary Arteries

RF Random Forest

RBF Radial Basis Functions

SE structuring element

sg Stop Gradient

SNN Shallow Neural Network

SSL Self-supervised Learning

SVM Support Vector Machine

TRAM Transfer and Marginalized

ViT Vision Transformer

XCA X-ray Coronary Angiography

ABSTRACT

The field of medical artificial intelligence (AI) has seen significant advancements with the availability of digitalized medical data. Machine learning (ML) and deep learning (DL) models have been developed to leverage these datasets, aiding in clinical decision-making and the delivery of evidence-based care. Medical imaging has particularly benefited from ML and DL algorithms, with successful applications in image classification, segmentation, and detection. Similarly, electronic health records (EHR) data analysis has facilitated risk prediction, disease phenotyping, and treatment outcome assessment. However, the field still faces practical challenges, such as the heterogeneity and missingness of data in EHR, and the scarcity of gold-standard labels in medical imaging.

This thesis aims to address these challenges and contribute to the field of medical AI by developing innovative techniques and methodologies. It focuses on building generalizable and explainable AI models with limited labeled data and leveraging privileged information for clinical decision support. To achieve these objectives, strategies such as bias mitigation, data augmentation, regularization, multi-source data integration, and ensembles are proposed or employed. Furthermore, the thesis investigates the utilization of privileged information, which refers to data or information accessible only during training and not during inference. In the medical field, privileged information is prevalent due to multiple data sources and the varying availability of modalities and variations in medical care protocols. By leveraging privileged infor-

mation, novel algorithms under the Learning Using Privileged Information (LUPI) paradigm and the Learning Using Partially Available Privileged Information (LUPAPI) paradigm are proposed to enhance model performance and address issues of data missingness in multimodal settings. These algorithms allow models to make predictions without relying on specific data during inference, while still benefiting from its inclusion.

The thesis consists of several chapters that tackle specific tasks and challenges. Chapter 2 presents an automated pipeline for segmenting coronary arteries in X-ray coronary angiography images. Chapter 3 focuses on the diagnosis of acute respiratory distress syndrome (ARDS) using EHR data, while Chapter 4 extends this work by applying the LUPI paradigm and LUPAPI paradigm. Chapter 5 addresses the challenge of label uncertainty in ARDS detection using chest X-ray images. Finally, Chapter 6 concludes the thesis by summarizing the key findings and discussing future directions.

In conclusion, this thesis contributes to the advancement of medical AI by developing techniques for robust and explainable decision-support models with limited labeled data. It also explores the utilization of privileged information to enhance model performance. The proposed methodologies have the potential to improve patient care and outcomes, paving the way for further research and development in the field of medical AI.

CHAPTER I

Introduction

1.1 Background

The field of medical Artificial Intelligence (AI) has undergone a transformative shift with the proliferation of digitalized medical data, including Electronic Health Records (EHR), waveforms, medical notes, and various imaging modalities [3, 4]. This rich availability of data has paved the way for the development of Machine Learning (ML) and Deep Learning (DL) models, which have demonstrated the capability to leverage large-scale medical datasets for various purposes, such as extracting meaningful patterns, identifying risk factors, predicting disease progression, and supporting personalized treatment strategies [5]. The utilization of these models in healthcare holds promise for augmenting clinical decision-making and ultimately improving patient outcomes with evidence-based care [6, 7].

In the realm of medical imaging, ML and DL algorithms have demonstrated remarkable capabilities in tasks such as image classification, segmentation, and detection [8]. Convolutional Neural Networks (CNN) have particularly demonstrated success in interpreting radiological images, such as X-rays [9], computed tomography scans [10, 11], and magnetic resonance imaging [12]. Additionally, deep learning models have shown promise in detecting abnormalities, aiding in early diagnosis, and guiding treatment planning in domains like histopathology [13, 14], gastroenterology

[15, 16], ophthalmology [17, 18], and dermatology [19, 20].

ML and DL techniques have not only been extensively applied to medical imaging but also to the analysis of EHR data [21, 22, 23, 24]. These approaches have shown great potential in extracting valuable insights from both structured and unstructured clinical data, enabling risk prediction [25], disease phenotyping [26], adverse event prediction [27], and treatment outcome assessment [28]. By integrating patient-specific information, such as demographics, medical history, laboratory results, and medication records, these models contribute to personalized medicine and support clinical decision-making [29, 30]. Furthermore, various methods have been employed for outcome prediction, risk prediction, and prognostic analysis using medical signals and waveform data, such as electroencephalography [31] and electrocardiography [32].

Despite the significant achievements and potential advantages of utilizing medical data for AI development, the field of medical AI still faces practical challenges [33, 34]. Some of the challenges relate to accountability, fairness, and ethical concerns, while others arise from the unique characteristics of medical data, which require careful consideration and specialized approaches to achieve robust and reliable AI solutions.

Take EHR as an example, the data derived from EHR exhibits heterogeneity [35] due to its inclusion of categorical, numerical, and hierarchical variables. This inherent heterogeneity presents challenges in the processing and analysis of the data. Furthermore, the temporal aspect of EHR data adds complexity [36], as variables can change over time, and some variables may have uneven recording frequency. Missing data is also prevalent in EHR, with instances of entry-level missing or patient-level missing. Importantly, the missing data in EHR, as well as in other medical modalities, do not occur randomly. Instead, there is a concept known as informative presence [37], which refers to the notion that the presence or absence of patient data can provide valuable information about their health condition. This highlights that the missing data, along with its timing, frequency, and rate in a patient's longitudinal

data, can carry meaningful insights [38]. Consequently, EHR data becomes a sparse data source that requires careful handling and consideration in analysis and modeling [39].

Medical images, on the other hand, also need specific processing techniques as they differ significantly from natural images [40]. Apart from some obvious discrepancies [41] in image format such as channels, sizes, and dimensions, the biggest difference lies in the labeling. Traditional machine learning and deep learning methods for image applications rely on large amounts of labeled data, often obtained on websites and social media with crowdsourcing annotations from non-experts [42]. However, this approach is impractical for medical images due to privacy concerns, institutional policies, and the need for expert understanding. Concepts and abnormalities in medical images are complex and specific, requiring clinical expertise for accurate annotation. Even when medical experts perform manual labeling, limited dataset sizes, labor-intensive labeling processes, and time constraints can hinder the availability of labeled data. Furthermore, there could be a notable presence of inter-observer variability among experts [43], with label uncertainty stemming from inherent case ambiguity, limitations within diagnostic criteria, or imperfections in the labeling process, leading to label noise that significantly impacts the performance of deep learning models in machine learning and computer vision applications [44]. Consequently, the scarcity of gold standard labels becomes a significant obstacle for supervised learning projects in medical imaging analysis.

1.2 Objective

In light of these remarkable advancements and the remaining challenges, this thesis aims to contribute to the advancement of medical AI by focusing on the development of innovative techniques and methodologies. The following subsections outline the specific objectives, which focus on enhancing the robustness, generalizability, and

interpretability of AI models in the context of limited labeled data and privileged information.

1.2.1 Building Generalizable and Explainable Medical AI Models with Limited Labeled Data

Improving the robustness, generalizability, and explainability of decision-making models is paramount in the healthcare field. A trustworthy decision-support system is essential for providing reliable guidance to healthcare professionals and improving the accountability of the models in use.

To achieve these objectives within the constraints of limited labeled data, this thesis adopts several key strategies. Firstly, addressing bias originating from imbalanced training sets is crucial for enhancing model generalizability. Therefore, various sampling techniques are employed to mitigate this bias, ensuring the representation of samples from different classes is more balanced. Data augmentation techniques are also utilized to increase the representation of underrepresented classes when necessary. To ensure fair evaluation, the testing set remains untouched during this process. Secondly, to prevent overfitting and improve generalizability, regularization techniques are incorporated during model construction. By introducing regularization terms in machine learning algorithms or applying techniques such as dropout in network training, the models are encouraged to generalize well on unseen data, leading to improved robustness and performance. Thirdly, the thesis explores the use of multi-source data whenever possible. Training and validating models on datasets collected from diverse sources contribute to improved generalizability and performance. When single-source data is unavoidable, knowledge transfer techniques are applied, leveraging knowledge from related tasks or domains to enhance model performance. In addition, rigorous evaluation processes are adopted, employing cross-validation in hyper-parameter selection and testing over different seeds to assess model perfor-

mance and generalizability. This comprehensive evaluation approach ensures reliable and valid results, providing a better understanding of model capabilities and limitations. Furthermore, robust modeling approaches such as tree ensembles are employed to enhance generalization and improve performance in the presence of limited data. Finally, in terms of explainability, feature selection methods are utilized to identify the most relevant variables for accurate diagnosis. This reduces the dimensionality of the data, improves model efficiency, and enhances interpretability. Moreover, there is a strong emphasis on developing explainable models that provide insights into the decision-making process, which allows clinicians to understand and trust the model's predictions, facilitating their acceptance and integration into clinical practice.

1.2.2 Leveraging Privileged Information in Clinical Decision Support Models

Privileged information refers to a specific set of data or information that is accessible during the model training stage but not during inference [45]. In the medical field, privileged information is prevalent for various reasons. One of the primary reasons is the existence of multiple sources of medical data, some of which are readily available while others are more challenging to acquire. To illustrate, consider a patient presenting symptoms of shortness of breath. While lab test results and vital waveform data are typically obtained shortly after hospitalization, an accurate diagnosis of potential lung disease often necessitates the analysis of Chest X-ray (CXR) images, which may not be immediately accessible. In a retrospective study aimed at developing a decision support model, the CXR data can be regarded as privileged information, while the lab results and waveform data are regarded as base information. Additionally, the availability of different modalities or medical care protocols can vary over time or among different healthcare institutions. Therefore, certain modalities may be unavailable when deploying the model, making them privileged information. Further-

more, comprehensive descriptions from alternative data modalities, images or videos generated through advanced protocols, as well as information about the annotators, can be deemed privileged information if they are not accessible during the inference stage but offer valuable insights into the clinical problem at hand.

By incorporating privileged information during the training process, models can be developed to make predictions without relying on this specific data during inference, while still benefiting from its inclusion. This paradigm is known as Learning Using Privileged Information (LUPI) [46]. Here, the concept of learning is analogous to how students benefit from the guidance of a teacher in a classroom setting, as machines and algorithms can enhance their performance by leveraging this additional information that is only available during the training stage.

In this thesis, novel algorithms will be proposed under the LUPI paradigm to address various challenges. Firstly, these algorithms aim to transfer knowledge from the privileged domain to the base domain, thereby enhancing the predictive capabilities of the models on the base domain. Secondly, the issue of label noise will be tackled by leveraging labels provided by multiple annotators as privileged information. Additionally, an extension of LUPI called Learning Using Partially Available Privileged Information (LUPAPI) [47] will be considered in algorithm development, allowing the models to handle data missing in the privileged domain and benefiting the fusion of data from multiple modalities. These advancements will contribute to the development of more robust and accurate models for handling real-world challenges in medical AI.

1.3 Dissertation Outline

In pursuit of the aforementioned objectives, this thesis presents a comprehensive framework that encompasses various novel approaches and techniques. The following sections provide an outline of the dissertation, highlighting the major contributions

of each chapter.

In Chapter II, an automated pipeline was proposed for segmenting coronary arteries in X-ray Coronary Angiography (XCA) images. As the crucial step in computer-aided Coronary Artery Disease (CAD) diagnosis and treatment planning, correct delineation of the coronary artery is challenging in XCA due to the low signal-to-noise ratio and confounding background structures. Additionally, the limited availability of labeled images and the imbalance between foreground and background data points further complicate the segmentation task. To address these challenges, a novel ensemble framework was developed, leveraging deep learning and filter-based features to construct ensemble models and treating the segmentation as a pixel-wise classification task. Moreover, hybrid under-sampling techniques were integrated into the pipeline to create a balanced and representative training dataset based on domain knowledge, avoiding possible model bias. The proposed method outperformed common deep convolutional neural networks in most evaluation metrics while yielding more consistent results. Such a method can be used to facilitate the assessment of stenosis and has the potential to improve the quality of care in patients with CAD.

Starting from Chapter III, this thesis places a significant emphasis on the diagnosis of Acute Respiratory Distress Syndrome (ARDS), a life-threatening respiratory failure that is frequently underestimated in critically ill patients. As a rapidly progressing disease, late diagnosis of ARDS adversely affects patient outcomes. Despite this, timely interventions are hard to achieve as the diagnosis relies upon a frontline provider to obtain Chest X-ray. The work in this chapter aimed at ARDS detection with EHR, a data modality that is routinely available compared to chest radiology, to expedite the diagnosis of ARDS. In this chapter, we consider the unique characteristics of EHR data, such as temporality and missing data, during the preprocessing stage. Then, various dimensionality reduction techniques and classifiers are employed and evaluated for model development. Additionally, to identify ARDS prior to inva-

sive mechanical ventilation (MV) or when MV information is not available, models excluding related variables were also developed. The results demonstrated that machine learning models utilizing EHR data can accurately detect ARDS before the use of mechanical ventilation, showing promise in enhancing the early detection of ARDS and improving patient outcomes. Furthermore, relevant features in EHR data that might be associated with ARDS development are identified.

In Chapter IV, the work in Chapter III has been extended by the advanced learning paradigm, LUPI. Specifically, since CXR and MV-related information are not always available at the point of decision-making, they are assigned to the privileged domain, while the routinely available EHR is put at the base domain. Then, a novel model called privileged logistic regression (PLR) is developed under the LUPI paradigm for ARDS detection. The objective function of PLR is carefully designed to incorporate data from the privileged domain and facilitate knowledge transfer between the privileged and base domains. Regularization techniques are employed in both the privileged and base domains to enhance the generalizability of the model. Asymptotic analysis is conducted, establishing sufficient conditions under which the inclusion of privileged information improves the convergence rate of the proposed model. In addition, the model can be naturally extended to the case of LUPAPI, where missing privileged data is handled. Results for ARDS detection show that PLR models achieve better classification performances than logistic regression models trained solely on the base domain EHR data, even when privileged information is partially available. Additionally, PLR models performed better than support vector machines and shallow neural networks adapted to the LUPI paradigm. As the proposed models are effective, easy to interpret, and highly explainable, they are ideal for other clinical applications where privileged information is at least partially available.

In Chapter V, the focus remains on ARDS detection, with a shift to the analysis of CXR images. Although CXR is considered the gold standard for ARDS diagnosis, its

interpretation can be difficult due to non-specific radiological features, uncertainty in disease staging, and inter-rater variability among clinical experts. To overcome these challenges, a novel approach was introduced to utilize the label uncertainty derived from multiple annotators as privileged information, aiming to improve the detection of ARDS in CXR images. By incorporating the Transfer and Marginalized network and employing effective knowledge transfer mechanisms, the detection model performed superior to various baselines and obtained impressive detection performance after removing equivocal testing cases. These findings highlight the effectiveness of the proposed methods in addressing label uncertainty and label noise in CXRs for ARDS detection, with potential for use in other medical imaging domains that encounter similar challenges.

Chapter VI concludes the research presented in this dissertation by summarizing the key findings and contributions. Alternative approaches and future directions for performing the tasks are discussed, taking into account the evolving landscape of the field. Furthermore, the chapter highlights the potential opportunities that lie ahead in the field of medical AI, emphasizing the importance of ongoing research and development.

CHAPTER II

Vessel Segmentation for X-ray Coronary Angiography using Ensemble Methods with Deep Learning and Filter-based Features

2.1 Introduction

As the most common type of heart disease, CAD is the leading cause of death globally, resulting in a yearly loss of 17.9 million lives with 330 million being affected [48, 49]. CAD is primarily caused by the narrowing of the lumen in coronary arteries due to plaque build-up [50]. This narrowing, or stenosis, restricts the blood flow to cardiac muscle, depriving the heart of oxygen and nutrient supplements, ultimately leading to myocardial ischemia and infarction [51].

X-ray coronary angiography (XCA) is the gold standard for CAD diagnosis [52]. By releasing dye into the coronary vessels and inspecting its flow through the vessel structure via 2D projections, XCA helps clinicians locate potential stenoses, visually measure their severity, and determine the appropriate interventional therapies [53].

Visual stenosis assessment, however, is often unreliable: it tends to overestimate severe blockages while underestimating mild ones [54, 55] and has high intra- and inter-observer variability [56, 57]. To evaluate the lumen diameter more objectively, Quantitative Coronary Angiography (QCA) was introduced [58] to offer a

semi-automatic analysis of XCA. QCA analysis involves frame selection, vessel segmentation, stenosis positioning, and quantitative measurement [59, 60]. The vessel segmentation step of QCA is a prerequisite for calculating the percentage of arterial stenosis. Moreover, the correct delineation of coronary arteries plays an important role in center-line extraction, which is used for 3D reconstruction of blood vessels [61], vessel tracking [62], and cardiac dynamics assessment [63].

Due to the nature of XCA images, segmenting vessels accurately is challenging. First, XCA images usually are of low resolution, have low signal-to-noise ratios, and exhibit low contrast between the vessel structure and background region [64, 65, 66]. Second, the presence of irrelevant structures such as the catheter, the diaphragm, and the spine is confounding and leads to non-uniform illumination within the images [67]. Third, the various angles from which the 3D vessel structure is projected to form 2D XCA images create twisted and overlapping vessels, making the segmentation even more challenging [68].

To overcome these difficulties and aid in the quantitative diagnosis of CAD, efforts have been made to develop both supervised and unsupervised methods for automatic coronary vessel segmentation.

Unsupervised methods can be primarily categorized as tracking-based, model-based, or filter-based [69]. Tracking-based methods [70] choose seed points on the edges and the center-lines of vessels, then take a small step in the direction of the vessel to look for the vessel edges or the center-lines nearby. When new edges are found, an estimate of vessel direction is made to take the next step in this search direction. Model-based methods [71, 72, 73, 74, 75, 76], use deformable models or region growing to evolve the segmentation towards the vessel-background boundaries based on the forces and constraints defined by energy functions. They may also apply growing conditions defined by similarity functions together with a threshold parameter. Both tracking-based and model-based methods require initial seeds for

segmentation and are therefore sensitive to initialization. Although they tend to maintain good segmentation continuity for the vessel tree structure, they may fail to handle confounding elements in the background that are adjacent to the vessels. Filter-based methods [64, 77] apply a variety of filters for non-uniform background intensity balancing, irrelevant structures suppression, noise reduction, and vessel enhancement. The filtered images can be later processed with thresholding techniques for segmentation mask generation. Due to their ease of implementation and their ability to mitigate illumination problems, filter-based methods have also been employed extensively as preprocessing steps in both supervised and unsupervised methods for automated coronary vessel segmentation [76, 78, 79, 80, 81, 82, 83]. However, they are usually insufficient for use on their own, as they are sensitive to background structures and may not perform well on vessel junctions and bifurcations [84].

Coronary artery segmentation using supervised methods can be considered as a pixel-wise classification problem, with most current methods utilizing Neural Networks. Cervantes-Sanchez et al. [78] trained a multilayer perceptron with XCA images enhanced by Gaussian-matched filters and Gabor filters. Nasr-Esfahani et al. [79] presented a multi-stage model where CNNs extract local, contextual, and edge-based information that were then combined via a final fully connected layer. Recently, DL approaches have gained popularity in segmenting both major arteries and full artery trees from XCA images. Samuel and Veeramalai [85] proposed a Vessel Specific Skip Chain Network by adding two vessel-specific layers to the VGG-16 network [86]. Jo et al. [80] developed a two-stage CNN specifically for left anterior descending artery segmentation, where the first stage located candidate areas of interest and the second stage generated the segmentation mask. Iyer et al. [87] designed an angiographic processing network that learned how to preprocess the XCA images with the most suitable filters for local contrast enhancement. The preprocessed images were then fed into DeeplabV3+ [88] for segmentation. Shi et al. [89] developed

a generative adversarial network for major branch segmentation with a U-Net generator and a pyramid-structure discriminator, reporting improved connectivity for the segmented mask. Yang et al. [90] replaced the backbone of U-Net with an ImageNet pre-trained ResNet [91], InceptionResNetv2 [92], or DenseNet [93] for main branch segmentation. Fan et al. [94] modified U-Net so that the proposed structure can receive both the target and registered background images before dye release as inputs for generating segmentation masks. The network structure proposed by [95] receives multi-channel inputs by adding a 3D convolution layer to the U-Net encoder, exploiting the temporal information using three consecutive frames from angiographic image sequences to produce a segmentation mask for the middle frame. Zhu et al. [96] applied the Pyramid Scene Parsing Network, a network proposed by [97], for coronary vessel segmentation. They took advantage of the network structure to incorporate features from multiple scales by pyramid pooling and used transfer learning to avoid overfitting on a small training set. Supervised methods for coronary artery segmentation may focus on the major coronary arteries for which clinicians would be more concerned, instead of the entire arterial tree. Network-based supervised methods have a number of drawbacks, including overfitting when the training set is small, weaker interpretability as compared to unsupervised filter-based methods, and an inability to ensure connectivity within their prediction masks. However, supervised methods require less manual input and are more robust in discriminating background structures such as the catheter and spine than unsupervised methods.

In this chapter, a novel ensemble framework for coronary artery segmentation is proposed that employs GBDT [98] and Deep Forest classifiers [99]. The GBDT is a popular ML technique that combines weak decision tree learners for loss function minimization. When constructing a GBDT model, a series of trees is built wherein each new weak decision tree attempts to correct errors from the previous stage. The Deep Forest classifier, on the other hand, is a deep ensemble model that uses non-

differentiable modules to form DL structures. Unlike deep neural networks, it does not apply back-propagation for training, but it still uses multiple layers (with cascade structures) for processing and applies in-model feature transformation. However, GBDT boosts the performance of weak learners gradually in a sequential and additive way, while in Deep Forest, random forests composed of decision trees are considered as a subroutine stacked by layers, with layer outputs feeding into another layer to create depth. Though both GBDT and Deep Forest have not been applied to XCA image segmentation, they have been recently employed in medical image analysis in different image modalities [100, 101, 102, 103, 104, 105]. The ensemble methods produced promising results in retinal vessel segmentation [106, 107, 108, 109, 110, 111] and have not been, as far as we know, applied on coronary artery segmentation yet. In this study, 16 DL features obtained from the last layer of the Dense-Net-backbone U-Net decoder were combined with 21 multi-scale statistics on responses to a diverse range of filters to construct a 37-dimensional feature vector for each pixel in the input XCA image for training coronary artery segmentation models with GBDT and Deep Forest.

The proposed work takes advantage of both decades of classical computer vision research along with contemporary ML and DL techniques by employing a diverse set of reliable, well-established, hand-crafted features together with features from a deep structure for ensemble model training. Additional novelties come from the extraction of multiple statistics from the scale-space profile of a filter response and the adoption of a deep ensemble model on coronary artery segmentation.

The remainder of the chapter is organized as follows. Section 2.2 introduces the datasets used in this study and describes the methods employed for feature extraction, the under-sampling of imbalanced training classes, and model training, testing, and evaluation. Section 2.3 reports the effect of under-sampling on the training set, the performance of models constructed using different classifiers, and the analysis

of feature importance, while Section 2.4 provides interpretations of the results and describes limitations and future directions of the current work.

2.2 Methods

In the following subsections, the datasets used for the study, the feature extraction techniques (using filter-based and DL methods) and the under-sampling methods employed are first introduced, after which the training of ensemble classifiers is explained. A schematic diagram of the proposed method for coronary artery segmentation is depicted in Figure 2.1.

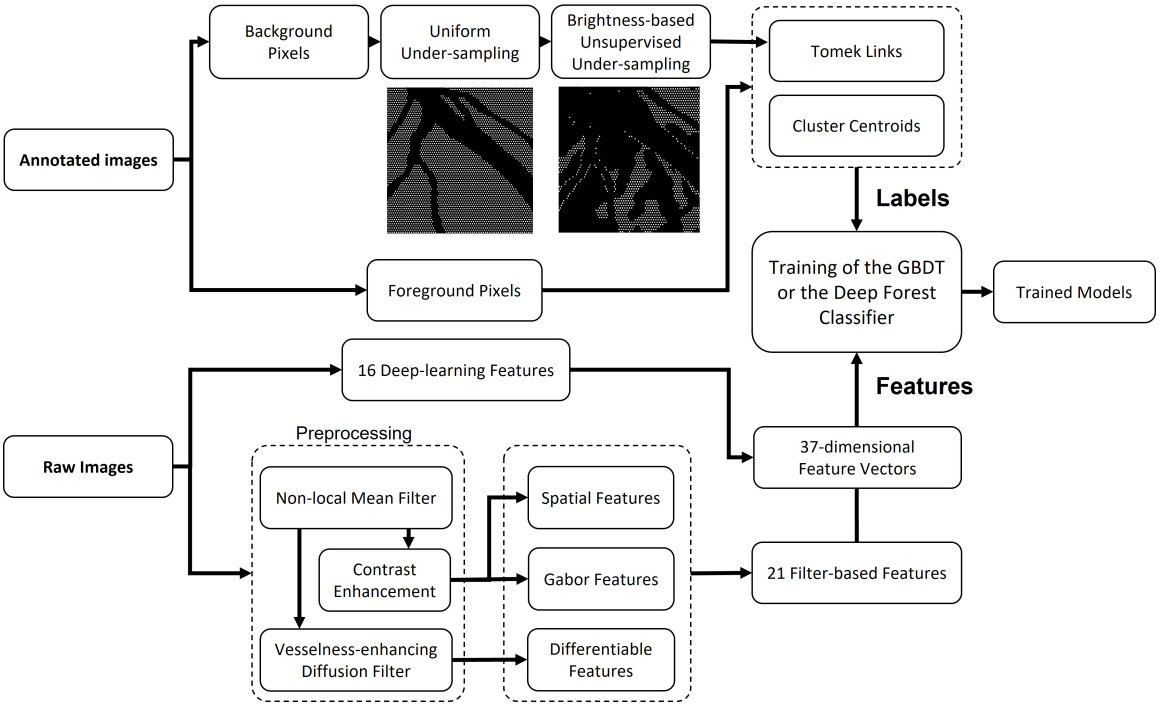


Figure 2.1: Schematic diagram of the training pipeline. Lower panel: features were extracted from raw images with deep-learning and filter-based methods. Upper panel: under-sampling methods were performed to balance the number of positive (vessel) and negative (background) training classes.

Table 2.1: Dataset Summary

Dataset Code	Total	LCA	RCA	With Acquisition Angles
1-17	98	68	30	0
1-19	8	4	4	8
1-AVI	10	0	10	0
2	14	8	6	14
Total Count	130	80	50	22

2.2.1 Dataset

The study was conducted with de-identified angiograms from two sources: "Dataset 1", collected from the University of Michigan Hospital, Ann Arbor, MI, and "Dataset 2", collected from a hospital in the United Kingdom. Both datasets are comprised of patients suspected of having coronary artery disease who underwent invasive coronary angiography. Dataset 1 contains three subsets: 1-17, 1-19, and 1-AVI. Angiograms within subsets 1-17 and 1-19 were collected in 2017 and 2019, respectively, and stored in Digital Imaging and Communications in Medicine (DICOM) format, while angiograms in 1-AVI were stored in Audio Video Interleave (AVI) format.

Patients were excluded if any of the following occurred: incomplete injection of contrast dye, percutaneous coronary intervention, an implanted pacemaker or cardioverter defibrillator, or the presence of artificial objects other than the dye injection catheter. Ultimately, 130 angiogram sequences from 130 patients were included in this study. 80 of them visualize the Left Coronary Arteries (LCA) while the remaining 50 depict the right coronary artery (RCA). The number of frames in each sequence ranges from 43 to 150, with an average of 86 frames per sequence.

As the entire vascular tree is not always visible in all frames, frames were selected from XCA videos according to three criteria: (1) the selected frame contains the full injection of contrast agent; (2) there is minimal cardiac motion between adjacent frames; and (3) the full coronary artery is visualized in the frame. The frames are gray-scale images with a resolution of 512×512 pixels. Segmentation masks used for

training and testing were first generated manually using Adobe Photoshop CS and later validated by experienced cardiologists. Catheter diameter size (measured by pixel number and referred to as "Cath" in later sections) was recorded along with the annotation. Only those vessels whose diameters were greater than or equal to 0.75× Cath were annotated. The mean diameter of Cath was 8.0 (± 1.2) pixels. A summary of the dataset information is listed in Table 2.1. For angiograms whose acquisition angles are available, the information is depicted in Figure 2.2.

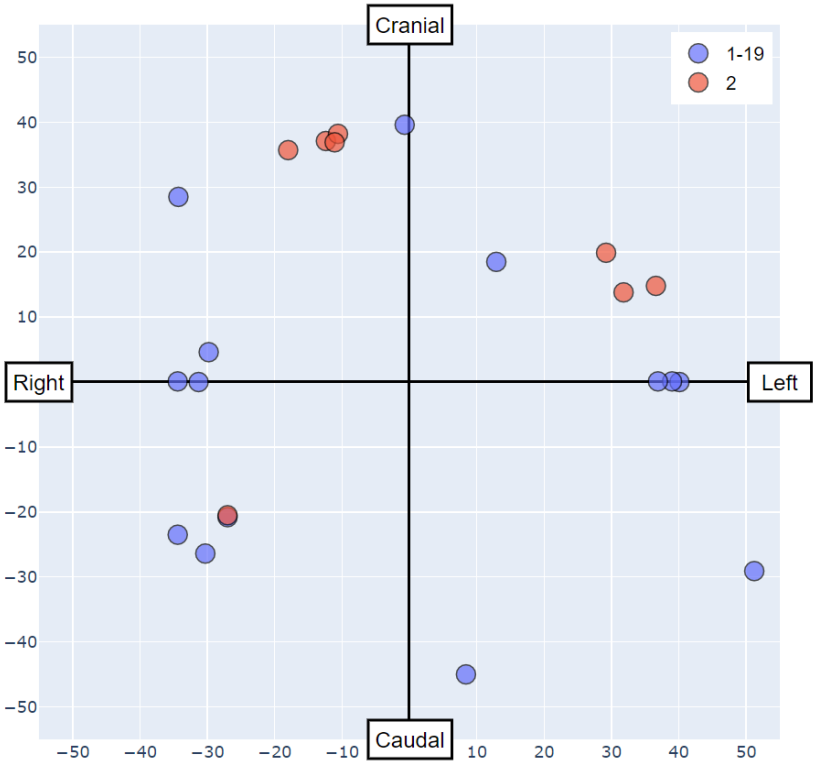


Figure 2.2: Image acquisition angles for Dataset 1-19 and Dataset 2. "Caudal" and "Cranial" refer to the caudal and cranial angulation of the X-ray.

2.2.2 Feature Extraction with Filters

2.2.2.1 Scale-space theory and the Z-profile of filter responses

Structures that a filter can extract from an XCA image depend on the scale of observation. A single scale is not always sufficient for capturing vessel structures of varying sizes. The scale-space theory [112] provides a framework for automatic scale selection in image filtering by applying multiple scales for image representation and summarizing filter responses across scales [113]. Based on this theory, the Z-profile of pixel-wise filter responses is constructed with four summary statistics: the maximum, mean, variance, and interquartile range of multi-scale responses. For example, Figure 2.3 illustrates the filter response of Frangi filters [114] over ten different scales and the Z-profile thus obtained.

Scale ranges ($\lambda \in \Lambda$) were selected to be relative to the physical constraints of coronary arteries, ranging from 0.66×Cath to 6.33×Cath on a logarithmic scale.

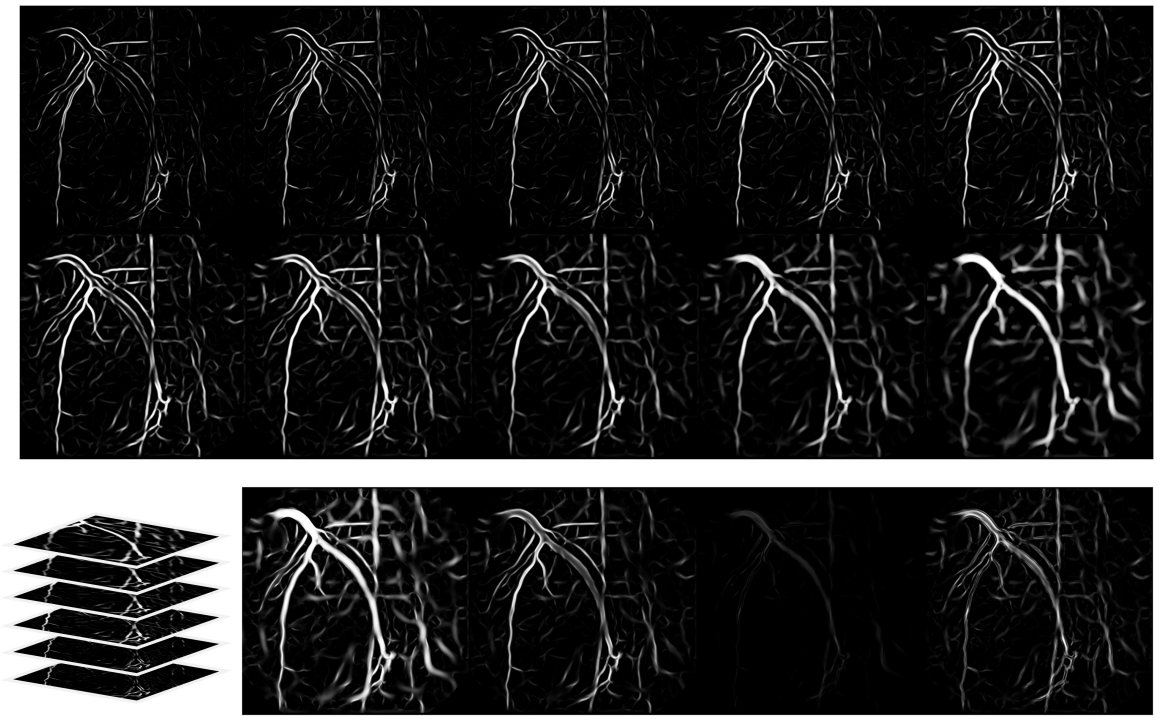


Figure 2.3: Multi-scale filtering with Frangi filter (upper panel) and the corresponding Z-profile of the max, mean, variance, and interquartile range of filtering responses (lower panel, from left to right).

2.2.2.2 Preprocessing

XCAs are often of poor quality due to image noise. To enhance the visibility of vessels within the frame, standard computer vision and image processing techniques were used to construct the preprocessing pipeline. First, metadata from the DICOM files was used to exclude the border regions (pixels outside of the imaging window) from analysis and to obtain catheter size information. Then, the filter scales were set as described in Section 2.2.2.1. In cases where the aforementioned metadata was unavailable, the mean Cath value (Section 2.2.1) was used. After that, a non-local mean filter [115] was applied for noise reduction. Following this step, contrast adjustment (Appendix A) using Top-bottom-hat filtering [116] was employed to reconstruct the image.

Let $I : \Omega \to \mathbb{R}$ be the $H \times W$ image with pixel coordinates given by $(x, y) \in \Omega = \{1, 2, \ldots, H\} \times \{1, 2, \ldots, W\}$ and SE_λ denote the structuring element (SE) with scale λ, then the Top-hat filtered image I_{top} is defined as the maximum of the differences between an input image I and its SE opening over $\lambda \in \Lambda$, while the Bottom-hat output I_{bottom} is defined as the maximum of the differences between SE closing with I over $\lambda \in \Lambda$, that is,

$$I_{top} = \max_{\lambda \in \Lambda} \left(I - (I \circ SE_\lambda)\right) \text{ and}$$
$$I_{bottom} = \max_{\lambda \in \Lambda} \left((I \bullet SE_\lambda) - I\right),$$

with $(\circ)$ and $(\bullet)$ denoting morphological opening and closing respectively (see Appendix A for the definitions of these operations.) The Top-bottom-hat enhanced image is then generated as

$$I_{enhanced} = I + m \cdot I_{top} - n \cdot I_{bottom}, \tag{2.1}$$

where m and n are the strengths of Top-hat and Bottom-hat transformations.

Vesselness-enhancing diffusion filtering [117] (see Appendix A for details) was also performed on the denoised images to enhance vascular structures as utilized in Section 2.2.2.3.

2.2.2.3 Filter-based Feature Extraction

A number of common vessel enhancement and segmentation filters were used to extract features that can be categorized into differentiable, spatial, and Gabor features.

In terms of differentiable features, the Z-profile of the Frangi filter [114], the Z-profile of the matched filter [118], the Gaussian-filter-smoothed Z-profile of the gradient magnitude, and the vessel confidence measure [119] were extracted, resulting in a total of 13 features. For spatial features, the granular decomposition of the top-bottom-hat image using the method given in [120] was obtained, producing a Z-profile with 4 features. The Gabor features were extracted from the Z-profile of the Gabor filter [64] responses on the complement of contrast enhancement output image.

2.2.3 Feature Extraction with DL

The DL networks described in this section were implemented in Python 3.7 using PyTorch 1.10 and the segmentation model package [121]. Each network was trained on a single NVIDIA Tesla V100 Graphics Processing Unit (GPU).

2.2.3.1 Data Partitioning for Model Construction

The dataset containing 130 XCA images was split into training, validation, and test sets in a 3:1:1 ratio. The partitions were stratified to ensure different subsets had approximately the same percentage of samples of RCA and LCA angiograms from different sources.

2.2.3.2 Network Structure

The network structures adopted in this study are common DL models developed for medical image segmentation, such as U-Net, DeepLabV3+, Inception ResNet-v2-backbone U-Net, ResNet101-backbone U-Net, and DenseNet121-backbone U-Net. The latter three were first applied for main branch segmentation in XCA images in [90] and achieved the best performance in terms of F1 score thus far. These structures were employed in this work for major branch segmentation with modified training logic to serve as a comparison to the ensemble models, and to obtain DL features for ensemble model training. For the DeepLabV3+ model, we used a DeepLabV3 encoder as mentioned in [88] and adapted the ImageNet pre-trained ResNet101[91] for dense features extraction in the encoder [122]. Details on model parameters can be found in Appendix C. For the modified U-Net models, the encoder and bottleneck sections of the U-Net were replaced with ImageNet pre-trained ResNet [91], InceptionResNet-v2 [92], or DenseNet [93], respectively, except for their average pooling layers and the fully connecting layers at the end. Skip connections were retained between the encoder and the decoder at different spatial resolutions.

2.2.3.3 Data Processing and Training Setting

Gray-scale XCA images were first preprocessed with 2-D min/max normalization. To increase the diversity of the training samples, data augmentation was employed at each training iteration before feeding data into the networks. Specifically, XCA images were randomly augmented by affine transformations (-20° to 20° rotation, 0-10% of image size translation shift on horizontal and vertical axes, or 0-10% zoom) with a probability of 0.7. The same augmentations were also applied to the corresponding ground-truth masks. The network was trained using a default-setting Adam optimizer with an initial learning rate of 10^{-3} and a mini-batch size of 8 images for up to 100 epochs. An early-stop mechanism was triggered if validation loss did not improve for

15 epochs.

2.2.3.4 Loss Function

To take into consideration class imbalance and class importance, the deep-learning model was trained using Generalized Dice loss (GD) [123]

$$GD = 1 - \frac{2\sum_{c=1}^{t} w_c \sum_{p=1}^{n} G_{cp} M_{cp}}{\sum_{c=1}^{t} w_c \sum_{p=1}^{n} (G_{cp} + M_{cp})},$$

where $w_c = (\sum_{p=1}^{n}(G_{cp}/t))^2 + \epsilon)^{-1}$ is the weight for class c, t is the total class number, p is the pixel location, and n the total number of pixels in an image. G_{cp} and M_{cp} are the ground truth image pixel and predicted mask pixel values from class c respectively.

2.2.3.5 Testing and Model Evaluation

For each network structure, the model that achieved the lowest validation loss was applied to the test set. The final layer of the network output was passed through an element-wise sigmoid activation function to generate the probability of each pixel belonging to either the vessel region or the background region. The same post-processing described in Section 2.2.6.3 was applied to generate the final binary segmentation masks. The quality of the generated masks was evaluated by precision, sensitivity, specificity, F1 score, Intersection over Union (IoU), and Area Under the Receiver

Operating Characteristic Curve (AUROC). The first five are defined as

$$
\begin{aligned}
Precision &= \frac{TP}{TP+FP} \\
Sensitivity\,(Recall) &= \frac{TP}{TP+FN} \\
Specificity &= \frac{TN}{TN+FP} \\
F1\ Score &= \frac{2 \times Sensitivity \times Precision}{Sensitivity + Precision} \\
IoU &= \frac{TP}{TP+FP+FN},
\end{aligned}
$$

where TP, TN, FP, and FN are the pixel counts of true positives, true negatives, false positives, and false negatives, respectively.

2.2.3.6 Deep Feature Extraction

The network structure that achieved the best test performance with respect to F1 score and AUROC was adopted for deep feature extraction. After normalization, the XCA images of dimension $1 \times 512 \times 512$ were fed into the network and the activation maps of the final decoder layer were extracted as the deep features with a dimension of $16 \times 512 \times 512$.

2.2.4 Feature Standardization and Training Samples

Table 2.2: Feature Domains and Types

Feature Domain	Feature Type	Feature Number
Differentiable features	Z-profile of Frangi filters	4
	Z-profile of matched filters	4
	Gaussian-filter-smoothed Z-profile of the gradient magnitude	4
	Vessel confidence measure	1
Spatial features	Z-profile of granular decomposition	4
Gabor features	Z-profile of Gabor features	4
Deep-learning features	Activation maps of the final decoder layer	16

After feature extraction as described in Sections 2.2.2 and 2.2.3, all features (Table 2.2) were concatenated (Figure 2.4). Each feature map outputted from a filter or a DL layer is standardized individually by subtracting its mean pixel value and dividing the result by the standard deviation. For each pixel p_i within an XCA image, a 37-dimensional feature vector $\boldsymbol{X_i}$ was extracted that yielded a training sample ($\boldsymbol{X_i}$, y_i) where the label y_i is 1 if the pixel is part of a vessel in the annotated XCA image and 0 otherwise. The 130 XCA images resulted a in total of $512 \times 512 \times 130 = 34078720$ samples before border removal and any under-sampling.

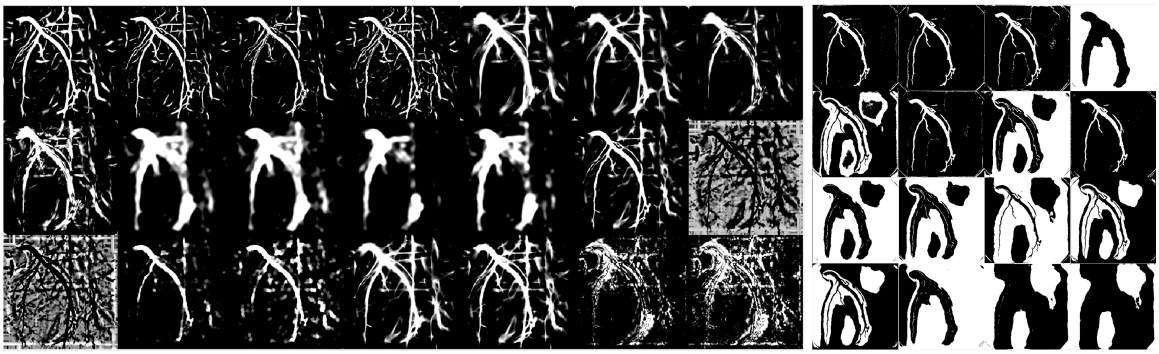


Figure 2.4: An example of the 37-dimensional feature maps extracted by filter-based methods (left panel; $3 \times 7 = 21$ features) and DL method (right panel; $4 \times 4 = 16$ features).

2.2.5 Under-sampling of Non-vessel Pixels

If vessel and background pixels are denoted as positive and negative classes respectively, the samples are highly imbalanced as the minority (positive) class only comprises an average of $5.58(\pm 1.99)\%$ in the XCA images. Moreover, the features extracted between neighboring pixels are highly correlated. Given these two facts, hybrid under-sampling of pixels at the image level was performed to (1) avoid overfitting due to redundant information across pixels; (2) ensure that the classifiers do not ignore the minority class; and (3) reduce training time.

2.2.5.1 Uniform and Unsupervised Under-sampling

The majority class consists of the background, non-vessel areas in the XCA images. Pixels from the majority class were first uniformly under-sampled via a mask (Figure 2.5) such that the 8-neighborhood of each sampled pixel was not sampled. Then, an intensity-based unsupervised under-sampling method was employed to further reduce the major class based on the contrast enhancement output (Figure 2.6). To affect this under-sampling, the histogram of the pixel intensity of the contrast-enhanced image was created with 256 bins, after which the discrete pdf (probability density function) was obtained and assigned to a one-dimensional median filter. The smoothed pdf was compared with the original one to identify the over-saturation peak generated by contrast enhancement. When these over-saturated pixels were excluded, the median value of pixel intensities was calculated as the binarization threshold for bright pixel removal (Figure 2.6, right panel). This step removes pixels that clearly belong to the background based solely on their intensity.

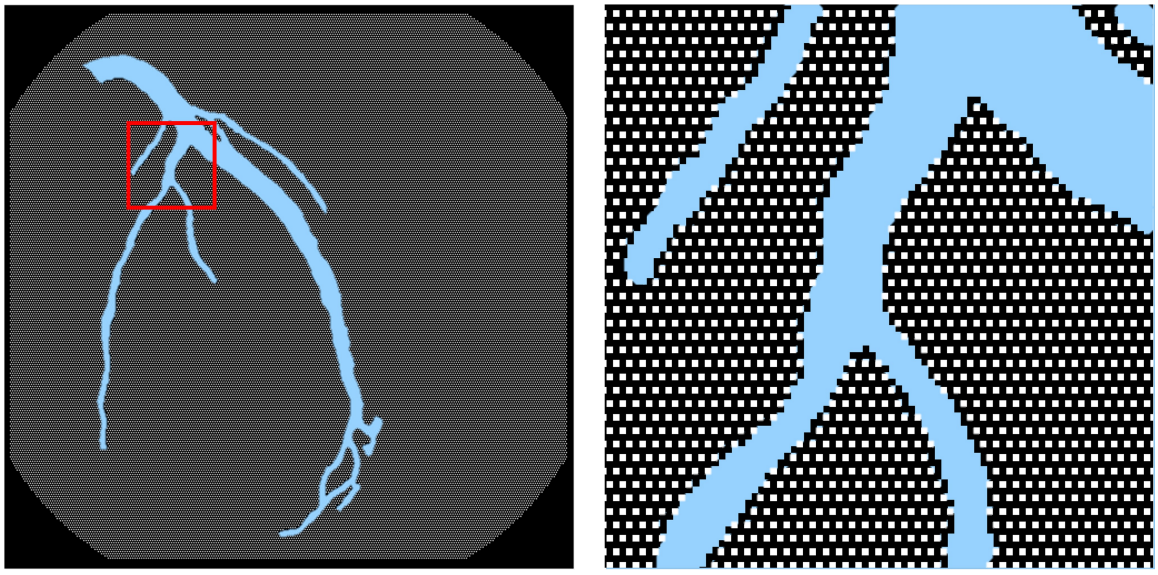


Figure 2.5: A uniform under-sampling mask of the majority class. Pixels from the minority class colored light blue are not involved. The mask image on the right is a magnified version of the selected red box on the left. Pixels colored white were retained after the mask was applied to the major class of the target image.

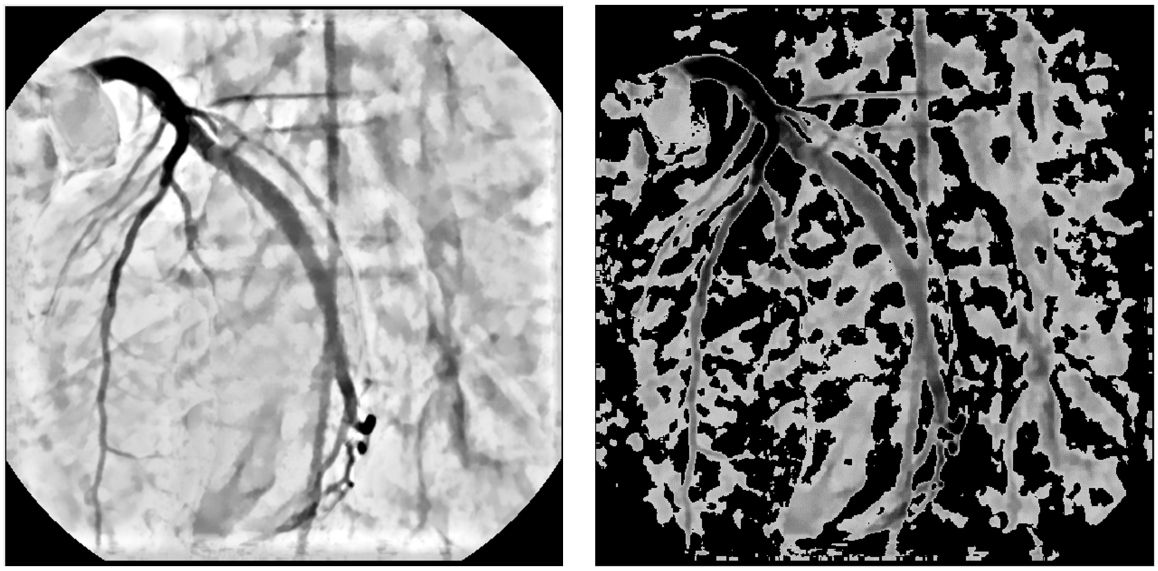


Figure 2.6: Unsupervised under-sampling. Left: the output image from contrast enhancement; Right: Pixels retained after under-sampling based on intensity.

2.2.5.2 Supervised Under-sampling

Supervised under-sampling was achieved using Tomek Links and Cluster Centroid for both the positive and the negative classes. Tomek Links [124] are defined as pairs of pixels from opposite classes that are the nearest neighbors of each other. As removing overlapping pixels between classes yields more well-defined boundaries for the classifiers [125], the Tomek Links of both classes were removed so that the minimally distanced nearest-neighbor pairs of vessel pixels belong to the same class. Figure 2.7 illustrates a Tomek Link under-sampling. Following the Tomek Links, the Cluster Centroid [126] under-sampling method was employed. This method first applies clustering algorithms such as k-nearest neighbors to generate cluster centroids and then uses these centroids to replace the original sample points. This method can reduce the number of pixels within each class to a fixed number, e.g., 4000, which is much smaller than the sample number in the original image. The Python implementation [127] of Tomek Links and Cluster Centroid methods were employed.

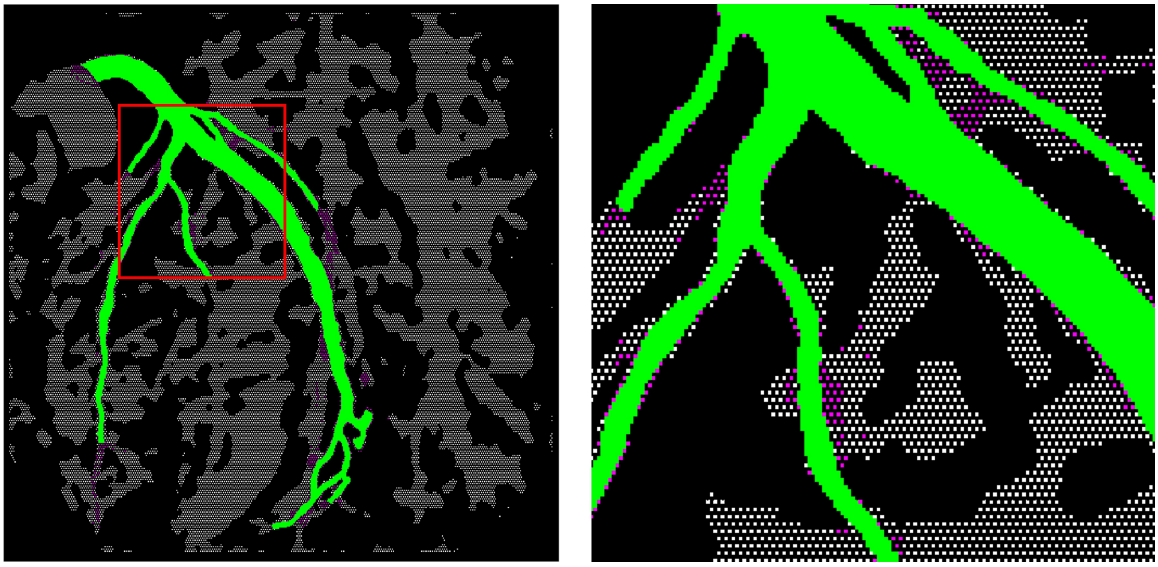


Figure 2.7: Tomek Links under-sampling. The image on the right is a magnified version of the red box on the left. Magenta: pixels removed by Tomek Link; Green: positive class, vessel pixels; White: negative class, background pixels.

2.2.6 Ensemble Learning for Coronary Vessel Segmentation

Two ensemble methods, Gradient Boost Decision Tree (GBDT) [98] and Deep Forest [99] were trained on the extracted 37-dimensional feature vectors with samples obtained by under-sampling. To explore how Tomek Links and Cluster Centroid affected the segmentation performance, models with and without these two under-sampling methods were constructed for comparison.

2.2.6.1 Data Partitioning for Model Construction

The dataset was split into training and test sets with a 4:1 ratio. The test set images are exactly the same as the DL test set mentioned in Section 2.2.3.1 and were not utilized until the test stage. All the partitions (including the cross-validation partitions in Section 2.2.6.2) were stratified to ensure different sets have approximately the same percentage of LCA and Right Coronary Arteries (RCA) images, and pixels from the same image were not split across sets. Moreover, standardization was applied to the training set with parameters being saved to transform the test set, and this

was also true for cross-validation.

2.2.6.2 Training and Testing Strategy

For the Deep Forest model, default hyper-parameters settings were used since hyper-parameters had minimal effect on the model performance [99], while for the GBDT model, hyper-parameters were tuned using 4-fold cross-validation, where fold compositions were changed under cyclic permutation with a 3:1 ratio. These hyper-parameters and their respective ranges included the learning rate ($\{0.01, 0.05, 0.1\}$), the number of boosting stages ($\{100, 500, 1000, 2000\}$), and the maximum depth of the individual regression estimators ($\{3, 5, 10, 20\}$). For other hyper-parameters, default values in the Sklearn package were applied. For example, the loss function to be optimized was the deviance and the split quality of trees were measured by the mean squared error with an improvement score by Friedman (Friedman MSE) [98]. Finally, the hyper-parameter combination that achieved the highest mean AUROC score in cross-validation was used to train the final GBDT model on the training set and applied to the test set for evaluation.

2.2.6.3 Post-processing and Model Evaluation

Features extracted from test images were passed into the trained ensemble models to generate masks. The masks were then binarized using Otsu's method [128] and post-processed by (1) removing border regions, (2) adding back unsupervised background masks, and (3) removing artifacts whose areas values were less than 50. The same evaluation metrics listed in Section 2.2.3.5 were calculated to evaluate the quality of the predicted masks across different models. The metrics were calculated image-wise and produced by calculating the mean and standard deviation over all tested images.

2.2.6.4 Feature Importance

The permutation importance [129] of the GBDT models was computed on the holdout test set for feature evaluation. The importance of a feature was calculated as the decrease of Friedman MSE (mentioned in Section 2.2.6.2) evaluated on the test set when permuting the feature column 10 times.

2.3 Results

2.3.1 Under-sampling

The counts of positive and negative pixels retained for model training after different under-sampling steps are listed in Table 2.3. Before under-sampling, the positive class comprised only 5.56% of the total samples after the border regions were removed (see Section 2.2.2.2). The uniform under-sampling selected 24.13% of the background pixels in the original labeled image, followed by the unsupervised under-sampling that further kept 45.98% of the negative pixel samples. These two steps resulted in a 34.78% share of vessel pixels in the overall samples and 4,000,602 samples of vessel and background pixels for model training. Tomek Links only altered this percentage level slightly, while Cluster Centroid completely balanced the positive and negative classes, reducing the training samples to 832,000 in total.

Table 2.3: Pixel Totals Resulting from Different Under-sampling Methods

Pixel Count After the Method	**% of Positive Class**	**% of Negative Class (The Majority Class)**	**Total Count (Training Samples)**
Original Image (Exclude border)	5.559%	94.441%	25002241
Uniform Under-Sampling	19.611%	80.489%	7087635
Unsupervised Under-sampling	34.663%	65.338%	4010003
Tomek Links	34.677%	65.323%	4000602
Cluster Centroid	50.000%	50.000%	832000

2.3.2 Performance Comparison of Deep-learning Models and Ensemble Models on the Test Set

Table 2.4: A Comparison of Model Performance

	Precision	**Sensitivity**	**Specificity**	**F1 Score**	**AUROC**	**IoU**
U-Net	0.867±0.073	0.810±0.122	0.993±0.005	0.831±0.082	0.902±0.060	0.719±0.115
DeepLabV3+	0.862±0.082	**0.828±0.096**	0.992±0.006	0.838±0.081	**0.909±0.047**	0.726±0.088
Inception-ResNet-v2 U-Net	**0.904±0.072**	0.805±0.133	**0.995±0.004**	0.842±0.089	0.900±0.066	0.737±0.120
DenseNet121 U-Net	0.891±0.053	0.824±0.145	0.994±0.004	**0.845±0.091**	**0.909±0.071**	**0.741±0.117**
Resnet101 U-Net	0.865±0.072	0.819±0.122	0.992±0.005	0.832±0.068	0.906±0.060	0.718±0.095
	Precision	**Sensitivity**	**Specificity**	**F1 Score**	**AUROC**	**IoU**
Unsupervised with Deep Forest	0.832±0.073	**0.911±0.096**	0.990±0.005	0.863±0.048	**0.95±0.046**	0.762±0.071
Tomek Links with Deep Forest	**0.884±0.061**	0.867±0.124	**0.993±0.004**	**0.867±0.066**	0.930±0.061	**0.770±0.094**
Cluster Centroid with Deep Forest	0.868±0.067	0.873±0.107	**0.993±0.004**	0.864±0.062	0.933±0.053	0.765±0.087
	Precision	**Sensitivity**	**Specificity**	**F1 Score**	**AUROC**	**IoU**
Unsupervised with GBDT	0.864±0.066	0.894±0.104	0.992±0.004	0.872±0.051	0.943±0.051	0.776±0.075
Tomek Links with GBDT	**0.885±0.06**	0.872±0.123	**0.994±0.004**	0.870±0.066	0.933±0.060	0.775±0.094
Cluster Centroid with GBDT	0.857±0.073	**0.902±0.084**	0.992±0.004	**0.874±0.048**	**0.947±0.041**	**0.779±0.072**
	Precision	**Sensitivity**	**Specificity**	**F1 Score**	**AUROC**	**IoU**
DenseNet121 U-Net [90]	0.858±0.071	0.873±0.109	0.991±0.006	0.858±0.057	0.926±0.068	0.755±0.082

Table 2.4 lists the performances of five DL models, six ensemble models and the state-of-the-art DL model on the test set in terms of their precision, sensitivity, specificity, F1 score, AUROC, and IoU. For the ensemble models, "Unsupervised" indicates that the uniform and unsupervised under-samplings were applied on the training samples while "Tomek Links" means that all the Tomek Links were also removed from the sample pixels. Moreover, "Cluster Centroid" indicates that the Cluster Centroid under-sampling method was further applied for reducing sample numbers.

For the DL models, Inception-ResNet-v2 U-Net achieved the highest precision and specificity, DeepLabV3+ obtained the best sensitivity and AUROC, while DenseNet121

U-Net had the best F1 score, AUROC, and IoU scores. For ensemble learning models that use Deep Forest as the classifier, the samples after Tomek Links under-sampling method yielded the highest score in precision, specificity, F1 score, and IoU, while the samples after unsupervised under-sampling gave the best test performance in terms of sensitivity and AUROC, achieving the highest AUROC of 0.95 among all models tested. For ensemble learning models that use GBDT as the classifier, the samples after Tomek Links under-sampling had the best precision and specificity. Moreover, the further application of the Cluster Centroid under-sampling method generated the highest sensitivity, F1 score, AUROC, and IoU scores. It also yielded the best F1 score and IoU of all models tested.

The state-of-the-art method[90] achieved higher sensitivity, F1 score, AUROC, and IoU than the five DL models. However, it can not beat the proposed ensemble models, which generally performed better than the DL methods in all metrics except for specificity. Moreover, GBDT classifiers performed better than the Deep Forest with higher mean value and lower standard deviation in terms of F1 score and IoU regardless of the under-sampling method employed.

2.3.3 The Permutation Feature Importance of GBDT models

Figure 2.8 illustrates the permutation importance of the 37 features used in GBDT model training when different under-sampling methods were applied. Statistics (maximum, mean, variance, and interquartile range) of the Z-profile are denoted as Z-max, Z-mean, Z-var, and Z-interq in the plot. For the GBDT trained with the unsupervised under-sampling method, Z-mean and Z-var of the gradient magnitude, Z-var of the granular decomposition, and Z-max of the Gabor filter are important filter-based features, as well as the deep features 1, 9, and 16. For the GBDT trained with Tomek Links under-sampling method, the Z-max and Z-mean of the Frangi filter, the Z-interq of gradient magnitude, and deep feature 4 have high permutation impor-

tance. In terms of the GBDT and Cluster Centroid combination, deep features 4, 9, and 15 show more permutation importance over other features. Filter-based features contribute less than DL features when evaluated on the test set. Overall, the GBDT model trained with different under-sampling methods assigned different contributions to features during test evaluation, with some features such as the Z-var and Z-interq of Gabor features, Z-var of Matched filters, Z-var and Z-interq of Frangi filters, deep features 5, 10, 12 and 14, having a minor influence when evaluated by permutation importance.

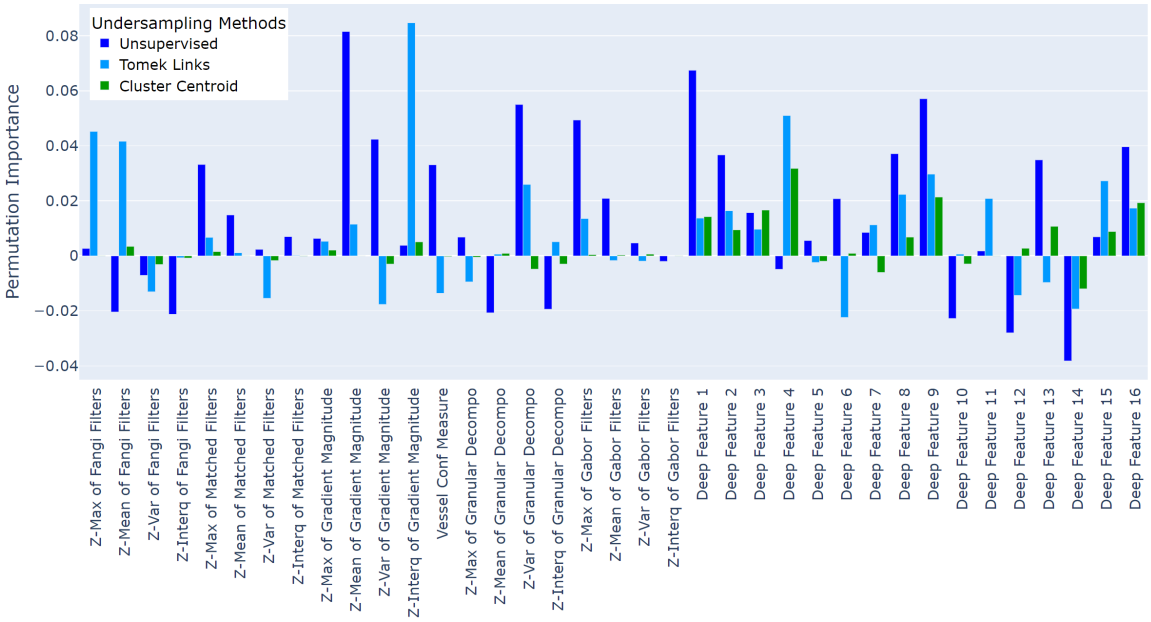


Figure 2.8: Permutation feature importance of GBDT models that were trained with different under-sampling methods. The smaller the value, the lower the importance.

2.4 Discussion

In this chapter, a novel ensemble framework for the automatic segmentation of coronary arteries in XCA was developed. The best-performing model utilized a GDBT classifier trained on samples generated by the Cluster Centroid under-sampling method, achieving a mean precision of 0.857, sensitivity of 0.902, specificity of 0.992,

F1 score of 0.874, AUROC of 0.947, and IoU of 0.779. The ensemble methods outperformed DeepLabV3+, various U-net-based models and the state-of-the-art method [90] on coronary vessel segmentation in almost all metrics and had a lower standard deviation in performance over test images.

From a clinical perspective, as more than 80% of Percutaneous Coronary Interventions (PCIs) are performed at the time of angiography [130], an accurate vessel segmentation method that improves the quality of QCA can greatly facilitate the assessment of stenosis; improve the quality of patient care; and avoid unnecessary PCIs, yield billions of dollars in savings at the national level [131]. In addition, correct delineation of the coronary vascular structures would be valuable in many types of CAD such as coronary endothelial dysfunction, where XCA serves as a testing technique [132].

From a technical perspective, this is the first time to our knowledge that ensemble methods, especially deep ensemble methods, have been applied to coronary artery segmentation. Ensemble methods are known for reducing the variance of predictions by gathering weak learners. In this study, we specifically used them for better predictive performance and robustness with limited data. In terms of the ensemble methods applied, GBDT is one of the leading boosting algorithms that employ decision tree weak learners. Compared with other decision tree boosting algorithms (e.g. [133]), it has more flexibility to handle various losses defined in different forms. Deep Forest, on the other hand, is an emerging ensemble method that creates stacked layers in ensemble training. Although the Deep Forest classifier did not achieve performance that was significantly better than GBDT, it still outperformed the DL models and was more consistent when evaluated over all test images. This suggests that the ensemble learning method and the training framework in which various features were extracted from both classic and DL filters are more suitable for a relatively small dataset where training samples are limited. The significant increase in sensitivity indicates that

the ensemble models have better recognition of the vessel area. This could be attributed to the employment of domain knowledge and a reduction in overfitting via the pixel-wise training scheme.

The proposed method intentionally incorporates redundancies in the feature vectors to enhance the model's ability to handle a broader range of possible scenarios effectively. The review of permutation importance suggests that certain features have limited contributions to the final prediction. Since permutation importance can be biased towards features that are correlated with one another [134], further investigation and refinement of the feature set will be essential to improve the model's performance and ensure the identification of the most representative features for accurate predictions.

Different from the state-of-the-art research on vessel segmentation [90], in which the dataset was run for 400 epochs with decreased learning rate on training loss saturation, we used an updated training logic to train the deep neural network model for feature extraction. Specifically, we reduced the training epoch's upper bounds and introduced an early-stop mechanism. Although the deep models trained as such have inferior performances compared to those obtained from [90], we believe it is not necessary to have prolonged training for the following two reasons. First, with the training logic currently applied, the state-of-the-art method by itself did not outperform our ensemble model in all metrics. Second, the trained deep neural networks are only used for feature extraction. Given that our dataset is relatively small, prolonged training on the deep feature extraction model may hinder the generalizability of the ensemble model built on its top. To summarize, the ensemble methods we proposed have strengths in predictive power compared to the deep-learning state-of-the-art on our datasets. Comparatively, our weaknesses lie in the complexity of preprocessing, feature extraction, and under-sampling pipeline prior to model training.

Recently, vision transformer networks [135] have been introduced to tackle seg-

mentation tasks[136] for their capability to capture long-range dependencies in images with the self-attention mechanism. However, most of the transformer-based networks are unable to be trained properly with a small-scale dataset. Current transformer-based structures designed for medical image segmentation either need thousands of annotated images [137] for training or require a large amount of computational resources[138]. Considering that we have limited training samples and a model that requires a lot of computational resources is less operational at the point of care in the cardiac catheterization lab, the transform-based network may not be a practical or optimal choice.

The proposed method has several limitations. First, the current Python implementation of Cluster Centroid under-sampling requires much more computational time than the Tomek Links and the unsupervised under-sampling methods (see Appendix B for details). Although a huge reduction in training samples should expedite the training process and yield better-performing prediction models given the relatively small size of the datasets utilized in this study 2.4, the prolonged under-sampling process increases training time and may hinder the method's efficiency and scalability in practice. Moreover, the choice of output pixel number for each class in the Cluster Centroid under-sampling method was determined through a number of trial experiments instead of a more comprehensive cross-validated grid search. Since this number affects the time required in under-sampling and the final performance, it is possible that better choices exist to reduce training time while maintaining a good performance.

CHAPTER III

Machine Learning Based Detection of Acute Respiratory Distress Syndrome using Electronic Health Records

3.1 Introduction

ARDS is a lung disease that develops in critically ill patients due to major trauma, pneumonia, aspiration, and sepsis, among other causes [139]. It is characterized by the accumulation of fluid in pulmonary alveoli, which results in decreased lung compliance and low blood oxygen [140]. Globally, ARDS affects more than 3 million people a year [141] of all ages, with a hospital mortality rate of around 40% [142, 143]. Recent studies indicate that ARDS is also a major complication related to COVID-19 [144, 145, 146] that is strongly associated with COVID-19 mortality [147, 148, 149].

As ARDS is a rapidly progressing disease, early intervention can improve patient outcomes [150]. Under-recognition of ARDS in clinical practice [151, 152], however, prevents evidence-based therapies, including lung protective Mechanical Ventilation (MV) and prone positioning [153], from being instituted. Therefore, establishing a machine-learning-based clinical support system to provide real-time ARDS surveillance for patients at risk is an urgent need. Such a system could help ensure clinicians provide patients with more consistent evidence-based care.

There have been previous EHR-based systems developed for detection [154, 155, 156] as well as patient risk stratification [157, 158] of ARDS. Reamaroon et al. [159] developed a machine-learning algorithm for ARDS detection by incorporating label uncertainty. Taoum et al. [160] used parameters extracted from four non-invasive physiological signals for real-time ARDS surveillance with belief functions theory, while the work in [161] only used raw ventilator waveform data to make predictions. Other Systems typically scan the dictated radiology reports of chest radiology studies for words consistent with the syndrome of ARDS [162, 163, 164, 165]. However, these systems rely on a frontline provider to obtain a chest radiograph at the time ARDS is developing, and rely on radiologists to rapidly describe the film using words these systems have been programmed to identify as consistent with "ARDS". A system that is designed to analyze other routinely available clinical data, including vital signs and clinical laboratory values, would be of significant benefit as they would be less reliant on chest radiology studies to accurately identify ARDS. In addition, given the variability in the early use of invasive MV for ARDS [166] and increasing use of other respiratory support modalities including high-flow nasal cannula [167], we also aimed to develop a model that was not reliant on variables closely related to invasive MV to detect ARDS.

ML approaches have been widely adopted in analyzing EHR data for disease prediction [168, 169], risk analysis [170] and classification [171]. For ARDS detection, classical algorithms like support vector machine (SVM) [159, 160], decision-tree-based classifiers [156, 172] and logistic regression (LR) [157, 162] have been used extensively.

The electronic health record contains a plethora of potential variables to consider for inclusion in a machine-learning model for ARDS diagnosis, including broad categories such as demographics, vital signs, respiratory support information, and laboratory results. Many of these features may be irrelevant to ARDS or redundant in terms of the information they provide. Using all available information may result

in the model overfitting the observed data while wasting computational power on redundant features during training. Thus, selecting an optimal number of features from EHR data that contribute to ARDS detection can help to construct a more consistent and generalizable model.

Feature dimension reduction techniques can be broadly categorized as either feature extraction or feature selection [173]. Feature extraction methods synthesize new features from extant ones, with Principle Component Analysis (PCA) [174] being a very common method. Feature selection methods, on the other hand, are further grouped into wrapper, embedded, and filter methods [175]. Least Absolute Shrinkage and Selection Operator (LASSO) [176], for instance, is an embedded method that performs feature selection in the linear model construction process and optimizes for it. Wrapper methods measure the utility of different feature subsets by the training/validation accuracy of a predictive model [177]. In contrast, filter methods are independent of the model or classifier and are defined by selection criteria. These selection criteria often employ information-theoretic concepts such as entropy [178], conditional entropy [179], and mutual information [180], a symmetric measure reflecting the information shared between X and Y, defined as

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(xy)log\frac{p(xy)}{p(x)p(y)} = I(Y,X). \tag{3.1}$$

The entropy-based filter methods used in this study are Minimum Redundancy Maximum Relevance (MRMR) [181] and Double Input Symmetrical Relevance (DISR) [182].

MRMR [181], as its name implies, seeks to maximize relevancy and minimize redundancy. Starting with an empty set, it uses mutual information to quantify the

usability of a potential feature X_i ($i \in 1, 2, \ldots, m$) in a feature set of size m as

$$f_{\text{MRMR}}(X_i) = I(Y, X_i) - \frac{1}{|S|} \sum_{X_s \in S} I(X_s, X_i) \tag{3.2}$$

in which Y is the class label, S is the set of all features selected with size $|S|$, and selected feature $X_s \in S$. X_i is a feature that is currently not selected, i.e., $X_i \notin S$.

At each forward selection step in MRMR, the feature that has the highest score is added to S:

$$\arg\max_{X_i \notin S} f_{MRMR}(X_i). \tag{3.3}$$

DISR [182] is a criterion defined based on symmetrical relevance [183] where a normalization term is applied to the mutual information. If the subset composed of X_i and X_j is denoted as $X_{i,j} = \{X_i, X_j\}$, then the DISR criterion is

$$f_{DISR}(X_i) = \sum_{X_j \in \mathcal{S}} \frac{I(X_{i,j}; Y)}{H(X_{i,j}, Y)} \tag{3.4}$$

where $I(X_{i,j}; Y) = I(X_j; Y|X_i) - I(X_i)$ by applying the chain rule of mutual information. Just as with MRMR, DISR utilizes forward selection, with the set S being updated by X_i:

$$\arg\max_{X_i \notin S} f_{DISR}(X_i.) \tag{3.5}$$

In this chapter, we present a comprehensive approach aiming to enhance real-time ARDS surveillance using EHR data. Our proposed methodology encompasses two key aspects: (1) employing various feature extraction and selection methods from EHR data, and (2) integrating them with different machine-learning models to facilitate ARDS diagnosis in real-time. The primary contributions of this work are as follows:

1. We introduced machine learning models based on EHR data for ARDS diagnosis, achieving promising discriminative results.

2. A set of models was generated, which do not rely on variables from mechanical ventilators, exploring the possibility of identifying ARDS before the onset of mechanical ventilation or in situations where this information is unavailable.

3. During the feature selection process, relevant factors and variables associated with ARDS were identified for further investigation into the factors influencing its development and progression.

The subsequent sections in this chapter are organized as follows. Section 3.3 reports the performance of multiple models and feature selection methods when trained and tested on different variable sets, with the best-performing model achieving an AUROC of 0.854($\pm$ 0.026). Section 3.2 describes the data preparation methodology and how the models were trained, selected, tested, and evaluated, while Section 3.4 provides interpretations of the features selected by different models and describes potential future applications. In particular, the results from this study show that the use of EHR-based ML models for ARDS detection may help to reduce the under-detection of ARDS and improve patient outcomes by enabling earlier intervention.

3.2 Methods

All computational methods were implemented in MATLAB. The implementation of MRMR used in this study was taken from [184], while the DISR code used was from the FEAST library [177].

3.2.1 Data Preparation

The dataset used in this study is composed of 426 encounters with patients who were hospitalized and developed acute respiratory failure. The Institutional Review Board approved this study with a waiver of informed consent. A group of 13 physicians reviewed hospitalizations to determine whether ARDS developed and if so the

time of ARDS onset, with 2 to 4 reviewers for each encounter. Non-ARDS and ARDS encounters were labeled by reviewers as 0 or 1 respectively. To determine the final ARDS status of each encounter, equal-weighted voting was applied to all reviewers (additional reviews were performed as needed to resolve ties). For those 105 encounters that were diagnosed as ARDS, the time of onset was chosen as the earliest time point when the ARDS diagnosis was made across all independent reviews.

Data samples of non-ARDS encounters and those collected from ARDS encounters before the onset of ARDS were labeled as non-ARDS. For samples after the onset of ARDS, only those within 48 hours of onset were labeled as ARDS, as treatment provided after diagnosis may result in later data becoming unrepresentative of the patient's disease status [185].

Vital signs, respiratory support information, and laboratory results were extracted from EHR, yielding 66 potential variables. All values were time-stamped, with observation intervals ranging from 15 min up to 24 hours. Previous data were carried forward until the next available value was observed. The missing values of each potential feature were imputed with its mean value after removing features whose number of missing values exceeded 1/6 of the total. 55 variables were preserved after this step. In order to develop an ARDS classification model that was less reliant on variables related to invasive MV, two datasets were created. The first included MV variables (55 variables) while the second excluded 10 variables related to invasive MV (41 variables). Variables related to MV include invasive MV status, the presence of supplemental oxygen, the preset respiratory rate for MV, the respiratory rate observed, the tidal volume set on the mechanical ventilator, the tidal volume observed, the Positive End-Expiratory Pressure (PEEP) used during MV, the plateau pressure, and the mean airway pressure delivered during invasive MV. Glasgow Coma Scale (GCS) related variables were also removed as they might be related to the provision of sedation once patients were placed on invasive MV. A comprehensive list of

variables extracted from the EHR, along with their abbreviations, meanings, missing condition, and summary statistics, can be found in Appendix D and D."

3.2.2 Data Partition and Sampling

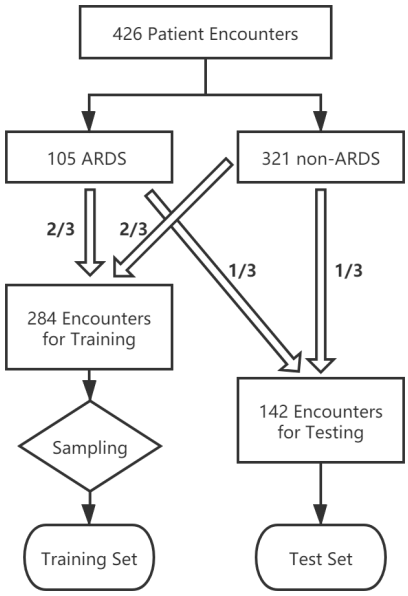


Figure 3.1: A diagram of data partition and sampling.

For both non-ARDS and ARDS encounters, an encounter-wise split was performed as shown in Figure 3.1. 2/3 of the encounters were randomly selected and assigned to training, while the remaining 1/3 were held out for testing. This ensured that the data samples from one encounter would not be partitioned across training and testing sets. In order to get the final training set, the sampling strategy proposed by [159] was applied to reduce the correlation between samples and generate a balanced training set with regard to the non-ARDS and ARDS samples. The thresholds η for sampling non-ARDS and ARDS examples were set to 0.572 and 0.994 respectively. The testing set, on the other hand, was not re-sampled nor used during training.

3.2.3 Training Strategy

The training set was divided into four folds. Fold compositions were changed in sequence under cyclic permutation with a 3:1 ratio of in-folds training and validation. When using PCA as a feature extraction method, normalization (see equation (3.6)) and PCA were first applied to the in-folds-training set with parameters being saved to transform the validation set. Normalization was also applied to models that used SVM as classification models. For entropy-based feature selection methods, all variables in the training set were discretized into 10 bins and then inputted into either MRMR or DISR to obtain a ranking of feature importance based on Equation (3.2) and (3.4).

The classification models evaluated were Support Vector Machine (SVM), Random Forest (RF), and Logistic Regression (LR). In order to determine the optimal Number of Principle Components (nPC) for PCA, the optimal Number of Features (nFea) for MRMR and DISR and other model hyperparameters, four-fold cross-validation was performed on the training set over different hyperparameter combinations as described in Table 3.1.

$$x' = \frac{2(x - min(x))}{max(x) - min(x)} - 1 \tag{3.6}$$

When Least Absolute Shrinkage and Selection Operator (LASSO) was used as a feature selection method, LASSO regularization for logistic regression (LR) [186] with 4-fold cross-validation was first employed on the training set. After that, the model coefficients corresponding to the λ that had the minimum expected deviance plus one standard deviation were chosen. The deviance here was the value of the mean square error of the model-averaged over the validation folds. Variables with nonzero model coefficients were "selected" as the optimal feature set derived from the regularized model. When combined with SVM or RF, these variables were used in training to run hyperparameter optimization along with details listed in Table 3.1.

3.2.4 Evaluation Metrics

The outputs from SVM and RF models were thresholded at zero to determine the resultant classification. The performances of different models were evaluated using the following metrics: AUROC, sensitivity (recall), specificity, accuracy, and F1 score.

The outputs from LR, on the other hand, were within the range of 0 and 1. These were directly passed into the AUROC calculating function. Sensitivity, specificity, accuracy, and F1 scores were computed at the point on the ROC curve where the sensitivity and specificity were approximately equal.

3.2.5 Hyperparameter Selection

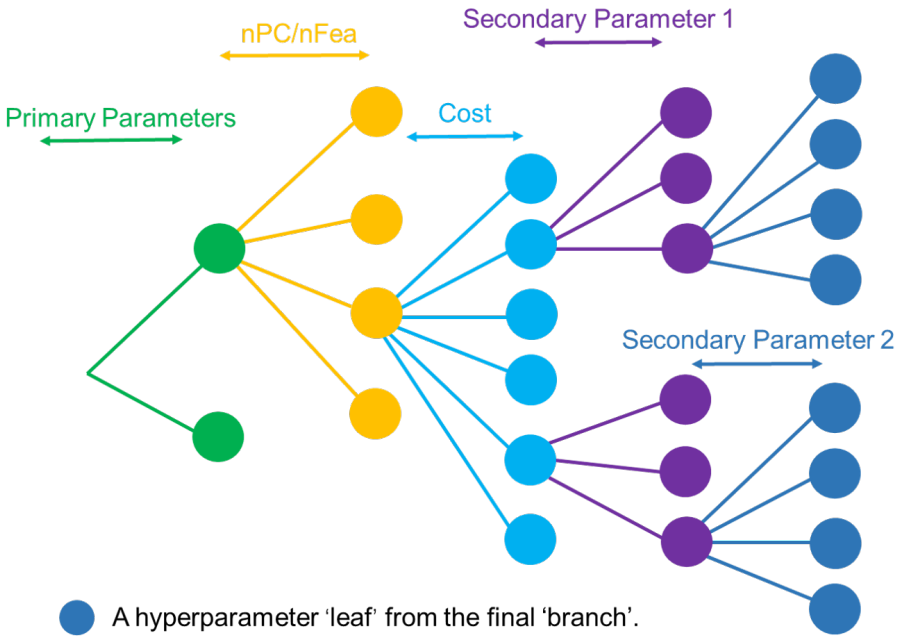


Figure 3.2: A tree representation of hyper-parameter combinations.

As depicted in Figure 3.2, hyperparameter combinations can be represented as a tree, where branches represent the choice of values for various hyperparameters. "Primary Parameters", "nPC/nFea" and "Cost" in Figure 3.2 corresponded to those in Table 3.1" and the "Secondary Parameter 1" and "Secondary Parameter 2" were consistent with "Secondary Parameters" in Table 3.1. The referenced order of "pri-

mary" and "secondary" comes from their position in the hyperparameter selection tree (Figure 3.2).

The optimal hyperparameter combination was chosen using the following three-step process. First, the mean and standard deviation of all metrics over four-fold cross-validation were calculated. Then, the first three hyperparameters ("Primary Parameters", "nPC/nFea" and "Cost") were fixed and the "Secondary Parameters" yielding the highest mean accuracy were chosen. Finally, the accuracy values over "nPC(/nFea)" and "Cost" for different Primary Parameters were examined as illustrated in Figure 3.3. Each point on the surface plot represented the best mean accuracy value obtained during cross-validation for specific hyperparameter combinations. By calculating the mean standard deviation of accuracy over all the points in a plot, the optimal "Primary Parameter" was chosen as the one that had a lower standard deviation, i.e., greater stability. The optimal number of principal components or the number of features were chosen to be the ones that achieved the highest accuracy value of the selected optimal "Primary Parameters", while the cost was chosen at the intersection of specificity and sensitivity when the optimal "Primary Parameters" and optimal "nPC(/nFea)" were fixed.

When LASSO was utilized for feature selection, the number of features was set prior to hyperparameter tuning. The rest of the hyperparameters were then chosen in the same manner as mentioned above.

3.2.6 Testing Strategy

For the SVM and RF models with PCA as a feature extraction method, the entire training set was retrained to obtain the final model after the optimal hyperparameters were selected. The test set was first transformed by normalization and then PCA using the transformation obtained from the training set prior to the final model being applied. For models using LASSO, MRMR, or DISR for feature selection, only the

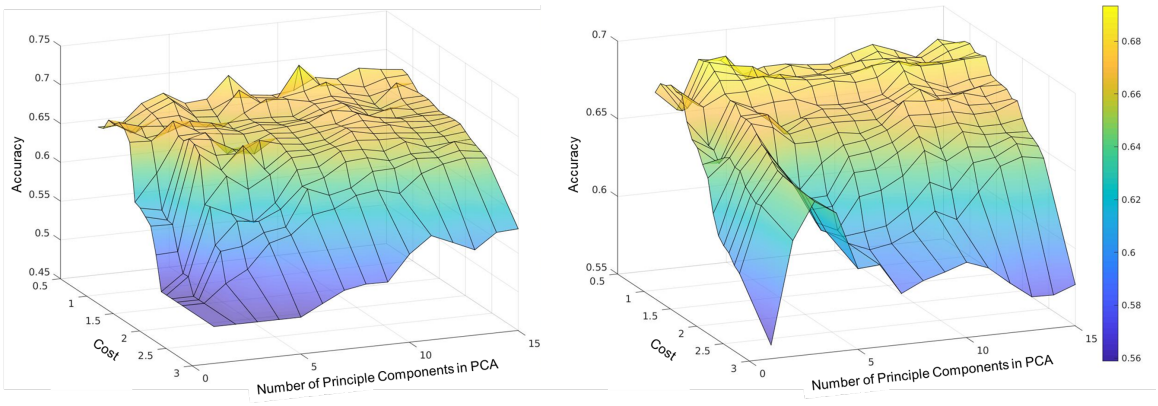


Figure 3.3: Mean accuracy plot over nPC and cost for PCA + SVM algorithm. Left panel: Linear kernel SVM; Right panel: RBF kernel SVM.

variables that had been selected in training would be retained in the testing set. For LASSO with LR, model coefficients from training were also used in testing to produce a predicted value ranging from 0 to 1.

Considering that the partition of the training and test set was random, the testing performance in one split could be unrepresentative for a model. Thus the whole partition-training-testing process was repeated four times to better evaluate the performance of these models.

3.2.7 ARDS with or without MV

To test the ability of the models to detect ARDS with or without variables related to invasive MV, two sets of models were trained using the two datasets that include or exclude these variables as described in Section 3.2.1.

3.3 Results

3.3.1 Classification Performance and Model Comparison

Table 3.2(a) shows the results of ARDS classification when MV-related variables were included. The mean and standard deviation of each performance metric were calculated over four random splits as described in Section 3.2.6. As shown in the table,

Table 3.1: Hyperparameter Configurations for Different Models

Models	**Primary Parameters**	**nPC[c]/ nFea[d]**	**Cost**	**Secondary Parameters**
PCA + SVM	Kernel= {Linear, RBF[a]}	{1:1:15}[e]	{0.8:0.1:2.4,3}	Box Constraints = {.1,1,10,100,1000}; Kernel Scale = {.5,1,50,100,500,1000};
PCA + RF	Split Criterion = {GDI[b], Cross-entropy}	{1:1:15}	{0.5:0.1:2.4,3}	Number of Trees = {5,25,50,100:25:175}; Minimal Leaf Size = {1,3,5,10,25,50};
LASSO + SVM	Kernel = {Linear, RBF}	-	{0.8:0.1:2.4,3}	Box Constraints = {.1,1,10,100,1000}; Kernel Scale = {.5,1,50,100,500,1000};
LASSO + RF	Split Criterion = {GDI, Cross-entropy}	-	{0.5:0.1:2.4,3}	Number of Trees = {5,25,50,100:25:175}; Minimal Leaf Size = {1,3,5,10,25,50};
MRMR/DISR + SVM	Kernel = {Linear, RBF}	{6:3:24}	{0.8:0.1:2.4,3}	Box Constraints = {.1,1,10,100,1000}; Kernel Scale = {.5,1,50,100,500,1000};
MRMR/DISR + RF	Split Criterion = {GDI, Cross-entropy}	{6:3:24}	{0.5:0.1:2.4,3}	Number of Trees = {5,25,50,100:25:175}; Minimal Leaf Size = {1,3,5,10,25,50};
MRMR/DISR + LR	-	{6:3:24}	-	-

[a] RBF: radial basis functions;
[b] GDI: Gini diversity index;
[c] nPC: the number of Principle Components when applying PCA;
[d] nFea: the number of features used.
[e] The notation $x : y : z$ indicates a range of values from x to z with y being the step size.

LASSO with LR performs the best in four out of five metrics (AUROC of 0.854, F1 score of 0.296, specificity of 0.764, and accuracy of 0.764), while MRMR with SVM obtains the highest mean sensitivity of 0.827. The model that has the lowest standard deviation (0.008) for AUROC is MRMR with RF. MRMR with SVM gives the lowest standard deviation (0.006) in the F1 score while MRMR with LR was the lowest with respect to sensitivity (0.011). LASSO with SVM exhibited steadier specificity and accuracy compared to other models. Its standard deviation for accuracy is only 0.003. For the worst-performing algorithms, PCA and RF had the lowest AUROC at 0.816

Table 3.2: Test Results across Four Random Splits

Models	AUROC	F1 Score	Sensitivity	Specificity	Accuracy
	(a). Test Results When Including MV-related Variables				
PCA+SVM	0.825±0.021	0.27±0.015	0.74±0.063	0.742±0.024	0.742±0.019
LASSO+SVM	0.837±0.024	0.259±0.013	0.803±0.063	0.697±0.004	0.704±0.003
MRMR+SVM	0.831±0.01	0.257±0.006	**0.828±0.044**	0.681±0.027	0.691±0.024
DISR+SVM	0.831±0.014	0.273±0.027	0.776±0.055	0.728±0.044	0.731±0.039
PCA+RF	0.816±0.034	0.267±0.036	0.728±0.069	0.742±0.022	0.741±0.021
LASSO+RF	0.829±0.023	0.255±0.018	0.825±0.032	0.679±0.029	0.688±0.028
MRMR+RF	0.83±0.008	0.251±0.015	0.817±0.041	0.675±0.032	0.684±0.028
DISR+RF	0.824±0.018	0.248±0.017	0.774±0.112	0.689±0.076	0.694±0.064
LASSO+LR	**0.854±0.026**	**0.296±0.025**	0.764±0.026	**0.764±0.026**	**0.764±0.026**
MRMR+LR	0.839±0.014	0.292±0.013	0.762±0.011	0.761±0.01	0.762±0.01
DISR+LR	0.837±0.019	0.284±0.01	0.754±0.016	0.754±0.016	0.754±0.016
	(b). Test Results When Excluding MV-related Variables				
Models	**AUROC**	**F1 Score**	**Sensitivity**	**Specificity**	**Accuracy**
PCA+SVM	0.817±0.03	0.259±0.025	0.723±0.05	0.732±0.035	0.731±0.031
LASSO+SVM	0.8±0.031	0.24±0.015	0.787±0.069	0.671±0.032	0.678±0.028
MRMR+SVM	0.794±0.023	0.245±0.007	0.762±0.052	0.692±0.028	0.697±0.025
DISR+SVM	0.791±0.031	0.248±0.02	0.732±0.053	0.712±0.043	0.713±0.039
PCA+RF	0.808±0.027	0.262±0.032	0.708±0.046	**0.742±0.043**	0.739±0.038
LASSO+RF	0.772±0.04	0.214±0.019	**0.797±0.056**	0.609±0.04	0.621±0.039
MRMR+RF	0.787±0.024	0.233±0.016	0.755±0.071	0.672±0.053	0.677±0.046
DISR+RF	0.792±0.023	0.238±0.016	0.718±0.064	0.703±0.032	0.703±0.028
LASSO+LR	**0.821±0.025**	**0.269±0.02**	0.74±0.023	0.74±0.023	**0.74±0.023**
MRMR+LR	0.8±0.021	0.255±0.008	0.726±0.017	0.726±0.017	0.726±0.017
DISR+LR	0.798±0.028	0.255±0.015	0.726±0.022	0.725±0.023	0.725±0.023

Bold indicates those with the highest mean value.
Underline signify those with the lowest standard deviation;

with the highest standard deviation of 0.034. DISR and RF had the lowest F1 score at 0.248. In general, the combination of LASSO/MRMR with SVM or LR yielded better performance and stability when compared to the models that used RF as a classifier.

3.3.2 Impact of MV-related Variables on Model Performance

Table 3.2(b) lists the mean and standard deviation of test metrics for different models over four random splits when MV-related variables were removed in training and testing. Based on these results, LASSO with LR works had the best performance

with respect to AUROC (0.821), F1 score (0.269) and accuracy (0.74). The highest sensitivity of 0.797 is achieved by LASSO with RF, while the best specificity of 0.742 is obtained by PCA with RF. MRMR with SVM has the lowest model variability with regard to the F1 score, while the lowest standard deviations for the rest of the metrics (AUROC, sensitivity, specificity and accuracy) are obtained by the combination of MRMR and LR. PCA with RF has the worst AUROC (0.772) and F1 score (0.248) of all models. Similar to the case when MV-related variables were present, the fluctuation of LASSO with LR is larger than MRMR/DISR with LR with respect to AUROC and F1 score.

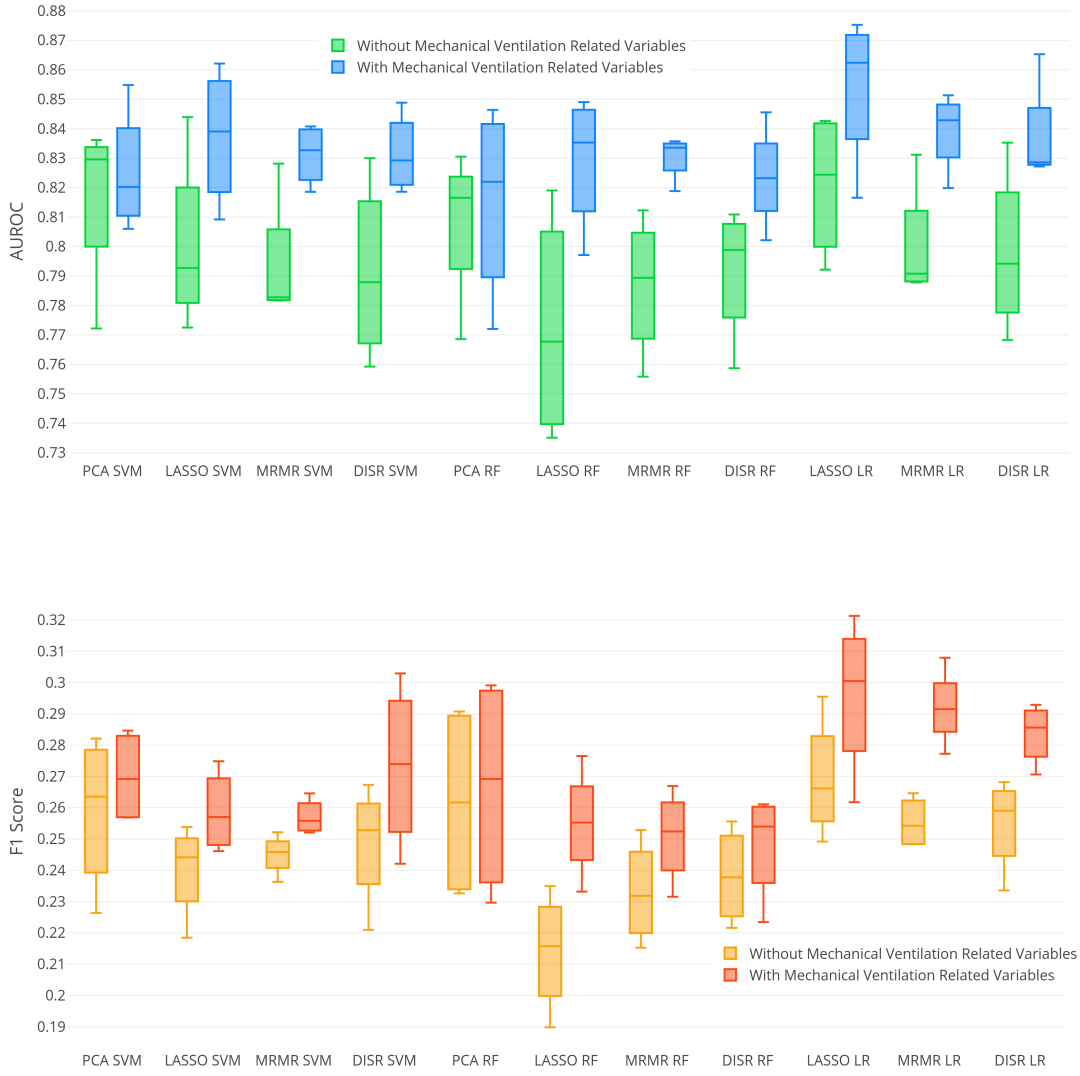


Figure 3.4: Test AUROC (upper panel) and F1 score (lower panel) for different methods with or without MV-related variables' presence.

Comparing the model performances between including (Table 3.2(a)) and excluding (Table 3.2(b)) MV variables, there are increases in performance for all the models and in almost all metrics. The elevation in performance indicates that MV-related variables are meaningful for ARDS classification. The increase in AUROC was around 3-4% except for PCA with SVM and PCA with RF, which are less than 1%. The increase in F1 score is bigger than 1% for all SVM-based models(2%), LASSO/MRMR with RF, and LR-based models(3%). This is also the case for sensitivity and speci-

ficity, suggesting that the performance boost for including MV-related variables is more obvious for LR models and then SVM models.

There are, however, some discrepancies for MRMR with the SVM model as to specificity and accuracy, and DISR with RF as to accuracy where these metrics decline when MV variables were added. By examining the performance in each of the four splits, the cause was identified as the low testing accuracy (and specificity) for the aforementioned models in the first split. As every split is random and the features selected in each dataset are different, such variation in performance is expected.

Figure 3.4 depicts the box and whisker plots of AUROC and F1 scores from both datasets. One can observe that even though LASSO with LR performed the best in terms of AUROC and F1 score, given its variability and the performance of other models, in particular, MRMR/DISR with LR, the superiority of LASSO with LR was not significant. Additionally, AUROC obtained from the dataset excluding MV variables appears to be less steady than its counterpart.

3.4 Discussion

In this chapter, different ML models for ARDS classification were examined with the proposed algorithms. Models were trained to identify ARDS in both cases where mechanical ventilators were used and cases where they were not.

The training set was re-sampled to produce a more balanced set for training so that 1) the inter-dependency of a patient's samples was reduced; 2) the total number of data samples was reduced to shorten training time; 3) to prevent the learning process from being compromised due to models that expect a balanced class distribution[187] and 4) to enable the use of accuracy as the hyperparameter selection metric. The testing set, on the other hand, was not down-sampled so that the testing results could better reflect how the methods would operate in clinical settings.

Classification costs were also incorporated to help mitigate the class imbalance

within the dataset. Though the training set was made to be almost balanced, ARDS cases are much less prevalent than non-ARDS cases in real-life scenarios (with a ratio of around 1:15 in the utilized dataset). By incorporating penalties for ARDS cases as specified by the cost matrix in cross-validation and testing, a more balanced performance of sensitivity and specificity was achieved when SVM and RF were used as the classification method.

In general, LASSO with LR performed the best in terms of AUROC, F1 score, and accuracy in all four testing splits, achieving an AUROC of 0.854 and accuracy of 0.764 when averaged over four random permutations of the dataset (Table 3.2(a)). Even when MV-related variables were not present, PCA + SVM and LASSO + LR achieved AUROC values over 0.81 and accuracies over 0.73 as shown in Table 3.2(b), suggesting that these algorithms can provide a reasonable classification of ARDS cases regardless of a patient's MV status. Though their performances did not surpass that of LASSO with LR, entropy-based feature selection methods (MRMR and DISR) were among the best-performing groups when paired with LR. Moreover, despite the fact that MV-related variables were relevant in ARDS classification, removing them seems to have less influence on models that use PCA as the feature combination method. That PCA with SVM performed the best under this scenario may indicate that the variables that were not related to MV can still provide enough information through feature recombination for models to accurately classify ARDS. The disadvantage of applying feature recombination in ARDS classification, however, is the difficulty in interpreting the classification results based on feature values.

Table 3.3 describes the optimal feature sets selected by LASSO, MRMR, and DISR over four splits (The details of the features are listed in Appendix D and D). Within each feature set, features are listed in decreasing importance as determined by the method used.

Despite the fact that some features might appear in a certain split with a specific

Table 3.3: Optimal Feature Set Selected by LASSO, MRMR, and DISR

Split 1	**Without MV-related variables**	**With MV-related variables**
LASSO	pf, pf_calc, sedated, total_out, Alb, HCO2, Plt, total_in, urine_out, dbp, oriented, Hgb, PTT, age, rass	pf, supl, PEEP, pf_calc, RRset, invasive, Plt, total_in, total_out, urine_out, Alb, HCO2, Plat, Vtset, sedated, PTT, gcs_total, oriented
MRMR	Plt, sedated, fiO2, alert, urine_out, age, total_in, pf_calc, rass, AST, oriented, total_out, iv_in, pf, plt_transf	Plt, supl, invasive, PEEP, sedated, fiO2, gcs_eye, urine_out, gcs_verbal, total_in, age, pf_calc, mAirP, alert, RRset, rass, Vtset, gcs_total, AST, total_out, oriented, iv_in, gcs_motor, pf
DISR	fiO2, urine_out, pf_calc, Tbili, sedated, total_in, alert, iv_in, total_out, rass, AST, pf, pH, Plt, plt_transf, HCO2, TP, PTT, temp, oriented, age, dbp, INR, Alb	supl, dialysis, invasive, urine_out, plt_transf, PEEP, fiO2, total_out, total_in, pf_calc, Tbili, mAirP
Split 2	**Without MV-related variables**	**With MV-related variables**
LASSO	pf, pf_calc, total_out, TP, sedated, total_in, HCO2, Plt, dbp, oriented, K, age, urine_out	pf, supl, invasive, mAirP, pf_calc, PEEP, RRset, total_in, total_out, Vtset, urine_out, Plat, Plt, TP, oriented, sedated, K, WBC, gcs_motor, spO2
MRMR	Plt, fiO2, sedated, urine_out, age, plt_transf, pf_calc, iv_in, oriented, rass, AST, total_out, alert, total_in, pf	Plt, supl, invasive, fiO2, PEEP, sedated, urine_out, age, iv_in, pf_calc, gcs_eye, mAirP, oriented, AST, rass
DISR	fiO2, total_out, total_in, pf_calc, urine_out, iv_in	supl, dialysis, invasive, ffp_transf, total_out, total_in, fiO2, urine_out, iv_in, PEEP, pf_calc, mAirP
Split 3	**Without MV-related variables**	**With MV-related variables**
LASSO	pf, pf_calc, dbp, total_out, HCO2, TP, alert, total_in, Alb, Hgb, Plt, urine_out, K, WBC, age, oriented, sedated	pf, supl, invasive, pf_calc, PEEP, RRset, total_out, Vtset, dbp, total_in, urine_out, HCO2, Hgb, spO2, Alb, K, Plat, Tbili, WBC, gcs_verbal
MRMR	Plt, fiO2, alert, urine_out, age, iv_in, rass, pf_calc, sedated, AST, total_out, total_in, TP, pf, plt_transf	Plt, supl, invasive, PEEP, fiO2, alert, urine_out, age, gcs_verbal, iv_in, pf_calc, rass
DISR	pf, pf_calc, dbp, total_out, HCO2, TP, alert, total_in, Alb, Hgb, Plt, urine_out, K, WBC, age, oriented, sedated	supl, dialysis, invasive, urine_out, total_out, total_in, fiO2, PEEP, ffp_transf, iv_in, pf_calc, Tbili
Split 4	**Without MV-related variables**	**With MV-related variables**
LASSO	pf, pf_calc, urine_out, HCO2, Plt, sedated, total_in, total_out, BUN, WBC, dbp, oriented	pf, supl, PEEP, pf_calc, RRset, invasive, urine_out, Vtset, total_in, total_out
MRMR	AST, fiO2, alert, urine_out, oriented, pf_calc, rass, total_in, sedated, age, pf, total_out	AST, supl, invasive, fiO2, PEEP, urine_out, gcs_verbal, Vtset, total_in, sedated, pf_calc, oriented, age, mAirP, rass, alert, RRset, total_out, gcs_total, pf, iv_in
DISR	fiO2, urine_out, pf_calc, Tbili, total_in, iv_in, total_out, sedated, alert, pf, rass, oriented, temp, plt_transf, AST, PTT, age, INR, Plt, HCO2, TP, dbp, WBC, rr	supl, dialysis, invasive, urine_out, total_out, fiO2, total_in, Tbili, PEEP, pf_calc, iv_in, gcs_verbal, plt_transf, ffp_transf, mAirP, gcs_total, sedated, RRset, AST, alert, noninvasive, pf, Plat, oriented

method but not in others, one can still observe a common set of features that occur repeatedly under different circumstances, indicating that they may be clinically relevant to ARDS diagnosis or disease development. For example, when MV-related variables are removed, 'pf', PaO_2/FiO_2, the ratio of blood oxygen to supplemental oxygen showed up in the selected set for all feature selection methods. 'total_in' and 'total_out', the total daily fluid in and out, were also listed frequently. When MV-related variables were included, 'pf', 'supl', 'PEEP', 'RRset', 'invasive' were almost always included in the selected set, while 'total_in', 'total_out', 'Vtset', 'urine_out' and 'mAirP' were often seen.

Additionally, as shown in Table 3.3, the three feature selection methods utilized in this study exhibited different tendencies of assigning feature importance, thus providing interesting viewpoints of clinical features that may have been overlooked by clinicians in current practice.

Among variables not related to respiratory function, 'Plt', the platelet count, was the most important feature with respect to the MRMR criterion in three out of four splits regardless of the inclusion of MV-related variables. A handful of previous studies had identified platelets as playing a pathophysiologic role in ARDS. Studies had shown that platelet count is related to ARDS risk and survival [188] and that it contributes to ARDS mortality by interacting with a genetic variant in gene leucine-rich repeat–containing 16A (LRRC16A)[189, 190]. Another variable, 'AST', or alanine aminotransferase, an enzyme that serves as an indicator of liver damage, was also a feature selected by MRMR and DISR in all four splits when MV was not present as predictive of ARDS. These results aligned with a recent study also identifying it to be a predictive factor of ARDS in COVID-19 patients[191]. Whether elevation of these liver enzymes simply identifies patients with more severe organ dysfunction, a scenario where ARDS also develops, or whether they are more directly related to ARDS pathogenesis, warrants further study.

The current model constitutes an important step for developing a clinical support system to detect ARDS. Apart from learning predictive EHR features, further work could be done by including physiological waveform signals and chest imaging data to develop an updated machine-learning base ARDS classifier.

CHAPTER IV

Learning Using Privileged Information with Logistic Regression on Acute Respiratory Distress Syndrome Detection

4.1 Introduction

ARDS is a serious lung disease that affects critically ill patients and has various causes [139], leading to fluid accumulation in the lungs and decreased oxygen levels [140]. Early detection of ARDS is crucial for improved patient outcomes, especially considering its high mortality rates [143]. Current clinical diagnosis methods rely on chest X-rays, which can introduce delays and hinder timely intervention. To address this, ML models [154, 155, 156, 157, 158, 159] using EHR data have been explored for ARDS detection and risk stratification. Additionally, alternative approaches involving physiological signals [160], mechanical ventilation waveform data [161], and chest radiology reports [192] have been investigated for real-time surveillance and image-based ARDS detection. However, previous studies have primarily focused on single-data modality models, while the integration of information from multiple modalities has the potential to significantly enhance the model's performance and provide stronger decision support, particularly if the additional information proves effective in diagnosing the disease.

In scenarios where there is additional data available during model training but not during testing, such data is referred to as *privileged information*. Conversely, the data consistently available during both model training and application is referred to as *base information* throughout this chapter.

Privileged information is particularly common in the medical field. In the case of ARDS, for patients who are suffering from shortness of breath, their EHR, including lab test results and vital waveform data, may likely be present upon hospital admission and can be used as data in the base domain. However, their CXR images, required for ARDS diagnosis, are typically not available at the early stage of hospitalization due to processing times and are considered privileged information in this case. In order to provide clinical decision support for ARDS condition upon admission, it would be advantageous to develop models that do not rely on CXR information in the test/inference stage but can still utilize it during training to supplement the EHR data and produce a better model. Mechanical ventilation-related information is another example of privileged information in ARDS. Due to the variability in the early use of invasive MV for ARDS [166] and increasing use of other respiratory support modalities such as high-flow nasal cannula [167], clinicians and diagnostic models alike may not have this information available when making decisions. In both examples, both privileged and base information could be effectively combined using the LUPI paradigm [45] to build a superior ARDS detection model. Similar to how students perform better in exams by learning in class with the help of teachers, under the LUPI paradigm, *Machine Students* can leverage privileged information provided by *Intelligent Teachers* during training [46] and thus yield an improved model on the base domain.

To formally present the LUPI paradigm, we first describe the classical paradigm of an ML classification problem as follows. Given a set of n independent and identically

distributed (i.i.d.) training samples generated by a probability distribution $p(x, y)$ as

$$(x_1, y_1), (x_2, y_2), ...(x_n, y_n),\ x_i \in \mathcal{X},\ y_i \in \mathcal{Y},$$

we want to find a decision function $\hat{f}$ from a family of functions $\mathcal{F}$ so that $\hat{y} = \hat{f}_n(x)$ gives the smallest error rate for classification among all $f \in \mathcal{F}$, for example, by minimizing the empirical ℓ-risk

$$\hat{R}_\ell(f) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f(x_i)\right) \tag{4.1}$$

defined with respect to a loss function ℓ.

Under the LUPI paradigm, however, we have n triplets of i.i.d. training samples generated based on a fixed but unknown probability distribution $p(x, x^*, y)$

$$(x_1, x_1^*, y_1), (x_2, x_2^*, y_2), ...(x_n, x_n^*, y_n),\ x_i \in \mathcal{X},\ x_i^* \in \mathcal{X}^*,\ y_i \in \mathcal{Y}$$

The goal is still to find the best decision function $\hat{f}_n$ and minimize the empirical ℓ-risk defined in (4.1). This time, we have (x, x^*, y) instead of (x, y) at the training stage, where $x^* \in \mathcal{X}^*$ contains additional information generated from x based on an unknown conditional distribution $p(x^*|x)$.

Sometimes, instead of having n triplets, we may only have privileged information in m out of n training samples, which looks like

$$(x_1, x_1^*, y_1), ...(x_m, x_m^*, y_m), (x_{m+1}, y_{m+1}), ..., (x_n, y_n),\ x_i \in \mathcal{X},\ x_i^* \in \mathcal{X}^*,\ y_i \in \mathcal{Y}$$

after possibly permuting and reordering the n samples. And as an extension of LUPI, this scenario is often referred to as LUPAPI, with the same goal of minimizing the empirical ℓ-risk defined in (4.1).

There has been a lot of previous work done on developing new algorithms under

the LUPI paradigm. Vapnik [45] developed the first SVM algorithm under the LUPI paradigm by introducing SVM+, in which privileged information is utilized to give an estimation of the slack variable ξ_i to accelerate model training. There have been follow-up discussions on the two base frameworks [46] used in LUPI, the *similarity-control* and *knowledge-transfer*, and on the theoretical foundations of the paradigm [193]. In addition, Li et al. [194] provided a fast implementation of the SVM+ base on the SVMLIB library [195]. Specifically for ARDS detection, Sabeti et al. [47] focused on the case when privileged information is partially available and extended the SVM model to incorporate both partially available privileged information and label uncertainty, where the privileged information came from the average reviewing score of chest X-ray images given by a group of clinicians.

In recent years, training Neural Networks (NN) under the LUPI paradigm has attracted considerable attention. Tang et al. [196] focus on data deployment by either using privileged information as an auxiliary prediction target or learning from the fused representation of data in base and privileged domains. Other studies modified the loss function under the knowledge-transfer framework [197, 198]. These methods mostly involved creating soft labels from the *Teacher*'s network and penalizing the difference between *Student*-learned and *Teacher*-given labels in the loss function to aid *Student* network training. Lambert et al. [199], on the other hand, extended the dropout, a well-formed regularization technique in NN training [200], to heteroscedastic dropout by making the variance of the multiplicative Gaussian dropout function a function of privileged information. This approach has stronger theoretical support based on previous research [201, 202, 203, 204] and has later been extended to graph neural networks [205] for disease gene prediction.

So far, most attention in LUPI research has been focused on developing SVM or NN models. Although other studies have sought to incorporate label uncertainties [206, 207] or noisy labels [208] in LR, LUPI with LR is seldom mentioned. In medical

fields, LR and regularized LR models have been widely applied [209] for EHR analysis in disease screening, onset prediction, and risk prediction because of their effectiveness and interpretability. They are also used to develop important clinical decision support tools for ARDS [157, 162] and other critical lung diseases [162]. Despite the development of more complex ML algorithms such as NNs, LR models have been proven to perform equally well, if not better, in EHR-related tasks [210, 211, 212, 213]. Moreover, they tend to be more robust against data imbalances [209], less biased [210], computationally more efficient, and have better explainability compared to other ML algorithms. This chapter, therefore, set out to build a Privileged Logistic Regression (PLR) model and examine its effectiveness in the context of ARDS detection.

The primary contributions of the work in this chapter are as follows:

1. To the best of our knowledge, we present the first PLR model following the LUPI paradigm and extended it for cases when privileged information is only partially available.

2. An asymptotic analysis was performed that delineates the sufficient conditions under which the addition of privileged information increases the rate of convergence in the proposed model.

3. Experiments were carried out for ARDS detection using the PLR model. Results show that the proposed model leveraged privileged information in training and displayed improved performances in testing compared to the models trained only on the base domain. In addition, PLR models yield superior performances compared with SVM+ or network-based models developed under the LUPI paradigm.

4. Based on the results, we provide insights and explanations for important risk factors for ARDS.

The remainder of this chapter is organized as follows. Section 4.2 first introduces the dataset used in the chapter, then covers the new PLR models and their asymptotic analysis. Following that, experimental setup, data processing details, model implementation, training strategy, and the evaluation metrics used are also presented. Section 4.3 reports the test performances of models trained with and without the presence of privileged information. After comparing performances across different LUPI models, it also presents the test results when experiments were carried out by varying the percentage of available training samples using the PLR. Section 4.4 provides interpretations of the results and describes the limitations of the current work.

4.2 Material and Method

4.2.1 Dataset

The ARDS dataset was collected at the University of Michigan Hospital from 2016 to 2017. It is comprised of 1081 encounters from 1041 patients hospitalized with acute respiratory failure ($PaO_2/FiO_2 < 300$ mm Hg under invasive MV) or moderate hypoxia (receive $>$ 3L O_2 by nasal cannula for $>$ 2 hours). Among these 1,081 encounters, 500 of them have at least one CXR taken during their stay, hereafter referred to as Cohort 1, while Cohort 2 contains the remaining 581 encounters without CXR data.

Thirteen clinicians were involved in the ARDS diagnosis procedures, where each encounter was reviewed independently by 2-4 clinicians to decide the ARDS condition and, if applicable, the time of onset. ARDS diagnoses were acquired by equally weighted voting of clinicians' opinions and ties were resolved by an additional reviewer. For those ARDS cases, the time point when the earliest ARDS diagnosis was made across reviewers was recorded as ARDS onset. In total, the dataset includes

220 ARDS cases and 861 non-ARDS cases. The number of ARDS and non-ARDS cases in each cohort are presented in Table 4.1, the ratio of non-ARDS to ARDS cases is 3.13 for Cohort 1 and 4.87 for Cohort 2.

Table 4.1: Number of ARDS and Non-ARDS Encounters in Cohorts 1 and 2

	Cohort 1	**Cohort 2**	**Total**
ARDS	121	99	220
Non-ARDS	379	482	861
Total	500	581	1,081

4.2.1.1 Base Information: EHR

A subset of the patients' EHRs was used as the base information in the study of this chapter. It contains 55 time-stamped numerical features covering various vital signs and lab results. The time interval between subsequent acquisitions of certain variables may range from 15 minutes to up to 24 hours. To fill in the missing data, previous entries were carried forward to the next available ones.

The ARDS status of each encounter was evaluated at each time point based on the diagnosis time previously acquired from reviewers. Specifically, for encounters labeled as having ARDS, each time point before the onset time of ARDS was assigned a non-ARDS label. Time points obtained after ARDS onset and for the following 48 hours were labeled as ARDS. The entries after 48 hours of ARDS onset were discarded since treatment may alter the intrinsic characteristic of the data and bias the model.

4.2.1.2 Privileged Information: Mechanical Ventilation Variables

The MV information in the dataset was also time-stamped and numerical. Therefore it was processed following the same imputation and labeling protocol as the EHR data described in Section 4.2.1.1, and likewise in later sections. It includes 10 features on the patients' invasive MV status, supplemental oxygen status, the preset respira-

tory rate for the ventilator, observed respiratory rate, tidal volume (set and observed), the positive end-expiratory pressure on the ventilator, the plateau pressure, and the mean airway pressure delivered during invasive MV. Four scores on Glasgow Coma Scale are also included in the privileged domain since they are closely related to the sedation caused by invasive MV, resulting in a total of 14 features in this privileged domain.

For all encounters in Cohorts 1 and 2, we have their MV information available, however, we still consider this privileged information due to the variability in respiratory support modalities utilized in clinical practice. Table 4.2 shows the demographic information together with the MV status grouped by ARDS diagnosis. The "True" in the attribute column for "Invasive" is an indication that the encounters have received invasive MV at some point during their hospitalization. While "False" means they have never been under the invasive mechanical ventilator during their stay. The attributes in the "Non-invasive" column follow the same rule.

As shown in the table, the median age for non-ARDS cases is 61, with 62% being male and 38% being female among those documented. 42% percent of the non-ARDS cases received non-invasive treatment and 74.4% were on invasive MV during their hospital stay. For ARDS cases, the median age of encounters is 59. There are 61.20% male and 38.8% female on the record. 37.7% of the encounters received non-invasive treatment, roughly 5% less than the non-ARDS encounters. Around 90% of the ARDS encounters received invasive MV, 15% higher compared with non-ARDS encounters.

4.2.1.3 Privileged Information: Chest X-ray Image Features

All encounters in Cohort 1 have CXR information, with a total of 2758 anterior-posterior CXRs obtained. We processed the image following the protocol in [192] and acquired a total of 2216 features for each image including 72 directionality measures, 72 histogram features, and 2048 deep features. Then the Chi-square test was

Table 4.2: EHR Data Characteristics

	Attribute	Format[1]	Overall	non-ARDS	ARDS
Encounter	-	N	1,081	861	220
Age	-	Median [Q1,Q3]	61.0 [51.0,70.0]	62.0 [52.0,71.0]	59.0 [45.0,67.0]
Gender[2]	Female	N (%)	191 (38.2)	144 (38.0)	47 (38.8)
	Male		309 (61.8)	235 (62.0)	74 (61.2)
Non-invasive	False	N (%)	636 (58.8)	499 (58.0)	137 (62.3)
	True		445 (41.2)	362 (42.0)	83 (37.7)
Invasive	False	N (%)	243 (22.5)	220 (25.6)	23 (10.5)
	True		838 (77.5)	641 (74.4)	197 (89.5)

[1] N: Number of encounters; Median: the median value; Q1: the first quartile; Q3: the third quartile;
[2] The gender information is missing for Cohort 2, therefore only 500 encounters' gender statistics are included in the table.

performed to select to top 100 most important features, which formed the privileged domain for CXR images.

The CXR features were time-aligned with the EHR data via the closest time point with a maximum time mapping discrepancy of one hour. The features for 2458 chest X-rays were successfully mapped to their corresponding EHR entries and reserved for further processing.

4.2.2 Data Partition

Cohort 1 was used as the training set since they have chest X-ray images as privileged information. For Cohort 2, a data partition with a ratio of 1:2 was performed to split the data into validation and test sets. The encounter-wise split was stratified based on ARDS diagnosis results. This brings about 500 encounters for training, 191 encounters for validation, and 390 encounters for testing. Table 4.3 displays the type of data modality presented in the training, validation, and test sets, together with the number of encounters coming from non-ARDS or ARDS in each set.

Table 4.3: Data Modalities and Encounter Composition in Training, Validation, and Testing

	Data Modality			*Number of Encounters*		
	EHR	**CXR**	**MV**	**Non-ARDS**	**ARDS**	**Total**
Training	Yes	Yes	Yes	379	121	500
Validation	Yes	No	Yes	158	33	191
Testing	Yes	No	Yes	324	66	390

4.2.3 Logistic Regression Models

Let the training dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, with each $\mathbf{x}_i = [x_{1i}, \ldots, x_{di}]^T \in \mathbb{R}^d$, together with a binary response label $y_i \in \{-1, 1\} = \mathcal{Y}$. Denote by $\boldsymbol{w} = [w_1, w_2, \ldots, w_d]^T \in \mathbb{R}^d$ the coefficients and $b \in \mathbb{R}$ the offset term, and represent the whole parameter vector as $\boldsymbol{\theta} = [b, w_1, w_2, \ldots, w_d]^T \in \mathbb{R}^{d+1}$ and the augmented feature vector as $\bar{\boldsymbol{x}}_i = [1, x_{1i}, x_{2i}, \ldots, x_{di}]^T \in \mathbb{R}^{d+1}$.

The LR model for y_i consists of the following parameterized family of decision functions

$$\mathcal{F} = \left\{ f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathcal{P}(\mathcal{Y}) \;\middle|\; \boldsymbol{\theta} \in \mathbb{R}^{d+1} \right\}$$

taking values in the space $\mathcal{P}(\mathcal{Y})$ of probability distributions on the label set $\mathcal{Y}$, defined by

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{1 + \exp(-y(\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b}))} = \frac{1}{1 + \exp(-y(\boldsymbol{\theta}^T\bar{\boldsymbol{x}}))}$$

with cross-entropy loss function

$$\ell\big(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\big) = -\sum_{y \in \mathcal{Y}} \delta_{y_i}(y) \log\big(p(y|\boldsymbol{x}_i, \boldsymbol{\theta})\big) = -\log\big(p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})\big) = \log(1+\exp(-y_i(\boldsymbol{\theta}^T\bar{\boldsymbol{x}}_i))$$

and empirical ℓ-risk function

$$\hat{R}_\ell(f_{\boldsymbol{\theta}}) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i(\boldsymbol{\theta}^T\bar{\boldsymbol{x}}_i)).$$

Alternatively, we could capture the same information using real-valued deterministic

decision functions $g_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}$ by taking the expectation

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{E}\big[f_{\boldsymbol{\theta}}(\boldsymbol{x})\big] = \frac{1}{1+\exp(-\boldsymbol{\theta}^T\bar{\boldsymbol{x}})} - \frac{1}{1+\exp(\boldsymbol{\theta}^T\bar{\boldsymbol{x}})} = \frac{\exp(\boldsymbol{\theta}^T\bar{\boldsymbol{x}})-1}{\exp(\boldsymbol{\theta}^T\bar{\boldsymbol{x}})+1}, \tag{4.2}$$

which recovers the usual sigmoid decision functions taking values between ± 1 and capable of representing intermediate predictive values of labels given the coefficients.

Defining the penalty function $J(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_q^q$ as the qth power of the l_q-norm of the whole parameter vector and letting $\phi(t) = \log(1+\exp(t))$, the LR objective $h(\boldsymbol{\theta})$ under l_q penalty can be expressed as

$$\begin{aligned} h(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \log(1+\exp(-y_i(\boldsymbol{\theta}^T\bar{\boldsymbol{x}}_i))) + \lambda J(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \phi(-y_i(\boldsymbol{\theta}^T\bar{\boldsymbol{x}}_i)) + \lambda J(\boldsymbol{\theta}) \end{aligned}$$

and the optimal parameter as $\widehat{\boldsymbol{\theta}}_n = \arg\min_{\boldsymbol{\theta}} h(\boldsymbol{\theta})$

In this paper, we put our attention on the l_1 and l_2 penalties, i.e., when $J(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1 = \sum_{i=1}^{d} |\theta_i|$ and $J(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2 = \sum_{i=1}^{d} \theta_i^2$, respectively. We refer to these two models as the l_1 regularized model and the l_2 regularized model. We also compare with the case when there is no penalty or regularization, i.e., $J(\boldsymbol{\theta}) = 0$, and we refer to this model as the standard model.

4.2.4 Privileged Logistic Regression Model

When privileged information is available, we consider the two different scenarios aforementioned in Section 4.1.

1. LUPI: Privileged information is fully available. For each feature vector $\boldsymbol{x}_i \in \mathcal{X}, i \in \{1,\ldots,n\}$, there is a corresponding l-dimensional privileged feature vector $\boldsymbol{x}_i^* = [x_{1i}^*, x_{2i}^*, \ldots, x_{li}^*]^T \in \mathbb{R}^l = \mathcal{X}^*$.

2. LUPAPI: Privileged information is partially available to a limited number of entries. Defining a subset $\boldsymbol{S} \subset \{1, \ldots, n\}$ whose cardinality is $|\boldsymbol{S}| = m \leq n$, we have the privileged features $\boldsymbol{x}_j^* \in \mathcal{X}^*, j \in \boldsymbol{S}$. Fully available privileged information corresponds to the case when $m = n$ and $j = i$.

In both cases, denoting by $\bar{\boldsymbol{x}}_j^* = [1, x_{1j}^*, x_{2j}^*, \ldots, x_{lj}^*]^T \in \mathbb{R}^{l+1}$ and $\boldsymbol{\theta}^* = [b^*, w_1^*, w_2^*, \ldots, w_l^*]^T \in \mathbb{R}^{l+1}$ the augmented feature vectors and LR parameters on the privileged domain, we formulate the objective of PLR as

$$\begin{aligned} h(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = & \sum_{i=1}^{n} \phi(-y_i(\boldsymbol{\theta}^T \bar{\boldsymbol{x}}_i)) + \lambda J(\boldsymbol{\theta}) \\ & + \beta \sum_{j \in S} \phi(-y_j(\boldsymbol{\theta}^{*T} \bar{\boldsymbol{x}}_j^*)) + \lambda^* J(\boldsymbol{\theta}^*) \\ & + \xi \sum_{j \in S} (\boldsymbol{\theta}^T \bar{\boldsymbol{x}}_j - \boldsymbol{\theta}^{*T} \bar{\boldsymbol{x}}_j^*)^2 \end{aligned} \tag{4.3}$$

in which λ, β, λ^* are hyperparameters, and the corresponding optimization problem is $\min_{\boldsymbol{\theta}, \boldsymbol{\theta}^*} h(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$.

Equation (4.3) gives a multi-objective function for optimization, with three targets combined together:

- regularized LR with coefficients $\boldsymbol{\theta}$ on the base domain $\mathcal{X}$,
- regularized LR with coefficients $\boldsymbol{\theta}^*$ on the privileged domain $\mathcal{X}^*$,
- minimizing the discrepancy across the privileged and base domains.

The first two targets aim at obtaining optimal parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$ that best fit the (regularized) LR models. Hyperparameters λ and λ^* control the penalty strengths on the base and privileged domain, respectively, while β regulates the effect of the privileged-domain LR objective. The third target, on the other hand, seeks to achieve

knowledge transfer from the privileged domain $\mathcal{X}^*$ to the base domain $\mathcal{X}$, and the hyperparameter ξ determines the penalizing strength of the across-domain discrepancies on the combined objective function.

Specifically for the last term in Equation (4.3), since $\boldsymbol{\theta}^T\bar{\boldsymbol{x}}_j$ and $\boldsymbol{\theta}^{*T}\bar{\boldsymbol{x}}_j^*$ completely determine the predictive values of labels, we expect to get identical labels from the information given in $\mathcal{X}$ and $\mathcal{X}^*$ when privileged information is available. Hence imposing an l_2-norm constraint on their differences $(\boldsymbol{\theta}^T\bar{\boldsymbol{x}}_j - \boldsymbol{\theta}^{*T}\bar{\boldsymbol{x}}_j^*)$ would help to pass information across domains.

Solving the optimization problem $\min_{\boldsymbol{\theta},\boldsymbol{\theta}^*} h(\boldsymbol{\theta},\boldsymbol{\theta}^*)$ would give an optimal $\widehat{\boldsymbol{\theta}}_n$ and an optimal $\widehat{\boldsymbol{\theta}}_n^*$. In the test/inference stage when privileged information is not available, predictions would be made by Equation (4.2) using only $\widehat{\boldsymbol{\theta}}_n$.

4.2.5 Asymptotic Analysis

Following the parameter and function declaration in Section 4.1, 4.2.3 and 4.2.4, we also use $X = [X_1, X_2, \ldots, X_d]^T$, $X^* = [X_1^*, X_2^*, \ldots, X_l^*]^T$ and Y to denote random variables taking values in $\mathcal{X}$, $\mathcal{X}^*$ and $\mathcal{Y}$ following the probability distribution $p(x, x^*, y)$ on $\mathcal{X} \times \mathcal{X}^* \times \mathcal{Y}$, and similarly $\bar{X}$ and $\bar{X}^*$ for the augmented random vectors.

Under some mild conditions (given in §4.6 of [214] and §5.3 of [215]), the empirical risk minimizer $\hat{\boldsymbol{\theta}}_n$ will converge to the expected risk minimizer

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}\Big[\phi\big(-Y(\boldsymbol{\theta}^T\bar{X})\big) + \lambda J(\boldsymbol{\theta})\Big]$$

as $n \to \infty$, and our goal is to compare the rate of convergence of $\hat{\boldsymbol{\theta}}_n \to \tilde{\boldsymbol{\theta}}$, either with or without privileged information.

For simplicity, we only consider the case of standard LR (sLR) and standard privileged LR (sPLR), i.e., $J(\boldsymbol{\theta}) = 0$, with fully available privileged information. Furthermore, we restrict to the case when the parameter ξ in Equation (4.3) is suffi-

ciently small, which corresponds to the infinitesimal benefit of introducing privileged information. The main results and intuitions are presented below.

The following is a set of sufficient conditions for the sPLR model to have an infinitesimally higher rate of convergence than the sLR model:

1. If the privileged features $\bar{X}^*$ contain a principal component that is uncorrelated with the base features $\bar{X}$, and

2. if the privileged features $\bar{X}^*$ are more predictive than the base features $\bar{X}$ in the sense of
$$\begin{cases} \tilde{\boldsymbol{\theta}}^{*T}\bar{X}^* \geq \tilde{\boldsymbol{\theta}}^T\bar{X} & \text{if } Y = 1 \\ \tilde{\boldsymbol{\theta}}^{*T}\bar{X}^* \leq \tilde{\boldsymbol{\theta}}^T\bar{X} & \text{if } Y = -1 \end{cases}$$
being true almost surely, or at least with a sufficiently high probability depending on the distribution of $(\bar{X}, Y)$.

The second condition holds in our case where the privileged features (CXR or MV) are more informative than the base features (EHR), and the main issue is the unavailability of privileged information during testing.

However, there are also interesting cases where the privileged features alone may not be more predictive than the base features. In such cases, we can replace the second condition with the following:

2†. if the $(d+1) \times (d+1)$ matrix whose ijth entry equals to
$$\mathbb{E}\Big[\phi'(-Y\tilde{\boldsymbol{\theta}}^T\bar{X})(\tilde{\boldsymbol{\theta}}^T\bar{X} - \tilde{\boldsymbol{\theta}}^{*T}\bar{X}^*)YX_iX_j\Big],$$
where the indices range over $0 \leq i, j \leq d$ with the convention that $X_0 = 1$, is positive semi-definite.

This result can be deduced from the following formulas for the asymptotic rate of

convergence of $\hat{\boldsymbol{\theta}}_n \to \tilde{\boldsymbol{\theta}}$:

$$\mathbb{E}\Big[\|\hat{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}\|^2 \;\Big|\; \text{sLR} \;\Big] \sim \frac{1}{n}\text{Tr}\big[H^{-1}GH^{-1}\big]$$

and

$$\mathbb{E}\Big[\|\hat{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}\|^2 \;\Big|\; \text{sPLR} \;\Big] \sim$$
$$\frac{1}{n}\Big(\text{Tr}\big[H^{-1}GH^{-1}\big] - 4\xi\text{Tr}\big[H^{-1}KH^{-1} + 2f^2H^{-2}GH^{-1}\big] + O(\xi^2)\Big),$$

where the definition for the matrices G, H, K and further details could be found in Appendix F.

The assumption and condition suggest that providing the privileged domain contains information that is not covered by the base domain, exploiting it would benefit the PLR model by increasing the rate of convergence.

4.2.6 Experimental Setup

Table 4.4 lists the experiments performed in the study of this chapter along with the involved models for evaluation or comparison when applicable. As shown in the table, Experiment 1, using both EHR in training and testing, gives baseline performances on the base domain. Experiments 2 and 3 use MV variables as privileged information but vary the availability for privileged models. Experiment 4 combines EHR with MV information in the base domain and uses them for both training and testing, while Experiment 5 uses chest X-ray features in the privileged domain. The results and interpretations of these experiments are detailed in Section 4.3.

4.2.7 Data Processing

For Experiments 1-4, the down-sampling strategy proposed in [159] is applied to reduce the correlation between time points and generate a balanced training set

Table 4.4: Experimental Setup

	Exp #	Training	PI Availability	Testing	LR Models	SVM Model	SNN
EHR	1	EHR	-	EHR	LR	SVM	Gaussian Dropout
MV	2	EHR + MV	10% - 90%	EHR	PLR	-	-
	3	EHR + MV	100%	EHR	PLR	SVM+	Hetero Dropout
	4	EHR + MV	-	EHR + MV	LR	-	-
CXR	5	EHR + CXR	100%	EHR	PLR	SVM+	Hetero Dropout

MV: Mechanical Ventilation Information; CXR: Chest X-ray; Hetero Dropout: Heteroscedastic Dropout. PI: privileged Information; SNN: Shallow Neural Network

with regard to the non-ARDS and ARDS entries. Down-sampling was performed so that: 1) the inter-dependency of a patient's time-dependent EHR and MV features introduced by carry-forward imputation was reduced to meet the i.i.d. assumption in Section 4.2.3 and 4.2.4; 2) the total number of data samples was reduced to shorten training time, and 3) the learning process would avoid being compromised due to the expectation of some employed models of having a balanced class distribution [187]. For Experiment 5, only the time points that have CXR features mapped are kept. The validation and test sets are left untouched.

For Experiment 2, privileged information at different levels of availability was generated in the training set so that the proposed model could be trained under the LUPAPI setting. To create such a dataset, the MV data on the privileged domain are split into 10 folds with random seed and the split was stratified by patients' ARDS status. For each specific seed, privileged data were added incrementally as the available percentage goes higher. (See Figure 4.1 for an illustration.)

Moreover, in all experiments, data in the training set were normalized feature-wise to [0, 1] before model training. Parameters for normalization were recorded to transform the validation and test sets.

4.2.8 Evaluation Metrics

The performance metrics involved in model evaluation are the AUROC, sensitivity, specificity, and F1 score.

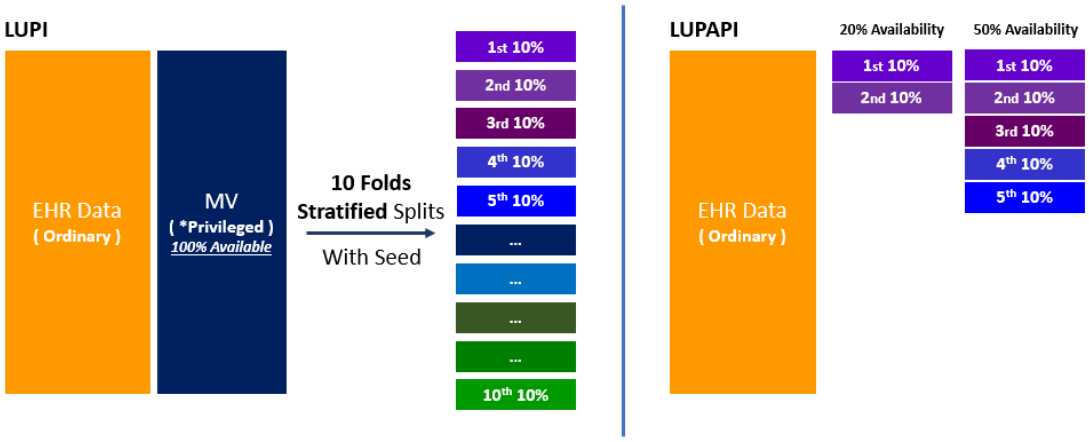


Figure 4.1: Training set generation when the privileged MV information is partially available. Left panel, the dataset in the LUPI scheme and the data split. Right panel, LUPAPI when privileged information has 20% and 50% availability.

In this chapter, specificity, and F1 scores were computed at the point on the ROC curve where sensitivity equals 70% so that different models give comparable results.

4.2.9 Training Strategy

The training/validation/test scheme stated in Section 4.2.2 is utilized in all experiments. Grid search on hyperparameters is performed on the training set over different combinations. The search space and hyperparameter details for each type of model are listed in Appendix E. The model trained with the hyperparameter combination that yields the best AUROC on the validation set is applied to the test set for final results.

For Experiment 2 where the privileged information is assigned with increasing availability as described in Section 4.2.7, repeated experiments are carried out using preset seeds in $\{1, 2, 3, 4, 5, 6\}$. Each seed gives a unique split of privileged data and the model thus generated would report a distinct test result after hyperparameter selection. In total, 6 sets of test results are obtained and an averaged measure of the performance metric is thus rendered.

For the Shallow Neural Network (SNN) models, preset seeds of $\{1, 2, 3, 4, 5, 6\}$

were used in Experiments 1, 3, and 5 for reproducible results. Moreover, different seeds provide unique initialization of the network parameters, thus yielding 6 sets of test results whose performance measures are then averaged. The SNN models were trained using the Adam optimizer with default parameter settings for up to 100 epochs, with early stopping triggered if validation loss did not improve for 15 epochs.

4.2.10 Model Implementation

The models introduced in Section 4.2.3 and 4.2.4 was implemented in Python 3.7 using solvers from the CVXPY library [216] (`https://www.cvxpy.org/`). The Gaussian and heteroscedastic dropout models on SNN were implemented in Python 3.7 and PyTorch 10.1 based on the previous version by [199]. For SVM+, the Matlab implementation from [194] was applied.

4.2.11 Explaining the Privileged Logistic Regression Results by Odds Ratio

Following the notation in Section 4.2.4, since the test results are obtained with Equation (4.2) by substituting $\widehat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$

$$p(y_i|\boldsymbol{x}_i, \widehat{\boldsymbol{\theta}}) = \frac{1}{1+\exp(-y_i(\widehat{\boldsymbol{\theta}}\bar{\boldsymbol{x}}_i))} = \frac{1}{1+\exp(-y_i(b+w_1x_{1i}+\cdots+w_dx_{di}))}. \quad (4.4)$$

Equation (4.4) can be used to interpret how different variables affect the final ARDS v.s. non-ARDS outcome by transforming it into

$$\log\left(\frac{\mathbb{P}(y=1)}{1-\mathbb{P}(y=1)}\right) = \log\left(\frac{\mathbb{P}(y=1)}{\mathbb{P}(y=-1)}\right) = b + w_1x_{1\bullet} + \cdots + w_dx_{d\bullet}.$$

Then the odds, i.e., the probability of ARDS divided by the probability of non-ARDS, is equal to

$$\text{odds} = \frac{\mathbb{P}(y=1)}{\mathbb{P}(y=-1)} = \exp(b + w_1 x_{1\bullet} + \cdots + w_d x_{d\bullet}).$$

and one unit change in the jth feature's value changes the odds ratio by

$$\begin{aligned}\frac{\text{odds}(x_{j\bullet}+1)}{\text{odds}(x_{j\bullet})} &= \frac{\exp(b + w_1 x_{1\bullet} + \cdots + w_j(x_{j\bullet}+1) + \cdots + w_d x_{d\bullet})}{\exp(b + w_1 x_{1\bullet} + \cdots + w_j x_{j\bullet} + \cdots + w_d x_{d\bullet})} \\ &= \exp(w_j(x_{j\bullet}+1) - w_j x_{j\bullet}) \\ &= \exp(w_j)\end{aligned} \tag{4.5}$$

When $w_j > 0$ and $\exp(w_j) > 1$ in Equation (4.5), an increase in $x_{j\bullet}$ would increase the odds of ARDS vs. non-ARDS with all other features held constant. When $w_j < 0$ and $\exp(w_j) < 1$, an increase in the feature's value would decrease the odds in a similar manner.

4.3 Results

4.3.1 Regularized Logistic Regression Models Show Better Testing Performances on ARDS Detection under the Classical Learning Scheme

Table 4.5 lists the test performances of Experiment 1 with standard Logistic Regression (sLR), l_1 regularized Logistic Regression (l_1 LR), l_2 regularized Logistic Regression (l_2 LR), SVM, and SNN using Gaussian Dropout (Gaussian SNN), with the Gaussian SNN results giving mean and standard deviation (std) of test metric over six different seeds as described in Section 4.2.9. Following the classical learning scheme, both training and testing within Experiment 1 were performed on the base domain using EHR data only. As shown in the table, the l_2 LR model performs the best with respect to AUROC (0.841), F1 score (0.266), and specificity (0.802). The results from l_1 LR, SVM, and Gaussian SNN models are comparable in terms of F1

score and specificity, but the Gaussian SNN model has slightly better performance with respect to AUROC. The LR model without any regularization had the lowest AUROC, F1 score, and specificity of all models.

Table 4.5: Comparison of Test Performances in Experiment 1: Classical Learning Paradigm

Models	**AUROC**	**F1 Score**	**Specificity (at 70% Sensitivity)**
sLR	0.772	0.208	0.721
l_1 **LR**	0.824	0.247	0.78
l_2 **LR**	**0.841**	**0.266**	**0.802**
SVM	0.815	0.25	0.783
Gaussian SNN	0.831±0.011	0.249±0.013	0.78±0.016

4.3.2 Privileged Logistic Regression Models are Effective and Outperform Other Methods on ARDS Detection under the LUPI Paradigm

Table 4.6: Test Performances in Experiment 3: LUPI Paradigm with MV as Privileged Information

Models	**AUROC**	**F1 Score**	**Specificity (at 70% Sensitivity)**
sPLR	0.83 (+0.058)	0.268 (+0.06)	0.804 (+0.55)
l_1 **PLR**	**0.856** (+0.032)	**0.299** (+0.052)	**0.835** (+0.055)
l_2 **PLR**	0.843 (+0.001)	0.266 (0)	0.802 (0)
SVM+ [194]	0.764 (-0.051)	0.205 (-0.045)	0.715 (-0.068)
Hetero SNN [199]	0.833±0.007 (+0.002)	0.254±0.008 (+0.005)	0.787±0.01 (+0.007)

The parenthetical numbers indicate the gain (+) or loss (-) in comparison to their counterparts under the classical learning paradigm.

Table 4.6 provides the model performances of Experiment 3, where MV variables are used as privileged information in training. Testing was carried out on EHR alone, following the LUPI paradigm. The parenthetical numbers for each in Table 4.6

Table 4.7: Test Performances in Experiment 5: LUPI Paradigm with CXR as Privileged Information

Models	**AUROC**	**F1 Score**	**Specificity (at 70% Sensitivity)**
sPLR	0.81 (+0.038)	0.242 (+0.034)	0.774 (+0.053)
l_1 **PLR**	0.831 (+0.007)	0.224 (-0.023)	0.788 (+0.008)
l_2 **PLR**	**0.851** (+0.009)	**0.278** (+0.012)	**0.815** (+0.013)
SVM+ [194]	0.792 (-0.023)	0.206 (-0.044)	0.717 (-0.066)
Hetero SNN [199]	0.827±0.007 (-0.004)	0.254±0.015 (+0.005)	0.787±0.018 (+0.007)

The parenthetical numbers indicate the gain (+) or loss (-) in comparison to their counterparts under the classical learning paradigm.

indicate the gain (+) or loss (-) in comparison to those in Table 4.5 of the classical learning paradigm.

Among the standard Privileged Logistic Regression (sPLR), l_1 regularized Privileged Logistic Regression (l_1 PLR), l_2 regularized Privileged Logistic Regression (l_2 PLR), SVM+, and SNN using Heteroscedastic Dropout (Hetero SNN), l_1 PLR achieve the best performance with respect to AUROC (0.856), F1 score (0.299), and specificity (0.835). Moreover, unregularized or regularized PLR models perform better than SVM+ and Hetero SNN in general, while the SVM+ model gives the worst performance on the task.

In terms of the performance changes, it can be seen that the sPLR models show improvements by leveraging the MV information. In terms of AUROC, there is around 6% increase in sPLR and 3% increase by using l_1 PLR. The same trend hold for F1 score and specificity, with more than 5% gain of performances on sPLR and l_1 PLR. The l_2 PLR, however, has insignificant variations in comparison to the l_2 PLR model. Moreover, using Hetero SNN slightly increase the test performances in contrast to Gaussian SNN. There are also decreases in the standard deviation, suggesting that

Hetero SNN may yield more robust models on testing. For the SVM-based models, the results given by the SVM+ model present a more than 4.5% decrease compared to those obtained with SVM.

The results for using CXR features as privileged information is illustrated in Table 4.7. Similar to Table 4.6, it covers the performance change in the parentheses with regard to the models under the classical learning paradigm. As shown in Table 4.7, the l_2 PLR model achieves the highest AUROC (0.81), F1 score (0.278), and specificity (0.815), with each of the metrics gaining around 1% compared to their l_2 LR counterpart. The l_1 PLR model has the same level of performance as Hetero SNN on AUROC and specificity, while the Hetero SNN shows a better F1 score. In addition, the performance of PLR is not the best of all but it achieves the most gain on every metric evaluated. On the contrary, the SVM+ model is inferior compared to other models and only achieves 0.792 on AUROC, 0.206 on the F1 score, and 0.717 on specificity. There are 2.3%, 4.4% and 6.6% decreases in using SVM+ instead of SVM.

4.3.3 The Proposed Privileged Logistic Regression Models are Effective in the Setting of LUPAPI

Table 4.8 mainly lists the test results of Experiment 2, summarizing the performances when models were trained with increasing availability of privileged information. The privileged information used is MV features. When the availability range from 10% to 90%, the mean and std of the evaluated metrics are presented by calculating through 6 independent repeats following the method in Section 4.2.9. Results from Experiments 1 and 3 are also included for easier comparison.

Table 4.8: Summary Results on Logistic Regression and Privileged Logistic Regression Models with Varying Availability of Privileged Information

Models	*Availability (MV as PI)*	**AUROC**	**F1 Score**	**Specificity (at 70% Sensitivity)**
sLR	0%	0.772	0.208	0.721
sPLR	10%	0.824±0.005	0.263±0.004	0.798±0.005
	20%	0.828±0.008	0.268±0.006	0.804±0.007
	30%	0.83±0.007	0.269±0.005	0.806±0.006
	40%	**0.832±0.006**	**0.271±0.005**	**0.808±0.005**
	50%	0.831±0.004	0.27±0.004	0.807±0.004
	60%	0.83±0.005	0.27±0.004	0.807±0.004
	70%	0.829±0.003	0.267±0.003	0.804±0.003
	80%	0.829±0.002	0.268±0.002	0.804±0.002
	90%	0.83±0.001	0.268±0.002	0.804±0.002
	100%	0.83	0.268	0.804
l_1 **LR**	0%	0.824	0.247	0.78
l_1 **PLR**	10%	0.849±0.002	0.277±0.005	0.813±0.005
	20%	0.847±0.002	0.272±0.004	0.809±0.005
	30%	0.849±0.001	0.277±0.002	0.814±0.002
	40%	0.848±0.002	0.274±0.003	0.811±0.004
	50%	0.849±0.001	0.275±0.003	0.812±0.003
	60%	0.848±0.002	0.275±0.003	0.812±0.003
	70%	0.848±0.001	0.276±0.001	0.813±0.001
	80%	0.849±0.001	0.276±0.001	0.813±0.001
	90%	0.855±0.001	0.293±0.001	0.829±0.001
	100%	**0.856**	**0.299**	**0.835**
l_2 **LR**	0%	0.842	0.266	0.802
l_2 **PLR**	10%	0.842±0.002	0.267±0.002	0.804±0.002
	20%	**0.844±0.003**	**0.27±0.004**	**0.806±0.004**
	30%	0.843±0.003	0.268±0.003	0.804±0.004
	40%	0.843±0.004	0.268±0.004	0.804±0.004
	50%	0.844±0.002	0.268±0.005	0.805±0.005
	60%	0.842±0.002	0.266±0.002	0.802±0.002
	70%	0.842±0.002	0.266±0.002	0.803±0.002
	80%	0.841±0.001	0.265±0.001	0.801±0.001
	90%	0.842±0.001	0.266±0.001	0.803±0.001
	100%	0.843	0.266	0.802

As shown in the table, the sPLR models achieve the best performance when the availability of privileged information is 40%. The AUROC, F1 score, and specificity increase first and then decrease. For the l_1 regularized models, the mean values of performance are rather steady as the available proportion varies from 10 - 80%. There is an obvious increase when the availability is 90% and the performance reaches its peak when all of the privileged information is present. The optimal metrics for the l_2

regularized models, on the other hand, were obtained when availability is 20%, but the variations of performance are not significant in general, especially for the AUROC and F1 scores. It's also interesting to notice that the standard deviation of testing is decreasing as availability goes higher for all the PLR-based models, indicating the models are becoming more stable when trained with more privileged information.

Comparing privileged models at a set availability, the l_1 PLR almost always performs better than the l_2 PLR and sPLR on all metrics. The l_2 PLR usually outperforms sPLR on AUROC and F1 score. This gap suggests the regularized PLR models are more predictive than those without regularization.

4.3.4 Privileged Logistic Regression Models Show Strong Ability in Knowledge Transfer

Table 4.9 specifically compares the results between Experiments 4 and 5, with the training and test data listed in the columns. Since the dataset used in this chapter contains MV-related variables in both training, validation, and testing, it can be utilized to show the ability of knowledge transfer in PLR-based models. In the upper panel, MV-related variables are only present in the training set as privileged information, and the PLR-based models were used. In the lower panel, MV-related variables are no longer regarded as privileged information. They are combined with EHR and feed into the LR-based models and the models thus obtained would be referred to as plain LR-based models in this section.

The results show that PLR-based models have either comparable or better performances than plain LR-based models. These improvements are obvious in F1 score and specificity when comparing the privileged models against plain models given a fixed regularization condition. For AUROC, sPLR and l_1 PLR both obtain higher values than their plain counterparts. Although the l_2 LR achieves the highest AUROC in the plain models and surpasses the l_2 PLR, the best-performing models among the

six still fall in the group of privileged models.

Table 4.9: Comparison of Test Performances of Logistic Regression Models between Experiments 4 and 5

Models	**Training Data**	**Test Data**	**AUROC**	**F1 Score**	**Specificity (at 70% Sensitivity)**
sPLR	EHR + MV (MV as PI)	EHR	0.83	0.268	0.804
l_1 **PLR**			**0.856**	**0.299**	**0.835**
l_2 **PLR**			0.843	0.266	0.802
sLR	EHR + MV (MV in base Domain)	EHR + MV	0.802	0.226	0.751
l_1 **LR**			0.851	**0.261**	**0.796**
l_2 **LR**			**0.854**	0.251	0.785

4.3.5 Privileged Logistic Regression Models: Interpretability

Using the l_1 PLR model to obtain $\widehat{\boldsymbol{\theta}}$, the conditions that increase the odds of ARDS v.s. non-ARDS are identified from the EHR and listed in Table 4.10. These conditions are consistent when the $\widehat{\boldsymbol{\theta}}$ is acquired from l_2 PLR model or PLR model with no regularization.

In addition, the sparsity introduced by the l_1-regularization can help to select important features from EHR in the privileged model. With the $\widehat{\boldsymbol{\theta}}$ given by the best performing l_1 PLR, the variables whose absolute values of w_i were larger than 10^{-5} were screened out and listed in Table 4.11.

4.4 Discussion

In this chapter, PLR models were proposed under the LUPI paradigm. Using either MV variables or CXR image features as privileged information, the trained models were applied to EHR for ARDS onset detection.

As shown in Section 4.3, the PLR models, compared to the LR models that were trained only on EHR, exhibited better performance on all evaluated metrics. The best-performing model that utilized MV variables as privileged information is the l_1 PLR, achieving an AUROC of 0.856, F1 score of 0.299, and 0.835 specificity at

Table 4.10: Conditions that increase the odds of ARDS v.s. non-ARDS

Higher temperature
Higher respiration rates
A higher level of unresponsiveness measured by AVPU (an acronym from "alert, verbal, pain, unresponsive")
Being sedated
On dialysis
Use of Norepinephrine / Epinephrine / Vasopressin / Phenylephrine / Dopamine / Dobutamine / Milrinone
A Higher lactate acid level obtained by blood gas
A Higher carbon dioxide level obtained by blood gas
A higher level of Na (Sodium) in the bloodstream
A higher level of K (Potassium) in the bloodstream
A higher Creatinine level in the bloodstream
Elevated Aspartate Aminotransferase (AST) level
Elevated Troponin level
Elevated brain natriuretic peptide level
A higher International Normalized Ratio (INR, used to measure clotting)

This table does not have a specific order in the listing.

70% sensitivity. When the CXR features were used as privileged information, the l_2 PLR model achieved the best performance, yielding an AUROC, F1 score, and specificity of 0.851, 0.278, and 0.815 respectively. These best-performing PLR models also displayed superior performances over other LUPI models, such as the SNN with heteroscedastic dropout and the SVM+ model. Moreover, in reviewing the results from Tables 4.6 and 4.7, it can be seen that PLR-based models can leverage both types of privileged information in ARDS detection. Though l_2 regularization does not necessarily result in performance gains, no degradation in performance was observed. These results are in contrast to the SNN-based and SVM-based models for incorporating privileged information, wherein the SNN models exhibited a minimal improvement in performance while the SVM-based model actually had *lower* performance.

Table 4.11: Important Variables in EHR for ARDS Detection Identified by the l_1 PLR Model

Temperature
Respiration rates
SpO2 (blood oxygen level)
Richmond Agitation and Sedation Scale (RASS)
Unresponsiveness measured by AVPU (an acronym from "alert, verbal, pain, unresponsive")
Sedation condition
Orientation levels
FiO2 (the fraction of inspired oxygen)
PaO2/FiO2, the recorded ratio of arterial oxygen partial pressure to fractional inspired oxygen
The calculated ratio of PaO2 v.s. FiO2
Use of Norepinephrine
Level of Na in the bloodstream
Bicarbonate level
Transfused plasma
Lactate acid level obtained by blood gas

The proposed PLR models also proved effective in the LUPAPI setting. For sPLR and l_1 PLR, a 10% presence of privileged information was sufficient to achieve significant improvements in model performance, achieving more than 5% increase using sPLR and 2-3% increase for l_1 PLR on all metrics. For the three PLR models (Table 4.8), l_1 PLR displays an increasing trend on the mean and a decreased trend on the standard deviation for performance when the availability of privileged information rises. With the help of sparsity introduced via l_1 regularization, l_1 PLR is still able to exploit information in the privileged domain until all the information is available. In contrast, sPLR and l_2 PLR reached their optimality at 40% and 20% availability, respectively, suggesting the models are saturated with a limited amount of privileged information. Given that the l_2 PLR saturates at 20% availability and constantly outperforms the sPLR model shows that has greater potential in leveraging privileged information. This could result from the use of the l_2 regularization.

Results from Table 4.9 indicate that the information contained in the privileged domain has been successfully transferred to the parameter $\boldsymbol{\theta}$ learned on the base do-

main. Therefore, when testing is performed based solely on $\boldsymbol{\theta}$ in Experiment 3, the models use data effectively as if the information from the privileged domain is present during testing (which is the case for Experiment 4). The gains in performance seem counter-intuitive since the total amount of knowledge in the base domain and the privileged domain is fixed. However, the reduced number of parameters in the privileged model may ease the overfitting problem, thus increasing model performance.

These results also provide insight into the proposed model from another perspective, in which privileged information can be seen as missing data at test time or missing data in the partially available case. One of the most common ways of handling missing data is mean imputation [217]. Under the hypothesis that MV is informative, the presence of MV variables in both training and testing is better than or at least equal to a mean imputation of non-available MV variables. Since the PLR models show improved performance consistently on F1 score and specificity compared to the model that includes MV variables as part of the base data, designating features as PI might be a preferable choice in contrast to imputation. Together with the asymptotic analysis in Section 4.2, where sufficient conditions are provided for exploiting the PI in training, the model possesses both empirical and theoretical support.

ARDS is characterized by pathological changes in the lungs that are difficult to measure continuously and objectively. Therefore, clinical criteria alone, such as timing, origin, imaging, and oxygenation, are used in the diagnostic criteria established in the 2012 Berlin definition for ARDS in adults[218]. Even with these clinical guidelines, ARDS can be difficult to distinguish from other diseases such as congestive heart failure and certain pneumonias[219]. In order to more accurately and quickly diagnose ARDS, clinical decision support systems, such as the one developed in this chapter and in [157, 162], can be employed to improve clinical outcomes. Our test results for l_1 PLR using MV as privileged information (AUROC=0.856) show an increase in performance and smaller standard deviation than the validation perfor-

mance of LR in Schmickl et al. (AUROC=0.800 $\pm$ 0.080) [162]. Likewise, our l_2 PLR model using CXR as privileged information (AUROC=0.851) outperformed the l_2 LR model test performance in Zeiberg et al. which used only EHR base information (AUROC=0.810 $\pm$ 0.075) [157]. For a disease as life-threatening and difficult to diagnose as ARDS, this increase in predictive performance is clinically significant and speaks to the power of using privileged learning models. Further work should explore additional opportunities to improve clinical decision support systems for ARDS by incorporating other kinds of privileged information.

There are three primary limitations to our current work. First, we did not employ the cross-validation/test scheme to validate our model but used the training/validation/test scheme. The primary reason for doing so is to maximize the use of the privileged information in Cohort 1 and maintain consistency on data splits across different privileged domains. This, in consequence, may reduce the confidence in the model's generalizability to an independent dataset. The data split scheme also results in the lack of statistical measures, such as mean and standard deviation, on performance metrics for the LR-based, PLR-base, and SVM-based models. However, given that the results have consistency over different experiments that were performed, we argue that the validation scheme used in this chapter should not reduce the reliability of the proposed model. Secondly, the CVXPY implementation is not efficient enough to handle large dimensional input, thus the scalability of the provided implementation is not guaranteed. Experimentally, we found that the current version is roughly 30 times slower than the Scikit-learn [220] implementation of logistic regression models. Thirdly, although the model is effective and explainable, it currently can include only one type of privileged information. The incorporation of multiple data modalities is feasible by modifying the model but would lead to more hyperparameters, followed by greater complexity in model training.

CHAPTER V

Leveraging Multi-Annotator Label Uncertainties as Privileged Information for Acute Respiratory Distress Syndrome Detection in Chest X-ray Images

5.1 Introduction

ARDS is an inflammatory lung injury characterized by diffuse alveolar damage. It occurs in critically ill patients due to various etiologies such as major trauma, pneumonia, and sepsis. As a prevalent medical condition worldwide, ARDS affects over 3 million people of all ages annually [141]. Due to the nonspecific manifestations of ARDS, patients easily go unrecognized until the severity worsens [221], which leads to a hospital mortality rate of approximately 45% [221, 222].

CXRs are key diagnostic criteria for ARDS, as the radiological presence of bilateral infiltrates is required for the definition of ARDS. CXRs usually demonstrate evidence of ARDS in the form of bilateral diffuse alveolar opacities, which may appear as consolidations as ARDS progresses. Nevertheless, the image findings may vary depending on the stage and severity of ARDS and may be subtle within the first 24 hours following the lung insult [223]. Additionally, radiological features alone are

nonspecific and may not correlate with clinical findings. As a result, poor agreements (Cohen's $\kappa < 0.27$) among clinicians on CXR interpretations for ARDS diagnosis have been reported [224, 225]. Given that ARDS has a fast-progressing nature, recognizing and treating this condition promptly is crucial for better patient outcomes. Therefore, approaches that can identify ARDS from CXR are urgently needed to provide patients with timely and evidence-based care.

Previous studies have used traditional ML and DL approaches to detect ARDS from CXR. Zaglam et al. [226] considered the image texture of intercostal patches for distinguishing between CXRs with ARDS and those without. After identifying the patches by semiautomatic segmentation of ribs, histogram features, co-occurrence matrix features, and spectral features were obtained and fed into a SVM for classification. Reamaroon et al. [192] employed SVM, random forest, and tree-based boosting classifiers to detect ARDS based on handcrafted features extracted from the entire CXRs. These features included directional-blur features that capture the cloudiness in the CXR, histogram features, co-occurrence matrix features, and features from pre-trained deep neural networks. Regarding DL approaches, Sjoding et al. [227] proposed an automatic ARDS detection network with Densenet [93], a widely used architecture for medical imaging analysis. They first pre-trained the network by supervised learning on public datasets, then fine-tuned it with ARDS images for a downstream classification task. In addition, they utilized GRAD-Cam to highlight the potential ARDS findings on CXRs through saliency maps. On the other hand, [228] developed the Dense-Ynet model for stratifying the severity of ARDS in CXR images by performing the segmentation and classification tasks simultaneously. A global ARDS severity score for the CXRs was provided based on the distribution of infiltrates in different lung quadrants.

Despite the effectiveness of existing approaches, previous research has not adequately addressed label uncertainty and label noise concerns given the high inter-

reviewer variability and poor agreements in ARDS diagnosis. For instance, Zaglam et al. [226] trained their model using image patches from only 9 CXR images without mentioning how the image level labels were generated, while Yahyatabar et al. [228] dropped images with labeling disagreements. In the studies conducted by Reamaroon et al. [192] and Sjoding et al. [227], although uncertain annotations from multiple clinicians were available in the dataset, only the mean-aggregated values were utilized as training and validation labels, potentially exposing the model to issues stemming from noisy labels.

In the field of DL for medical image analysis, several strategies [44] have been proposed to address the challenge of label noise, including label smoothing [229], network structure modification [230], and data reweighting [43, 231]. However, none of the existing approaches have fully utilized label uncertainty from multiple experts, and the prevailing practice of label averaging persists when multiple annotations are available. Notably, two studies, namely Confusion Estimation [232] and the Transfer and Marginalized (TRAM) network [233], have emerged as promising solutions to this challenge. Confusion Estimation addresses observer confusion by simultaneously estimating correct labels and annotator confusion matrices during network training. This method has demonstrated significant improvements in tasks such as natural image classification and ultrasound cardiac view classification, a medical imaging task. On the other hand, the TRAM network incorporates the annotator's information as privileged information, which is available only during training and not during inference [45]. By employing a two-branch network architecture consisting of the base and privileged branches, and updating the base feature extractor solely through the privileged branch during training, the TRAM network encourages the inclusion of knowledge from the privileged branch in the base branch during testing when the privileged branch is no longer needed. In [233], the privileged branch utilizes multi-annotator labels and one-hot encoded annotator IDs as privileged information,

leading to enhanced performance in natural image tasks. However, its applicability to medical data has not been investigated.

In this Chapter, we present a novel deep-learning model inspired by the TRAM [233] to enhance the detection of ARDS in CXR images by leveraging label uncertainty from multiple annotators as privileged information. We propose three distinct encoding methods and a simple yet effective measure of uncertainty. By incorporating a mechanism to provide the model with privileged information only when necessary and refining the privileged branch to apply ordinal regression on its output, the proposed model facilitates more effective knowledge transfer from the privileged branch to the base branch. As a result, the model achieves superior testing performance compared to the original TRAM, Confusion Estimation, and other baseline models.

The main contributions of this work can be summarized as follows:

1. We introduce a novel DL model that leverages label uncertainty from multiple annotators to enhance the discriminative performance in identifying ARDS from CXR images. This model addresses the challenge of label uncertainty in medical image analysis and provides a valuable approach to improving ARDS detection.

2. We introduce effective encoding methods and a measure of uncertainty for handling multi-annotator uncertain labels.

3. This work represents the first attempt to apply and improve the TRAM network, originally proposed for natural image tasks, to medical images, specifically for ARDS detection. By incorporating specially designed mechanisms to encourage effective knowledge transfer within TRAM, we demonstrate the potential of enhancing the model's performance compared to existing approaches and approaches that rely solely on aggregated labels.

Overall, the significance of this work lies in its approach to addressing the problem of multi-annotator label uncertainty in medical image analysis, particularly in the

context of ARDS detection in CXR images. It has the potential to mitigate the impact of label noise and improve the accuracy of ARDS detection, which can ultimately lead to more effective diagnosis and treatment of this life-threatening condition.

The remainder of this Chapter is organized as follows. Section 5.2 provides an introduction to the dataset utilized in this Chapter. Section 5.3 begins by covering the encoding methods and measure of uncertainty, followed by details on implementation, experiment setup, training strategy, and test evaluation. Section 5.4 presents the test performances of the models on all test cases or stratified test cases. In Section 5.5, we provide interpretations of the results and discuss the limitations of the current work, offering potential directions for future research.

5.2 Dataset

5.2.1 Inclusion Criteria

The study cohort was formed by retrospectively identifying adult patients admitted to intensive care units at Michigan Medicine during the period of 2016-2017 who met either of the following criteria: (1) acute hypoxic respiratory failure, defined by PaO2/FiO2 ratio $<$ 300 mm Hg while receiving invasive mechanical ventilation, or (2) moderate hypoxia, requiring more than 3 L of supplemental oxygen by nasal cannula for at least 2 hours.

These inclusion criteria were designed to encompass a diverse patient population representative of real-world clinical settings, which included patients with potential lung disease phenotypes other than ARDS. As such, the objective of the work was to accurately identify ARDS in patients presenting with a range of respiratory illnesses, rather than differentiate between healthy and ARDS patients.

Table 5.1: Number of Patients and CXR Images (ARDS and Non-ARDS) in Training and Testing Sets. All Numbers Shown Are Counts.

	Patient	**ARDS CXRs**	**Non-ARDS CXRs**	**Total CXR**
Train	333	606	1,444	2,050
Test	167	327	678	1,005
Total	500	933	2,122	3,055

5.2.2 Characteristics

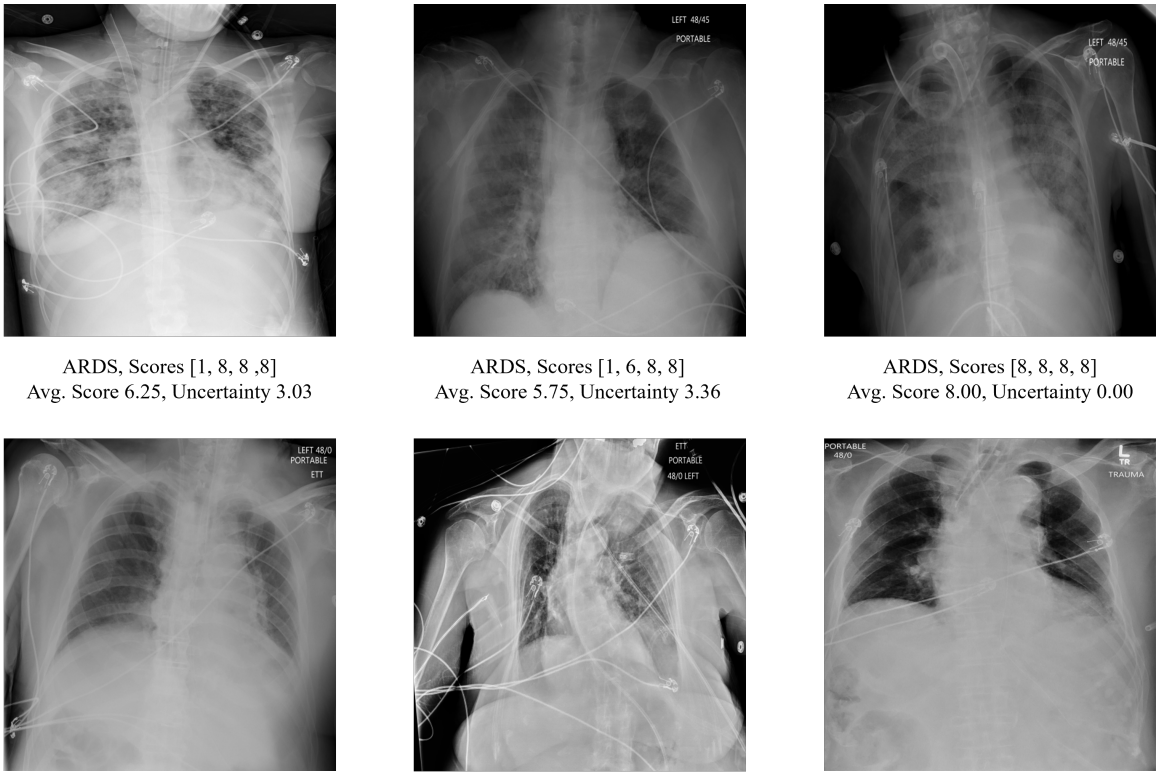
Examples of CXRs in this dataset are shown in Fig. 5.1. Since the CXRs were obtained from hospitalized settings, they exhibit a wide range of variations and complexities. These include variations in image quality such as dynamic range and sharpness, the presence of medical devices or implants, and the manifestation of the disease itself. In total, the cohort consisted of 3,055 anteroposterior (AP) CXRs from 500 patients. As depicted in Table 5.1, 2,050 CXRs from 333 patients admitted in 2016 were used in training, while 1,005 CXRs from 167 patients admitted in 2017 were designated as the hold-out test set, with no patient overlap in the data split. Among these 500 patients, 309 were male and 191 were female. The average age of the patients was 57.65 years, with a standard deviation of 16.32 years. Further information regarding patient demographics can be found in Table 5.2.

Table 5.2: Demographics of Patients.

	Patients (N)	**Age (yrs)**
Male	309	57.16 ± 16.72
Female	191	58.46 ± 15.71
Total	500	57.65 ± 16.32

5.2.3 Label Scheme

Fourteen physicians trained in critical care medicine independently evaluated the CXRs, with each image receiving two to four evaluations. The evaluations primarily relied on the presence of bilateral opacities, supplemented by reviewing other clinical



ARDS, Scores [1, 8, 8 ,8]
Avg. Score 6.25, Uncertainty 3.03

ARDS, Scores [1, 6, 8, 8]
Avg. Score 5.75, Uncertainty 3.36

ARDS, Scores [8, 8, 8, 8]
Avg. Score 8.00, Uncertainty 0.00

Non-ARDS, Scores [1, 2, 3, 8]
Avg. Score 3.50, Uncertainty 3.44

Non-ARDS, Scores [1, 1, 2, 2]
Avg. Score 1.50, Uncertainty 1.00

Non-ARDS , Scores [1, 1, 1, 1]
Avg. Score 1.00, Uncertainty 0.00

Figure 5.1: The upper panel displays CXR scans of patients diagnosed with ARDS, while the lower panel shows scans of patients without ARDS. The score array represents the annotation score provided by multiple reviewers, together with the averaged score and the corresponding measurement of uncertainty. (defined in Section 5.3.2)

information during the patient's hospitalization. As illustrated in Figure 5.2, the physicians used an ordinal scale ranging from 1 to 8 to rate the presence of ARDS, with a rating of 1 indicating high confidence that the CXR did not show ARDS, a rating of 8 indicating high confidence of ARDS presence, and a rating of 4 or 5 indicating equivocal findings. Detailed information regarding the distribution of the number of reviewers per image and the total images reviewed per reviewer can be found in Appendix G

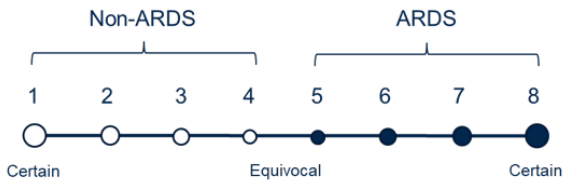


Figure 5.2: Diagram for different labeling scores on a scale of 1 to 8. Solid circles indicate diagnoses of ARDS, while empty ones represent non-ARDS. The size of the circles represents the certainty level of an assigned score.

5.2.4 Label Agreement

To assess the agreement in labeling among different reviewers, the evaluations provided by each reviewer were binarized by applying a threshold of 4.5 to the annotated score. Cohen's κ coefficient was subsequently computed between each pair of reviewers based on the images that were reviewed by both reviewers. In cases where there were no shared images for a specific pair of reviewers, the resulting κ value was set as NaN (Not a Number). The mean Cohen's κ value is around 0.366, indicating only a fair level of agreement between reviewers [234]. Figure 5.3 displays a heatmap depicting the pairwise Cohen's κ values among the 14 reviewers.

5.2.5 Mean Label Aggregation

The CXR labels, y, were determined by averaging the annotated scores assigned by different physicians. If the average score was below 4.5, the CXR was labeled as non-ARDS; otherwise, it was labeled as ARDS. By employing this approach, a total of 933 CXR images were identified as meeting the criteria for ARDS, while 2122 images were labeled as non-ARDS. As listed in Table 5.1, there were 606 ARDS CXR images and 1444 non-ARDS images within the training set. In the holdout test set, the numbers stood at 327 ARDS CXR images and 687 non-ARDS images. However, due to the high level of label disagreements among the physicians, this approach inherently introduced noisy labels. In Section 5.3, methods would be introduced to measure the uncertain levels associated with these labels and to provide a more

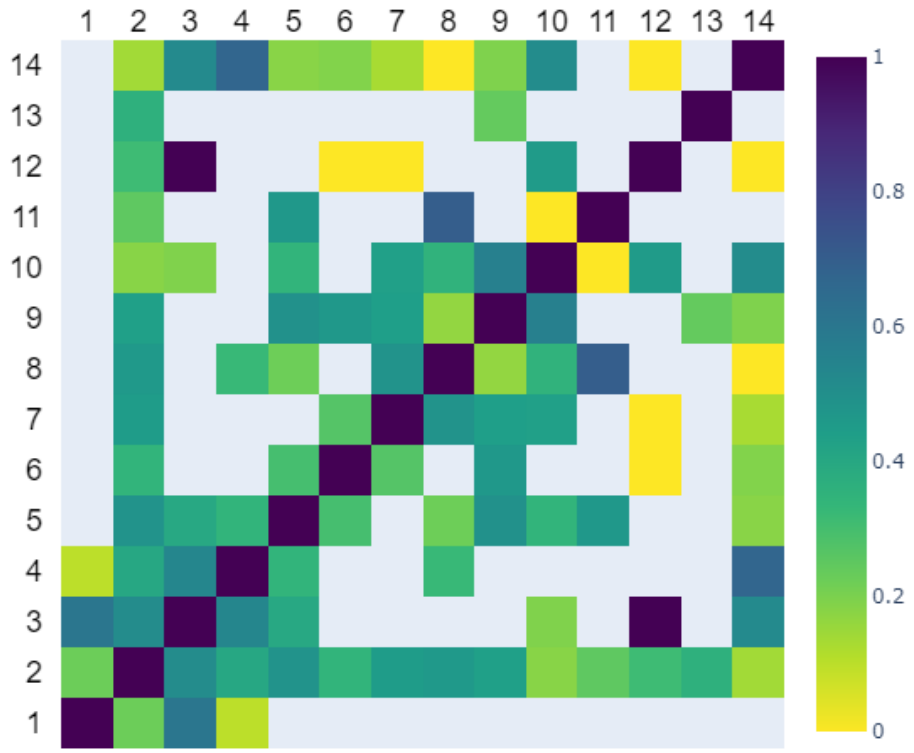


Figure 5.3: Cohen Kappa score between pairs of 14 independent reviewers' agreement of ARDS diagnosis from CXR images

reliable assessment of the labels during the testing phase.

5.3 Method

5.3.1 Encoding of Multi-Annotator Information

Assuming there are k annotations on a CXR image x, where $k \in \{2, 3, 4\}$, each annotation is represented by an annotation score $S_i \in \mathcal{S} = \{1, 2, \ldots, 8\}$, and is associated with a reviewer's ID denoted by $T_i \in \mathcal{T} = \{1, 2, 3, \ldots, 14\}$. In this context, a sequence $\{(S_1, T_1), \ldots, (S_k, T_k)\}$ corresponds to the annotation scores provided by k reviewers, with each S_i linked to the respective reviewer's ID, T_i. To illustrate this, consider a sequence $\{(6, 8), (2, 6), (2, 12)\}$, which represents three reviewers with reviewer IDs 8, 6, and 12, and corresponding annotation scores of 6, 2, and 2, re-

spectively. In order to incorporate this information into the training of our proposed methods, we introduce three encoding protocols as follows.

- **Score Encoding** (Score. E.): Only the annotation score is encoded. The encoder vector is represented as $\mathbf{E} = [E_1, E_2, \ldots, E_7, E_8] \in \mathbb{R}^8$, where each element

$$E_s = \sum_{i=1}^{k} \begin{cases} 1, & \text{if } S_i = s \\ 0, & \text{otherwise} \end{cases}, \quad s \in \mathcal{S}$$

  The value of E_s represents the count of occurrence of the corresponding score s among the k annotations. $\mathcal{S}$ represents the set of all possible scores.

- **Separate Encoding** (Separ. E.): Both the annotation scores and the annotator IDs are encoded. The encoder vector for annotation scores is the same as that in the Score Encoding, while the one for annotator ID is represented as $\mathbf{A} = [A_1, A_2, \ldots, A_{13}, A_{14}] \in \mathbb{R}^{14}$, where

$$A_t = \begin{cases} 1, & \text{if } t \in \{T_1, \ldots, T_K\} \\ 0, & \text{otherwise} \end{cases}, \quad t \in \mathcal{T}$$

  and $\mathcal{T}$ represents the set of all possible annotator IDs. $\mathbf{A}$ and $\mathbf{E}$ are then concatenated to form the final encoder vector in $\mathbb{R}^{22}$.

- **Combine Encoding** (Comb. E.): Both the annotation score and the annotator ID are encoded. The encoder vector is $\mathbf{C} = [C_1, C_2, \ldots, C_{13}, C_{14}] \in \mathbb{R}^{14}$. If an annotation T_i is provided by annotator t with score S_i, then C_t takes the value

of S_i. Otherwise, it is assigned a value of 0. The formulation is

$$C_t = \begin{cases} S_i, & \text{if } t \in \{T_1, \ldots, T_K\} \\ 0, & \text{otherwise} \end{cases}, \quad t \in \mathcal{T}$$

5.3.2 Measure of Uncertainty

The uncertainty of an ARDS diagnosis from a CXR image arises from two sources: the annotation score provided by a reviewer and the agreements or disagreements among reviewers. As discussed earlier, a rating of 1 or 8 indicates higher certainty from the physician regarding the presence or absence of ARDS findings in the CXR. However, uncertainty is not solely dependent on the annotation score but also on the level of agreement between reviewers. Higher reviewer disagreements generally indicate a higher level of uncertainty for a given case.

Therefore, following the notions described in the previous section, we have designed the following measure of uncertainty

$$D = \frac{1}{k} \sum_{i}^{K} g(S_i) + \sigma_{S_1,\ldots,S_k}$$

to quantify the uncertainty at the image level, where $\sigma_{S_1,\ldots,S_k}$ is the standard deviation component that takes into account the variability in the scores S_i assigned by different reviewers and the function $g(s) : \mathcal{S} \to \mathbb{R}$ is defined as:

$$g(s) = -|s - 4.5| + 3.5, s \in \mathcal{S}$$

The function $g(s)$ captures the degree of uncertainty associated with each annotation score. It assigns lower values of uncertainty to scores farther away from the threshold of 4.5, with a minimum value of 0 when the score is at the extremes of 1 or 8.

Table 5.3 presents the summary statistics of D on the training and testing sets,

providing a comprehensive overview of its distribution and variability. In the training set, the mean measurement of uncertainty is 1.95 with a standard deviation of 1.29, ranging from a minimum of 0.00 to a maximum of 3.92. The 25th percentile (Q1) is 1.00, the median is 2.00, and the 75th percentile (Q3) is 3.25. Similarly, in the testing set, the mean measurement of uncertainty is 1.86 with a standard deviation of 1.23, ranging from 0.00 to 3.83. The 25th percentile, median, and 75th percentile values are consistent with those of the training set.

Although D may not serve as an unbiased estimator as those in [235] and [43], it is proved to be effective when combined with the thresholding mechanism described in Section 5.3.5. This combination successfully promotes knowledge transfer in the proposed models.

Table 5.3: Summary Statistics of Uncertainty Measurement on Training and Testing Sets

Uncertainty D	Mean	Std	Min	Q1[a]	Median	Q3[b]	Max
Training	1.95	1.29	0.00	1.00	2.00	3.25	3.92
Testing	1.86	1.23	0.00	1.00	2.00	3.22	3.83

[a] 25th Percentile, [b] 75th Percentile

5.3.3 Supervised Per-Trained Encoder

The encoder used in the work in this Chapter is a ResNet50 [91] model pretrained using supervised learning and the weight is obtained from the TorchXrayVision repository (via https://github.com/mlmed/torchxrayvision/) [236]. By leveraging the knowledge learned from diverse publicly available datasets, including the RSNA Pneumonia Challenge (https://www.kaggle.com/c/rsna-pneumonia-detection-challenge), NIH Chest X-ray8 [237], PadChest [238], CheXpert [239], and MIMIC-CXR datasets [240], the pretrained ResNet50 encoder provides a strong foundation for our model to extract meaningful features from the CXR images.

To ensure compatibility with the pretrained encoder, the CXR images are pro-

cessed accordingly. They are resized to a dimension of 512 × 512 and then normalized to a range of [-1024, 1024].

5.3.4 Proposed Method

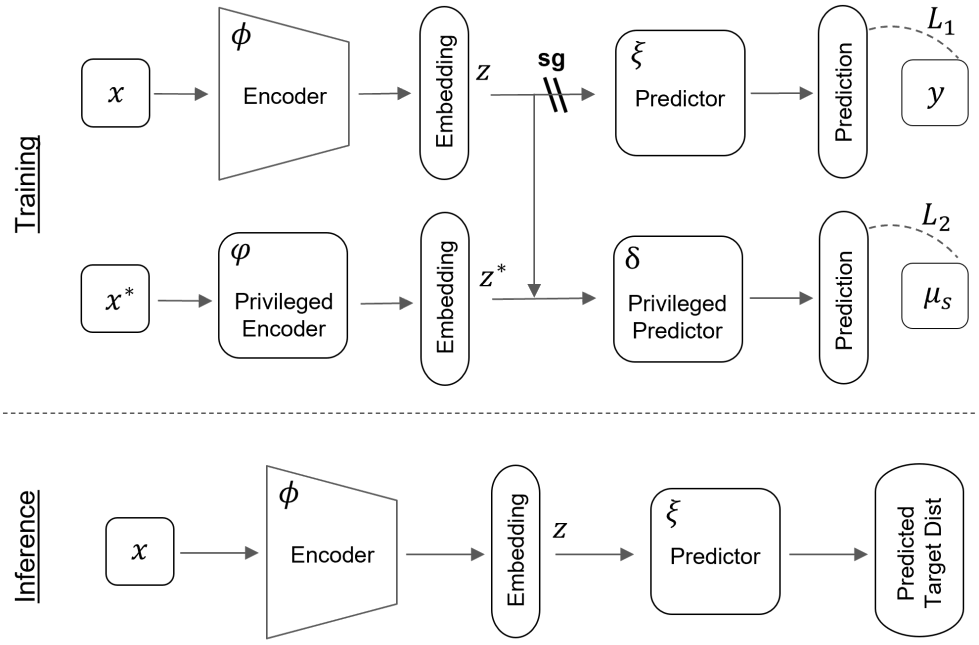


Figure 5.4: Diagram of the Training and Inference Network Structure.

Figure 5.4 depicts the diagrammatic representation of the proposed method, which incorporates two branches in its network architecture. The base branch comprises an encoder labeled as ϕ and a predictor denoted as ξ. The privileged branch, on the other hand, consists of a privileged encoder represented as φ and a privileged predictor labeled as δ.

In training, the encoder $\phi(x)$ generates an embedding from the input CXR images x. The resulting embedding, denoted as z, serves two purposes. Firstly, it is passed to the predictor $\xi(z)$ on the base branch for the primary task. Secondly, it is concatenated with the privileged annotation information encoded as $z^* = \varphi(x^*)$ and utilized by the predictor $\delta(z^*, z)$ on the privileged branch. With a Stop Gradient (sg) operator applied to the base branch, the updates on the encoder ϕ occur through the

privileged branch, allowing the privileged information to influence and modify the encoder. This mechanism, initially introduced in [233] and named TRAM, enables the exploitation of the privileged annotation information x^* to enhance the learning process of the base branch. During the testing phase, only the base branch is retained for making predictions.

The TRAM mechanism may encounter limitations when the annotation information obtained from the privileged branch contains excessive details about the target label [241]. This situation arises as the model may heavily rely on the embedded score annotations z^* in conjunction with $\delta(z^*, z)$ on the privileged branch, overshadowing the importance of learning associations between the input data x and the target label through the encoder ϕ. Consequently, the models may struggle to generalize well to unseen examples or exhibit limited performance in the testing phase, where privileged information is not available. Therefore, it is crucial to strike a balance in utilizing the privileged information while ensuring that the base branch also learns from the input data to obtain an encoder that produces robust and meaningful representations.

Two strategies were proposed to address the aforementioned issue. Firstly, the model is provided with the privileged annotation information, which is encoded as described in Section 5.3.1, only when the measure of uncertainty, introduced in Section 5.3.2, exceeds a certain threshold. When the uncertainty falls below this threshold, an all-zero vector is used as a substitute for the privileged annotation information. By employing this strategy, the model is encouraged to utilize the multi-annotator privileged information primarily in cases where the label may be noisy, while also promoting the learning of associations between clean samples and their corresponding labels within the encoder. Secondly, instead of using binarized labels as the prediction target for the privileged branch, a rank-consistent ordinal prediction approach is employed, where the averaged scores among annotators are rounded up and used as targets. This approach creates a more nuanced prediction target that effectively

captures the ordinal nature of the labels. Together with the thresholding mechanism, it will further encourage the learning of the clean instances during network training and help with knowledge transfer across branches.

5.3.5 Implementation Details and Training Logic

The models described below were implemented using PyTorch 1.10 and Python 3.7. The experiments were conducted on two Tesla V100 GPUs, each equipped with 16 GB of memory.

In our proposed model, the encoder ϕ is a ResNet50 model (described in Section 5.3.3) with its final prediction layer removed. The resulting embeddings have a dimension of 2048. For the predictor ξ, we employ a linear layer to map the embeddings to the 2-dimensional output. On the privileged branch, the privileged encoder φ is implemented as a linear layer with 64 units, followed by batch normalization and ReLU activation to facilitate effective information flow and non-linearity. The output of φ is then concatenated with the embeddings z and passed into the privileged predictor, which is a network consisting of two layers. Each layer has 128 units, with batch normalization and ReLU activation applied between the layers. Notably, the final layer of the privileged predictor is specifically modified to align with a rank-consistent ordinal regression framework known as CORN [242], which has proved its efficiency on various datasets.

The loss function $\mathcal{L}$ for training the network is defined as follows:

$$\mathcal{L} = \mathcal{L}_1 \left(\xi \left[\text{sg}(\phi(x)) \right], y \right) + \beta \mathcal{L}_2 \left(\delta \left[\varphi(x^*), \phi(x) \right], \mu_S \right)$$

Here, $\mathcal{L}_1$ represents the cross-entropy loss on the base branch, which measures the discrepancy between the predicted label distribution and the mean aggregated labels y. The function $\text{sg}(\cdot)$ denotes the stop gradient operation applied to the output of

the encoder $\phi(x)$, ensuring that no gradients flow through the base branch during backpropagation. On the other hand, $\mathcal{L}_2$ represents the CORN loss on the privileged branch. The target values $\mu_S \in \mathcal{S}$ are obtained by averaging the scores provided by multiple annotators and rounding them to the closest integer. The weight parameter β determines the relative significance of the privileged branch loss compared to the base branch loss. In the experiments, β was consistently set to 0.5, considering that the search for the learning rate could adequately incorporate the impact of both losses, as mentioned in [233].

In addition, the following models were implemented to provide a basis for comparison. The encoder architectures in these models remain unchanged from the previous description. In addition, the three encoding methods were independently applied in experiments for models that require annotator information encoding to evaluate the effectiveness.

1. **Linear Probing**: The encoder is frozen, and the predictor ξ is a linear layer with an input feature size of 2048 and an output dimension of 2. The objective is to minimize the cross-entropy loss between the predicted label distribution and the mean-aggregated labels.

2. **Fine Tuning**: The predictor architecture and the loss remain the same as in Linear Probing, but with a trainable encoder.

3. **Confusion Estimation**: This model follows the same architecture as Linear Probing, but introduces trainable confusion matrices specific to each of the 14 reviewers. The prediction targets are obtained by binarizing the scores provided by each reviewer, using a threshold of 4.5. Other details follow the approach outlined in [232].

4. **TRAM**: The architecture is identical to the proposed model, except that no thresholding (Thresh.) is applied when supplying privileged annotation infor-

mation, and no ordinal regression (Ord.Reg.) is used in the privileged branch.

5. **TRAM w/ Thresh.**: This model builds upon the TRAM framework but incorporates the thresholding mechanism, where a threshold is applied to determine whether to use privileged annotation information.

6. **TRAM w/ Ord. Reg.**: This model builds upon the TRAM framework but has the privileged branch that uses an ordinal regression for prediction.

The Adam optimizer [243] with default parameters was used in all the conducted experiments. The hyperparameters of interest were the learning rates for the encoder, denoted as α, and the learning rate β for the rest of the network. To determine the optimal learning rates, a grid search was performed over the values $\alpha \in \{1e-4, 4e-4, 1e-5, 5e-5, 1e-6\}$ and $\beta \in \{1e-3, 5e-3, 1e-4, 5e-4, 1e-5\}$. To ensure consistency in comparing encoding methods and avoid extensive hyperparameter tuning, the threshold level that distinguishes between more uncertain and less uncertain cases was not considered a hyperparameter in our experiment. Instead, the median value of 2 was selected as the threshold based on the statistics provided in Table 5.3. However, the choice of threshold level did have an impact on the performance of the proposed models. As the threshold increased from 0 to its maximum, the validation and testing performance initially improved and then started to decline. Details on the influence of applying different thresholds to validation and testing outcomes are listed in Appendix H.

Two separate random seeds were utilized to carry out the experiments. The first seed was employed for hyperparameter selection, where the model was trained using three-fold cross-validation on the training set. The data splits were performed in a patient-wise manner. Each fold was trained up to 40 epochs using a batch size of 64, and early stopping was triggered if the validation loss did not decrease for 10 consecutive epochs. Among the models trained on each fold, the one with the lowest

validation loss was identified as the optimal model. By calculating the mean statistics of the validation loss across the optimal models from all three folds, we were able to determine the optimal combination of hyperparameters. The second seed was used to repeat the three-fold cross-validation process with the optimal hyperparameters. The optimal models obtained from each fold were applied to the holdout test set, and the mean test metrics and standard deviation were reported.

Furthermore, while the training loss, target label, and network architecture may differ among different methods, the validation process was consistently conducted on the same architecture depicted in the lower panel of Figure 5.4. This architecture utilized mean-aggregated labels and cross-entropy loss. To ensure the reliability of the validation set within each fold and prevent potential misleading results, only cases with an uncertain level of 2 or lower were included in the validation set after their assignment during cross-validation. This filtering process ensured that the validation set consisted of cases with relatively low uncertain levels.

5.3.6 Test Evaluation

The performance metrics involved in model evaluation are precision, accuracy, AUROC, Area Under the Precision-Recall Curve (AUPRC), sensitivity, specificity, and F1 score.

Due to the absence of gold standard labels for the test set, we employed two evaluation approaches. The first approach utilized mean-aggregated labels, while the second approach categorized the predictions based on their uncertainty into two distinct ranges, [0, 2) and [2, 4), which allows us to analyze the model's performance over different levels of uncertainty. We paid more attention to cases with lower uncertainty, as they were more likely to have accurate labels.

Table 5.4: Testing Performances Across Different Methods on Test Set with Mean Aggregated Labels

(a). Baselines	**Precision**	**Accuracy**	**AUPRC**	**AUROC**	**Sensitivity**	**Specificity**	**F1 Score**
Linear Probing	0.768 ± 0.006	0.771 ± 0.006	0.838 ± 0.010	0.850 ± 0.008	0.776 ± 0.007	0.765 ± 0.006	0.772 ± 0.006
Fine-tuning	0.771 ± 0.015	0.772 ± 0.010	0.855 ± 0.008	0.856 ± 0.012	0.775 ± 0.014	0.769 ± 0.009	0.773 ± 0.012
Confusion Estimation [232]	0.785 ± 0.010	0.788 ± 0.009	**0.870 ± 0.001**	0.871 ± 0.001	0.794 ± 0.008	0.782 ± 0.010	0.789 ± 0.009
TRAM [233] + Score. E.	0.763 ± 0.016	0.766 ± 0.016	0.835 ± 0.018	0.842 ± 0.017	0.773 ± 0.015	0.760 ± 0.016	0.768 ± 0.015
TRAM [233] + Separ. E.	0.758 ± 0.012	0.762 ± 0.013	0.836 ± 0.018	0.842 ± 0.015	0.767 ± 0.013	0.756 ± 0.012	0.763 ± 0.013
TRAM [233] + Comb. E.	0.764 ± 0.014	0.770 ± 0.014	0.834 ± 0.015	0.846 ± 0.012	0.781 ± 0.013	0.758 ± 0.014	0.772 ± 0.014
TRAM w/ Thresh. + Score. E.	0.785 ± 0.013	0.788 ± 0.014	0.860 ± 0.011	0.866 ± 0.009	0.795 ± 0.016	0.782 ± 0.013	0.790 ± 0.015
TRAM w/ Thresh. + Separ. E.	**0.792 ± 0.011**	**0.796 ± 0.012**	0.866 ± 0.016	**0.872 ± 0.011**	**0.802 ± 0.014**	**0.789 ± 0.010**	**0.797 ± 0.012**
TRAM w/ Thresh. + Comb. E.	0.786 ± 0.031	0.789 ± 0.033	0.859 ± 0.020	0.865 ± 0.021	0.796 ± 0.034	0.783 ± 0.031	0.791 ± 0.033
(b). Proposed Models	**Precision**	**Accuracy**	**AUPRC**	**AUROC**	**Sensitivity**	**Specificity**	**F1 Score**
TRAM w/ Ord. Reg. + Score. E.	0.790 ± 0.007	0.790 ± 0.007	0.852 ± 0.017	0.861 ± 0.014	0.789 ± 0.008	0.790 ± 0.007	0.789 ± 0.007
TRAM w/ Ord. Reg. + Separ. E.	0.792 ± 0.009	0.792 ± 0.009	0.852 ± 0.016	0.860 ± 0.013	0.791 ± 0.009	0.792 ± 0.009	0.791 ± 0.009
TRAM w/ Ord. Reg. + Comb. E.	0.780 ± 0.014	0.780 ± 0.013	0.850 ± 0.014	0.858 ± 0.012	0.779 ± 0.012	0.780 ± 0.015	0.780 ± 0.013
Proposed + Score. E.	**0.798 ± 0.007**	**0.797 ± 0.006**	**0.868 ± 0.012**	**0.873 ± 0.010**	**0.796 ± 0.006**	**0.798 ± 0.007**	**0.797 ± 0.006**
Proposed + Separ. E.	0.796 ± 0.008	0.795 ± 0.007	0.864 ± 0.015	0.871 ± 0.012	0.793 ± 0.006	0.796 ± 0.008	0.794 ± 0.007
Proposed + Comb. E.	0.789 ± 0.003	0.789 ± 0.003	0.863 ± 0.014	0.868 ± 0.010	0.788 ± 0.004	0.789 ± 0.003	0.789 ± 0.003

5.4 Results

5.4.1 Performance Analysis and Comparison of Baseline and Proposed Models on Test Set with Mean Aggregated Labels

Table 5.4 presents the testing statistics for various models. The model names and the encoding methods employed adhere to the abbreviations outlined in Section 5.3.1 and 5.3.5. The upper panel (Table 5.4.(a)) showcases the performance of the baseline models, while the lower panel highlights the results of the proposed models incorporating ordinal regression in the privileged task. The best-performing metric in each panel is indicated in bold. Among all the tested models, the proposed network with Score Encoding achieved the highest precision, accuracy, AUROC, specificity, and F1 score. On the other hand, TRAM with Thresholding and Separate Encoding attained the highest sensitivity, whereas the Confusion Estimation model yielded the highest AUPRC.

For tested baselines, the Confusion Estimation model achieved the highest AUPRC. However, the TRAM models with the threshold mechanism and Separate Encoding surpassed other models across all the rest metrics. It achieved a precision of 0.792, accuracy of 0.796, AUROC of 0.872, sensitivity of 0.802, specificity of 0.789, and F1

score of 0.797. While the performance differences between the Confusion Estimation and the TRAM models may not be statistically significant based on the error bars defined by the standard deviation, the presence of the thresholding mechanism had a notable impact on the TRAM models' testing performance, resulting in a 2% - 3% increase across all metrics compared to the original TRAM models.

Moreover, the performance of linear probing was found to be comparable to that of fine-tuning, with only a slight difference observed in AUPRC. This indicates that the self-supervised pretrained encoder is capable of capturing meaningful embeddings from the CXR images. It also suggests that fine-tuning may lead to overfitting the training data, resulting in similar performances to linear probing. Additionally, it is worth noting that disregarding the thresholding mechanism in TRAM can have a detrimental effect on testing performances. When comparing with fine-tuning and linear probing as baselines, the original TRAM model performs unfavorably on most metrics, except for sensitivity and F1 score when utilizing Combine Encoding.

Among the proposed models in Table 5.4.(b), the best results were achieved when using the proposed model with Score Encoding. This approach yielded a precision of 0.798, accuracy of 0.797, AUPRC of 0.868, AUROC of 0.873, sensitivity of 0.796, specificity of 0.798, and F1 score of 0.797. What's more, although all the models in panel (b) incorporated ordinal regression in the privileged branch, the proposed models that additionally utilized the thresholding mechanism demonstrated a performance improvement of up to 1% in all the testing metrics compared to those without the thresholding mechanism. This finding reinforces the importance of the thresholding mechanism in achieving effectiveness in TRAM models.

By comparing the results in panel (a) and panel (b) of Table 5.4, we can observe that incorporating ordinal regression in the privileged branch leads to performance improvements ranging from 1% to 3% in the testing stage, particularly when the thresholding mechanism is not present. Furthermore, the use of ordinal regression

helps mitigate the performance drop in the absence of the thresholding mechanism in TRAM. When comparing the proposed model in panel (b) with TRAM models that incorporates the thresholding technique in panel (a), we observed performance enhancements in terms of precision, accuracy, AUPRC, AUROC, and specificity, while achieving similar performance levels in the F1 score. Although most of these improvements are subtle, there is a noticeable decrease in the standard deviation for the proposed models compared to the TRAM models regardless of the encoding applied. This decrease in standard deviation indicates better stability and robustness of the proposed models across the three-fold cross-validation.

5.4.2 Performance Evaluation on Stratified Testing Set: Clean and Equivocal Test Cases

Table 5.5: Testing Performances Across Different models on the Test Set Stratified by Uncertainty.

(a). Uncertainty ∈ [0, 2), n=477	**Precision**	**Accuracy**	**AUPRC**	**AUROC**	**Sensitivity**	**Specificity**	**F1 Score**
Linear Probing	0.882 ± 0.009	0.886 ± 0.007	0.955 ± 0.006	0.958 ± 0.005	0.892 ± 0.005	0.881 ± 0.009	0.887 ± 0.007
Fine-tuning	0.887 ± 0.006	0.889 ± 0.007	0.956 ± 0.003	0.956 ± 0.004	0.892 ± 0.008	0.887 ± 0.006	0.890 ± 0.007
Confusion Estimation [232]	0.900 ± 0.004	0.903 ± 0.005	0.965 ± 0.002	0.965 ± 0.002	0.907 ± 0.007	0.899 ± 0.004	0.904 ± 0.005
TRAM w/ Thresh. + Score. E.	0.908 ± 0.008	0.911 ± 0.008	0.961 ± 0.009	0.965 ± 0.008	0.914 ± 0.008	0.907 ± 0.008	0.911 ± 0.008
TRAM w/ Thresh. + Separ. E.	0.911 ± 0.007	0.914 ± 0.007	0.966 ± 0.011	0.969 ± 0.007	0.918 ± 0.008	0.910 ± 0.007	0.914 ± 0.007
TRAM w/ Thresh. + Comb. E.	0.902 ± 0.035	0.905 ± 0.035	0.962 ± 0.012	0.964 ± 0.012	0.909 ± 0.034	0.901 ± 0.035	0.906 ± 0.034
Proposed + Score. E.	**0.921 ± 0.006**	**0.921 ± 0.005**	**0.971 ± 0.005**	**0.973 ± 0.003**	**0.920 ± 0.004**	**0.921 ± 0.006**	**0.921 ± 0.005**
Proposed + Separ. E.	0.920 ± 0.008	0.920 ± 0.008	0.969 ± 0.007	0.972 ± 0.005	0.919 ± 0.009	0.920 ± 0.008	0.920 ± 0.008
Proposed + Comb. E.	0.915 ± 0.003	0.915 ± 0.004	0.969 ± 0.005	0.971 ± 0.003	0.915 ± 0.004	0.915 ± 0.003	0.915 ± 0.004
(b). Uncertainty ∈ [2, 4], n=528	**Precision**	**Accuracy**	**AUPRC**	**AUROC**	**Sensitivity**	**Specificity**	**F1 Score**
Linear Probing	0.664 ± 0.005	0.666 ± 0.006	0.693 ± 0.008	0.724 ± 0.008	0.672 ± 0.008	0.660 ± 0.003	0.668 ± 0.006
Fine-tuning	0.666 ± 0.013	0.667 ± 0.013	0.720 ± 0.013	0.731 ± 0.010	0.670 ± 0.014	0.664 ± 0.013	0.668 ± 0.014
Confusion Estimation [232]	0.681 ± 0.015	0.684 ± 0.013	**0.737 ± 0.006**	**0.748 ± 0.003**	0.691 ± 0.009	0.677 ± 0.017	0.686 ± 0.012
TRAM w/ Thresh. + Score. E.	0.675 ± 0.019	0.678 ± 0.021	0.715 ± 0.015	0.732 ± 0.011	0.688 ± 0.026	0.669 ± 0.017	0.681 ± 0.022
TRAM w/ Thresh. + Separ. E.	0.685 ± 0.015	**0.689 ± 0.016**	0.718 ± 0.022	0.738 ± 0.018	**0.698 ± 0.019**	0.680 ± 0.014	**0.691 ± 0.017**
TRAM w/ Thresh. + Comb. E.	0.681 ± 0.031	0.685 ± 0.033	0.715 ± 0.021	0.736 ± 0.026	0.694 ± 0.035	0.676 ± 0.030	0.687 ± 0.033
Proposed + Score. E.	**0.686 ± 0.009**	0.686 ± 0.009	0.715 ± 0.017	0.737 ± 0.018	0.684 ± 0.009	**0.687 ± 0.009**	0.685 ± 0.009
Proposed + Separ. E.	0.683 ± 0.008	0.682 ± 0.007	0.709 ± 0.020	0.733 ± 0.018	0.679 ± 0.006	0.684 ± 0.009	0.681 ± 0.007
Proposed + Comb. E.	0.675 ± 0.004	0.675 ± 0.004	0.712 ± 0.025	0.731 ± 0.018	0.673 ± 0.004	0.676 ± 0.005	0.674 ± 0.004

Table 5.5 displays the performance of various models on the stratified testing set described in Section 5.3.6. The test cases are categorized based on their uncertain levels, as defined in Section 5.3.2, and evaluated separately using different models. The upper panel (a) presents the results for 477 CXRs with uncertain levels smaller than 2, referred to as clean test cases. The lower panel (b) shows the results for 528

cases with higher uncertainty, denoted as equivocal test cases. In line with the findings discussed in Section 5.4.1, the models incorporating the thresholding mechanism demonstrate superior performance compared to those without it. Therefore, only the models utilizing the thresholding mechanism, along with linear probing, fine-tuning, and Confusion Estimation, are included in the table for comparison.

When evaluating the clean test cases, both linear probing and fine-tuning exhibited similar levels of performance, with differences of less than 1%. The Confusion Estimation model performed on par with or slightly worse than TRAM with thresholding. Among the TRAM models incorporating the thresholding mechanism, those utilizing Separate Encoding achieved the highest performance on the clean test cases compared to the other two encoding techniques. However, their overall performance was 1% to 2% lower than that of the proposed models across most metrics, and they exhibited higher standard deviations.

The proposed model with Score Encoding demonstrated the highest performance across all evaluated metrics. It achieved a precision of 0.921, accuracy of 0.921, AUPRC of 0.971, AUROC of 0.973, sensitivity of 0.92, specificity of 0.921, and F1 score of 0.921. The model utilizing Separate Encoding achieved the same level of performance while using Combined Encoding showed slightly inferior results. Comparing the proposed models with linear probing and fine-tuning, the proposed models showed a 1% increase in AUROC and AUPRC, as well as a 3%-4% improvement across other metrics.

When considering the equivocal test cases, fine-tuning showed higher AUPRC and AUROC values compared to linear probing, although the differences were subtle for other metrics. Among the models tested, Confusion Estimation achieved the highest AUPRC of 0.737 and AUROC of 0.728. TRAM with thresholding and Separate Encoding demonstrated the best performance in terms of accuracy, sensitivity, and F1 score. The proposed model with Score Encoding achieved the highest precision

and specificity. Overall, the proposed model did not consistently outperform the other models and often yielded lower scores.

It is important to note that in the context of equivocal test cases potentially being assigned with incorrect labels, higher values could indicate overfitting to the noisy labels. Therefore, we are more concerned with the performances on the clean test cases as they can provide a more accurate reflection of how each model performs. Interestingly, although the Confusion Estimation model achieved the best AUPRC overall, as shown in Table 5.4, its performances on clean test cases are worse than the proposed method. Another interesting observation is that the performance on the equivocal test cases was generally worse compared to the clean cases, exhibiting a decrease of approximately 25%. Additionally, the standard deviations were generally larger for the equivocal test cases. These findings indicate that the presence of uncertainty in test cases can significantly impact model performance and increase variability in the results.

5.5 Discussion

CXRs are commonly used for ARDS diagnosis, but their interpretation can be challenging and subjective. Previous studies have utilized traditional ML and DL approaches to detect ARDS from CXR images. However, these approaches have not adequately addressed label uncertainty and noise, which can affect model performance. In this Chapter, inspired by the TRAM network, we propose a DL model that leverages label uncertainty from multiple annotators as privileged information to improve ARDS detection in CXR images. We introduce three different encoding methods and a simple, but effective, measure of uncertainty to supply the model with privileged information when necessary. Additionally, we apply ordinal regression to the privileged branch of the model to encourage knowledge transfer across branches. Our proposed model achieves an AUROC of 0.873, AUPRC of 0.868, and an F1 score

of 0.797 on test examples. Moreover, it achieves an AUROC of 0.973, AUPRC of 0.971, and an F1 score of 0.921 on cases with more certain and cleaner labels, while fine-tuning the encoder only gets AUROC 0f 0.956, AUPRC of 0.956, and F1 score of 0.890. In comparison to the two previous studies [192, 227] which primarily focused on developing models for ARDS detection and used ARDS datasets that have similar attributes to ours, our work specifically addresses the challenge of leveraging multi-annotator label uncertainty to enhance performance.

This Chapter also presents findings and insights regarding the TRAM mechanism. As the first application of TRAM in medical image analysis, our experiments highlight its utility in the identification of ARDS from CXR images, while validating previous findings [241] that excessive privileged information can hinder model generalizability. Specifically, we demonstrated the critical role of the thresholding mechanism in the success of our proposed model. Although we used a median value of the uncertainty measurements in the training set as the determined threshold, this value can be regarded as a hyperparameter when using the proposed method on other datasets. In Appendix H, we explore the impact of different threshold values on cross-validation and testing performance. It is observed that increasing the threshold from 0 to 4 in increments of 0.5 initially enhances testing performance but subsequently leads to a decline. Another interesting observation is that, despite the maximum measurement of uncertainty being 3.96 and applying a threshold of 4 results in the TRAM network receiving no privileged information, the results presented in Table 5.4 and Appendix H Table H show that using a threshold of 4 outperforms both the approach of supplying extensive privileged information without thresholding and fine-tuning the network. This performance improvement using the TRAM mechanism, even in the absence of additional information, could be attributed to two key factors. First, the privileged prediction head in our experimental setup exhibits stronger learning capability. While the base network employs a single-layer prediction head, the privileged branch

incorporates a larger prediction head with enhanced learning capacity. Consequently, knowledge from the prediction head, rather than privileged information itself, can be learned and transferred to the base network. Second, the presence of two branches and the stop-gradient operation in TRAM may contribute to mitigating overfitting tendencies. We observe more stable training loss behavior and less overfitting when employing the TRAM-based network compared to fine-tuning.

While fine-tuning a supervised pretrained feature encoder is the most common approach for transfer learning in medical imaging tasks, recent studies [244, 245, 246, 247] have explored the effectiveness of self-supervised pretraining in CXR image analysis and some [244, 247] have shown that self-supervised pretrained feature encoders generate more informative embeddings compared to their supervised counterparts. To assess if our proposed model consistently achieves superior performance with different pretrained encoders, and to explore whether self-supervised pretraining can yield better encoders for ARDS detection than their supervised counterparts, we conducted additional experiments using Boost Your Own Latent (BYOL) [248] and distillation with no labels (DINO) [249] pretrained encoders. Detailed information regarding the background, training protocol, and results of these experiments can be found in Appendix I. In summary, our findings demonstrate that utilizing DINO pretrained encoders can enhance the performance of ARDS detection compared to supervised pretrained encoders. Moreover, while the quality of the pretrained encoder and its architecture are crucial factors influencing downstream fine-tuning performance, the methods proposed in this Chapter consistently yielded matching or superior test performance compared to other baselines, regardless of the specific pretrained encoder employed.

Our work has certain limitations that should be acknowledged, with the primary limitation relating to interpretability. Firstly, the proposed models do not provide insight into how different annotators contribute to label noise or how their annota-

tions impact the final results, whereas the Confusion Estimation model [232] offers a potential solution by estimating the skill level of each annotator based on their confusion matrix's average diagonal elements. Furthermore, the encoding methods that performed best in testing for the proposed models favor Score Encoding and Separate Encoding, which do not rely on the correspondence between the score and its annotator. This observation suggests that the model's performance may not be dependent on this correspondence, and the mechanism by which it utilizes multi-annotator information still lacks explainability.

In a recent study by Farzaneh et al. [250], which investigated collaborative strategies between physicians and an AI model in ARDS diagnosis, it was discovered that AI and physician expertise complemented each other. The AI model exhibited higher and more consistent accuracy on less challenging chest X-rays, while physicians demonstrated higher accuracy on difficult chest X-rays. These findings endorse the strategy of having the AI model review chest X-rays initially and involve clinicians when uncertainty arises. This highlights the significance of identifying cases with uncertainty and guides the future direction of our work. Specifically, our focus will be on enhancing the interpretability of the uncertain level associated with each case and integrating strategies to handle noisy labels at both the annotator and sample levels. By doing so, we aim to further support the identification of ARDS patients and provide them with evidence-based care.

CHAPTER VI

Conclusion

The thesis presented several novel machine-learning and deep-learning models that address practical challenges in clinical decision support. These challenges are diverse and depend on the specific data type being analyzed. The following is a summary of the challenges and the corresponding solutions presented in the thesis, grouped by the target medical data modalities.

In Chapter II, the focus was on medical image analysis, specifically the segmentation of coronary arteries in XCA images. The challenges in this task included the limited availability of labeled images and the class imbalance of the training data. To address these, an automated pipeline was proposed, leveraging an ensemble framework that combined deep learning and filter-based features. Additionally, multiple under-sampling methods were incorporated based on domain knowledge to create a balanced training dataset. The proposed approach outperformed common deep CNNs and yielded more consistent results in coronary artery segmentation. Moving to Chapter V, the emphasis remained on medical image analysis, with a focus on improving ARDS detection in CXR images. The main obstacle in this context was the lack of gold-standard labels. Therefore, a novel approach was introduced to utilize the label uncertainty derived from multiple annotators as privileged information. By incorporating the latest progress in LUPI and effective knowledge transfer mech-

anisms, the proposed network outperformed various baselines in the testing phase, proving its ability to address label uncertainty and noise in CXR images for ARDS detection.

Chapter III shifted the focus to the diagnosis of ARDS using EHR data. In this task, the challenges involved dealing with the unique characteristics of EHR data and balancing the efficiency and explainability of the ML models. In addition to the careful handling of EHR data in preprocessing, various ML models and feature selection methods were employed and evaluated, with the most relevant clinical variables diagnosis identified. The results demonstrated that ML models utilizing EHR data alone could accurately detect ARDS. This capability allows for the early detection of ARDS before using MV, which can potentially improve patient outcomes. Chapter IV built upon the work in Chapter III by considering the use of EHR data as the base data modality, while incorporating CXRs and MV-related information as privileged information. It is an attempt to integrate multiple sources of medical data with the LUPI paradigm. Specifically, the objective was to develop a LUPI model that transfers knowledge from the privileged domain to the base domain for improved ARDS detection. The proposed PLR model, with regularization techniques employed in both the privileged and base domains, showed improved classification performance even when privileged information was only partially available.

It is important to note that the models and techniques developed in this research were based on the state-of-the-art at the time of their development. Since the field of medical AI is rapidly evolving, with new advancements and approaches continuing to emerge, there are more advanced and effective solutions to address some of the aforementioned challenges and they could reshape the methodology proposed in this thesis.

For example, to address the challenge of limited labeled data discussed in Chapter II, an emerging trend is to utilize generative models, such as generative adversarial

networks and Denoising Diffusion Probabilistic (DDP) models. These models have shown significant potential in generating synthetic medical images, thereby supplementing small datasets and enabling effective data augmentation techniques. Notely, a recent work [251] has used diffusion adversarial representation learning model, a combination of DDP with adversarial learning, to generate synthetic vessel images and vessel segmentation masks at the same time, achieving significantly better performance than the supervised counterpart in segmentation of coronary angiography and retinal images. In addition, the application of generative models in the medical domain has extended well beyond image-generation tasks. It now encompasses a wide range of applications, including image-to-image translation, image reconstruction, image classification, image segmentation, and abnormalities detection.

In recent years, the field of medical AI has witnessed significant advancements beyond traditional supervised learning methods, with a growing focus on weakly supervised learning [252, 253] and self-supervised learning [254]. These approaches have gained popularity for their capacity to leverage large amounts of unannotated data and enhance model performance. Weakly supervised learning tackles the challenges associated with incomplete, inexact, or inaccurate label supervision [255], while self-supervised learning trains models by predicting image transformations or contextually related image patches, enabling the learning of valuable representations without manual annotations. By harnessing the inherent information within the data itself to uncover meaningful representations, self-supervised learning techniques have shown promise in pre-training deep neural networks using unlabeled medical images, as demonstrated in the experiments conducted in Chapter V. Recently, there have been endeavors [246] to combine self-supervised learning with semi-supervised learning, enabling the model to evolve through self-training and achieve enhanced performance in lung disease detection from chest X-rays. This represents a potential future direction to the methods proposed in Chapter V for addressing label noise in ARDS detection

from chest X-rays.

Furthermore, there has been significant research progress focused on analyzing EHR data [256, 257] and establishing standards for its processing [35, 258]. Researchers have proposed techniques [259, 260] to address the challenges of temporality and missingness in EHR data such as imputation with masking and incorporating time intervals. These studies highlight areas that need improvement in the work presented in Chapter III and Chapter IV. Specifically, although temporality was addressed by under-sampling the longitudinal EHR data and assuming it to be i.i.d. samples, the informative presence captured by frequencies and other factors could be overlooked. Considering the utilization of more advanced temporal models or incorporating temporal information in other ways could potentially result in better-performing models for EHR data analysis in Chapter III and Chapter IV.

Still, this thesis provided valuable insights that can guide future research endeavors in medical AI. Chapter IV sheds light on the advantages of integrating multiple data modalities to build robust decision support models. This highlights the immense potential of multimodal learning approaches in medical AI [261], where models can leverage diverse types of data inputs to make predictions, mirroring the way human clinicians rely on multiple sources of information in their decision-making processes. Notable efforts have already demonstrated the effectiveness of integrating different data modalities in medical AI applications [262], such as integration of continuous ECG data and discrete clinical data for decompensation prediction [263], the fusion of radiological images and EHR data for pulmonary disease detection [264], and the utilization of pathology-radiology fusion for prostate cancer classification [265]. However, a major limitation of these approaches is the assumption of complete patient records across all modalities, which is often unrealistic in real-world clinical scenarios due to various factors [266], such as the risks associated with certain examinations or invasive procedures, patient preferences, or limitations in data collection across

departments and institutions. Currently, approaches like deleting incomplete cases or using imputation techniques are commonly employed to handle missing data in multimodal settings [267]. However, these techniques may reduce the available training data or introduce biases and inaccuracies in the imputed values, ultimately affecting the performance of the models. To address the challenge of missing data and enhance multimodal learning, the LUPI paradigm emerges as a promising approach, together with the others [268]. Chapter V and other research [269] have demonstrated that the LUPI framework often outperforms imputation methods when dealing with missing data in multimodal settings. While a comprehensive systematic analysis is yet to be conducted, it appears that leveraging multiple data modalities within the LUPI framework not only enables the inclusion of privileged or additional information from different sources but also provides a more consistent way to handle missing data with the LUPAPI approach. By leveraging principles from information theory and multi-view learning [270], the LUPI paradigm holds promise as a potential avenue to enhance the performance and generalizability of multimodal models in medical AI.

Looking toward the future, it is evident that the concept of knowledge transfer, encompassing ideas like transfer learning [271] and knowledge distillation [272], will remain essential in the development of medical AI models. With the emergence of very large imaging models [273] and the availability of publicly accessible datasets [274, 275] on medical imaging, EHR, and others, it could be beneficial to leverage their power while incorporating domain knowledge to tailor solutions to the specific data at hand. Beyond the technical aspects, there are numerous other facets of medical AI that are worth discussing [33]. For example, exploring the interaction between medical AI models and clinicians is an intriguing area of investigation, as it can enhance collaboration, decision-making processes, and the overall quality of patient care. Furthermore, ethical considerations [33] pertaining to data use, privacy, security, and bias are of utmost importance in the field of medical AI to ensure the

fair deployment of AI systems, promoting trust, transparency, and equitable access to healthcare advancements.

In conclusion, as we move forward, a multidisciplinary approach that combines technical advancements with considerations of ethics and human interaction will drive the future of medical AI. By navigating these challenges, we can harness the full potential of AI technologies to transform healthcare and improve patient outcomes.

APPENDICES

APPENDIX A

Enhancing X-ray Coronary Angiography Images through Pre-processing with Filters

Top-bottom-hat Filtering

Top-hat and bottom-hat filters are morphological filters that combine dilation and erosion operations with a structuring element (SE). For a gray-scale image, dilation and erosion operation (Figure A.1) of a pixel return the minimum and the maximum, respectively, of the pixel intensities in its neighborhood defined by SE, and hence, the former is often used for gaps filling and region connections, while the latter is applied for detail elimination. Denoting the image matrix as I and the SE with scale λ as SE_λ, the morphological opening operation ($\circ$) is defined as an erosion ($\ominus$) followed by a dilation ($\oplus$) operation and the morphological closing operation ($\bullet$) first perform dilation and then erosion.

$$I \circ SE_\lambda = (I \ominus SE_\lambda) \oplus SE_\lambda$$

$$I \bullet SE_\lambda = (I \oplus SE_\lambda) \ominus SE_\lambda$$

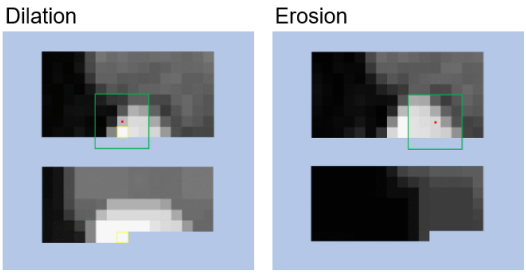


Figure A.1: Examples of dilation (left) and erosion (right) on a grayscale image using a 5×5 flat SE [1, 2]. The top and bottom parts illustrate the position and results of the structuring element window when applied on specific pixels of the original images.

Contrast-enhancing Filtering

Algorithm: Contrast Enhancement

$BG = I \bullet \text{SE}(\text{disk}, \lfloor \Lambda_{n-1}/1.5 \rfloor)$ ▷ Λ_{n-1}, the second largest scale, $\bullet$, the closing operation

$FG = I - BG$ ▷ FG, the foreground; BG, the background.

$\mu_{BG} = \min(\text{mean}(BG) + \text{std}(BG)/2,\ 0.95)$

procedure FLAT-FIELD CORRECTION

  for $\lambda \in \Lambda_1, \ldots, \Lambda_n$ **do** ▷ Iterate over scales

$$I_{flat}(\lambda) = I \cdot \frac{\mu_{BG}}{I * g_{\frac{\lambda}{2}}}$$

  ▷ g_σ is the Gaussian kernel of scale σ, $*$ is the convolution operation

  end for

  return $BG_{adjust} = \max_{\lambda \in \Lambda} I_{flat}$

end procedure

Recenter the mean of BG_{adjust} to 0.75

$BG_{blur} = I \bullet SE(\text{disk}, \lfloor \Lambda_n \rfloor) * g_{10}$

$FG_{adjust} = min(0, FG \cdot (1 - BG_{blur}))$

$FG_{adjust} = \text{mean}(FG_{adjust})/\text{std}(FG_{adjust})$

Reconstruct image as $I_{reconstruct} = BG_{adjust} + FG_{adjust}/10$

Normalize $I_{reconstruct}$

Top-bottom-hat enhancement using Equation 2.1, with $m = n = 0.25$ and $\Lambda_i = \Lambda_i/2$

Diffusion Filtering

Diffusion is a time-dependent process from physics that models the concentration change. This process evolves the input frame, I, by introducing a 'time' variable and generating images, $I(t)$, evolved via the diffusion equation:

$$I(t) = \nabla \cdot (D \nabla I)$$

with respect to time t. Here, ∇ represents the divergence operator; $\nabla I = (I_x, I_y)$ denotes the image gradient; and D is a diffusion tensor that describes the diffusion process. The diffusion tensor is constructed to enhance the vascular structures in the image using a variant of Frangi's vesselness filter with continuous derivatives. The filter V_σ is given by

$$V_\sigma = \begin{cases} 0, & \lambda_2 < 0 \\ exp\left(-\frac{R^2}{2r^2}\right) \times \left(1 - exp(\frac{S^2}{2s^2})\right) \times \left(-\frac{2c^2}{\lambda_2^2}\right), & \lambda_2 \geq 0 \end{cases}$$

where λ_1 and λ_2 are the eigenvectors of the Hessian matrix, $|\lambda_2| > |\lambda_1|$, $R = \lambda_1/\lambda_2$ and $S = \sqrt{\lambda_1^2 + \lambda_2^2}$, the parameter σ denotes convolution with a Gaussian Kernel of radius σ. The smoothing parameter, c, ensures that the derivative of V_σ remains continuous at all points, ensuring a smooth function. The diffusion tensor must be smooth, positive definite, and symmetric. Due to the addition of the smoothing parameter in the above filter, the diffusion tensor can be defined as

$$D = QGQ^T$$

where Q is the matrix given by the eigenvectors of the Hessian of the image and

$$G = \begin{bmatrix} \lambda_1^{'} & 0 \\ 0 & \lambda_2^{'} \end{bmatrix} = \begin{bmatrix} 1 + (\omega - 1) \cdot V^{s^{-1}} & 0 \\ 0 & 1 + (\epsilon - 1) \cdot V^{s^{-1}} \end{bmatrix}$$

In this definition, ω, ϵ, and s denote additional tuning parameters.

Vessel-enhancing diffusion is performed on each image using scale-space theory for scale invariance [276] with scaling parameters selected for the LCA and RCA based on expected vessel width ranges [277]. The differential filters are optimized for rotation invariance [278]. Finally, the vesselness response of the diffused image I' is given by the maximum vesselness response over all scales,

$$V = max_{\sigma} V_{\sigma}$$

APPENDIX B

Computational Time for Different Under-sampling Procedures

Table B.1 shows the mean and standard deviation of the computational times when applying different under-sampling methods over all images in our dataset. Unsupervised under-sampling is implemented by Matlab, while Tomek Links and Cluster Centroid are carried out with Python Imbalanced-learn library [127]. Cluster Centroid under-sampling takes 67312 times more computational time than the unsupervised method and uses 163 times more computational time than the Tomek Links under-sampling.

Table B.1: Computational Time Statistics

Under-sampling Methods	**Computational Time (Seconds)**
Unsupervised	0.08 ± 0.005
Tomek Links	33.36 ± 8.62
Cluster Centroid	5385.73 ± 207.27

APPENDIX C

Training Details on DeepLabV3+

We applied a backbone of ImageNet pre-trained Resnet101, an encoder depth of 5, an encoder output stride of 16, a decoder atrous rates of (12, 24, 36), and a decoder channel of 256 in the DeepLabV3+ model construction. These are the preferable parameter choices described in the original paper when training the model on images of roughly the same resolution as ours.

APPENDIX D

Supplementary Tables for Extracted Variables from Electronic Health Records

Comprehensive Listing of Variables

Table D.1: Abbreviations and Meanings of Variables from EHR

Variable Abbr.	**Meaning**	**Mechanical Ventilation Related or Not**
temp	Temperature	No
sbp	Systolic blood pressure	No
dbp	Diastolic blood pressure	No
hr	Heart rate	No
rr	Respiratory rate	No
sp02	Pulse oximetry value	No
gcs_total	Total score of Glasgow Coma Scale	Maybe
gcs_motor	Glasgow Coma Scale of motor response	Maybe
gcs_eye	Glasgow Coma Scale of eye-opening response	Maybe
gcs_verbal	Glasgow Coma Scale of verbal response	Maybe

rass	Richmond Agitation and Sedation Scale (RASS), a validated and reliable method to assess a patient's level of sedation in the intensive care unit.	No
alert	An AVPU scale, given when the patient is fully awake (although not necessarily oriented)	No
unresponsive	AVPU scale, is recorded if the patient does not give any eye, voice or motor response to voice or pain.	No
sedated	If patient is sedated	No
oriented	Orientation Levels	No
invasive	Patient currently receiving invasive mechanical ventilation (1 = yes, 0 = no)	**Yes**
noninvasive	Patient currently receiving non-invasive mechanical ventilation (1 = yes, 0 = no)	**Yes**
hfnc	High flow nasal cannula (HFNC) oxygen delivery, a relatively new non-invasive ventilation therapy that seems to be well tolerated in neonates and adults with hypoxemic respiratory failure	Removed
supl	Supplemental oxygen	Yes
fiO2	Fraction of inspired oxygen, a percentage indicator of supplemental oxygen level	No
pf	PaO2/FiO2, ratio of blood oxygen to supplemental oxygen	No
pf_calc	pf, but calculated rather than recorded	No
RRset	Preset respiratory rate for mechanical ventilation	**Yes**
RRobs	Preset respiratory rate observed	**Yes**

Vtset	Tidal volume (VT) set on the mechanical ventilator	**Yes**
Vtobs	Tidal volume observed	**Yes**
PEEP	Positive end-expiratory pressure. PEEP maintains the patient's airway pressure above the atmospheric in mechanical ventilation	**Yes**
Plat	Plateau pressure, a measurement of lung compliance made on patients receiving invasive mechanical ventilation	**Yes**
mAirP	Mean airway pressure, measure of pressure delivered during invasive mechanical ventilation, typically higher pressure required when lung injury severe	**Yes**
iv_in	Intravenous fluid given (mL)	No
urine_out	Urine output (mL)	No
dialysis	A binary variable indicating kidney dialysis.	No
norepi	Norepinephrine	Removed
epi	Epinephrine	Removed
vasso	Vasopressin	Removed
phenyl	Phenylephrine	Removed
dopa	Dopamine	Reomved
dobu	Dobutamine	Removed
mil	Milrinone	Removed
lactate	Lactate acid level obtained by blood gas	No
pH	pH level obtained by blood gas	No
pCO2(PaCO2)	Carbon dioxide level obtained by blood gas	No
pO2	Oxygen level obtained by blood gas	No
Na	Sodium	No
K	Potassium	No

HCO2	Another measure of bicarbonate	No
BUN	Blood urea nitrogen, measure of kidney function	No
Cr	Creatinine, measure of kidney function	No
WBC	White blood cell count level	No
Hgb	Hemoglobin	No
Plt	Platelet Count	No
TP	Transfused plasma	No
Tbili	Total bilirubin	No
Alb	Albumin level	No
AST	Aspartate aminotransferase, indicator of liver damage	No
INR	International Normalized Ratio, used to measure clotting.	No
lipase	Lipase level, an indicator of pancreatic function	Removed
trop	Troponin level, elevated in myocardial infarction or heart failure, which are potential “mimicers” of ARDS	Removed
BNP	Brain natriuretic peptide level, elevated in heart failure which is a “mimicer” of ARDS	Removed
PTT	Partial Thromboplastin Time, the time that takes a blood clot to form.	No
rbc_transf	Red blood cell transfusion	No
plt_transf	Platelet transfusion	No
ffp_transf	Fresh frozen plasma transfusion	No
total_out	Total fluids out, likely daily total	No
total_in	Total fluids in, likely daily total	No
age	Age in years	No
bicarb	Bicarbonate level	Removed

pressor	Whether or not patient currently receive vasopressor support for hemodynamic insufficiency	Removed
net_in	Net IV fluids, the amount of total IV fluids given, i.e., total urinary output to date	Removed
shock	Ratio of heart rate over systolic blood pressure	Removed

Summary Statistics of Variables

Table D.2: Summary Statistics with Mean and Standard Deviation

Variable Names	**No. of Missing** (out of 306292)	**Statistics, mean (std)**
temp	273736	98.3 (1.7)
sbp	233954	122.9 (27.7)
dbp	233959	64.7 (15.0)
hr	207399	91.6 (20.3)
rr	232388	20.8 (7.2)
fiO2	187503	41.8 (22.1)
pf	294084	247.1 (119.7)
pf_calc	263561	227.9 (93.6)
RRset	267765	18.0 (5.7)
RRobs	263016	21.5 (7.0)
Vtset	273101	420.0 (74.7)
Vtobs	264552	414.2 (141.8)
PEEP	259108	7.4 (3.3)
Plat	298112	22.0 (6.7)
mAirP	283083	11.9 (4.6)
vasso	305524	0.0 (0.2)
phenyl	305601	85.9 (77.7)

lactate	292173	2.4 (2.8)
pH	292165	7.4 (0.1)
pCO2	293894	40.5 (10.7)
pO2	293900	108.8 (62.4)
Na	293252	141.2 (5.5)
K	293264	4.2 (0.6)
HCO2	293291	26.7 (5.7)
BUN	292810	32.5 (23.2)
Cr	292993	1.4 (1.2)
WBC	294920	12.1 (8.4)
Hgb	294924	9.1 (2.0)
Plt	294950	218.3 (165.1)
TP	301694	5.6 (1.1)
Alb	301375	3.0 (0.6)
INR	301932	1.5 (0.9)
lipase	306012	79.8 (194.0)
trop	304921	7.9 (37.7)
BNP	305939	688.0 (987.1)
PTT	303782	37.6 (20.0)
rbc_transf	166185	1.8 (26.3)
plt_transf	166185	0.5 (13.0)
ffp_transf	166185	0.4 (11.7)
total_out	166185	52.1 (159.7)
total_in	166185	58.4 (137.4)

Table D.3: Summary Statistics with Median, Lower, and Upper Quartiles

Variable Names	**No. of Missing** (out of 306292)	**Statistics median [Q1, Q3]**
spO2	214196	97.0 [94.0,99.0]
gcs_total	296165	15.0 [14.0,15.0]
gcs_motor	296148	6.0 [6.0,6.0]

gcs_eye	296135	4.0 [4.0,4.0]
gcs_verbal	296154	5.0 [4.0,5.0]
rass	288049	0.0 [-2.0,0.0]
iv_in	166185	0.0 [0.0,10.0]
urine_out	166185	0.0 [0.0,0.0]
norepi	298700	0.1 [0.0,0.2]
epi	305753	0.1 [0.0,0.5]
dopa	306214	5.0 [2.6,13.7]
dobu	306007	4.7 [1.0,5.0]
mil	306178	0.2 [0.2,0.2]
AST	301715	44.0 [27.0,110.0]
Tbili	301691	0.8 [0.4,2.0]

Table D.4: Summary Statistics based on Number of Count and Percentage

Variable Names	**Values**	**No. of Missing** (out of 306292)	**Statistics, n (%)**
alert	0	299867	2857 (44.5)
	1		3568 (55.5)
unresponsive	0	299867	6122 (95.3)
	1		303 (4.7)
sedated	0	299867	5731 (89.2)
	1		694 (10.8)
oriented	0	300584	2603 (45.6)
	1		3105 (54.4)
invasive	0	186475	59096 (49.3)
	1		60721 (50.7)
noninvasive	0	186475	117043 (97.7)
	1		2774 (2.3)
noninvasive	0	186475	63495 (53.0)
	1		56322 (47.0)
hfnc	1	303108	3184 (100.0)

dialysis	0	166185	139878 (99.8)
	1		134 (0.1)
	2		95 (0.1)

APPENDIX E

Hyperparameter Searching Range for the Privileged Logistic Regression Models

The hyperparameter searching ranges of different models are listed in Table E.1.

Table E.1: Hyperparameter Searching Range for Different Models

Models	**Hyperparameter**	**Searching Range**
LR	λ	{0.001, 0.005, 0.01, 0.05, 1, 5, 10, 50, 100, 500, 1000}
PLR	λ	{0.01, 0.05, 1, 5, 10, 50}
	λ^*	{0.01, 0.05, 1, 5, 10, 50}
	β	{0.05, 0.1, 0.5, 1, 5, 10, 50}
	ξ	{0.1, 0.5, 1, 5, 10}
SVM+	C	{0.001, 0.01, 0.1, 1, 10, 100, 1000}
	γ	{0.5, 1, 5,10, 50, 100, 500}
Hetero SNN	batch size	32, 64, 128
	learning rate	$\{10^{-4}, 10^{-5}, 10^{-6}\}$
	hidden layer	{1, 2, 3}
	hidden nodes	{5, 10, 20, 50}

APPENDIX F

Asymptotic Analysis

In this section, general results from §4.6 of [214] and §5.3 of [215] on the asymptotic normality of empirical risk minimizing (ERM) estimators were used. For simplicity, we only consider the cases of sLR and sPLR.

Apart from the notions in the main text, we denote Hessian matrix as $H(\boldsymbol{\theta}) = \mathcal{R}''(\boldsymbol{\theta})$ whose ijth entry is given by

$$\frac{\partial^2}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_j} \mathbb{E}\big[\ell(Y, f_{\boldsymbol{\theta}}(\bar{X}))\big]$$

and G-matrix as $G(\boldsymbol{\theta}) = \mathbb{E}\big[\nabla_{\boldsymbol{\theta}} \ell(Y, f_{\boldsymbol{\theta}}(\bar{X})) \nabla_{\boldsymbol{\theta}} \ell(Y, f_{\boldsymbol{\theta}}(\bar{X}))^T\big]$ whose ijth entry is given by

$$\mathbb{E}\Big[\frac{\partial}{\partial \boldsymbol{\theta}_i} \ell(Y, f_{\boldsymbol{\theta}}(\bar{X})) \frac{\partial}{\partial \boldsymbol{\theta}_j} \ell(Y, f_{\boldsymbol{\theta}}(\bar{X}))\Big].$$

Theorem F.1. *If the loss function ℓ is sufficiently differentiable with respect to the second variable, and if the Hessian matrix $H(\tilde{\boldsymbol{\theta}})$ is positive definite (i.e. invertible) at the expected risk minimizer $\tilde{\boldsymbol{\theta}}$, then*

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}) \sim N\big(0, H(\tilde{\boldsymbol{\theta}})^{-1} G(\tilde{\boldsymbol{\theta}}) H(\tilde{\boldsymbol{\theta}})^{-1}\big)$$

and

$$\mathbb{E}\big[\|\hat{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}\|^2\big] \sim \frac{1}{n} \mathrm{Tr}\big[H(\tilde{\boldsymbol{\theta}})^{-1} G(\tilde{\boldsymbol{\theta}}) H(\tilde{\boldsymbol{\theta}})^{-1}\big]$$

asymptotically as $n \to \infty$.

Without privileged information *(sLR model)* First evaluate the G-matrix. To this end, we have

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\theta}_i}\ell(Y, f_{\boldsymbol{\theta}}(\bar{X})) &= \frac{\partial}{\partial \boldsymbol{\theta}_i}\Big(\phi(-Y\boldsymbol{\theta}^T\bar{X})\Big)\\ &= -\phi'(-Y\boldsymbol{\theta}^T\bar{X})YX_i,\end{aligned}$$

hence the ijth entry of the G-matrix is equal to

$$\begin{aligned}G(\boldsymbol{\theta})_{ij} &= \mathbb{E}\Big[\Big(-\phi'(-Y\boldsymbol{\theta}^T\bar{X})YX_i\Big)\Big(-\phi'(-Y\boldsymbol{\theta}^T\bar{X})YX_j\Big)\Big]\\ &= \mathbb{E}\Big[\phi'(-Y\boldsymbol{\theta}^T\bar{X})^2X_iX_j\Big].\end{aligned}$$

Next evaluate the Hessian matrix. For this we make a further simplifying assumption that the distribution of $(\bar{X}, Y)$ is sufficiently smooth to permit interchanging partial differentiation and taking expectation, hence

$$\begin{aligned}H(\boldsymbol{\theta})_{ij} &= \frac{\partial^2}{\partial \boldsymbol{\theta}_i\partial \boldsymbol{\theta}_j}\mathbb{E}\big[\ell(Y, f_{\boldsymbol{\theta}}(\bar{X}))\big]\\ &= \mathbb{E}\Big[\frac{\partial^2}{\partial \boldsymbol{\theta}_i\partial \boldsymbol{\theta}_j}\ell(Y, f_{\boldsymbol{\theta}}(\bar{X}))\Big]\\ &= \mathbb{E}\Big[\frac{\partial}{\partial \boldsymbol{\theta}_j}\Big(-\phi'(-Y\boldsymbol{\theta}^T\bar{X})YX_i\Big)\Big]\\ &= \mathbb{E}\Big[\phi''(-Y\boldsymbol{\theta}^T\bar{X})X_iX_j\Big].\end{aligned}$$

With privileged information *(sPLR model)* First evaluate the G-matrix. By a similar calculation as before, we have

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\theta}_i}\ell(Y, f_{\boldsymbol{\theta}}(\bar{X})) &= -\phi'(-Y\boldsymbol{\theta}^T\bar{X})YX_i + 2\xi(\boldsymbol{\theta}^T\bar{X} - \boldsymbol{\theta}^{*T}\bar{X}^*)X_i\\ \frac{\partial}{\partial \boldsymbol{\theta}^*_j}\ell(Y, f_{\boldsymbol{\theta}}(\bar{X})) &= -\beta\phi'(-Y\boldsymbol{\theta}^{*T}\bar{X}^*)YX^*_j - 2\xi(\boldsymbol{\theta}^T\bar{X} - \boldsymbol{\theta}^{*T}\bar{X}^*)X^*_j\end{aligned}$$

hence the G-matrix is a 2×2-block matrix whose ijth entry is equal to

$$\begin{aligned}
G(\boldsymbol{\theta}, \boldsymbol{\theta})_{ij} &= \mathbb{E}\Big[\phi'(-Y\boldsymbol{\theta}^T\bar{X})^2 X_i X_j\Big] \\
&\quad - 4\xi\mathbb{E}\Big[\phi'(-Y\boldsymbol{\theta}^T\bar{X})(\boldsymbol{\theta}^T\bar{X} - \boldsymbol{\theta}^{*T}\bar{X}^*)YX_iX_j\Big] \\
&\quad + 4\xi^2\mathbb{E}\Big[(\boldsymbol{\theta}^T\bar{X} - \boldsymbol{\theta}^{*T}\bar{X}^*)^2 X_iX_j\Big] \\
G(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*)_{ij} &= \beta^2\mathbb{E}\Big[\phi'(-Y\boldsymbol{\theta}^{*T}\bar{X}^*)^2 X_i^*X_j^*\Big] \\
&\quad + 4\beta\xi\mathbb{E}\Big[\phi'(-Y\boldsymbol{\theta}^{*T}\bar{X}^*)(\boldsymbol{\theta}^T\bar{X} - \boldsymbol{\theta}^{*T}\bar{X}^*)YX_ijX_j^*\Big] \\
&\quad + 4\xi^2\mathbb{E}\Big[(\boldsymbol{\theta}^T\bar{X} - \boldsymbol{\theta}^{*T}\bar{X}^*)^2 X_i^*X_j^*\Big] \\
G(\boldsymbol{\theta}, \boldsymbol{\theta}^*)_{ij} &= G(\boldsymbol{\theta}^*, \boldsymbol{\theta})_{ji} \\
&= \beta\mathbb{E}\Big[\phi'(-Y\boldsymbol{\theta}^T\bar{X})\phi'(-Y\boldsymbol{\theta}^{*T}\bar{X}^*)X_iX_j^*\Big] \\
&\quad + 2\xi\mathbb{E}\Big[\Big(\phi'(-Y\boldsymbol{\theta}^T\bar{X}) - \beta\phi'(-Y\boldsymbol{\theta}^{*T}\bar{X}^*)\Big)(\boldsymbol{\theta}^T\bar{X} - \boldsymbol{\theta}^{*T}\bar{X}^*)YX_iX_j^*\Big] \\
&\quad - 4\xi^2\mathbb{E}\Big[(\boldsymbol{\theta}^T\bar{X} - \boldsymbol{\theta}^{*T}\bar{X}^*)^2 X_iX_j^*\Big].
\end{aligned}$$

Next, we evaluate the Hessian matrix. Under the same simplifying assumption which permits interchanging partial differentiation and taking expectation, the ijth entry of the Hessian is equal to

$$\begin{aligned}
H(\boldsymbol{\theta}, \boldsymbol{\theta})_{ij} &= \mathbb{E}\Big[\phi''(-Y\boldsymbol{\theta}^T\bar{X})X_iX_j\Big] + 2\xi\mathbb{E}\Big[X_iX_j\Big] \\
H(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*)_{ij} &= \beta\mathbb{E}\Big[\phi''(-Y\boldsymbol{\theta}^{*T}\bar{X}^*)X_i^*X_j^*\Big] + 2\xi\mathbb{E}\Big[X_i^*X_j^*\Big] \\
H(\boldsymbol{\theta}, \boldsymbol{\theta}^*)_{ij} &= H(\boldsymbol{\theta}^*, \boldsymbol{\theta})_{ji} \\
&= -2\xi\mathbb{E}\Big[X_iX_j^*\Big].
\end{aligned}$$

Asymptotic comparison By Theorem F.1 we have the following:

$$\mathbb{E}\big[\|\hat{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}\|^2 \mid \text{sLR}\big] \sim \frac{1}{n}\mathrm{Tr}\big[H(\tilde{\boldsymbol{\theta}})^{-1}G(\tilde{\boldsymbol{\theta}})H(\tilde{\boldsymbol{\theta}})^{-1}\big]$$

and

$$\begin{aligned}
&\mathbb{E}\big[\|\hat{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}\|^2 \;\big|\; \text{sPLR}\;\big] \\
&\sim \frac{1}{n}\mathrm{Tr}\left[\left[\begin{bmatrix} H(\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\theta}}) & H(\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\theta}}^*) \\ H(\tilde{\boldsymbol{\theta}}^*,\tilde{\boldsymbol{\theta}}) & H(\tilde{\boldsymbol{\theta}}^*,\tilde{\boldsymbol{\theta}}^*) \end{bmatrix}^{-1} \begin{bmatrix} G(\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\theta}}) & G(\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\theta}}^*) \\ G(\tilde{\boldsymbol{\theta}}^*,\tilde{\boldsymbol{\theta}}) & G(\tilde{\boldsymbol{\theta}}^*,\tilde{\boldsymbol{\theta}}^*) \end{bmatrix} \begin{bmatrix} H(\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\theta}}) & H(\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\theta}}^*) \\ H(\tilde{\boldsymbol{\theta}}^*,\tilde{\boldsymbol{\theta}}) & H(\tilde{\boldsymbol{\theta}}^*,\tilde{\boldsymbol{\theta}}^*) \end{bmatrix}^{-1}\right]_{\boldsymbol{\theta},\boldsymbol{\theta}}\right]
\end{aligned}$$

where $A_{\boldsymbol{\theta},\boldsymbol{\theta}}$ denotes the top-left block of a 2×2-block matrix.

To proceed further with the analysis, we make the following additional assumptions:

$$\begin{aligned}
\mathbb{E}[X_i X_j] &= \begin{cases} \sigma^2 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \\
\mathbb{E}[X_i^* X_j^*] &= \begin{cases} \sigma^{*2} & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \\
\mathbb{E}[X_i X_j^*] &= 0,
\end{aligned}$$

which could be achieved with a suitable linear transformation in the base and privileged feature spaces, provided the privileged features $\bar{X}^*$ contain a principal component that is uncorrelated with the base features $\bar{X}$. Under this additional simplifying assumption, we have

$$\mathbb{E}\big[\|\hat{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}\|^2 \;\big|\; \text{sPLR}\;\big] \sim \frac{1}{n}\mathrm{Tr}\big[\big(H(\tilde{\boldsymbol{\theta}}) + 2\xi\sigma^2\big)^{-1} G(\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\theta}}) \big(H(\tilde{\boldsymbol{\theta}}) + 2\xi\sigma^2\big)^{-1}\big].$$

Finally, we compare the asymptotic rate of convergence of the empirical risk minimizer $\hat{\boldsymbol{\theta}}_n$ to the expected risk minimizer $\tilde{\boldsymbol{\theta}}$ with and without privileged information. For this, we assume that $\tilde{\boldsymbol{\theta}}$ remains the same with or without privileged information. Furthermore, we will restrict to the case when the parameter ξ is sufficiently small, which corresponds to the infinitesimal benefit of introducing privileged information.

To simplify notations, let H denote the matrix whose ijth entry is equal to

$$\mathbb{E}\Big[\phi''(-Y\tilde{\boldsymbol{\theta}}^T\bar{X})X_i X_j\Big],$$

let G denote the matrix whose ijth entry is equal to

$$\mathbb{E}\Big[\phi'(-Y\tilde{\boldsymbol{\theta}}^T\bar{X})^2 X_i X_j\Big]$$

and let K denote the matrix whose ijth entry is equal to

$$\mathbb{E}\Big[\phi'(-Y\tilde{\boldsymbol{\theta}}^T\bar{X})(\tilde{\boldsymbol{\theta}}^T\bar{X}-\tilde{\boldsymbol{\theta}}^{*T}\bar{X}^*)YX_iX_j\Big].$$

Hence

$$\mathbb{E}\big[\|\hat{\boldsymbol{\theta}}_n-\tilde{\boldsymbol{\theta}}\|^2 \mid \text{sLR }\big] \sim \frac{1}{n}\mathrm{Tr}\big[H^{-1}GH^{-1}\big]$$

and

$$\begin{aligned}
&\mathbb{E}\big[\|\hat{\boldsymbol{\theta}}_n-\tilde{\boldsymbol{\theta}}\|^2 \mid \text{sPLR }\big] \\
&\sim \frac{1}{n}\mathrm{Tr}\big[(H+2\xi\sigma^2)^{-1}\big(G-4\xi K+O(\xi^2)\big)(H+2\xi\sigma^2)^{-1}\big] \\
&= \frac{1}{n}\mathrm{Tr}\big[\big(H^{-1}-4\xi\sigma^2H^{-2}+O(\xi^2)\big)\big(G-4\xi K+O(\xi^2)\big)\big(H^{-1}-4\xi\sigma^2H^{-2}+O(\xi^2)\big)\big] \\
&= \frac{1}{n}\Big(\mathrm{Tr}\big[H^{-1}GH^{-1}\big]-4\xi\mathrm{Tr}\big[H^{-1}KH^{-1}+2\sigma^2H^{-2}GH^{-1}\big]+O(\xi^2)\Big),
\end{aligned}$$

where the last equality follows from the identity

$$\mathrm{Tr}[ABC]=\mathrm{Tr}[CAB]=\mathrm{Tr}[BCA].$$

Therefore, the introduction of privileged information will lead to an infinitesimal increase in the rate of convergence of $\hat{\boldsymbol{\theta}}_n \to \tilde{\boldsymbol{\theta}}$ if

$$\mathrm{Tr}\big[H^{-1}KH^{-1}+2\sigma^2H^{-3/2}GH^{-3/2}\big]>0.$$

Since H and G are both positive definite, it would be sufficient if

$$K=\mathbb{E}\Big[\phi'(-Y\tilde{\boldsymbol{\theta}}^T\bar{X})(\tilde{\boldsymbol{\theta}}^T\bar{X}-\tilde{\boldsymbol{\theta}}^{*T}\bar{X}^*)YX_iX_j\Big]$$

is positive semidefinite. Since $\phi'<0$, one possible sufficient condition would be if

$$\mathbb{P}\big((\tilde{\boldsymbol{\theta}}^T\bar{X}-\tilde{\boldsymbol{\theta}}^{*T}\bar{X}^*)Y\le 0\big)=1,$$

in other words if

$$\begin{cases}
\tilde{\boldsymbol{\theta}}^{*T}\bar{X}^* \ge \tilde{\boldsymbol{\theta}}^T\bar{X} & \text{if } Y=1 \\
\tilde{\boldsymbol{\theta}}^{*T}\bar{X}^* \le \tilde{\boldsymbol{\theta}}^T\bar{X} & \text{if } Y=-1
\end{cases}$$

holds almost surely, or at least holds with a sufficiently high probability which depends on the

distribution of $(\bar{X}, Y)$.

APPENDIX G

Reviewer Assignment and Review Distribution

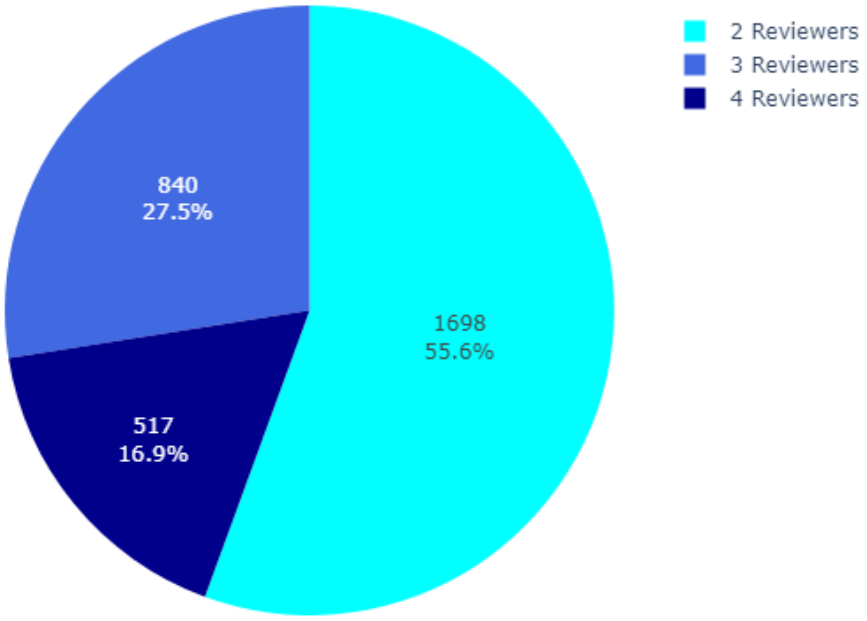


Figure G.1: Pie chart depicting the distribution of the number of reviewers on each image.

Figure G.1 depicts the distribution of the number of reviewers assigned to each image. Approximately 55.6% (1698) of the images were reviewed by two independent reviewers, while 27.5% (840) were reviewed by three reviewers. The remaining 16.9% (517) of the images underwent review by four reviewers. Based on Figure G.2, the reviewer with the highest number of review cases examined 1558 chest X-ray images, while the reviewer with the fewest number of images reviewed had 95 records.

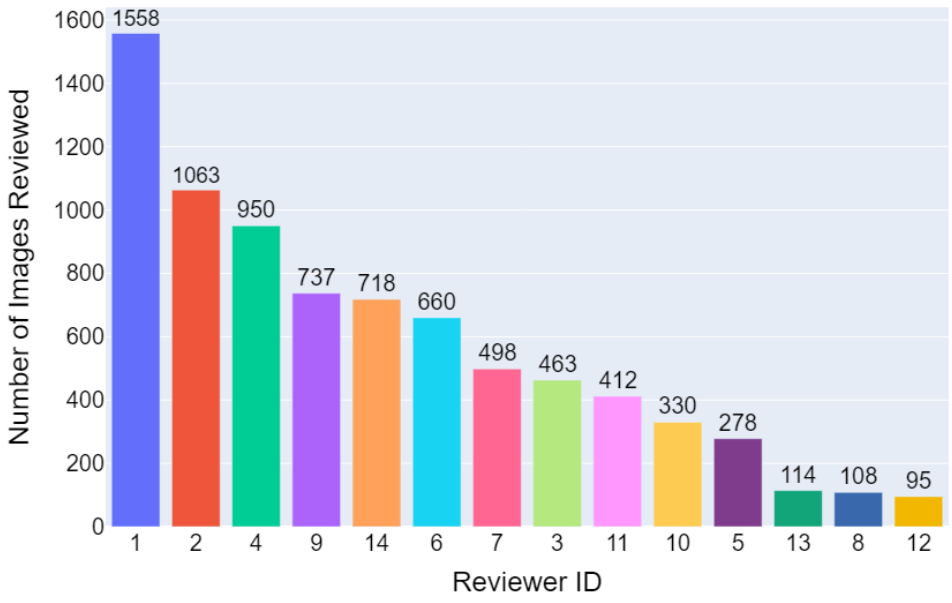


Figure G.2: Bar plot illustrating the number of images reviewed by each reviewer in descending order.

APPENDIX H

Impact of Uncertainty Threshold Levels on Validation and Testing Results

Table H.1: Cross-Validation and Testing Outcomes for the Proposed Model Using Scale Encoding with Different Thresholds

	Cross-Validation Outcomes				**Test Outcomes**					
	Loss	AUROC	AUPRC	F1 Score	Precision	AUPRC	AUROC	Sensitivity	Specificity	F1 Score
w/o Thred.	0.468±0.003	0.952±0.010	0.946±0.010	0.884±0.007	0.790±0.007	0.852±0.017	0.861±0.014	0.789±0.008	0.790±0.007	0.789±0.007
Thred. 0.5	0.449±0.006	0.958±0.010	0.954±0.013	0.901±0.015	0.792±0.007	0.857±0.013	0.867±0.009	0.791±0.006	0.792±0.008	0.791±0.006
Thred. 1.0	0.445±0.003	0.960±0.009	0.956±0.011	0.902±0.013	0.796±0.006	0.863±0.009	0.870±0.008	0.793±0.007	0.797±0.005	0.794±0.006
Thred. 1.5	0.442±0.008	0.958±0.013	0.953±0.016	0.899±0.012	**0.797±0.006**	0.865±0.012	0.871±0.010	**0.796±0.007**	**0.798±0.005**	**0.797±0.006**
Thred. 2.0	0.439±0.008	0.960±0.011	0.955±0.014	0.901±0.015	0.796±0.007	0.868±0.012	0.873±0.010	0.795±0.006	0.796±0.007	0.795±0.006
Thred. 2.5	0.435±0.007	**0.961±0.011**	**0.956±0.014**	0.903±0.014	0.794±0.005	**0.871±0.011**	**0.875±0.010**	0.793±0.005	0.795±0.005	0.794±0.005
Thred. 3.0	0.437±0.010	0.960±0.013	0.955±0.016	0.901±0.014	0.792±0.007	0.869±0.010	0.873±0.009	0.790±0.005	0.793±0.008	0.791±0.006
Thred. 3.5	**0.433±0.019**	0.959±0.019	0.954±0.023	**0.904±0.022**	0.794±0.010	0.869±0.014	0.873±0.011	0.791±0.009	0.795±0.011	0.793±0.010
Thred. 4.0	0.443±0.014	0.958±0.017	0.952±0.024	0.903±0.016	0.789±0.004	0.864±0.013	0.870±0.010	0.787±0.004	0.790±0.005	0.788±0.004

Table H.1 presents the results of the proposed models with Score Encoding as an example, illustrating the impact of threshold levels on cross-validation and testing outcomes. As the threshold level increases from 0 to 4 with a 0.5 basis, the validation performance initially improves and then declines. The lowest validation loss and highest F1 score were achieved when thresholding at 3. However, these metrics exhibited large standard deviations. Considering the trend, a threshold of 2.5 could be an optimal value as it demonstrates a similar averaged metric and lower standard deviation. The best validation AUROC and AUPRC were also achieved at the threshold of 2.5. The testing performance exhibits a similar trend of initially increasing and then decreasing. The best precision, sensitivity, specificity, and F1 score were achieved with a threshold of 1.5, while the best AUPRC and AUROC were obtained when thresholding at 2.5. These findings suggest that providing the model with more privileged information relative to non-privileged information initially enhances performance, but beyond a certain threshold, the performance begins to deteriorate. Furthermore,

the results indicate that the threshold used in our experiments may not be the optimal choice but can be deemed valid and appropriate within the context of this work.

APPENDIX I

Results with Self-supervised Pretrained Encoders

Self-supervised Learning (SSL) [279] is a powerful approach that utilizes large unlabeled datasets to train models in a task-agnostic manner. Unlike traditional supervised learning, which relies on labeled data, SSL derives its supervisory signal from the inherent structure and patterns within the data itself. SSL can be broadly categorized into two main types: contrastive and non-contrastive methods. Both aim to capture meaningful and discriminative features that are valuable for downstream tasks. Contrastive SSL involves training the model to bring similar images closer together in the embedding space while pushing dissimilar images apart. By optimizing the embeddings based on similarity, the model learns to extract informative visual representations. On the other hand, non-contrastive SSL, represented by self-distillation methods, encourages the model to learn consistent embeddings from different views of the same image. This process of learning from the model's own predictions fosters the development of robust and informative embeddings. Notable works in the realm of non-contrastive SSL include BYOL [248] and DINO [249].

BYOL utilizes two networks, the "online" and the "target", where each network is presented with a different view of the same image through image transformations. During training, the online network is updated using gradient descent based on its predictions of the representation for the differently augmented view. At the same time, the target network, serving as a reference, is updated using exponential moving average updates of the weights from the online network. This process encourages the encoder to learn meaningful embeddings that are robust to different data augmentations, enabling it to capture useful features in the desired image domain. In DINO, two networks with identical Vision Transformer (ViT) models are employed: one acts as the student network, and the other as the teacher network. These networks receive input from two sets of views

obtained by cropping the same image, allowing them to capture the semantic relationship between the local and global crops through self-attention mechanisms. Both BYOL and DINO have shown promising results in learning powerful visual representations without the need for manual annotations and contrastive examples and have recently been applied as pretraining methods for chest disease classification in CXR [244, 245, 246].

In the following section, we provide details of the experiments conducted using BYOL and DINO pretrained encoders. Unless stated otherwise, the implementation details closely follow those described in Section 5.3.

Table I.1: Testing Performance with BYOL Pretrained Encoder on All Test Cases and Clean Test Cases

All Test Cases, n=1005	Precision	Accuracy	AUPRC	AUROC	Sensitivity	Specificity	F1 Score
Linear Probing	0.670 ± 0.006	0.694 ± 0.008	0.764 ± 0.005	0.770 ± 0.005	0.766 ± 0.018	0.622 ± 0.011	0.714 ± 0.009
Fine-tuning	0.743 ± 0.007	0.754 ± 0.007	0.835 ± 0.010	0.838 ± 0.009	0.777 ± 0.010	0.731 ± 0.008	0.759 ± 0.007
Confusion Estimation	0.747 ± 0.007	0.762 ± 0.007	0.840 ± 0.011	0.841 ± 0.010	0.794 ± 0.008	0.730 ± 0.007	0.770 ± 0.007
TRAM w/ Thresh. + Scale. E.	0.727 ± 0.009	0.750 ± 0.010	0.833 ± 0.007	0.836 ± 0.006	0.799 ± 0.011	0.700 ± 0.010	0.761 ± 0.010
TRAM w/ Thresh. + Separ. E.	0.732 ± 0.009	0.753 ± 0.009	0.837 ± 0.005	0.839 ± 0.005	0.800 ± 0.012	0.707 ± 0.011	0.764 ± 0.009
TRAM w/ Thresh. + Comb. E.	0.741 ± 0.005	0.755 ± 0.003	0.839 ± 0.005	0.842 ± 0.002	0.785 ± 0.010	0.725 ± 0.010	0.762 ± 0.004
Proposed + Scale. E.	**0.755 ± 0.009**	**0.770 ± 0.007**	0.838 ± 0.008	**0.850 ± 0.004**	0.798 ± 0.003	**0.741 ± 0.012**	**0.776 ± 0.006**
Proposed + Separ. E.	0.739 ± 0.005	0.761 ± 0.006	**0.843 ± 0.009**	0.849 ± 0.006	**0.808 ± 0.009**	0.715 ± 0.005	0.772 ± 0.007
Proposed + Comb. E.	0.749 ± 0.002	0.766 ± 0.001	0.838 ± 0.010	0.848 ± 0.005	0.801 ± 0.003	0.731 ± 0.004	0.774 ± 0.001
Clean Test Cases, n=477	**Precision**	**Accuracy**	**AUPRC**	**AUROC**	**Sensitivity**	**Specificity**	**F1 Score**
Linear Probing	0.733 ± 0.013	0.765 ± 0.016	0.857 ± 0.006	0.858 ± 0.008	0.834 ± 0.027	0.695 ± 0.017	0.780 ± 0.017
Fine-tuning	0.854 ± 0.009	0.867 ± 0.007	0.943 ± 0.003	0.944 ± 0.005	0.885 ± 0.004	0.849 ± 0.010	0.869 ± 0.006
Confusion Estimation	0.872 ± 0.010	0.886 ± 0.007	**0.952 ± 0.006**	0.953 ± 0.007	0.906 ± 0.003	0.867 ± 0.012	0.889 ± 0.006
TRAM w/ Thresh. + Scale. E.	0.819 ± 0.018	0.846 ± 0.019	0.929 ± 0.015	0.932 ± 0.012	0.887 ± 0.018	0.804 ± 0.020	0.852 ± 0.018
TRAM w/ Thresh. + Separ. E.	0.831 ± 0.016	0.855 ± 0.016	0.930 ± 0.015	0.933 ± 0.012	0.891 ± 0.018	0.818 ± 0.018	0.860 ± 0.016
TRAM w/ Thresh. + Comb. E.	0.839 ± 0.009	0.853 ± 0.010	0.936 ± 0.005	0.938 ± 0.003	0.873 ± 0.013	0.832 ± 0.010	0.855 ± 0.010
Proposed + Scale. E.	**0.873 ± 0.007**	**0.887 ± 0.007**	0.951 ± 0.003	**0.954 ± 0.001**	**0.906 ± 0.010**	**0.868 ± 0.008**	**0.889 ± 0.007**
Proposed + Separ. E.	0.849 ± 0.001	0.872 ± 0.006	0.948 ± 0.007	0.949 ± 0.004	0.905 ± 0.014	0.839 ± 0.002	0.876 ± 0.007
Proposed + Comb. E.	0.860 ± 0.008	0.878 ± 0.010	0.948 ± 0.005	0.949 ± 0.004	0.904 ± 0.013	0.853 ± 0.009	0.881 ± 0.010

BYOL Pretrained Encoder

Dataset The CheXpert [239] dataset consists of 224,316 chest radiographs from 65,240 patients with both frontal and lateral projections available. For self-supervised pre-training in this study, only frontal projections with either anteroposterior (AP) or posteroanterior (PA) chest view from the original training set were retained, resulting in n=191,010 chest x-rays from N=64,534 patients. The dataset was then split patient-wisely, with 80% (N=51,628, n=153,813) of the patients assigned to the training set and the remaining 20% (N=12,906, n=37,197) to the validation.

Pre-training Protocol The ResNet50 network was employed as the encoders and pretrained using the BYOL. The projectors and the predictor followed the original BYOL implementation, comprising a linear layer with an output size of 4,096, a batch normalization layer, a ReLU activation function, and a linear layer with an output size of 256. By referencing [280], we use random resized

cropping of scale (3/4, 4/3) and random contrast and brightness adjustments as the augmentation strategy for BYOL training. The image augmentation was carried out using the kornia library [281]. In addition, the pre-training utilized a batch size of 128, employing the SGD optimizer with a learning rate of 0.03, momentum of 0.9, and a weight decay of 0.0004. To facilitate the training process, a linear warm-up cosine annealing scheduler was applied for the initial 5 epochs. Subsequently, the training continues for a total of 20 epochs, and the epoch that yields the lowest validation loss is selected for the downstream task. The implementation of BYOL was based on the code repository available at https://github.com/lucidrains/byol-pytorch. However, certain modifications were made by removing the hook registration, while leaving the remaining code unchanged.

Table I.2: Testing Performance with DINO Pretrained Encoder on All Test Cases and Clean Test Cases

All Test Cases, n=1005	Precision	Accuracy	AUPRC	AUROC	Sensitivity	Specificity	F1 Score
Linear Probing	0.672 ± 0.000	0.672 ± 0.000	0.609 ± 0.002	0.682 ± 0.002	0.672 ± 0.000	0.672 ± 0.000	0.672 ± 0.000
Fine-tuning	0.792 ± 0.007	0.789 ± 0.004	0.871 ± 0.006	0.872 ± 0.004	**0.783 ± 0.002**	0.794 ± 0.010	0.788 ± 0.003
Confusion Estimation	0.802 ± 0.006	0.795 ± 0.003	**0.873 ± 0.004**	**0.873 ± 0.003**	**0.783 ± 0.005**	0.806 ± 0.008	**0.792 ± 0.002**
TRAM w/ Thresh. + Scale. E.	**0.829 ± 0.011**	0.784 ± 0.004	0.864 ± 0.002	0.866 ± 0.003	0.718 ± 0.023	**0.851 ± 0.016**	0.769 ± 0.009
TRAM w/ Thresh. + Separ. E.	0.816 ± 0.010	0.790 ± 0.002	0.868 ± 0.001	0.869 ± 0.002	0.748 ± 0.015	0.832 ± 0.014	0.781 ± 0.004
TRAM w/ Thresh. + Comb. E.	0.822 ± 0.005	0.792 ± 0.004	0.861 ± 0.003	0.867 ± 0.003	0.745 ± 0.015	0.838 ± 0.009	0.781 ± 0.007
Proposed + Scale. E.	0.813 ± 0.010	0.790 ± 0.007	0.869 ± 0.001	0.870 ± 0.003	0.753 ± 0.010	0.827 ± 0.011	0.782 ± 0.007
Proposed + Separ. E.	0.811 ± 0.014	**0.797 ± 0.004**	0.872 ± 0.003	**0.873 ± 0.005**	0.774 ± 0.013	0.819 ± 0.020	**0.792 ± 0.003**
Proposed + Comb. E.	0.824 ± 0.007	0.795 ± 0.003	0.868 ± 0.002	0.873 ± 0.004	0.750 ± 0.008	0.839 ± 0.008	0.785 ± 0.004
Clean Test Cases, n=477	Precision	Accuracy	AUPRC	AUROC	Sensitivity	Specificity	F1 Score
Linear Probing	0.711 ± 0.000	0.711 ± 0.000	0.613 ± 0.003	0.713 ± 0.002	0.711 ± 0.000	0.711 ± 0.000	0.711 ± 0.000
Fine-tuning	0.923 ± 0.009	0.922 ± 0.005	0.969 ± 0.003	0.971 ± 0.003	0.920 ± 0.004	0.923 ± 0.010	0.922 ± 0.004
Confusion Estimation	0.933 ± 0.008	0.927 ± 0.003	0.972 ± 0.002	0.973 ± 0.001	0.921 ± 0.004	0.934 ± 0.008	0.927 ± 0.003
TRAM w/ Thresh. + Scale. E.	**0.946 ± 0.004**	0.922 ± 0.005	0.970 ± 0.002	0.970 ± 0.002	0.896 ± 0.015	**0.949 ± 0.005**	0.920 ± 0.006
TRAM w/ Thresh. + Separ. E.	0.943 ± 0.005	0.929 ± 0.003	0.972 ± 0.002	0.972 ± 0.002	0.914 ± 0.006	0.945 ± 0.005	0.928 ± 0.003
TRAM w/ Thresh. + Comb. E.	0.943 ± 0.003	0.924 ± 0.004	0.968 ± 0.002	0.969 ± 0.002	0.903 ± 0.011	0.945 ± 0.004	0.923 ± 0.005
Proposed + Scale. E.	0.937 ± 0.006	0.923 ± 0.005	0.970 ± 0.001	0.970 ± 0.002	0.908 ± 0.006	0.939 ± 0.007	0.922 ± 0.005
Proposed + Separ. E.	0.938 ± 0.004	**0.932 ± 0.002**	0.973 ± 0.002	0.973 ± 0.002	**0.925 ± 0.009**	0.939 ± 0.005	**0.931 ± 0.003**
Proposed + Comb. E.	0.943 ± 0.003	0.928 ± 0.003	**0.974 ± 0.001**	**0.974 ± 0.002**	0.912 ± 0.006	0.945 ± 0.003	0.927 ± 0.003

DINO Pretrained Encoder

For the DINO pretrained encoder, we utilized the implementation from the paper [246], and the weights were obtained from the repository available at https://github.com/sangjoon-park/AI-Can-Self-Evolve. According to the paper, the encoder architecture is a small ViT model with 12 layers and 6 heads, using a patch size of 8x8. The pretraining dataset is still CheXpert. During the pretraining process, an Adam optimizer with a learning rate of 0.0001 was utilized. The encoder was pretrained for a total of 5 epochs, and a step decay scheduler was employed. The batch size for the pretraining phase was set to 16. In terms of data augmentation, weak transformations such as random flipping, rotation, and translation were applied to enhance the training diversity. For more detailed information on image preprocessing and the implementation of the DINO method, we

recommend referring to the method section of the original paper [246].

Results

Table I.1 presents the results obtained using a BYOL pre-trained encoder. Among all the listed models, the proposed model with Scale Encoding achieved the highest precision, accuracy, AUROC, specificity, and F1 score, while the best AUPRC and Sensitivity were attained by the proposed model with Separate Encoding. TRAM with Threshold Mechanism demonstrated similar levels of performance as fine-tuning, except for higher sensitivity, F1 score, and lower specificity. The Confusion Estimation method achieved slightly lower performance than the proposed models. On the other hand, the linear probing approach performed significantly worse compared to the other models.

When evaluating on the clean cases, the proposed method with Scale Encoding demonstrated the highest values on almost all metrics, while the Confusion Estimation also achieved the same level of performance, exhibiting the best AUPRC. The TRAM with thresholding showed 2-4% lower performance compared to the proposed methods and Confusion Estimation across almost all metrics. And it did not perform as well as fine-tuning in terms of overall performance.

Table I.2 presents the results of the DINO pretrained encoders. The proposed model with Separate Encoding achieved the highest accuracy, AUROC, and F1 score, while the Confusion Estimation obtained the optimal AUPRC, AUROC, sensitivity, and F1 score. TRAM with thresholding and Score Encoding exhibited the best precision and specificity. However, there is no single model that consistently outperformed the others across all metrics. Overall, the Confusion Estimation method and the proposed methods with Separate Encoding demonstrated the best performances. When tested on clean cases, TRAM with thresholding and Score Encoding achieved the highest precision and specificity. The proposed methods covered the optimal performance in the remaining metrics, achieving an F1 score of 0.931 with Separate Encoding, and an AUPRC of 0.974 together with an AUROC of 0.974 for Combined Encoding. It is worth noting that TRAM with thresholding models tended to have higher precision and specificity but lower F1 scores and sensitivity compared to the proposed models and Confusion Estimation models.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Wikipedia contributors. Erosion (morphology). `https://en.wikipedia.org/w/index.php?title=Erosion_(morphology)`, 2020. Accessed: 2021-3-23.

[2] Wikipedia contributors. Dilation (morphology). `https://en.wikipedia.org/w/index.php?title=Dilation_(morphology)`, 2020. Accessed: 2021-3-23.

[3] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019.

[4] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[5] Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56, 2019.

[6] Ziad Obermeyer and Ezekiel J Emanuel. Predicting the future—big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216, 2016.

[7] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[8] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[9] Daniel A Moses. Deep learning applied to automatic disease detection using chest x-rays. *Journal of Medical Imaging and Radiation Oncology*, 65(5):498–517, 2021.

[10] Peng Huang, Cheng T Lin, Yuliang Li, Martin C Tammemagi, Malcolm V Brock, Sukhinder Atkar-Khattra, Yanxun Xu, Ping Hu, John R Mayo, Heidi Schmidt, et al. Prediction of lung cancer risk at follow-up screening with low-dose ct: a training and validation study of a deep learning method. *The Lancet Digital Health*, 1(7):e353–e362, 2019.

[11] Diego Ardila, Atilla P Kiraly, Sujeeth Bharadwaj, Bokyung Choi, Joshua J Reicher, Lily Peng, Daniel Tse, Mozziyar Etemadi, Wenxing Ye, Greg Corrado, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, 25(6):954–961, 2019.

[12] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019.

[13] Yu Fu, Alexander W Jung, Ramon Viñas Torne, Santiago Gonzalez, Harald Vöhringer, Artem Shmatko, Lucy R Yates, Mercedes Jimenez-Linan, Luiza Moore, and Moritz Gerstung. Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis. *Nature cancer*, 1(8):800–810, 2020.

[14] Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Miraflor, Vitor Werneck Krauss Silva, Klaus J Busam, Edi Brogi, Victor E Reuter, David S Klimstra, and Thomas J Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8):1301–1309, 2019.

[15] Dejun Zhou, Fei Tian, Xiangdong Tian, Lin Sun, Xianghui Huang, Feng Zhao, Nan Zhou, Zuoyu Chen, Qiang Zhang, Meng Yang, et al. Diagnostic evaluation of a deep learning model for optical diagnosis of colorectal cancer. *Nature communications*, 11(1):2961, 2020.

[16] Dexin Gong, Lianlian Wu, Jun Zhang, Ganggang Mu, Lei Shen, Jun Liu, Zhengqiang Wang, Wei Zhou, Ping An, Xu Huang, et al. Detection of colorectal adenomas with a real-time computer-aided system (endoangel): a randomised controlled study. *The lancet Gastroenterology & hepatology*, 5(4):352–361, 2020.

[17] Dan Milea, Raymond P Najjar, Zhubo Jiang, Daniel Ting, Caroline Vasseneix, Xinxing Xu, Masoud Aghsaei Fard, Pedro Fonseca, Kavin Vanikieti, Wolf A Lagrèze, et al. Artificial intelligence to detect papilledema from ocular fundus photographs. *New England Journal of Medicine*, 382(18):1687–1695, 2020.

[18] Zhaoran Wang, Pearse A Keane, Michael Chiang, Carol Y Cheung, Tien Yin Wong, and Daniel Shu Wei Ting. Artificial intelligence and deep learning in ophthalmology. In *Artificial Intelligence in Medicine*, pages 1519–1552. Springer, 2022.

[19] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[20] Pranav Puri, Nneka Comfere, Lisa A Drage, Huma Shamim, Spencer A Bezalel, Mark R Pittelkow, Mark DP Davis, Michael Wang, Aaron R Mangold, Megha M Tollefson, et al. Deep learning for dermatologists: Part ii. current applications. *Journal of the American Academy of Dermatology*, 87(6):1352–1360, 2022.

[21] Barbara J Kenner, Natalie D Abrams, Suresh T Chari, Bruce F Field, Ann E Goldberg, William A Hoos, David S Klimstra, Laura J Rothschild, Sudhir Srivastava, Matthew R Young, et al. Early detection of pancreatic cancer: applying artificial intelligence to electronic health records. *Pancreas*, 50(7):916, 2021.

[22] Young Juhn and Hongfang Liu. Artificial intelligence approaches using natural language processing to advance ehr-based clinical research. *Journal of Allergy and Clinical Immunology*, 145(2):463–469, 2020.

[23] Seyedeh Neelufar Payrovnaziri, Zhaoyi Chen, Pablo Rengifo-Moreno, Tim Miller, Jiang Bian, Jonathan H Chen, Xiuwen Liu, and Zhe He. Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review. *Journal of the American Medical Informatics Association*, 27(7):1173–1185, 2020.

[24] Mollie Hobensack, Jiyoun Song, Danielle Scharp, Kathryn H Bowles, and Maxim Topaz. Machine learning applied to electronic health record data in home healthcare: a scoping review. *International Journal of Medical Informatics*, page 104978, 2022.

[25] Ruowang Li, Yong Chen, Marylyn D Ritchie, and Jason H Moore. Electronic health records and polygenic risk scores for predicting disease risk. *Nature Reviews Genetics*, 21(8):493–502, 2020.

[26] Hadeel Alzoubi, Raid Alzubi, Naeem Ramzan, Daune West, Tawfik Al-Hadhrami, and Mamoun Alazab. A review of automatic phenotyping approaches using electronic health records. *Electronics*, 8(11):1235, 2019.

[27] Nenad Tomašev, Natalie Harris, Sebastien Baur, Anne Mottram, Xavier Glorot, Jack W Rae, Michal Zielinski, Harry Askham, Andre Saraiva, Valerio Magliulo, et al. Use of deep learning to develop continuous-risk models for adverse event prediction from electronic health records. *Nature Protocols*, 16(6):2765–2787, 2021.

[28] Binyam Tilahun, Kassahun Dessie Gashu, Zeleke Abebaw Mekonnen, Berhanu Fikadie Endehabtu, and Dessie Abebaw Angaw. Mapping the role of digital health technologies in the case detection, management, and treatment outcomes of neglected tropical diseases: a scoping review. *Tropical medicine and health*, 49:1–10, 2021.

[29] Noura S Abul-Husn and Eimear E Kenny. Personalized medicine and the power of electronic health records. *Cell*, 177(1):58–69, 2019.

[30] Cris Martin P Jacoba, Leo Anthony Celi, and Paolo S Silva. Biomarkers for progression in diabetic retinopathy: expanding personalized medicine through integration of ai with electronic health records. In *Seminars in ophthalmology*, volume 36, pages 250–257. Taylor & Francis, 2021.

[31] Jan Claassen, Kevin Doyle, Adu Matory, Caroline Couch, Kelly M Burger, Angela Velazquez, Joshua U Okonkwo, Jean-Rémi King, Soojin Park, Sachin Agarwal, et al. Detection of brain activation in unresponsive patients with acute brain injury. *New England Journal of Medicine*, 380(26):2497–2505, 2019.

[32] Mihaela Porumb, Saverio Stranges, Antonio Pescapè, and Leandro Pecchia. Precision medicine and artificial intelligence: a pilot study on deep learning for hypoglycemic events detection based on ecg. *Scientific reports*, 10(1):170, 2020.

[33] Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J Topol. Ai in health and medicine. *Nature medicine*, 28(1):31–38, 2022.

[34] Gaël Varoquaux and Veronika Cheplygina. Machine learning for medical imaging: methodological failures and recommendations for the future. *NPJ digital medicine*, 5(1):48, 2022.

[35] Shengpu Tang, Parmida Davarmanesh, Yanmeng Song, Danai Koutra, Michael W Sjoding, and Jenna Wiens. Democratizing ehr analyses with fiddle: a flexible data-driven preprocessing pipeline for structured clinical data. *Journal of the American Medical Informatics Association*, 27(12):1921–1934, 2020.

[36] Feng Xie, Han Yuan, Yilin Ning, Marcus Eng Hock Ong, Mengling Feng, Wynne Hsu, Bibhas Chakraborty, and Nan Liu. Deep learning for temporal data representation in electronic health records: A systematic review of challenges and methodologies. *Journal of biomedical informatics*, 126:103980, 2022.

[37] Rose Sisk, Lijing Lin, Matthew Sperrin, Jessica K Barrett, Brian Tom, Karla Diaz-Ordaz, Niels Peek, and Glen P Martin. Informative presence and observation in routine health data: a review of methodology for clinical risk prediction. *Journal of the American Medical Informatics Association*, 28(1):155–166, 2021.

[38] Jessica Chubak, Ronit R Dalmat, Noel S Weiss, V Paul Doria-Rose, Douglas A Corley, and Aruna Kamineni. Informative presence in electronic health record data: A challenge in implementing study exclusion criteria. *Epidemiology*, 34(1):29–32, 2023.

[39] Christopher M Sauer, Li-Ching Chen, Stephanie L Hyland, Armand Girbes, Paul Elbers, and Leo A Celi. Leveraging electronic health records for data science: common pitfalls and how to avoid them. *The Lancet Digital Health*, 4(12):e893–e898, 2022.

[40] Jürgen Weese and Cristian Lorenz. Four challenges in medical image analysis from an industrial perspective, 2016.

[41] Lia Morra, Luca Piano, Fabrizio Lamberti, and Tatiana Tommasi. Bridging the gap between natural and medical images through deep colorization. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 835–842. IEEE, 2021.

[42] Djamila Romaissa Beddiar, Mourad Oussalah, Tapio Seppänen, and Rachid Jennane. Acapmed: Automatic captioning for medical imaging. *Applied Sciences*, 12(21):11092, 2022.

[43] Lie Ju, Xin Wang, Lin Wang, Dwarikanath Mahapatra, Xin Zhao, Quan Zhou, Tongliang Liu, and Zongyuan Ge. Improving medical images classification with label noise using dual-uncertainty estimation. *IEEE transactions on medical imaging*, 41(6):1533–1546, 2022.

[44] Davood Karimi, Haoran Dou, Simon K Warfield, and Ali Gholipour. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical image analysis*, 65:101759, 2020.

[45] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

[46] Vladimir Vapnik and Rauf Izmailov. Learning with intelligent teacher: Similarity control and knowledge transfer. In *International Symposium on Statistical Learning and Data Sciences*, pages 3–32. Springer, 2015.

[47] Elyas Sabeti, Joshua Drews, Narathip Reamaroon, Elisa Warner, Michael W Sjoding, Jonathan Gryak, and Kayvan Najarian. Learning using partially available privileged information and label uncertainty: Application in detection of acute respiratory distress syndrome. *IEEE journal of biomedical and health informatics*, 25(3):784–796, 2020.

[48] Haidong Wang, Mohsen Naghavi, Christine Allen, Ryan M Barber, Zulfiqar A Bhutta, Austin Carter, Daniel C Casey, Fiona J Charlson, Alan Zian Chen, Matthew M Coates, et al. Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the global burden of disease study 2015. *The lancet*, 388(10053):1459–1544, 2016.

[49] Theo Vos, Christine Allen, Megha Arora, Ryan M Barber, Zulfiqar A Bhutta, Alexandria Brown, Austin Carter, Daniel C Casey, Fiona J Charlson, Alan Z Chen, et al. Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990–2015: a systematic analysis for the global burden of disease study 2015. *The lancet*, 388(10053):1545–1602, 2016.

[50] Peter Libby and Pierre Theroux. Pathophysiology of coronary artery disease. *Circulation*, 111(25):3481–3488, 2005.

[51] Kristian Thygesen, Joseph S Alpert, Harvey D White, and Joint ESC/ACCF/AHA/WHF Task Force for the Redefinition of Myocardial Infarction. Universal definition of myocardial infarction. *Journal of the American College of Cardiology*, 50(22):2173–2195, 2007.

[52] Donald S Baim and William Grossman. Coronary angiography. In *Cardiac catheterization and angiography. Third edition.* 1986.

[53] Coronary angiography — national heart, lung, and blood institute (NHLBI). `https://www.nhlbi.nih.gov/health-topics/coronary-angiography`. Accessed: 2021-3-3.

[54] Richard M Fleming, Richard L Kirkeeide, Richard W Smalling, K Lance Gould, and Yvonne Stuart. Patterns in visual interpretation of coronary arteriograms as detected by quantitative coronary arteriography. *Journal of the American College of Cardiology*, 18(4):945–951, 1991.

[55] Haibo Zhang, Lin Mu, Shuang Hu, Brahmajee K Nallamothu, Alexandra J Lansky, Bo Xu, Georgios Bouras, David J Cohen, John A Spertus, Frederick A Masoudi, et al. Comparison of physician visual assessment with quantitative coronary angiography in assessment of stenosis severity in china. *JAMA internal medicine*, 178(2):239–247, 2018.

[56] LEONARD M Zir, STEPHEN W Miller, ROBERT E Dinsmore, JP Gilbert, and JW Harthorne. Interobserver variability in coronary angiography. *Circulation*, 53(4):627–632, 1976.

[57] TA DeRouen, JA Murray, and WILLIAM Owen. Variability in the analysis of coronary arteriograms. *Circulation*, 55(2):324–328, 1977.

[58] Patrick W Serruys, Johan HC Reiber, William Wijns, Marcel vd Brand, Cornelis J Kooijman, J Harald, and Paul G Hugenholtz. Assessment of percutaneous transluminal coronary angioplasty by quantitative coronary angiography: diameter versus densitometric area measurements. *The American journal of cardiology*, 54(6):482–488, 1984.

[59] Patrick W Serruys, David P Foley, and Pim J De Feyter. *Quantitative coronary angiography in clinical practice*, volume 145. Springer Science & Business Media, 1993.

[60] Paolo Garrone, GIUSEPPE BIONDI-ZOCCAI, Ilaria Salvetti, Noemi Sina, Imad Sheiban, Peter R Stella, and Pierfrancesco Agostoni. Quantitative coronary angiography in the current era: principles and applications. *Journal of interventional cardiology*, 22(6):527–536, 2009.

[61] Christophe Blondel, Grégoire Malandain, Régis Vaillant, and Nicholas Ayache. Reconstruction of coronary arteries from a single rotational x-ray projection sequence. *IEEE Transactions on Medical Imaging*, 25(5):653–663, 2006.

[62] Guy Shechter, Frédéric Devernay, Eve Coste-Manière, Arshed Quyyumi, and Elliot R McVeigh. Three-dimensional motion tracking of coronary arteries in biplane cineangiograms. *IEEE Transactions on Medical Imaging*, 22(4):493–503, 2003.

[63] Zheng Sun and Ya Zhou. Assessing cardiac dynamics based on x-ray coronary angiograms. *Journal of Multimedia*, 8(1), 2013.

[64] Ivan Cruz-Aceves, Faraz Oloumi, Rangaraj M Rangayyan, Juan G Avina-Cervantes, and Arturo Hernandez-Aguirre. Automatic segmentation of coronary arteries using gabor filters and thresholding based on multiobjective optimization. *Biomedical Signal Processing and Control*, 25:76–85, 2016.

[65] Hamid R Fazlali, Nader Karimi, SM Reza Soroushmehr, Shahram Shirani, Brahmajee K Nallamothu, Kevin R Ward, Shadrokh Samavi, and Kayvan Najarian. Vessel segmentation and catheter detection in x-ray angiograms using superpixels. *Medical & biological engineering & computing*, 56(9):1515–1530, 2018.

[66] Banafsheh Felfelian, Hamid R Fazlali, Nader Karimi, S Mohamad R Soroushmehr, Shadrokh Samavi, B Nallamothu, and Kayvan Najarian. Vessel segmentation in low contrast x-ray angiogram images. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 375–379. IEEE, 2016.

[67] N Maglaveras, K Haris, SN Efstratiadis, J Gourassas, and G Louridas. Artery skeleton extraction using topographic and connected component labeling. In *Computers in Cardiology 2001. Vol. 28 (Cat. No. 01CH37287)*, pages 17–20. IEEE, 2001.

[68] Ali Zifan and Panos Liatsis. Patient-specific computational models of coronary arteries using monoplane x-ray angiograms. *Computational and mathematical methods in medicine*, 2016, 2016.

[69] Binjie Qin, Mingxin Jin, Dongdong Hao, Yisong Lv, Qiegen Liu, Yueqi Zhu, Song Ding, Jun Zhao, and Baowei Fei. Accurate vessel extraction via tensor completion of background layer in x-ray coronary angiograms. *Pattern recognition*, 87:38–54, 2019.

[70] Zhou Shoujun, Yang Jian, Wang Yongtian, and Chen Wufan. Automatic segmentation of coronary angiograms based on fuzzy inferring and probabilistic tracking. *Biomedical engineering online*, 9(1):1–21, 2010.

[71] Faten M'hiri, Luc Duong, Christian Desrosiers, and Mohamed Cheriet. Vesselwalker: Coronary arteries segmentation using random walks and hessian-based vesselness filter. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pages 918–921. IEEE, 2013.

[72] Tianling Lv, Guanyu Yang, Yudong Zhang, Jian Yang, Yang Chen, Huazhong Shu, and Limin Luo. Vessel segmentation using centerline constrained level set method. *Multimedia Tools and Applications*, 78(12):17051–17075, 2019.

[73] Guangkun Ma, Jinzhu Yang, and Hong Zhao. A coronary artery segmentation method based on region growing with variable sector search area. *Technology and Health Care*, (Preprint):1–10, 2020.

[74] Shaoyan Xia, Haogang Zhu, Xiaoli Liu, Ming Gong, Xiaoyong Huang, Lei Xu, Hongjia Zhang, and Jialong Guo. Vessel segmentation of x-ray coronary angiographic image sequence. *IEEE Transactions on Biomedical Engineering*, 67(5):1338–1348, 2019.

[75] Maryam Taghizadeh Dehkordi, Ali Mohamad Doost Hoseini, Saeed Sadri, and Hamid Soltanianzadeh. Local feature fitting active contour for segmenting vessels in angiograms. *IET Computer Vision*, 8(3):161–170, 2013.

[76] Asma Kerkeni, Asma Benabdallah, Antoine Manzanera, and Mohamed Hedi Bedoui. A coronary artery segmentation method based on multiscale analysis and region growing. *Computerized Medical Imaging and Graphics*, 48:49–61, 2016.

[77] Fernando Cervantes-Sanchez, Ivan Cruz-Aceves, Arturo Hernandez-Aguirre, Sergio Solorio-Meza, Teodoro Cordova-Fraga, and Juan Gabriel Aviña-Cervantes. Coronary artery segmentation in x-ray angiograms using gabor filters and differential evolution. *Applied Radiation and Isotopes*, 138:18–24, 2018.

[78] Fernando Cervantes-Sanchez, Ivan Cruz-Aceves, Arturo Hernandez-Aguirre, Martha Alicia Hernandez-Gonzalez, and Sergio Eduardo Solorio-Meza. Automatic segmentation of coronary arteries in x-ray angiograms using multiscale analysis and artificial neural networks. *Applied Sciences*, 9(24):5507, 2019.

[79] Ebrahim Nasr-Esfahani, Nader Karimi, Mohammad H Jafari, S Mohamad R Soroushmehr, Shadrokh Samavi, BK Nallamothu, and Kayvan Najarian. Segmentation of vessels in angiograms using convolutional neural networks. *Biomedical Signal Processing and Control*, 40:240–251, 2018.

[80] Kyungmin Jo, Jihoon Kweon, Young-Hak Kim, and Jaesoon Choi. Segmentation of the main vessel of the left anterior descending artery using selective feature mapping in coronary angiography. *IEEE Access*, 7:919–930, 2018.

[81] Arso M Vukicevic, Serkan Çimen, Nikola Jagic, Gordana Jovicic, Alejandro F Frangi, and Nenad Filipovic. Three-dimensional reconstruction and nurbs-based structured meshing of coronary arteries from the conventional x-ray angiography projection images. *Scientific reports*, 8(1):1–20, 2018.

[82] Tao Wan, Xiaoqing Shang, Weilin Yang, Jianhui Chen, Deyu Li, and Zengchang Qin. Automated coronary artery tree segmentation in x-ray angiography using improved hessian based enhancement and statistical region merging. *Computer methods and programs in biomedicine*, 157:179–190, 2018.

[83] Salma Sameh, Mostafa Abdel Azim, and Ashraf AbdelRaouf. Narrowed coronary artery detection and classification using angiographic scans. In *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, pages 73–79. IEEE, 2017.

[84] Asma Kerkeni, Asma Benabdallah, and Mohamed Hedi Bedoui. Coronary artery multiscale enhancement methods: A comparative study. In *International Conference Image Analysis and Recognition*, pages 510–520. Springer, 2013.

[85] Pearl Mary Samuel and Thanikaiselvan Veeramalai. Vssc net: Vessel specific skip chain convolutional network for blood vessel segmentation. *Computer Methods and Programs in Biomedicine*, 198:105769, 2021.

[86] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[87] Kritika Iyer, Cyrus P Najarian, Aya A Fattah, Christopher J Arthurs, SM Reza Soroushmehr, Vijayakumar Subban, Mullasari A Sankardas, Raj R Nadakuditi, Brahmajee K Nallamothu, and C Alberto Figueroa. Angionet: A convolutional neural network for vessel segmentation in x-ray angiography. *medRxiv*, 2021.

[88] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[89] Xiaotong Shi, Tianming Du, Shuang Chen, Honggang Zhang, Changdong Guan, and Bo Xu. Uenet: A novel generative adversarial network for angiography image segmentation. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1612–1615. IEEE, 2020.

[90] Su Yang, Jihoon Kweon, Jae-Hyung Roh, Jae-Hwan Lee, Heejun Kang, Lae-Jeong Park, Dong Jun Kim, Hyeonkyeong Yang, Jaehee Hur, Do-Yoon Kang, et al. Deep learning segmentation of major vessels in x-ray coronary angiography. *Scientific reports*, 9(1):1–11, 2019.

[91] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[92] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[93] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[94] Jingfan Fan, Jian Yang, Yachen Wang, Siyuan Yang, Danni Ai, Yong Huang, Hong Song, Aimin Hao, and Yongtian Wang. Multichannel fully convolutional network for coronary artery segmentation in x-ray angiograms. *Ieee Access*, 6:44635–44643, 2018.

[95] Lu Wang, Dongxue Liang, Xiaolei Yin, Jing Qiu, Zhiyun Yang, Junhui Xing, Jianzeng Dong, and Zhaoyuan Ma. Coronary artery segmentation in angiographic videos utilizing spatial-temporal information. *BMC Medical Imaging*, 20(1):1–10, 2020.

[96] Xiliang Zhu, Zhaoyun Cheng, Sheng Wang, Xianjie Chen, and Guoqing Lu. Coronary angiography image segmentation based on pspnet. *Computer Methods and Programs in Biomedicine*, page 105897, 2020.

[97] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[98] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[99] Zhi-Hua Zhou and Ji Feng. Deep forest. *arXiv preprint arXiv:1702.08835*, 2017.

[100] Fan Guo, Weiqing Li, Jin Tang, Beiji Zou, and Zhun Fan. Automated glaucoma screening method based on image segmentation and feature extraction. *Medical & Biological Engineering & Computing*, 58(10):2567–2586, 2020.

[101] Vinícius Veloso de Melo, Daniela Mayumi Ushizima, Salety Ferreira Baracho, and Regina Célia Coelho. Gradient boosting decision trees for echocardiogram images. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[102] Alexander Rakhlin, Alexey Shvets, Vladimir Iglovikov, and Alexandr A Kalinin. Deep convolutional neural networks for breast cancer histology image analysis. In *international conference image analysis and recognition*, pages 737–744. Springer, 2018.

[103] Wei Chen, Boqiang Liu, Suting Peng, Jiawei Sun, and Xu Qiao. Computer-aided grading of gliomas combining automatic segmentation and radiomics. *International journal of biomedical imaging*, 2018, 2018.

[104] Liang Sun, Zhanhao Mo, Fuhua Yan, Liming Xia, Fei Shan, Zhongxiang Ding, Bin Song, Wanchun Gao, Wei Shao, Feng Shi, et al. Adaptive feature selection guided deep forest for covid-19 classification with chest ct. *IEEE Journal of Biomedical and Health Informatics*, 24(10):2798–2805, 2020.

[105] Alexander Katzmann, Alexander Muehlberg, Michael Suehling, Dominik Nörenberg, Julian Walter Holch, and Horst-Michael Gross. Deep random forests for small sample size prediction with medical imaging data. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1543–1547. IEEE, 2020.

[106] Carmen Alina Lupascu, Domenico Tegolo, and Emanuele Trucco. Fabc: retinal vessel segmentation using adaboost. *IEEE Transactions on Information Technology in Biomedicine*, 14(5):1267–1274, 2010.

[107] Muhammad Moazam Fraz, Alicja R Rudnicka, Christopher G Owen, and Sarah A Barman. Delineation of blood vessels in pediatric retinal images using decision trees-based ensemble classification. *International journal of computer assisted radiology and surgery*, 9(5):795–811, 2014.

[108] Muhammad Moazam Fraz, Paolo Remagnino, Andreas Hoppe, Bunyarit Uyyanonvara, Alicja R Rudnicka, Christopher G Owen, and Sarah A Barman. An ensemble classification-based approach applied to retinal blood vessel segmentation. *IEEE Transactions on Biomedical Engineering*, 59(9):2538–2548, 2012.

[109] Shuangling Wang, Yilong Yin, Guibao Cao, Benzheng Wei, Yuanjie Zheng, and Gongping Yang. Hierarchical retinal blood vessel segmentation based on feature and ensemble learning. *Neurocomputing*, 149:708–717, 2015.

[110] Shahab Aslani and Haldun Sarnel. A new supervised retinal vessel segmentation method based on robust hybrid features. *Biomedical Signal Processing and Control*, 30:1–12, 2016.

[111] Jiong Zhang, Yuan Chen, Erik Bekkers, Meili Wang, Behdad Dashtbozorg, and Bart M ter Haar Romeny. Retinal vessel delineation using a brain-inspired wavelet transform and random forest. *Pattern Recognition*, 69:107–123, 2017.

[112] Andrew P Witkin. Scale-space filtering. In *Readings in Computer Vision*, pages 329–332. Elsevier, 1987.

[113] Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[114] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. In *International conference on medical image computing and computer-assisted intervention*, pages 130–137. Springer, 1998.

[115] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[116] Rafsanjany Kushol, Md Hasanul Kabir, Md Sirajus Salekin, and ABM Ashikur Rahman. Contrast enhancement by top-hat and bottom-hat transform with optimal structuring element: Application to retinal vessel segmentation. In *International Conference Image Analysis and Recognition*, pages 533–540. Springer, 2017.

[117] Rashindra Manniesing, Max A Viergever, and Wiro J Niessen. Vessel enhancing diffusion: A scale space representation of vessel structures. *Medical image analysis*, 10(6):815–825, 2006.

[118] Subhasis Chaudhuri, Shankar Chatterjee, Norman Katz, Mark Nelson, and Michael Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on medical imaging*, 8(3):263–269, 1989.

[119] Xiaoning Qian, Matthew P Brennan, Donald P Dione, Wawrzyniec L Dobrucki, Marcel P Jackowski, Christopher K Breuer, Albert J Sinusas, and Xenophon Papademetris. A non-parametric vessel detection method for complex vascular structures. *Medical image analysis*, 13(1):49–61, 2009.

[120] J Andrew Bangham, Richard W Harvey, Paul D Ling, and Richard V Aldridge. Morphological scale-space preserving transforms in many dimensions. *Journal of Electronic Imaging*, 5(3):283–299, 1996.

[121] Pavel Yakubovskiy. Segmentation models pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2020.

[122] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[123] William R Crum, Oscar Camara, and Derek LG Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE transactions on medical imaging*, 25(11):1451–1461, 2006.

[124] Ivan Tomek et al. Two modifications of cnn. 1976.

[125] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

[126] M Mostafizur Rahman and D Davis. Cluster based under-sampling for unbalanced cardiovascular data. In *Proceedings of the World Congress on Engineering*, volume 3, pages 3–5, 2013.

[127] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[128] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[129] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[130] Edward L Hannan, Zaza Samadashvili, Gary Walford, David R Holmes, Alice Jacobs, Samin Sharma, Stanley Katz, and Spencer B King, 3rd. Predictors and outcomes of ad hoc versus non-ad hoc percutaneous coronary interventions. *JACC Cardiovasc. Interv.*, 2(4):350–356, 2009.

[131] Steve Sternberg and Geoff Dougherty. Angioplasty: Risks and benefits. `https://www.usnews.com/news/articles/2015/02/11/angioplasty-risks-and-benefits`, 2015. Accessed: 2021-3-28.

[132] Yasushi Matsuzawa and Amir Lerman. Endothelial dysfunction and coronary artery disease: assessment, prognosis, and treatment. *Coron. Artery Dis.*, 25(8):713–724, 2014.

[133] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[134] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8(1):1–21, 2007.

[135] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[136] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021.

[137] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[138] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M Patel. Medical transformer: Gated axial-attention for medical image segmentation. *arXiv preprint arXiv:2102.10662*, 2021.

[139] Michael A Matthay, Lorraine B Ware, Guy A Zimmerman, et al. The acute respiratory distress syndrome. *The Journal of clinical investigation*, 122(8):2731–2740, 2012.

[140] Lorraine B Ware and Michael A Matthay. The acute respiratory distress syndrome. *New England Journal of Medicine*, 342(18):1334–1349, 2000.

[141] Eddy Fan, Daniel Brodie, and Arthur S Slutsky. Acute respiratory distress syndrome: advances in diagnosis and treatment. *Jama*, 319(7):698–710, 2018.

[142] Marya D Zilberberg and Scott K Epstein. Acute lung injury in the medical icu: comorbid conditions, age, etiology, and hospital outcome. *American journal of respiratory and critical care medicine*, 157(4):1159–1164, 1998.

[143] Jesús Villar, Jesús Blanco, and Robert M Kacmarek. Current incidence and outcome of the acute respiratory distress syndrome. *Current opinion in critical care*, 22(1):1–6, 2016.

[144] Wei-jie Guan, Zheng-yi Ni, Yu Hu, Wen-hua Liang, Chun-quan Ou, Jian-xing He, Lei Liu, Hong Shan, Chun-liang Lei, David SC Hui, et al. Clinical characteristics of coronavirus disease 2019 in china. *New England journal of medicine*, 382(18):1708–1720, 2020.

[145] Chaolin Huang, Yeming Wang, Xingwang Li, Lili Ren, Jianping Zhao, Yi Hu, Li Zhang, Guohui Fan, Jiuyang Xu, Xiaoying Gu, et al. Clinical features of patients infected with 2019 novel coronavirus in wuhan, china. *The lancet*, 395(10223):497–506, 2020.

[146] Dawei Wang, Bo Hu, Chang Hu, Fangfang Zhu, Xing Liu, Jing Zhang, Binbin Wang, Hui Xiang, Zhenshun Cheng, Yong Xiong, et al. Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in wuhan, china. *Jama*, 323(11):1061–1069, 2020.

[147] Chaomin Wu, Xiaoyan Chen, Yanping Cai, Xing Zhou, Sha Xu, Hanping Huang, Li Zhang, Xia Zhou, Chunling Du, Yuye Zhang, et al. Risk factors associated with acute respiratory distress syndrome and death in patients with coronavirus disease 2019 pneumonia in wuhan, china. *JAMA internal medicine*, 2020.

[148] Qiurong Ruan, Kun Yang, Wenxia Wang, Lingyu Jiang, and Jianxin Song. Clinical predictors of mortality due to covid-19 based on an analysis of data of 150 patients from wuhan, china. *Intensive care medicine*, 46(5):846–848, 2020.

[149] T Guo, Y Fan, M Chen, et al. Association of cardiovascular disease and myocardial injury with outcomes of patients hospitalized with 2019-coronavirus disease (covid-19). *JAMA Cardiol*, 2020.

[150] Acute Respiratory Distress Syndrome Network. Ventilation with lower tidal volumes as compared with traditional tidal volumes for acute lung injury and the acute respiratory distress syndrome. *New England Journal of Medicine*, 342(18):1301–1308, 2000.

[151] Stephen Fröhlich, Noelle Murphy, Aoife Doolan, Orla Ryan, and John Boylan. Acute respiratory distress syndrome: underrecognition by clinicians. *Journal of critical care*, 28(5):663–668, 2013.

[152] Niall D Ferguson, Fernando Frutos-Vivar, Andrés Esteban, Pilar Fernández-Segoviano, José Antonio Aramburu, Laura Nájera, and Thomas E Stewart. Acute respiratory distress syndrome: underrecognition by clinicians and diagnostic accuracy of three clinical definitions. *Critical care medicine*, 33(10):2228–2234, 2005.

[153] Dale M Needham, Ting Yang, Victor D Dinglas, Pedro A Mendez-Tellez, Carl Shanholtz, Jonathan E Sevransky, Roy G Brower, Peter J Pronovost, and Elizabeth Colantuoni. Timing of low tidal volume ventilation and intensive care unit mortality in acute respiratory distress syndrome. a prospective cohort study. *American journal of respiratory and critical care medicine*, 191(2):177–185, 2015.

[154] Max T Wayne, Thomas S Valley, Colin R Cooke, and Michael W Sjoding. Electronic "sniffer" systems to identify the acute respiratory distress syndrome. *Annals of the American Thoracic Society*, 16(4):488–495, 2019.

[155] Nicolas W Chbat, Weiwei Chu, Monisha Ghosh, Guangxi Li, Man Li, Caitlyn M Chiofolo, Srinivasan Vairavan, Vitaly Herasevich, and Ognjen Gajic. Clinical knowledge-based inference model for early detection of acute lung injury. *Annals of biomedical engineering*, 40(5):1131–1141, 2012.

[156] 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). *medRxiv*, 2020.

[157] 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.

[158] A Agrawal, I Shaheen, M Narasimhan, M Qiu, J Hirsch, M Zhang, and N Hajizadeh. Combining machine learning and traditional statistical modeling to identify risk factors of hospital mortality and directionality for severe ards. In *A25. CRITICAL CARE: THE WIND IN THE WILLOWS-ARDS: OF SWINE AND MEN*, pages A1144–A1144. American Thoracic Society, 2019.

[159] Narathip Reamaroon, Michael W Sjoding, Kaiwen Lin, Theodore J Iwashyna, and Kayvan Najarian. Accounting for label uncertainty in machine learning for detection of acute respiratory distress syndrome. *IEEE journal of biomedical and health informatics*, 23(1):407–415, 2018.

[160] Aline Taoum, Farah Mourad-Chehade, and Hassan Amoud. Evidence-based model for real-time surveillance of ards. *Biomedical Signal Processing and Control*, 50:83–91, 2019.

[161] JY Adams, GB Rehm, I Cortes-Puch, BT Kuhn, JI Nguyen, NR Anderson, and C-N Chuah. A machine learning classifier for early detection of ards using raw ventilator waveform data. In *B24. CRITICAL CARE: GONE WITH THE WIND-MECHANICAL VENTILATION: HFNC, NIV AND INVASIVE*, pages A2745–A2745. American Thoracic Society, 2019.

[162] Christopher N Schmickl, Khurram Shahjehan, Guangxi Li, Rajanigandha Dhokarh, Rahul Kashyap, Christopher Janish, Anas Alsara, Allan S Jaffe, Rolf D Hubmayr, and Ognjen Gajic. Decision support tool for early differential diagnosis of acute lung injury and cardiogenic pulmonary edema in medical critically ill patients. *Chest*, 141(1):43–50, 2012.

[163] Andrew C McKown, Ryan M Brown, Lorraine B Ware, and Jonathan P Wanderer. External validity of electronic sniffers for automated recognition of acute respiratory distress syndrome. *Journal of intensive care medicine*, 34(11-12):946–954, 2019.

[164] Meliha Yetisgen-Yildiz, Cosmin Adrian Bejan, and Mark Wurfel. Identification of patients with acute lung injury from free-text chest x-ray reports. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*, pages 10–17, 2013.

[165] Helen C Koenig, Barbara B Finkel, Satjeet S Khalsa, Paul N Lanken, Meeta Prasad, Richard Urbani, and Barry D Fuchs. Performance of an automated electronic acute lung injury screening system in intensive care unit patients. *Critical care medicine*, 39(1):98–104, 2011.

[166] Kirsten Neudoerffer Kangelaris, Lorraine B Ware, Chen Yu Wang, David R Janz, Zhuo Hanjing, Michael A Matthay, and Carolyn S Calfee. Timing of intubation and clinical outcomes in adults with ards. *Critical care medicine*, 44(1):120, 2016.

[167] Jonathan Messika, Karim Ben Ahmed, Stéphane Gaudry, Romain Miguel-Montanes, Cédric Rafat, Benjamin Sztrymf, Didier Dreyfuss, and Jean-Damien Ricard. Use of high-flow nasal cannula oxygen therapy in subjects with ards: a 1-year observational study. *Respiratory care*, 60(2):162–169, 2015.

[168] Pacharmon Kaewprag, Cheryl Newton, Brenda Vermillion, Sookyung Hyun, Kun Huang, and Raghu Machiraju. Predictive models for pressure ulcers from intensive care unit electronic health records using bayesian networks. *BMC medical informatics and decision making*, 17(2):81–91, 2017.

[169] Chengyin Ye, Tianyun Fu, Shiying Hao, Yan Zhang, Oliver Wang, Bo Jin, Minjie Xia, Modi Liu, Xin Zhou, Qian Wu, et al. Prediction of incident hypertension within the next year: prospective study using statewide electronic health records and machine learning. *Journal of medical Internet research*, 20(1):e22, 2018.

[170] Andrew J Steele, Spiros C Denaxas, Anoop D Shah, Harry Hemingway, and Nicholas M Luscombe. Machine learning models in electronic health records can outperform conventional survival models for predicting patient mortality in coronary artery disease. *PLoS One*, 13(8):e0202344, 2018.

[171] Shang-Ming Zhou, Fabiola Fernandez-Gutierrez, Jonathan Kennedy, Roxanne Cooksey, Mark Atkinson, Spiros Denaxas, Stefan Siebert, William G Dixon, Terence W O'Neill, Ernest Choy, et al. Defining disease phenotypes in primary care electronic health records by a machine learning approach: a case study in identifying rheumatoid arthritis. *PloS one*, 11(5):e0154515, 2016.

[172] Pratik Sinha, Matthew M Churpek, and Carolyn S Calfee. Machine learning classifier models can identify ards phenotypes using readily available clinical data. *American Journal of Respiratory and Critical Care Medicine*, 2019.

[173] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*, pages 372–378. IEEE, 2014.

[174] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[175] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

[176] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[177] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The journal of machine learning research*, 13(1):27–66, 2012.

[178] Amir Dembo, Thomas M Cover, and Joy A Thomas. Information theoretic inequalities. *IEEE Transactions on Information theory*, 37(6):1501–1518, 1991.

[179] Thomas M Cover and Joy A Thomas. Entropy, relative entropy and mutual information. *Elements of information theory*, 2:1–55, 1991.

[180] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[181] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[182] Patrick E Meyer and Gianluca Bontempi. On the use of variable complementarity for feature selection in cancer classification. In *Workshops on applications of evolutionary computation*, pages 91–102. Springer, 2006.

[183] YY Yao, SK Michael Wong, and Cory J Butz. On information-theoretic measures of attribute importance. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 133–137. Springer, 1999.

[184] Hanchuan Peng. mrmr feature selection (using mutual information computation), 2020.

[185] Mark JD Griffiths, Danny Francis McAuley, Gavin D Perkins, Nicholas Barrett, Bronagh Blackwood, Andrew Boyle, Nigel Chee, Bronwen Connolly, Paul Dark, Simon Finney, et al. Guidelines on the management of acute respiratory distress syndrome. *BMJ open respiratory research*, 6(1):e000420, 2019.

[186] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. Efficient L_1 regularized logistic regression. In *Aaai*, volume 6, pages 401–408, 2006.

[187] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[188] Tiehua Wang, Zhuang Liu, Zhaoxi Wang, Meili Duan, Gang Li, Shupeng Wang, Wenxiong Li, Zhaozhong Zhu, Yongyue Wei, David C Christiani, et al. Thrombocytopenia is associated with acute respiratory distress syndrome mortality: an international study. *PLoS One*, 9(4):e94124, 2014.

[189] Yongyue Wei, Zhaoxi Wang, Li Su, Feng Chen, Paula Tejera, Ednan K Bajwa, Mark M Wurfel, Xihong Lin, and David C Christiani. Platelet count mediates the contribution of a genetic variant in lrrc 16a to ards risk. *Chest*, 147(3):607–617, 2015.

[190] Yongyue Wei, Paula Tejera, Zhaoxi Wang, Ruyang Zhang, Feng Chen, Li Su, Xihong Lin, Ednan K Bajwa, B Taylor Thompson, and David C Christiani. A missense genetic variant in lrrc16a/carmil1 improves acute respiratory distress syndrome survival by attenuating platelet count decline. *American journal of respiratory and critical care medicine*, 195(10):1353–1361, 2017.

[191] Xiangao Jiang, Megan Coffee, Anasse Bari, Junzhang Wang, Xinyue Jiang, Jianping Huang, Jichan Shi, Jianyi Dai, Jing Cai, Tianxiao Zhang, et al. Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity. *CMC: Computers, Materials & Continua*, 63:537–51, 2020.

[192] 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.

[193] Dmitry Pechyony and Vladimir Vapnik. On the theory of learnining with privileged information. *Advances in neural information processing systems*, 23, 2010.

[194] Wen Li, Dengxin Dai, Mingkui Tan, Dong Xu, and Luc Van Gool. Fast algorithms for linear and kernel svm+. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2258–2266, 2016.

[195] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[196] Fengyi Tang, Cao Xiao, Fei Wang, Jiayu Zhou, and Li-wei H Lehman. Retaining privileged information for multi-task learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1369–1377, 2019.

[197] Hao Yang, Joey Tianyi Zhou, Jianfei Cai, and Yew Soon Ong. Miml-fcn+: Multi-instance multi-label learning via fully convolutional networks with privileged information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1577–1585, 2017.

[198] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

[199] John Lambert, Ozan Sener, and Silvio Savarese. Deep learning under privileged information using heteroscedastic dropout. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8886–8895, 2018.

[200] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[201] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28:2575–2583, 2015.

[202] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2897–2905, 2018.

[203] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[204] Daniel Hernández-Lobato, Viktoriia Sharmanska, Kristian Kersting, Christoph H Lampert, and Novi Quadrianto. Mind the nuisance: Gaussian process classification using privileged noise. *Advances in Neural Information Processing Systems*, 27:837–845, 2014.

[205] Juan Shu, Yu Li, Sheng Wang, Bowei Xi, and Jianzhu Ma. Disease gene prediction with privileged information and heteroscedastic dropout. *Bioinformatics*, 37(Supplement_1):i410–i417, 2021.

[206] Laurence S Magder and James P Hughes. Logistic regression when the outcome is measured with uncertainty. *American journal of epidemiology*, 146(2):195–203, 1997.

[207] Quang Nguyen, Hamed Valizadegan, and Milos Hauskrecht. Learning classification models with soft-label information. *Journal of the American Medical Informatics Association*, 21(3):501–508, 2014.

[208] Paul G Byrnes and Francisco A DiazDelaO. Kernel logistic regression: A robust weighting for imbalanced classes with noisy labels. In *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, pages 30–34. IEEE, 2018.

[209] Jionglin Wu, Jason Roy, and Walter F Stewart. Prediction modeling using ehr data: challenges, strategies, and a comparison of machine learning approaches. *Medical care*, pages S106–S113, 2010.

[210] Evangelia Christodoulou, Jie Ma, Gary S Collins, Ewout W Steyerberg, Jan Y Verbakel, and Ben Van Calster. A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal of clinical epidemiology*, 110:12–22, 2019.

[211] Anita L Lynam, John M Dennis, Katharine R Owen, Richard A Oram, Angus G Jones, Beverley M Shields, and Lauric A Ferrat. Logistic regression has similar performance to optimised machine learning algorithms in a clinical setting: application to the discrimination between type 1 and type 2 diabetes in young adults. *Diagnostic and prognostic research*, 4(1):1–10, 2020.

[212] Simon Nusinovici, Yih Chung Tham, Marco Yu Chak Yan, Daniel Shu Wei Ting, Jialiang Li, Charumathi Sabanayagam, Tien Yin Wong, and Ching-Yu Cheng. Logistic regression was as good as machine learning for predicting major chronic diseases. *Journal of clinical epidemiology*, 122:56–69, 2020.

[213] Xuan Song, Xinyan Liu, Fei Liu, and Chunting Wang. Comparison of machine learning and logistic regression models in predicting acute kidney injury: A systematic review and meta-analysis. *International Journal of Medical Informatics*, 151:104484, 2021.

[214] Francis Bach. Learning theory from first principles. *Draft of a book, version of Sept*, 6:2021, 2021.

[215] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[216] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[217] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[218] V. M. ARDS Definition Task Force, Ranieri, G. D. Rubenfeld, B. T. Thompson, N. D. Ferguson, E. Caldwell, E. Fan, L. Camporota, and A. S. Slutsky. Acute respiratory distress syndrome: the berlin definition. *JAMA*, 307(23):2526–2533, 2012.

[219] C. Guérin, T. Thompson, and R. Brower. The ten diseases that look like ards. *Intensive Care Med*, 41(6):1099–1102, 2015.

[220] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[221] Giacomo Bellani, John G Laffey, Tài 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.

[222] Yub Raj Sedhai, Mengdan Yuan, Scott W Ketcham, Ivan Co, Dru D Claar, Jakob I McSparron, Hallie C Prescott, and Michael W Sjoding. Validating measures of disease severity in acute respiratory distress syndrome. *Annals of the American Thoracic Society*, 18(7):1211–1218, 2021.

[223] Sarah Sheard, Praveen Rao, and Anand Devaraj. Imaging of acute respiratory distress syndrome. *Respiratory care*, 57(4):607–612, 2012.

[224] Jin-Min Peng, Chuan-Yun Qian, Xiang-You Yu, Ming-Yan Zhao, Shu-Sheng Li, Xiao-Chun Ma, Yan Kang, Fa-Chun Zhou, Zhen-Yang He, Tie-He Qin, et al. Does training improve diagnostic accuracy and inter-rater agreement in applying the berlin radiographic definition of acute respiratory distress syndrome? a multicenter prospective study. *Critical Care*, 21(1):1–8, 2017.

[225] Shannon L Goddard, Gordon D Rubenfeld, Venika Manoharan, Shelly P Dev, John Laffey, Giacomo Bellani, Tai Pham, and Eddy Fan. The randomized educational acute respiratory distress syndrome diagnosis study: a trial to improve the radiographic diagnosis of acute respiratory distress syndrome. *Critical care medicine*, 46(5):743–748, 2018.

[226] Nesrine Zaglam, Philippe Jouvet, Olivier Flechelles, Guillaume Emeriaud, and Farida Cheriet. Computer-aided diagnosis system for the acute respiratory distress syndrome from chest radiographs. *Computers in biology and medicine*, 52:41–48, 2014.

[227] Michael W Sjoding, Daniel Taylor, Jonathan Motyka, Elizabeth Lee, Dru Claar, Jakob I McSparron, Sardar Ansari, Meeta Prasad Kerlin, John P Reilly, Michael GS Shashaty, et al. Deep learning to detect acute respiratory distress syndrome on chest radiographs: a retrospective study with external validation. *The Lancet Digital Health*, 3(6):e340–e348, 2021.

[228] Mohammad Yahyatabar, Philippe Jouvet, Donatien Fily, Jérome Rambaud, Michaël Levy, Robinder G Khemani, and Farida Cheriet. A web-based platform for the automatic stratification of ards severity. *Diagnostics*, 13(5):933, 2023.

[229] Hieu H Pham, Tung T Le, Dat Q Tran, Dat T Ngo, and Ha Q Nguyen. Interpreting chest x-rays via cnns that exploit hierarchical disease dependencies and uncertainty labels. *Neurocomputing*, 437:186–194, 2021.

[230] Yair Dgani, Hayit Greenspan, and Jacob Goldberger. Training a neural network based on unreliable human annotation of medical images. In *2018 IEEE 15th International symposium on biomedical imaging (ISBI 2018)*, pages 39–42. IEEE, 2018.

[231] Cheng Xue, Qi Dou, Xueying Shi, Hao Chen, and Pheng-Ann Heng. Robust learning at noisy labeled medical images: Applied to skin lesion classification. In *2019 IEEE 16th International symposium on biomedical imaging (ISBI 2019)*, pages 1280–1283. IEEE, 2019.

[232] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11244–11253, 2019.

[233] Mark Collier, Rodolphe Jenatton, Effrosyni Kokiopoulou, and Jesse Berent. Transfer and marginalize: Explaining away label noise with privileged information. In *International Conference on Machine Learning*, pages 4219–4237. PMLR, 2022.

[234] Anthony J Viera, Joanne M Garrett, et al. Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363, 2005.

[235] Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Bobby Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. Direct uncertainty prediction for medical second opinions. In *International Conference on Machine Learning*, pages 5281–5290. PMLR, 2019.

[236] Joseph Paul Cohen, Joseph D. Viviano, Paul Bertin, Paul Morrison, Parsa Torabian, Matteo Guarrera, Matthew P Lungren, Akshay Chaudhari, Rupert Brooks, Mohammad Hashir, and Hadrien Bertrand. TorchXRayVision: A library of chest X-ray datasets and models. In *Medical Imaging with Deep Learning*, 2022.

[237] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[238] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria De La Iglesia-Vaya. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical image analysis*, 66:101797, 2020.

[239] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

[240] 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):317, 2019.

[241] Guillermo Ortiz-Jimenez, Mark Collier, Anant Nawalgaria, Alexander D'Amour, Jesse Berent, Rodolphe Jenatton, and Effrosyni Kokiopoulou. When does privileged information explain away label noise? *arXiv preprint arXiv:2303.01806*, 2023.

[242] Xintong Shi, Wenzhi Cao, and Sebastian Raschka. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *arXiv preprint arXiv:2111.08851*, 2021.

[243] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[244] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Self-knowledge distillation based self-supervised learning for covid-19 detection from chest x-ray images. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1371–1375. IEEE, 2022.

[245] Matej Gazda, Ján Plavka, Jakub Gazda, and Peter Drotar. Self-supervised deep convolutional neural network for chest x-ray classification. *IEEE Access*, 9:151972–151982, 2021.

[246] Sangjoon Park, Gwanghyun Kim, Yujin Oh, Joon Beom Seo, Sang Min Lee, Jin Hwan Kim, Sungjun Moon, Jae-Kwang Lim, Chang Min Park, and Jong Chul Ye. Self-evolving vision transformer for chest x-ray diagnosis through knowledge distillation. *Nature Communications*, 13(1):3848, 2022.

[247] Tuan Truong, Sadegh Mohammadi, and Matthias Lenga. How transferable are self-supervised features in medical image classification tasks? In *Machine Learning for Health*, pages 54–74. PMLR, 2021.

[248] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[249] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[250] Negar Farzaneh, Sardar Ansari, Elizabeth Lee, Kevin R Ward, and Michael W Sjoding. Collaborative strategies for deploying artificial intelligence to complement physician diagnoses of acute respiratory distress syndrome. *NPJ Digital Medicine*, 6(1):62, 2023.

[251] Boah Kim, Yujin Oh, and Jong Chul Ye. Diffusion adversarial representation learning for self-supervised vessel segmentation. *arXiv preprint arXiv:2209.14566*, 2022.

[252] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.

[253] Veronika Cheplygina, Marleen de Bruijne, and Josien PW Pluim. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis*, 54:280–296, 2019.

[254] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 6(12):1346–1352, 2022.

[255] Zeyu Ren, Shuihua Wang, and Yudong Zhang. Weakly supervised machine learning. *CAAI Transactions on Intelligence Technology*, 2023.

[256] Yuqi Si, Jingcheng Du, Zhao Li, Xiaoqian Jiang, Timothy Miller, Fei Wang, W Jim Zheng, and Kirk Roberts. Deep representation learning of patient data from electronic health records (ehr): A systematic review. *Journal of biomedical informatics*, 115:103671, 2021.

[257] Ali Amirahmadi, Mattias Ohlsson, and Kobra Etminani. Deep learning prediction models based on ehr trajectories: A systematic review. *Journal of Biomedical Informatics*, page 104430, 2023.

[258] Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*, pages 222–235, 2020.

[259] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

[260] Hans-Christian Thorsen-Meyer, Annelaura B Nielsen, Anna P Nielsen, Benjamin Skov Kaas-Hansen, Palle Toft, Jens Schierbeck, Thomas Strøm, Piotr J Chmura, Marc Heimann, Lars Dybdahl, et al. Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: a retrospective study of high-frequency data in electronic patient records. *The Lancet Digital Health*, 2(4):e179–e191, 2020.

[261] Farida Mohsen, Hazrat Ali, Nady El Hajj, and Zubair Shah. Artificial intelligence-based methods for fusion of electronic health records and imaging data. *Scientific Reports*, 12(1):17981, 2022.

[262] Muhammad Adeel Azam, Khan Bahadar Khan, Sana Salahuddin, Eid Rehman, Sajid Ali Khan, Muhammad Attique Khan, Seifedine Kadry, and Amir H Gandomi. A review on multimodal medical image fusion: Compendious analysis of medical modalities, multimodal databases, fusion techniques and quality metrics. *Computers in biology and medicine*, 144:105253, 2022.

[263] Yanbo Xu, Siddharth Biswal, Shriprasad R Deshpande, Kevin O Maher, and Jimeng Sun. Raim: Recurrent attentive and intensive model of multimodal patient monitoring data. In *Proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining*, pages 2565–2573, 2018.

[264] 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):22147, 2020.

[265] Pegah Khosravi, Maria Lysandrou, Mahmoud Eljalby, Qianzi Li, Ehsan Kazemi, Pantelis Zisimopoulos, Alexandros Sigaras, Matthew Brendel, Josue Barnes, Camir Ricketts, et al. A deep learning approach to diagnostic classification of prostate cancer using pathology–radiology fusion. *Journal of Magnetic Resonance Imaging*, 54(2):462–471, 2021.

[266] Qiuling Suo, Weida Zhong, Fenglong Ma, Ye Yuan, Jing Gao, and Aidong Zhang. Metric learning on healthcare data with incomplete modalities. In *IJCAI*, volume 3534, page 3540, 2019.

[267] Can Cui, Zuhayr Asad, William F Dean, Isabelle T Smith, Christopher Madden, Shunxing Bao, Bennett A Landman, Joseph T Roland, Lori A Coburn, Keith T Wilson, et al. Multimodal learning with missing data for cancer diagnosis using histopathological and genomic data. In *Medical Imaging 2022: Computer-Aided Diagnosis*, volume 12033, pages 357–364. SPIE, 2022.

[268] Chaohe Zhang, Xu Chu, Liantao Ma, Yinghao Zhu, Yasha Wang, Jiangtao Wang, and Junfeng Zhao. M3care: Learning with missing modalities in multimodal healthcare data. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2418–2428, 2022.

[269] Yufeng Zhang, Zijun Gao, Emily Wittrup, Jonathan Gryak, and Kayvan Najarian. Increasing efficiency of svmp+ for handling missing values in healthcare prediction. *PLOS Digital Health*, 2(6):e0000281, 2023.

[270] Changqing Zhang, Yajie Cui, Zongbo Han, Joey Tianyi Zhou, Huazhu Fu, and Qinghua Hu. Deep partial multi-view learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2402–2415, 2020.

[271] Padmavathi Kora, Chui Ping Ooi, Oliver Faust, U Raghavendra, Anjan Gudigar, Wai Yee Chan, K Meenakshi, K Swaraja, Pawel Plawiak, and U Rajendra Acharya. Transfer learning techniques for medical image analysis: A review. *Biocybernetics and Biomedical Engineering*, 42(1):79–107, 2022.

[272] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

[273] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[274] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[275] Jin Yang, Yuanjie Li, Qingqing Liu, Li Li, Aozi Feng, Tianyi Wang, Shuai Zheng, Anding Xu, and Jun Lyu. Brief introduction of medical database and data mining technology in big data era. *Journal of Evidence-Based Medicine*, 13(1):57–69, 2020.

[276] Tony Lindeberg. Scale-space. *Wiley Encyclopedia of Computer Science and Engineering*, pages 2495–2504, 2007.

[277] J Theodore Dodge Jr, B Greg Brown, Edward L Bolson, and Harold T Dodge. Lumen diameter of normal human coronary arteries. influence of age, sex, anatomic variation, and left ventricular hypertrophy or dilation. *Circulation*, 86(1):232–246, 1992.

[278] Joachim Weickert and Hanno Scharr. A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. *Journal of Visual Communication and Image Representation*, 13(1-2):103–118, 2002.

[279] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

[280] Rogier Van der Sluijs, Nandita Bhaskhar, Daniel Rubin, Curtis Langlotz, and Akshay Chaudhari. Exploring image augmentations for siamese representation learning with chest x-rays. *arXiv preprint arXiv:2301.12636*, 2023.

[281] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3674–3683, 2020.