IMPERIAL

ENHANCING CLINICAL DECISION MAKING WITH INTERPRETABLE AI

Author
Anish Narain
CID: 02045150

Supervised by
Dr Sonali Parbhoo

A thesis submitted in fulfillment of requirements for the degree of
Bachelor of Engineering in Electronic and Information Engineering

Department of Electrical and Electronic Engineering
Imperial College London
2024

Abstract

The digitisation of medical data has created opportunities to leverage Artificial Intelligence (AI) to enhance healthcare delivery. However, the opaque nature of this technology poses challenges with interpretability, which is imperative in clinical settings. Concept Bottleneck Models (CBMs) offer a solution by using human-understandable concepts as an intermediate layer between the input data and the final prediction, allowing for clearer insights into the decision-making process. This project focuses on implementing and evaluating CBMs to improve transparency and effectiveness in clinical prediction tasks, specifically predicting patient mortality and diagnosing Acute Respiratory Distress Syndrome (ARDS) using the Medical Information Mart for Intensive Care (MIMIC) database. By fine-tuning CBMs on electronic health records (EHRs), this study achieved 83% accuracy in mortality prediction, matching standard regression models while providing superior interpretability. Additionally, CBMs trained on EHRs for ARDS diagnosis outperformed baseline regression models by 7%, achieving 68% accuracy. The incorporation of a large language model (LLM) pipeline to process free-text clinical notes and generate additional concepts further enhanced CBM performance, improving ARDS diagnosis accuracy by up to 12%. It also increased completeness scores, thus reducing the likelihood of information leakage. This research underscores the potential of CBMs and LLMs to enhance clinical AI applications through improved interpretability and performance, fostering trust among healthcare professionals and facilitating better clinical decision-making.

Declaration of Originality

1. I affirm that I have submitted an electronic copy of my final year project report to the provided EEE link.
2. I affirm that I have submitted an identical electronic copy of my final year project to the provided Blackboard module for plagiarism checking.
3. I affirm that I have provided explicit references for all the material in my thesis that is not authored by me, but is represented as my own work.
4. I acknowledge the use of ChatGPT v4 and Microsoft Editor in preparing my report. These tools have helped improve the quality of my English throughout. However, all technical content and references come from my original text.

Acknowledgments

Firstly, I would like to express my deepest gratitude to Dr Sonali Parbhoo. Working with her was a privilege because of her exceptional mentorship and unwavering optimism. She worked tirelessly to create an ideal environment for my project. I always felt comfortable voicing my concerns and consistently received the help I needed.

I am also grateful for the opportunity to be part of Dr Parbhoo's AI for Actionable Impact (AI4AI) lab. Thanks to her dedication in arranging regular tea meetings and group discussions, I had the chance to explore topics beyond my own thesis. This experience has truly been a highlight of my time at Imperial, and I have already begun recommending my friends on the MEng course to work with Dr Parbhoo next year.

I would also like to thank my fantastic AI4AI colleagues, especially Dr Dominic Marshall, Chang Liu, Nikita Narayanan, Jared Joselowitz, and Angelos Ragkousis, for their invaluable support throughout this project.

Finally, I extend my gratitude to my friends for being my source of inspiration over the past three years and, as always, my family for their unconditional love. Projects like this were individual, but thanks to you, I never felt alone.

Contents

Abstract **i**

Declaration of Originality **iii**

Acknowledgments **v**

1 Introduction **1**

1.1 Motivation 1

1.2 Project Objectives 2

1.3 Chapter Summaries 3

1.4 Project Contributions 4

2 Background **5**

2.1 Interpretability 5

2.2 Concept Bottleneck Models 6

2.3 CBM Training Methods 8

2.4 Large Language Models 9

3 Related Work **13**

3.1 LLMs Trained on Medical Data 13

3.2 Considerations of Using LLMs for Concept Generation 14

3.3 Leakage in CBMs 16

3.4 Relevant MIMIC Projects 18

3.4.1 MIMIC-IV Overview 18

3.4.2 Predicting Intervention Onset in the ICU 19

3.4.3 CBM for Predicting Vasopressor Onset 19

3.4.4 Google LLMs for Extracting Insights from Clinical Notes 19

3.4.5 Evaluating LLM Tasks for Extracting Insights from Clinical Notes 20

3.4.6 LLMs Trained on MIMIC 20

3.5 Mortality Prediction using MIMIC 21

3.6 Identifying Acute Respiratory Distress Syndrome 24

4 Mortality Prediction with CBM **27**

4.1 Cohort Selection 28

4.2 Dataset Construction 29

4.2.1 Concepts from SOFA Scores 29

4.2.2 Concepts from Anion Gap, Albumin, and Bilirubin 31

4.2.3 Data Pre-processing 31

4.3 CBM Training 33

4.3.1 Independent Training 33

4.3.2 Sequential Training 34

4.3.3 Joint Training 34

4.4 Investigations 34

4.4.1 Fine-tuning 34

4.4.2 Comparison Against Logistic Regression Models 40

4.4.3 Analysing Model Weights 41

4.5 Future Work 48

5 ARDS Identification with CBM **51**

5.1 Cohort Selection 52

5.1.1 Berlin Definition 52

5.1.2 Screening Process 53

5.2 Dataset Construction 54

5.2.1 Concepts Derived from SOFA Scores 54

5.2.2 Concepts Derived from Sepsis and Shock 55

5.2.3 Concepts Derived from Ventilation Status 55

5.2.4 Concepts Derived from Pre-existing Respiratory Conditions 56

5.2.5 Data Pre-processing 57

5.3 CBM Training 57

5.4 Investigations 57

5.4.1 Fine-tuning 57

5.4.2 Comparison Against Logistic Regression Models 61

5.4.3 Analysing Model Weights 61

5.5 Future Work 62

6 Enhancing ARDS CBM with Labels from LLM **65**

6.1 Dataset Construction 66

6.1.1 Data Pre-processing 66

6.1.2 LLM Concept Labels 67

6.2 Implementing LLM for Clinical Concept Labelling 69

6.2.1 Evaluating Open Source LLMs for Concept Labelling 70

6.2.2 Scaling Up and Further Investigations 74

6.2.3 Final LLM Implementation 77

6.2.4 Random Sampling Validation 77

6.3 Augmenting Concept Set with LLM Labels 78

6.3.1 Initial Attempt 78

6.3.2 Using Altered Architecture 80

6.3.3 Investigating Completeness Score 82

6.4 Future Work 83

Conclusions **87**

A Ethical, Legal, and Safety Plan **89**

B Additional CBM Information **91**

B.1 CBM Training Algorithms 91

B.2 Mortality Prediction CBM 94

B.2.1 Full List of Features and Concepts 94

B.2.2 Architecture 94

B.3 ARDS Identification CBM 96

B.3.1 Full List of Features and Concepts for Vanilla CBM 96

B.3.2 Vanilla CBM Loss Plots 96

B.3.3 Vanilla CBM Heat Maps 96

B.3.4 Augmented Concept Set Comparison 96

C Project Code **107**

Bibliography **109**

1
Introduction

Contents

1.1 Motivation 1
1.2 Project Objectives 2
1.3 Chapter Summaries 3
1.4 Project Contributions 4

1.1 Motivation

The digitisation of medical data presents a prime opportunity to apply Artificial Intelligence (AI) technology such as Machine Learning (ML) and Natural Language Processing (NLP) to enhance healthcare delivery. ML algorithms have demonstrated great promise, particularly in predicting patient outcomes and aiding early disease diagnosis. Additionally, Large Language Models (LLMs) represent a major advancement in NLP with impressive reasoning capabilities using contextual data. For instance, Google's MedPaLM2, which is specifically trained on medical literature, has matched licensed physicians' performance on benchmark medical exams.

However, the inherent problem with these models is their black-box nature. It is challenging to interpret the complex layers within an ML model or understand how an LLM makes its decisions. Interpretability is crucial in the medical domain for clinicians to identify spurious correlations and make necessary corrections. Additionally, due to the human-centric nature of healthcare, it is essential to identify any biases within the models and ensure they align with high standards of

clinical behaviour.

Recently, there has been a significant push for interpretable models in medicine. A prominent example is the Concept Bottleneck Model (CBM). Unlike standard models that produce final clinical decisions directly from medical data, CBMs first predict intermediate concepts before making the final prediction. These concepts translate raw input data into understandable, high-level features, which is crucial in clinical settings where understanding the reasoning behind a decision is as important as the decision itself. Their transparency helps build trust in AI systems among healthcare professionals. Furthermore, concepts can highlight areas where the model is underperforming or misdiagnosing, allowing for targeted interventions to improve predictions.

While CBMs offer significant benefits, their performance depends highly on the quality of the data they are trained on. Medical information comes in various forms, including electronic health records, medical imaging, and clinical text. This project explores how CBMs can be implemented on relevant medical data from patients in intensive care units (ICUs) for clinical prediction tasks. The results show that CBMs can match or even exceed the predictive performance of standard regression models, while displaying superior interpretability.

Finally, as discussed in Google's MedPaLM paper [1], medicine is a human endeavour where language enables key interactions for and between clinicians, researchers, and patients. However, current clinical ML models fail to fully utilise language. Hence, this project also explores generating insights from unstructured clinical text with LLMs. By augmenting the concept set of CBMs with LLM-generated labels, these enhanced CBMs exhibit notable performance improvements.

1.2 Project Objectives

This project explored two clinical tasks using the Medical Information Mart for Intensive Care (MIMIC) database: predicting patient mortality and diagnosing Acute Respiratory Distress Syndrome (ARDS). The MIMIC database is an invaluable resource for machine learning research, containing comprehensive health-related information on patients admitted to the ICUs of the Beth Israel Deaconess Medical Center [2]. It includes hospital-wide electronic health records (EHRs), which store detailed patient information such as medical history, laboratory test results, vital signs, and diagnoses. In January 2023, a collection of free-text clinical notes written by healthcare providers in the ICU was added to the MIMIC-IV database [3]. These notes are crucial for understanding a patient's clinical course.

The first project objective was to use MIMIC's EHRs to fine-tune a concept bottleneck model (CBM) for predicting mortality across a broad cohort of patients. A targeted dataset of lab measurements and vital signs was selected to capture generalisable trends within the cohort. The interpretability of the CBM was leveraged for evaluation, uncovering valuable insights about the data and the model's behaviour. One challenge with using CBMs is the issue of leakage, where the model inadvertently uses information from downstream tasks or future data to predict intermediate concepts, leading to artificially high performance. This project also investigated the role of leakage in the mortality prediction CBM.

The second project objective was to train a CBM to diagnose ARDS. This study was conducted in collaboration with Dr Dominic Marshall, a licensed physician in the AI4AI lab, who provided clinical expertise for the task. Dr Marshall had pre-screened a small cohort of patients with the potential of having ARDS and manually reviewed extensive patient information for a robust diagnosis. Identifying ARDS was challenging because it required reviewing various sources of medical information—not just lab measurements and vital signs, but also ventilation status and clinical notes, such as radiology reports, discharge summaries, and electrocardiogram studies. To address this challenge, this project implemented a CBM to identify ARDS in Dr Marshall's cohort. Initially, the CBM was trained on EHR data alone and the performance was evaluated. Subsequently, the CBM was enhanced with additional insights from clinical notes using Meta's Llama3 large language model. Having observed Dr Marshall's manual review process, new concept labels were selected from the notes, to identify conditions not explicitly found in the EHR data. These concepts were integrated into the CBM, and the impact on performance and leakage was investigated.

1.3 Chapter Summaries

The work done in this thesis is presented in the following chapters.

- **Chapter 2**: Explains the fundamental theory behind concept bottleneck models (CBMs) and large language models (LLMs).
- **Chapter 3**: Reviews the relevant literature for this project. This includes papers on CBMs, LLMs for medical information extraction, and ML research on the MIMIC database. Additionally, it considers studies on predicting patient mortality and identifying ARDS.

- **Chapter 4**: Describes the construction of the dataset for mortality prediction. It details the training of CBMs on this dataset, the investigation of the model's predictions, and outlines future research possibilities.
- **Chapter 5**: Details the construction of the dataset for ARDS identification. Similar to Chapter 4, it covers the training of CBMs on this dataset, the investigation of the model's predictions, and suggests further research opportunities.
- **Chapter 6**: Evaluates various open-source LLMs for the concept labelling task. It describes how clinical text was processed by the LLM pipeline, the incorporation of the new concept set into the original CBM model, and examines the effects of this enhancement along with potential future work.

1.4 Project Contributions

1. Implemented interpretable concept bottleneck models to predict mortality using a small concept set on a broad patient cohort. The highest performing, fine-tuned CBM achieved an accuracy of 83%, comparable to a regression baseline.
2. Developed similar interpretable CBMs for diagnosing ARDS in a pre-screened patient cohort. The highest performing, fine-tuned CBM achieved an accuracy of 68%, outperforming the regression baseline by 7%.
3. Confirmed the interpretability of the models by analysing predictions. Insights from this analysis helped evaluate the CBMs' clinical reasoning and explore the impact of leakage on final predictions.
4. Implemented an LLM pipeline using Meta's Llama3 model to process a variety of clinical texts and produce additional concept labels.
5. Augmented the concept set of the original ARDS CBMs with these labels and identified the ideal training architecture to benefit from the new information.
6. Demonstrated that the enhanced ARDS CBMs were more accurate at identifying ARDS. The largest accuracy improvement was 12%. Furthermore, the enhanced CBMs showed a substantial increase in completeness score, indicating that the new concept labels helped reduce the CBMs' potential for leakage.

2

Background

Contents

2.1 Interpretability 5
2.2 Concept Bottleneck Models 6
2.3 CBM Training Methods 8
2.4 Large Language Models 9

2.1 Interpretability

Interpretability in machine learning is the ability of a system to explain or present its decisions in understandable terms to humans. In [4], the authors explain when a ML system requires interpretability, and their reasoning emphasises the importance of interpretability for ML models operating in the medical domain. Firstly, and most importantly, models need to be interpretable to ensure safety. Interpretability allows medical professionals to understand and verify a system's decisions, which is important when stakes are high, and consequence of errors can be severe. Secondly, the paper considers ethical considerations. In healthcare, it is important to guard against certain kinds of discrimination. But without interpretability, the concept of fairness is too abstract to incorporate into the system. Finally, the authors discuss the importance of interpretability to help boost scientific understanding and to avoid mismatched objectives between user and model.

Many state-of-art models, like deep neural networks, are inherently black-box. They consist of multiple layers and complex structures that transform raw input data into target outputs without

revealing the decision-making process. This lack of interpretability makes them difficult to adopt for healthcare. However, work has been done to meet this demand for transparency, with concept bottleneck models being a notable example.

2.2 Concept Bottleneck Models

Supervised learning is a type of machine learning where the model is trained using data that is labelled. This means the data contains examples of both inputs and correct outputs [5]. The model learns from these examples to become better at predicting the desired output when provided new data. Concept bottleneck models are a specific type of supervised learning model which not only train on raw inputs and target outputs, but also pre-defined human-understandable concepts. The approach involves mapping raw inputs x to concepts c and then mapping c to target labels y, [6].

The approach of models initially predicting concepts and then using those concepts to make further predictions was not originally proposed by [7]. However, this paper's formulation of the CBM architecture, demonstration of strong task accuracy and subsequent discussion of future work has inspired a wave of papers exploring the nuances of concept bottleneck models. Hence this paper serves as a strong basis for understanding CBMs.

In the paper, the authors utilise an example of a radiologist using a machine learning model to examine the knee x-ray of a patient for arthritis. A traditional model would take in the raw pixels input x and produce the target output y, which is the severity of arthritis. Concept bottleneck models, on the other hand, first predict "an intermediate set of concepts c like joint space narrowing and bone spurs." These concepts c would then be used to predict target y. Figure 2.1 from the paper illustrates this process.

By providing users with such concepts, they get an insight into the model's reasoning. This means it would be easier for healthcare providers, who are using CBMs, to understand the rationale behind clinical predictions. In [8], researchers created a concept bottleneck model which predicted how long, after administration, it would take vasopressors (blood medication) to start having effects on a patient. They presented the concepts generated by the model to an expert and reported that the concepts allowed for meaningful clinical insights about the prediction. The paper stated that even for concepts that did not show immediate medical significance, having this concept framework made it much easier to evaluate the model's decision making process.

Furthermore, [7] describes how concepts in CBMs provide the ability to make interventions

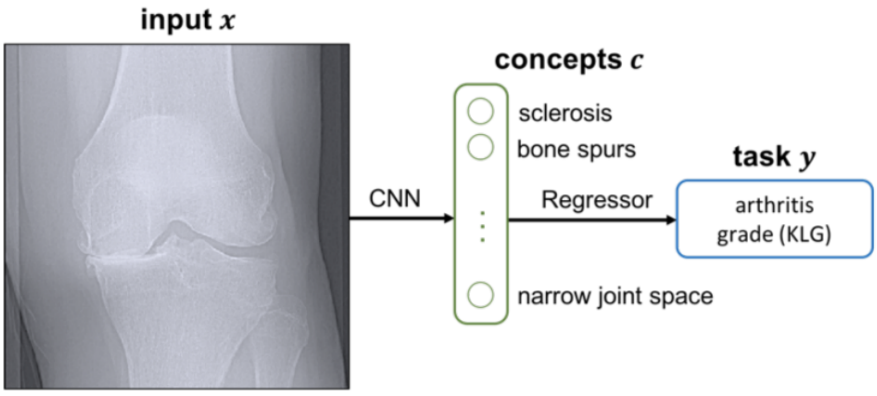

Figure 2.1: Illustration from [7]. The concept bottleneck model first predicts "an intermediate set of human-specified concepts c, then uses c to predict final output y".

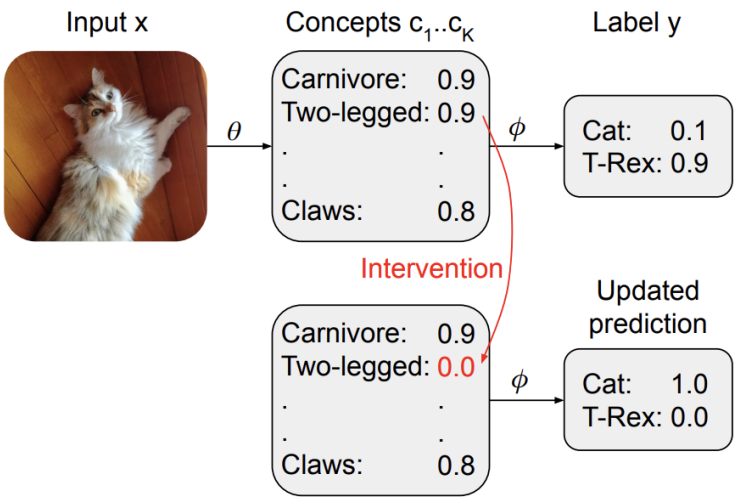

Figure 2.2: Illustration from [9] which explains: "when the model mispredicts a concept, one can intervene by setting the concept probability to 0 or 1 and obtain an updated prediction".

(Figure 2.2). Consider a CBM which is trained on data points (x, c, y). At test time, given an input x, the CBM would predict concepts $\hat{c}$ and use those concepts to predict the target label $\hat{y}$. This offers the opportunity for users to edit $\hat{c}$ and propagate those changes to $\hat{y}$. Going back to the radiologist examining the knee x-ray example, if let's say the radiologist recognises the model is misidentifying bone spurs, they can update the model's prediction by changing $\hat{c}$. Testing this on actual applications allowed the paper to conclude that partially correcting concept mistakes at test time improved the accuracy of their CBM significantly compared to the standard baseline model.

2.3 CBM Training Methods

Concept bottleneck models can be formulated in the following manner [6]:

1. The training data points are $\{(x^{(i)}, y^{(i)}, c^{(i)})\}_{i=1}^{n}$. In this case, the input is $x \in \mathbb{R}^d$, the target label is $y \in \mathbb{R}$, and $c \in \mathbb{R}^k$ is a vector of k concepts.
2. Concept bottleneck models are of form $f(g(x))$, where g maps an input x into the concept space and f maps concepts into a final prediction.
3. At actual test time (execution time) CBMs produce prediction $\hat{y}$ where $\hat{y} = f(g(x))$. The test-time concepts are $\hat{c}$ where $\hat{c} = g(x)$.
4. *Task accuracy* is how accurately $f(g(x))$ predicts y. *Concept accuracy* is how accurately $g(x)$ predicts c.

There are three well-defined training methods for CBMs: independent, sequential, and joint training. In machine learning, the loss function is a useful metric because it evaluates how well the model predicts the actual values. During training, the goal is to adjust the parameters to minimise a model's loss function. Let L_{C_j} be a loss function that measures the difference between the predicted and true j-th concept [7]. Let L_Y be the loss function that measures the difference between the predicted and true target label y values. Finally, let the learning concept bottleneck model $f(g(x))$ be represented using $\hat{f}$ and $\hat{g}$.

Independent

An independent bottleneck CBM learns $\hat{g}$ and $\hat{f}$ independently. This means at training time, $\hat{f}$ is optimised to predict the labels from the true concepts. At test time, $\hat{g}$ is used to generate the concepts from new inputs which are fed into $\hat{f}$.

Concept predictor:

$$\hat{g} = \arg\min_{g} \sum_{i,j} L_{C_j}(g_j(x^{(i)}); c_j^{(i)}) \tag{2.1}$$

Label predictor:

$$\hat{f} = \arg\min_{f} \sum_{i} L_Y(f(c^{(i)}); y^{(i)}) \tag{2.2}$$

Sequential

In a sequential bottleneck CBM, $\hat{g}$ is first trained to predict concepts and then $\hat{f}$ is trained to predict the target from the concepts predicted by $\hat{g}$. In this case $\hat{f}$ is not trained with the true concepts but with the concepts predicted by $\hat{g}$.

Concept predictor (learns it in the same way as independent bottleneck):

$$\hat{g} = \arg\min_{g} \sum_{i,j} L_{C_j}(g_j(x^{(i)}); c_j^{(i)}) \tag{2.3}$$

Label predictor:

$$\hat{f} = \arg\min_{f} \sum_{i} L_Y(f(\hat{g}(x^{(i)})); y^{(i)}) \tag{2.4}$$

Joint

The joint bottleneck CBM combines aspects of both the independent and sequential approaches. It trains $\hat{f}$ and $\hat{g}$ simultaneously and optimises a weighted sum of the losses for predicting concepts and the target label.

Concept and label predictor:

$$\hat{g}, \hat{f} = \arg\min_{f,g} \sum_{i} \Big[L_Y(f(g(x^{(i)})); y^{(i)}) + \sum_{j} \lambda L_{C_j}(g(x^{(i)}); c^{(i)}) \Big] \tag{2.5}$$

2.4 Large Language Models

Large Language Models (LLMs) are artificial intelligence programs capable of understanding and generating human-like text. They achieve this because they are trained on vast amounts of data [10]. Instead of being limited to domain-specific datasets, these models are trained on massive amounts of text from the internet, encompassing a wide variety of topics. At the fundamental level, LLMs are based on a transformer model, which processes data by converting it into tokens and applies mathematical equations to it to identify the relationships between the tokens [11]. This allows models to recognise patterns like humans. LLMs can understand context, reply to human prompts, summarise text, generate code, etc.

Domains	Model Structures	Models	# Params	Pre-train Data Scale
General-domain (Large) Language Models	Encoder-only	BERT [70]	110M/340M	3.3B tokens
		ERNIE [77]	110M	173M sentences
		ALBERT[78]	12M/18M/60M/235M	16GB
		ELECTRA [79]	14M/110M/335M	33B tokens
		RoBERTa [80, 81]	355M	161GB
		DeBERTa[73]	1.5B	160GB
	Decoder-only	XLNet [80]	110M/340M	158GB
		GPT-2[82]	1.5B	40GB
		Vicuna[83]	7B/13B	LLaMA + 70K dialogues
		Alpaca[84]	7B/13B	LLaMA+ 52K IFT
		LLaMA [4]	7B/13B/33B/65B	1.4T tokens
		LLaMA-2 [5]	7B/13B/34B/70B	2T tokens
		Galactica [85]	6.7B/30.0B/120.0B	106B tokens
		GPT-3[6]	6.7B/13B/175B	300B tokens
		InstructGPT [86]	175B	-
		PaLM [3]	8B/62B/540B	780B tokens
		FLAN-PaLM [10]	540B	-
		Bard [87]	-	-
		GPT-4[7]	-	-
	Encoder-Decoder	BART [88]	140M/400M	160GB
		ChatGLM [8, 9]	6.2B	1T tokens
		T5 [76]	11B	1T tokens
		Flan-T5 [76]	3B/11B	780B tokens
		mT5 [76]	1.2B/3.7B/13B	1T tokens
		UL2 [89]	19.5B	1T tokens
		GLM [9]	130B	400B tokens

Figure 2.3: Table from [12]. It categorises some general LLMs based on the three different types of architectures. It also provides a quantitative summary of the number of parameters and the size of the datasets used for model training.

Existing general LLMs have three main types of architectures: encoder-only, decoder-only, encoder-decoder [12]. Figure 2.3 offers a comprehensive list of existing general LLMs.

1. *Encoder-only*: Utilise stacks of transformer encoder layers which analyse input text bidirectionally, meaning they consider context from both before and after the token being processed. These LLMs are suited for getting a deep understanding of text which is useful for tasks like document classification and sentiment analysis. Examples include BERT and DeBERTa.

2. *Decoder-only*: Comprised of transformer decoder layers, these models process text unidirectionally (left-to-right), focusing on generating text by predicting the next token based on the preceding ones. These LLMs are suited for text generation tasks. Examples include the GPT-series developed by OpenAI, the Llama-series developed by Meta, PaLM and Bard (Gemini) developed by Google.

3. *Encoder-decoder*: Feature a combination of bidirectional encoder layers that understand input sequences and unidirectional decoder layers that produce output sequences. These LLMs are suited for both the comprehension and generation of text. Examples include Flan-T5 and ChatGLM.

Many leading companies have released LLMs which are open to use by the public [13]. However, the consensus about these models is that they have strong global performance but are weaker with

specific task-oriented problems [14]. Hence, it is important to fine-tune these LLMs to obtain better performance for specific tasks. There are several different types of fine-tuning methods [15]. They can be split into two broad categories: supervised fine-tuning and RLHF (reinforcement learning from human feedback).

Supervised fine-tuning, like supervised learning, uses inputs and their corresponding labels. Common techniques include:

- *Transfer learning*: The goal of this fine-tuning technique is to transfer knowledge from the source task to the target task. This requires taking the model and fine-tuning it on a smaller, task-specific dataset. Domain-specific fine-tuning is a type of transfer learning approach where a model is fine-tuned on a dataset composed of text from a target domain to adapt its ability to understand domain-specific context.
- *Few-shot learning*: Shots are examples of the required task. In this method of learning, the model is presented with numerous sets of input-output demonstrations to guide the model's predictions by providing examples directly in the prompt.
- *Multi-task learning*: This involves tuning the model on multiple tasks simultaneously. This is useful when the tasks are closely related, because the model can develop a robust understanding of the data by considering the similarities and differences across the tasks.

RLHF, on the other hand, involves training the model by receiving feedback from humans [16]. This allows humans to share their expertise with the model and encourages the model to adapt and evolve from feedback. Common techniques include:

- *Reward modelling*: This involves the model generating different possible outputs or actions, which are then rated or ranked by humans based on quality. The model learns from these ratings and adjusts its behaviour to try to get better ratings in the future.
- *Proximal policy optimisation*: This is a step-by-step method used to improve a language model by making small, careful changes to its behaviour to increase rewards. The main idea is to improve the model without making big, potentially harmful changes, ensuring updates are gradual and beneficial.
- *Parameter efficient fine-tuning*: This fine-tuning method adjusts only part of the model's parameters in a task-specific manner, making it easier and faster to adapt the model to new tasks.

3

Related Work

Contents

3.1 LLMs Trained on Medical Data **13**
3.2 Considerations of Using LLMs for Concept Generation **14**
3.3 Leakage in CBMs **16**
3.4 Relevant MIMIC Projects **18**
3.4.1 MIMIC-IV Overview 18
3.4.2 Predicting Intervention Onset in the ICU 19
3.4.3 CBM for Predicting Vasopressor Onset 19
3.4.4 Google LLMs for Extracting Insights from Clinical Notes 19
3.4.5 Evaluating LLM Tasks for Extracting Insights from Clinical Notes . . . 20
3.4.6 LLMs Trained on MIMIC 20
3.5 Mortality Prediction using MIMIC **21**
3.6 Identifying Acute Respiratory Distress Syndrome **24**

3.1 LLMs Trained on Medical Data

The motivation for this project was to utilise recent advancements in AI technology on medical data to derive new insights and make useful predictions. Clinical LLMs show promise in this area by demonstrating impressive reasoning on certain forms of evaluation. However, in this section, some key limitations of these models are discussed, which prevent them from being deployed at a wide scale in clinical settings. Clinical LLMs are general LLMs which have been fine-tuned for the medical domain. They are trained on a high-quality medical corpus to conduct fine-tuning through transfer learning. The current leading models were obtained from fine-tuning Meta's

Llama models and studies have demonstrated that these models are not only getting better at understanding and generating medical text, but they are improving their ability to make clinical decisions [12]. Furthermore, researchers at Google have combined different types of prompting methods like few-shot learning and prompt turning to create a new technique called "Instruction Prompt Tuning" [17]. This was used to develop MedPaLM-2, an improved version of their previous model, MedPaLM. The researchers reported "a competitive score of 86.5% compared to human experts 87.0% on the US Medical Licensing Examination" (USMLE) [12].

According to [12], although existing clinical LLMs are showing promising results, there are some key issues which are limiting their ability to be widely adopted in healthcare. Most existing medical LLMs report their performance on answering medical questions like USMLE, however these examinations "do not capture the complexity of realistic clinical cases." When the authors of [18] constructed new datasets on more challenging clinical cases, LLMs like MedPaLM-2 and Llama-2 did not perform nearly as well. Furthermore, it is expensive and inefficient to inject new knowledge into an LLM through re-training [12]. However, in the medical domain, it is necessary to update LLMs with things like a new adverse effect of a medication or a novel disease. Additionally, LLMs are black-box, and their lack of interpretability (Section 2.1) makes it challenging to align their behaviours with the ethics and objectives of a medical task.

This reveals a gap in LLMs that can be addressed using concepts. In this project, a CBM will be trained initially, and then an LLM's insights will be augmented on top of it. It is important to recognise that the reverse is also possible, where an LLM is trained with concepts. Instead of having a feed-forward or NLP-based architecture in the LLM's last layer, a concept layer could be used. However, this approach has a limitation: it eliminates the ability to access the intermediate layers of the model. The internal decision-making of an LLM cannot be easily interpreted or intervened upon. Hence, this project leverages the strengths of both models: CBMs enhance interpretability and ease of updating with new medical knowledge, while the LLM's medical text processing capabilities improve performance by incorporating insights from previously unexplored clinical notes.

3.2 Considerations of Using LLMs for Concept Generation

In the seminal CBM paper [7], one of the final points of discussion was the big limitation of concept bottleneck models (Section 2.2): they require annotated concepts at training time (not only do

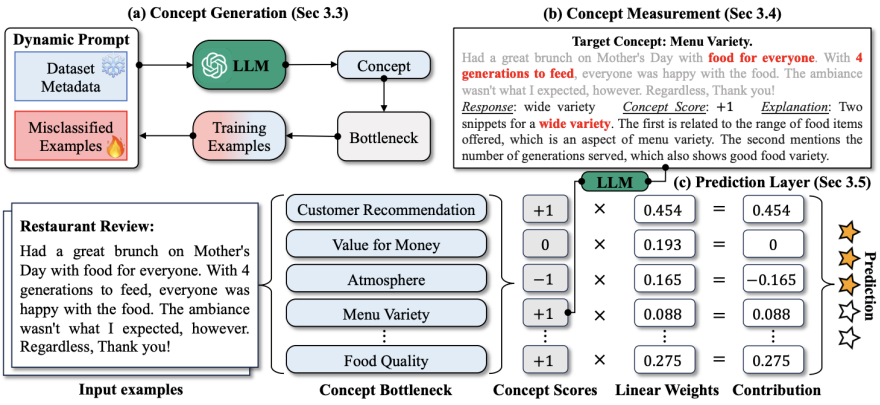

Figure 3.1: Demonstration of the three modules in Textual Bottleneck Models from [19]: Concept Generation, Concept Measurement, and Prediction Layer.

they need the x and y but also c). Traditional approaches of CBMs involve manually annotating the data to try and construct a set of concepts which is informative for the model's task. This requires a significant amount of time to be spent by individuals who are domain experts. This restricts the scalability of CBMs, especially in domains such as medicine which would require licensed physicians taking time out to do annotations.

Some studies have been conducted to try and address this issue. Work has been done to automate the task of generating concepts using LLMs. In [19], the authors proposed a Textual Bottleneck Model (TBM). This extends a CBM and uses concepts to make text classification interpretable with 3 steps: concept generation, concept measurement and final prediction, see Figure 3.1. Firstly, the model generates a set of concepts from the text by prompting an LLM to discover new concepts from the data that help tell the difference between examples that have been classified incorrectly. Each concept it generates contains 3 key pieces of information: the concept name, concept description and a question which has a predefined set of possible responses such "Positive", "Negative", "Uncertain" or "Not applicable". There is also a numerical score associated with each response which means in the TBM's concept measurement stage, an LLM is provided the text data and the concept question. Its response to the question allows the mapping of the concept to a score which assigns the value of the concept to the model. Finally, the concept scores are put together to make a final prediction.

The paper lays the groundwork for using LLMs for concept generation. However, a human evaluation of the concept sets produced in the concept generation step revealed some key limitations. The authors found that the module had almost no problem discovering concepts relevant to the task label, but it occasionally accepted unnecessary concepts that were too similar to the

previously generated ones. Furthermore, they observed that some concepts were directly leaking the task label. For certain datasets, instances of redundancy and leakage were especially high. Redundancy unnecessarily increases the size of the concept space and leaky concepts can undermine the faithfulness of the provided explanations. Since this project aims to generate concepts using an LLM from clinical notes, it is important to consider how the two issues will be tackled. Section 3.3 contains a detailed discussion on how to address leakage. The rest of this section looks at how to ensure concept quality and the work done in Section 3.4.3 provides further ideas.

In [20], the authors developed a medical image classifier which incorporates natural language concepts to improve interpretability. They used GPT-4 to generate medical concepts. Rather than applying GPT-4 to any particular data, they explored the efficacy of the general model by asking a series of questions about radiology which became increasingly specific until eventually they were able to generate a list of concepts. The study found collecting ideal concepts which aligned with experts required "a few rounds of revision of the prompts and interaction with the model." Furthermore, "the generation results of GPT-4 come with randomness and are not perfectly controllable." The quality of the generated concepts heavily relied on the clarity and effectiveness of the questions posed to the model, indicating that some input and guidance from medical experts was still necessary. This reveals the need for a comprehensive evaluation process for this project to establish a robust concept labelling pipeline with an LLM. During implementation it was found that a refined prompt structure was sufficient to achieve an acceptable level of accuracy. However the labelling task was mainly based on standard Named Entity Recognition (discussed in Sections 3.4.4 and 3.4.5) and did not need to apply much reasoning. Due to time and resource constraints, a small-scale validation study was conducted for evaluation, but more formal investigations are suggested as future work in Section 6.4.

3.3 Leakage in CBMs

There are two ways of passing concepts to the label predictor $\hat{f}$ (Section 2.3) in concept bottleneck models: hard concepts (binary numbers 0 or 1) and soft concepts (any number between 0 and 1).

While evaluating concept bottleneck models, it was found that soft CBMs consistently outperform hard CBMs in predictive performance. Furthermore, when the concept predictor $\hat{g}$ and label predictor $\hat{f}$ are trained independently, they do not perform as well as a black-box predictor [9]. The best performing CBMs do not use the independent bottleneck training method. Instead, the

inputs to the label predictor f are concept probabilities produced from the concept predictor.

However, soft CBMs, which are trained sequentially or jointly, suffer from information 'leakage' as discussed in [9]. Hard CBMs, on the other hand, are unaffected by this issue. Leakage is a phenomenon where the concept predictor unintentionally conveys extra information about the target label that should not be available to the label predictor. The paper uses the example of an animal classification task where the target class labels are 'cat' or 'dog'. The model might identify subtle differences in the predicted probabilities for concepts like 'fur' and 'tail' which could allow the model to distinguish between cats and dogs, even though these features alone should not actually be enough for a reliable classification. This leakage makes it difficult to trust the model's predictions and undermines its interpretability because it is unclear whether predictions are based on genuine concepts present in the data or just hidden encodings to guess the class label.

One of the first discussions on leakage in CBMs was in [6]. In the paper, leakage was mainly associated with joint training of CBMs, where the concept and label predictors are trained together, potentially leading the concept predictor to encode additional label information. The authors suggested that techniques like task-blind training or Concept Whitening, which decorrelates concept representations, could mitigate leakage. However, research by [21] suggests that leakage is a more widespread issue that persists even with natural mitigation strategies, particularly in models that use soft concept representations. This indicates a deeper, systemic problem within high-performing current CBMs. Having considered this research, the authors of [9] conducted studies to address the performance gap between hard and soft CBMs. By doing so, they could attain the robustness of hard CBMs against leakage while still achieving the higher accuracy of soft CBMs. They found that the gaps observed can largely be attributed to inadequate concept sets and inflexible concept predictors.

Looking more closely at the first reason, the paper suggests that hard CBMs will under-perform compared to a soft CBM predictor when the Markovian assumption is not met. The Markovian assumption is that the concept set c, which the CBM will be trained on, contains all the information about y present in x. A completeness score can be used to determine when the assumption is fulfilled. In Chapters 4 and 5, the CBMs were trained using independent, sequential, and joint training methods. It was observed that there was variation in the concept predictions across all three training methods, which is attributed to leakage. In Chapter 6, the evaluation focuses on whether augmenting the concept set reduces leakage by comparing the completeness scores of the original and enhanced CBMs.

3.4 Relevant MIMIC Projects

This section provides an overview of the MIMIC-IV database followed by summaries of previous work done on the MIMIC dataset which is related to this project.

3.4.1 MIMIC-IV Overview

In order to access MIMIC-IV, certain credentials had to be obtained (A). According to the MIMIC Docs [22], "MIMIC-IV contains data from 2008-2019. The data was collected from Metavision bedside monitors. MIMIC-III contains data from 2001-2012. The data was collected from Metavision and CareVue bedside monitors." Most of the works discussed in this section use MIMIC-III which has some subtle differences from MIMIC-IV. MIMIC-IV includes some of the patient data from MIMIC-III during the years after the CareVue system stopped being used. Additionally, "a number of improvements have been made [to the MIMIC-IV database], including simplifying the structure, adding new data elements, and improving the usability of previous data elements." MIMIC-IV is split into distinct modules. These include:

1. Hosp - contains the hospital level data for patients including their admission information, lab measurements, microbiology and medication administration.
2. ICU - contains information collected from the system used within the ICU. This includes ventilator settings and intravenous (IV) administrations.
3. ED - contains the information collected for patients who were admitted to the emergency department including triage assessment, vital signs and medicine reconciliation.
4. CXR - contains information about patient chest x-rays.
5. ECG - contains electrocardiogram data for patients. An electrocardiogram is a medical test that records the electrical activity of the heart over a period of time.
6. Note - contains four files, with the main ones being discharge and radiology notes, which detail patient admission reasons, hospital courses, discharge instructions, and radiography imaging reports, supplemented by discharge detail and radiology detail files providing additional information about the primary notes.

The Hops, ICU, ECG and Note modules were used for this project.

3.4.2 Predicting Intervention Onset in the ICU

This study [23] worked on training "unsupervised switching state autoregressive models" to identify patterns in ICU patients' vital signs and predict one of five treatments for the patients based on their data. This model was trained on the MIMIC-III dataset and the study offers an insight into how to effectively select a cohort of patients for training a model. It also highlights the useful data types for predicting treatments:

- Clinical Observations: This is a time series of different clinical measurements taken at various times during the patient's stay in the ICU.
- Clinical Intervention Labels: This array indicates when and which medical interventions were given to the patient.
- Static Observations: These are observations that do not change over time, such as demographic information or initial clinical status.

3.4.3 CBM for Predicting Vasopressor Onset

This paper [8] is a valuable reference for developing a CBM for MIMIC. The authors created a concept bottleneck model which predicted how long, after administration, it would take vasopressors (blood pressure medication) to start having effects on a patient. Their description of cohort selection, feature choices and evaluation approach is very insightful for this project. For data-preprocessing, pre-existing MIMIC-III data pipelines were used. This inspired a search for a similar MIMIC-IV data pipeline for this project, and the following were found [24][25]. These tools used extensively for the dataset construction in Chapters 4, 5, and 6.

3.4.4 Google LLMs for Extracting Insights from Clinical Notes

To train NLP models for understanding and interpreting clinical notes in EHRs, large amounts of labelled data are required. Relying only on human annotators is time consuming and expensive. This paper [26] proposes using LLMs along with human annotators. They use Google's medical LLM Med-PaLM to do "base annotations" and then bring in actual domain experts to refine them.

The authors propose a few-shot learning approach (Section 2.4) and demonstrate promising results from quantitative evaluation. They lacked a comprehensive qualitative study, which this

Task	Description	Example Text	Answer	Data
Clinical sense disambiguation	Given a note and an abbreviation, expand the abbreviation (classification)	*[...] was sent to IR for thrombolysis. Post IR, ultrasound showed that [...]*	Interventional radiology	41 acronyms from 18,164 notes from CASI (Moon et al., 2014) and 8912 notes from MIMIC (Adams et al., 2020)
Biomedical evidence extraction	Given an abstract, list interventions (multi-span identification/generation)	*[...] paliperidone extended- release tablets and [...] with risperidone [...]*	-paliperidone extended-release tablets -risperidone	187 abstracts (token-level) and 20 newly annotated abstracts (arm identification) from EBM-NLP (Nye et al., 2018)
Coreference resolution	Given a note and a pronoun, identify the antecedent (span identification)	*[...] Did develop some tremors, however. These were well managed [...]*	some tremors	105 newly annotated examples from CASI (Moon et al., 2014) with one pronoun-antecedent pair each
Medication status extraction	Given a note, extract medications and their status, e.g. active (NER + classification)	*[...] have recommended Citrucel [...] discontinue the Colace. [...]*	-Citrucel: *active* -Colace: *discontinued*	105 newly annotated examples from CASI (Moon et al., 2014) with 340 medication-status pairs
Medication attribute extraction	Given a note, extract medications and 5 attributes, e.g. dosage, reason (NER + relation extraction)	*[...] she was taking 325 mg of aspirin per day for three years for a TIA. [...]*	aspirin: {dose: 325 mg, freq: per day, duration: three years, reason: TIA}	105 newly annotated examples from CASI (Moon et al., 2014) with 313 medications and 533 attributes

Figure 3.2: Overview of the five tasks studied in [28].

project aims to fulfil. The paper has two major contributions for this project; firstly, it provides an insight into how to refine the LLM's prompt structure and schema. The template described in this study served as a starting point for the template used in Section 6.2. Secondly, the authors used their medical information extraction pipeline on the MIMIC-IV clinical notes (specifically the discharge data), and they have produced annotations for the text within the CSV files (things like REASON, MEDICATION, MODE, DURATION), available on PhysioNet [27]. This is the firstly publicly available set of annotated labels for MIMIC-IV clinical notes and could serve as a useful baseline for comparison in future work.

3.4.5 Evaluating LLM Tasks for Extracting Insights from Clinical Notes

In this paper [28], the authors set a benchmark for how LLMs such as GPT-3 perform at clinical NLP tasks using different medical data including MIMIC. They documented their evaluation on 5 diverse medical information extraction tasks 3.2. The study's biomedical evidence and medication status extraction tasks were used as references for the scoring criteria applied to evaluate open-source LLMs in Section 6.2.

3.4.6 LLMs Trained on MIMIC

T5 stands for "Text-To-Text Transfer Transformer," a type of encoder-decoder large language model architecture (Section 2.4). The authors of [29] trained four different T5 models in their

study. Two of these models were initialised from previous T5 models, while the other two were trained from scratch. Below is a more detailed description from [29]:

1. Clinical-T5-Base: This model was initialised from the T5-Base architecture, which was previously trained on a variety of general text using the masked language modelling (MLM) training scheme. The authors then fine-tuned this model on clinical text from MIMIC-III and MIMIC-IV notes.
2. Clinical-T5-Sci: This model was initialised from SciFive, which uses the T5-Base architecture as its starting point. SciFive further pretrains the model on PubMed abstracts and PubMed Central (these are a large collection of biomedical literature). The authors then fine-tuned this model on clinical text from MIMIC-III and MIMIC-IV notes.
3. Clinical-T5-Scratch: This model uses the same architecture as T5-Base but with randomly initialised weights. The authors constructed a vocabulary for the model based on MIMIC notes and then trained it from scratch using the MLM task with chunks of text from MIMIC.
4. Clinical-T5-Large: This model uses the same architecture as T5-Large but with randomly initialised weights. Similarly, the authors constructed a vocabulary for the model based on MIMIC notes and trained it from scratch using the MLM task with chunks of text from MIMIC.

The models have been made available of PhysioNet and form useful sources of comparison for future work discussed in Section 6.4.

3.5 Mortality Prediction using MIMIC

Mortality (death) prediction is one of the most common machine learning tasks in the medical domain. In [30], the authors conducted a systematic review on articles that used ML on the MIMIC dataset. A majority of studies in their review explored predicting mortality. Some papers predicted in-hospital mortality while others looked at mortality within a certain time frame after discharge. All the papers in the review used demographics, vital signs, and laboratory test variables for training their ML models. Furthermore, "clinical information such as medications, intake/output variables, risk scores, and comorbidities were also utilized."

[30] went on to discuss the two main types of mortality predicting models: traditional ML models and deep learning (DL) models. DL models were more complex and performed better

	Score				
	0	1	2	3	4
Respiratory system					
PaO_2/FiO_2 (mmHg)	≥400	<400	<300	<200 with respiratory support	<100 with respiratory support
Hepatic system					
Bilirubin (mg/dL)	<1.2	1.2–1.9	2.0–5.9	6.0–11.9	>12.0
Cardiovascular system					
	MAP ≥70 mmHg	MAP <70 mmHg	Dopamine <5 or dobutamine (any dose)[a]	Dopamine 5.1–15 or epinephrine ≤0.1 or norepinephrine ≤0.1[a]	Dopamine >15 or epinephrine >0.1 or norepinephrine >0.1[a]
Coagulation					
Platelets $\times 10^3/\mu L$	≥150	<150	<100	<50	<20
Central nervous system					
Glasgow coma scale	15	13–14	10–12	6–9	<6
Renal system					
Creatinine (mg/dL)	<1.2	1.2–1.9	2.0–3.4	3.5–4.9	>5.0
Urine output (mL/d)				<500	<200

Notes: [a]All catecholamine doses represent μg/kg/min. Organ dysfunction is identified as an increase in the SOFA score of ≥2 points. In patients with not known preexisting organ dysfunction, the baseline SOFA score is assumed to be zero. *Intensive Care Med.* The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure. On behalf of the Working Group on Sepsis-Related Problems of the European Society of Intensive Care Medicine. 22(7), 1996, 707–710, Vincent JL, Moreno R, Takala J, et al. With permission of Springer.[17]
Abbreviations: PaO_2, partial pressure of oxygen; FiO_2, fraction of inspired oxygen; MAP, mean arterial pressure.

Figure 3.3: Table for Sequential Organ Failure Assessment Score from [35].

than traditional models because they were trained on large amounts of data and had "many levels of features and hidden layers to predict outcomes." However traditional models were more interpretable because of the black-box nature of DL models, which is a worthy trade-off when it comes to clinical decision making. One of the most cited studies for mortality prediction is [31]. It was one of the first studies to use ensemble machine learning for mortality prediction, which the authors called "Super Learner." Ensemble ML combines multiple individual models to create a single, more accurate and robust predictive model. The paper is an interesting case study because the authors made an effort to make their results easily interpretable by clinicians which boosted the popularity of the paper. Furthermore, efforts have been made to improve the interpretability of DL models, using visualisation techniques [32][33]. These examples demonstrate an appetite for more interpretable mortality predictors.

Hence, a concept bottleneck model would be useful for predicting mortality. In [34], the authors used "high-level concepts" within their deep level framework. Although they did not have an intervenable bottleneck layer, their model has strong parallels with a CBM. The paper used Sequential Organ Failure Assessment (SOFA) scores for their concepts. The SOFA score is used to assess how well a patient's organ is functioning in the ICU. It looks at six different organ systems: respiratory, cardiovascular, hepatic, coagulation, renal and neurological. Each system is scored from 0 to 4, with higher scores indicating greater dysfunction, and the total score reflects overall organ failure severity, Figure 3.3. The study chose the SOFA score because it made it easier to interpret which aspects of the patient's health contributed to their deteriorating condition and probability of mortality.

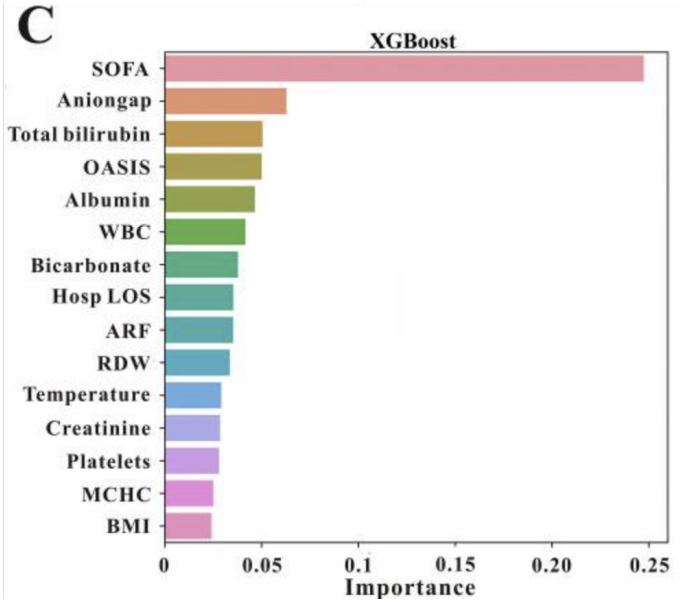

Figure 3.4: Feature importance analysis from [38].

The SOFA score is well placed to be a member of the concept set in a CBM because it needs to be derived from raw clinical variables like platelet count or creatinine level (for coagulation and renal score respectively). These variables would be features (x) and the scores would be concepts (c). The SOFA score is useful for generalising mortality trends for a broad cohort of patients because it quantifies the degree of organ dysfunction, which correlates with patient outcomes, providing a consistent measure across diverse clinical conditions. Hence, SOFA-related features and concepts served as the basis for the dataset constructed in Chapter 4.

However, as discussed in Section 4.5, it does not capture every aspect of the patient's health. The authors explain that SOFA organ system scores "only evaluates a few organs and sometimes markers of damage to one organ actually represent markers of damage to another." Furthermore, "chronic/underlying causes that may impact SOFA scores are not accounted for in many of the studies used for clinical validation." Other raw features and scoring systems need to be considered to capture the entire patient story. In [36], the mortality prediction model used 17 different raw features, including multiple components of the Glasgow coma scale, or GCS (neurological assessment tool used to evaluate a patient's level of consciousness), rather than just the one combined score. [37] and [38] used SOFA but also the Charlson comorbidity index (CCI), oxford acute severity of illness score (OASIS), the acute physiology score III (APSII), and logistic organ dysfunction score (LODS). In [38], the authors also explored which features had the most impact on the model output. A snippet can be seen in Figure 3.4.

Overall, these papers provide a strong basis for constructing a comprehensive dataset for train-

ing a mortality prediction model. Since the objective of the CBM for predicting mortality was to data to capture general trends in the patient cohort and demonstrate the enhancing of interpretability, the concept set was kept simple. Hence, the dataset used the SOFA score, aniongap, albumin, and total bilirubin values for patients (inspired by Figure 3.4).

3.6 Identifying Acute Respiratory Distress Syndrome

Acute respiratory distress syndrome (ARDS) is a serious condition where the lungs cannot get enough oxygen into the blood, and they become stiff and hard to expand [39]. It happens because of damage to the tiny blood vessels and air sacs in the lungs. This damage can lead to high blood pressure in the lungs and acute respiratory failure. Most patients who develop ARDS are usually already ill due to a serious illness or injury. ARDS can be triggered by several conditions including infection like pneumonia or severe flu, sepsis, severe chest injury, near drowning and acute pancreatitis.

ARDS is dangerous and often fatal; it is associated with a mortality rate as high as 43%. According to [40], nearly 25% of patients that are on mechanical ventilation develop ARDS. However, "40% of patients with ARDS are not identified, and 35% do not receive low–tidal volume ventilation despite its proven mortality benefit." ARDS often goes unrecognised because it involves complex symptoms and requires the integration of distinct types of medical data (laboratory measurements, ventilation information, radiology images, clinical notes, and disease characteristics) within a short time. Healthcare providers struggle to process all this data quickly enough, leading to missed diagnoses and missed opportunities for effective treatment [41][42].

Machine Learning tools have been developed to identify ARDS. ML can help catch cases of ARDS that doctors might miss during regular patient care, providing alerts to clinicians. Additionally, ML allows researchers to study large groups of ARDS patients using detailed data from ICU databases. This includes patients who would not typically be included in clinical trials, giving a more comprehensive view of the condition. The ARDS identification part of this project was done in collaboration with Dr Dominic Marshall from the AI4AI lab. The following discussion is based on and inspired by Dr Marshall's ongoing literature review. In [40], the authors performed a systematic review of studies which have worked on detecting ongoing ARDS with ML using data from EHRs. These ML algorithms were called "sniffer systems." The paper found studies that use a combination of vital signs, blood gas data and radiology reports to identify ARDS. In [43], the

authors did an external validation on the algorithms in [40]. They reported that the algorithms demonstrated high sensitivity but poor specificity for the diagnosis of ARDS. This indicated that the sniffer systems are more useful as a screening tool rather than a diagnostic one.

Some of the sniffer systems used a keyword search on radiology notes [44][45] to identify ARDS in patients, looking for words like 'edema' or 'bilateral infiltrates'. However, these keywords often vary between institutions so the approach is not generalisable. Additionally, these methods struggle to accurately identify the diverse types of ARDS and often make mistakes in classification. This creates an opportunity for pre-trained large language models to provide a more generalised approach to interpreting free text. In [46], the authors used an LLM without any "specific list of key words." Their aim was to create an ARDS identification algorithm that was based "solely on the language patterns being used in the radiology notes of patients, without relying on mapping to a specific definition or metathesaurus." The paper used versatile BERT-based model with a Hierarchical Attention Network, designed to work effectively with various datasets. The authors found that previous models often overfit and failed to generalise, but their model achieved superior performance. Tested on two datasets, it achieved a 74.5% F1-score, significantly outperforming other models that scored 46.13%. The model requires minimal training data, avoids overfitting, and can integrate with other data types like x-rays and metadata to improve ARDS identification.

This project aims to similarly address some of the limitations above using an CBM. This approach begins with training a standard CBM on structured Electronic Health Record (EHR) data, leveraging its ability to process and analyse well-defined clinical variables. Building on this, an LLM will be used to analyse free text notes from radiology, discharge, and echocardiography reports. By integrating insights extracted from these unstructured data sources into the original CBM, the enhanced model can better capture the complex and multifaceted nature of ARDS. This hybrid model capitalises on the strengths of both structured and unstructured data processing, improving the detection and classification accuracy of ARDS. Additionally, it addresses the issue of overfitting seen in previous models by being more generalised and adaptable to different datasets with minimal training data. This comprehensive approach not only improves the reliability of ARDS identification but also ensures that the results are interpretable for clinicians.

4

Mortality Prediction with CBM

Contents

4.1 Cohort Selection **28**
4.2 Dataset Construction **29**
4.2.1 Concepts from SOFA Scores 29
4.2.2 Concepts from Anion Gap, Albumin, and Bilirubin 31
4.2.3 Data Pre-processing 31
4.3 CBM Training **33**
4.3.1 Independent Training 33
4.3.2 Sequential Training 34
4.3.3 Joint Training 34
4.4 Investigations **34**
4.4.1 Fine-tuning 34
4.4.2 Comparison Against Logistic Regression Models 40
4.4.3 Analysing Model Weights 41
4.5 Future Work **48**

Following Section 3.5, a concept bottleneck model was implemented to predict mortality. This chapter illustrates an approach to enhancing the interpretability of this well-studied prediction task.

After obtaining the necessary credentials to access the MIMIC database A, credentials for working with cloud services were also secured. The MIMIC-IV dataset is available on Google BigQuery, a distributed relational database management system that can be queried using a free web tool. All datasets for this project were constructed using BigQuery, offering a fast and storage-efficient method for handling the 100GB+ MIMIC dataset. Google Colab was used to run the code for the mortality prediction CBM.

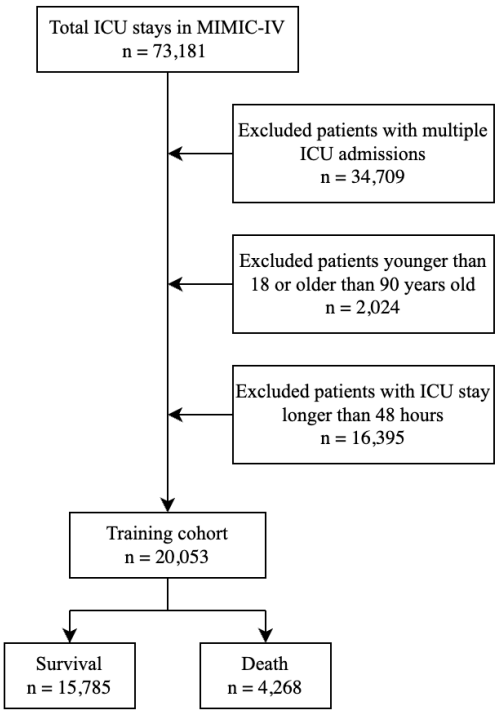

Figure 4.1: Flow chart for the cohort selection criteria.

The MIMIC-IV EHR is divided into two modules: Hosp and ICU. The Hosp module contains hospital-wide EHR data, including administration information, medications, lab measurements, microbiology, and billed diagnoses. The ICU module includes data from clinical systems used in the ICU, such as ventilator settings, procedures, and intravenous administrations. These two modules were the sole sources of data used for the mortality prediction dataset.

4.1 Cohort Selection

In the MIMIC-IV dataset there are 73,181 records of ICU stays. From this, a collection of 20,053 ICU stays were chosen based on the criteria that (i) the patient had only once ICU admission, (ii) the patient was between ages 18 and 90, and (iii) the patient was only in the ICU for a maximum of 48 hours, Figure 4.1.

Patients with only one ICU admission were chosen to simplify the analysis. Multiple ICU admissions can complicate the dataset with repeated measurements, potentially introducing biases or confounding factors. Furthermore, the clinical course of patients with multiple ICU admissions can differ significantly from those with a single admission, possibly indicating different underlying health conditions or treatment paths. Restricting to one ICU admission helps in isolating the

factors contributing to mortality within a single ICU stay.

The age range was restricted between 19 and 90 years old. This is because the medical condition and responses to treatment can vary significantly between children, young adults, and the elderly. By excluding those under 18, the analysis can focus on adult patients, leading to more homogeneous data. Moreover, in MIMIC, the ages of patients older that 90 are often shifted to protect privacy, which can cause inaccuracies. Hence those patients were also excluded.

The time frame of the patient's stay in the ICU was restricted to maintain the simplicity of the dataset. This decision was made to reduce the complexity and volume of data processing required, as many vital signs and lab measurements are recorded throughout a patient's ICU stay. Also, it eliminates the variability that might arise from different lengths of ICU stays, where longer stays might lead to more complicated medical histories and varying data points. Furthermore, the SOFA score was identified as a primary predictor of mortality and only the first day lab measurements and corresponding SOFA scores were considered. Focusing on the initial SOFA score is critical as it provides an early indication of the patient's condition upon ICU admission. Restricting the time frame to 48 hours was a strategic decision to minimise the likelihood of drastic changes in the SOFA score, thereby ensuring that the first day values provide a stronger representation of the patient's condition in the ICU.

4.2 Dataset Construction

A concept bottleneck model requires three sets of values: features, concepts, and labels. In this study, the label was mortality, defined as a binary value indicating whether the patient died within the year. Based on the literature on mortality prediction reviewed in Section 3.5, the SOFA score, Anion Gap, Albumin, and Total Bilirubin were used to guide the feature and concept choices. Ultimately, 15 features and 23 concepts were identified for the models, appendix B.2.1.

4.2.1 Concepts from SOFA Scores

As described in Figure 3.3, the total SOFA score has components for 6 different organ systems. The individual organ score ranges from 0 to 4 and indicates the severity of organ dysfunction. For this study, the features chosen were the specific measurements and data used to calculate the individual organ scores, Table 4.1. Concepts were then created based on these features. The SOFA

score was categorised into two types of concepts for each organ component: *moderate organ failure* (scores 1-3) and *severe organ failure* (score 4). This categorisation provides clinicians with a quick reference to understand the model's predictions.

Concepts are intended to offer useful insights to users, working best when information is combined to yield new perspectives. With this in mind, some more concepts were created. Following medical definitions, these concepts were meant to indicate the existence of possible conditions. The criteria for these conditions was loosely defined based solely on SOFA scores to enhance the insights provided by the model, Table 4.2. Here is a short summary of each of the conditions as explained by the Cleveland Clinic [47]:

- Septic Shock - This is the most severe and dangerous stage of sepsis. Sepsis itself is a life-threatening condition where the immune system overreacts to an infection, causing widespread inflammation. This condition can escalate to severe sepsis, where the inflammation leads to organ malfunction due to low blood pressure. Septic shock is characterised by critically low blood pressure that persists despite the administration of large amounts of intravenous (IV) fluids, making it the most critical phase requiring urgent medical intervention.

- ARDS - Explained in Section 3.6.

- Hepatorenal Syndrome - HRS is a severe complication of advanced liver disease that impacts kidney function, causing sudden prerenal acute kidney failure. This condition arises without prior kidney disease or structural changes to the kidneys. Despite being physically healthy, the kidneys lose their ability to function due to a compromised blood supply.

- Coagulopathy-Related Organ Dysfunction - This condition involves impaired blood clotting, which can lead to excessive bleeding or thrombosis. It is often associated with liver disease, sepsis, or disseminated intravascular coagulation (DIC), and can result in dysfunction or failure of multiple organs due to poor blood flow or bleeding.

- Multiple Organ Dysfunction Syndrome - MODS is a severe, potentially fatal condition characterised by the progressive failure of two or more organ systems. It often results from severe infections, trauma, or other critical illnesses, and can lead to a cascade of deteriorating organ functions.

- CNS and Renal Failure - This condition involves the simultaneous failure of the central nervous system (CNS) and the kidneys. CNS failure can manifest as altered mental status,

SOFA Organ Component	Lab Measurements
Respiratory System	PaO2/FiO2 ratio for ventilated and non-ventilated patients
Liver (Hepatic System)	Total bilirubin
Cardiovascular	Mean blood pressure and Dopamine, Dobutamine, Epinephrine and Norepinephrine rate
Coagulation	Platelet count
Central Nervous System	Glasgow Coma Scale score
Renal System	Urine output and creatinine level

Table 4.1: Lab measurements used for each of the six SOFA organ components.

coma, or seizures, while renal failure results in the kidneys' inability to filter waste, maintain electrolyte balance, and regulate fluid levels.

- Liver and Coagulation Failure - This condition involves simultaneous liver failure and impaired blood clotting. Liver failure affects the organ's ability to process toxins, produce essential proteins, and regulate metabolism. Coagulation failure leads to increased risk of bleeding and clotting abnormalities, further complicating the clinical picture.

4.2.2 Concepts from Anion Gap, Albumin, and Bilirubin

Lastly, features and concepts were derived from the anion gap, albumin, and total bilirubin measurements. The features were the raw lab values, while the concepts indicated whether these values were abnormal, signaling a poor patient condition. Upon reviewing medical literature from [47], it was learned that a high anion gap is clinically relevant because it suggests that the patient has acidosis (higher-than-normal level of acid in the blood) which can lead to more serious conditions, whereas a low anion gap is uncommon. Similarly, a high bilirubin level could indicate issues with the liver, but low levels are not generally a cause for concern. Albumin is used to assess liver function and both high and low levels are useful indicators for liver and kidney failure. From these insights, the following concepts were chosen, Table 4.3.

4.2.3 Data Pre-processing

All missing values for features and concepts were replaced with 0. The dataset was then split in a two step process. It was first divided into a temporary set (80% of data) and test set (20% of data). The temporary set was then further split into a training (75% of temporary set) set and

Potential Condition	**Description**
Organ failure moderate	This is for each of the 6 individual organ scores. The flag is 1 when the corresponding *Organ* score is between 1-3
Organ failure severe	Also for each of the 6 individual organ scores. The flag is 1 when the corresponding *Organ* score is 4
Septic Shock	Flagged when cardiovascular score $\geq$ 3 and any other organ score $\geq$ 2
ARDS	Flagged when respiratory score $\geq$ 3 and cardiovascular score $\geq$ 1 and CNS score $\geq$ 2
Hepatorenal Syndrome	Flagged when liver score $\geq$ 2 and renal score $\geq$ 2
Coagulopathy-Related Organ Dysfunction	Flagged when coagulation score $\geq$ 3 and liver score $\geq$ 2
Multiple Organ Dysfunction Syndrome	Flagged when failure scores $\geq$ 2 in three or more organ systems
CNS and Renal Failure	Flagged when CNS score $\geq$ 3 and renal score $\geq$ 2
Liver and Coagulation Failure	Flagged when liver score $\geq$ 3 and coagulation score $\geq$ 3

Table 4.2: Concepts derived from the SOFA scores.

Concept	**Criteria**
High Anion Gap	Anion gap $>$ 12 mEq/L
Low Albumin	Albumin $<$ 3.4 g/dL
High Albumin	Albumin $>$ 5.4 g/dL
High Bilirubin	Bilirubin $>$ 1.2 mg/dL

Table 4.3: Other concepts derived from lab measurements.

validation set (25% of temporary set). The concept bottleneck models were trained on the training set. During training, the validation set was used to monitor the model's performance and evaluate loss on unseen data. The test set was held out and used only for the final evaluation to assess the model's generalisation ability.

4.3 CBM Training

Six different mortality-predicting CBMs were implemented. For each of the three training methods—Independent, Sequential, and Joint (Section 2.3)—a CBM was trained for both classification and regression.

4.3.1 Independent Training

The same neural network was used for the concept predictor (Equation 2.1) and label predictor (Equation 2.2) for both of the independent CBMs. To start with, the neural network contained three hidden layers.

For the classification CBM, the neural network was designed to handle multiple binary labels. It used the ReLU activation function, $\text{ReLU}(x) = \max(0, x)$, in the hidden layers of the model to introduce non-linearity. This would allow the model to learn complex patterns. Then, a sigmoid-activation function, $\sigma(x) = \frac{1}{1+e^{-x}}$, was used in the output layer to convert the output into probabilities for binary classification. Binary Cross-Entropy (BCE) loss, was used to measure the difference between the predicted probabilities and actual binary labels, defined as: $\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$.

For the regression CBM, ReLU was used again for the hidden layers. The output layer did not require an activation function, it had a linear output layer to produce raw values suitable for regression. Mean Squared Error (MSE) Loss, $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$, was used to measure the difference between the predicted and actual continuous values.

For both CBMs, the Adam optimiser was used. Algorithm 1 describes how both independent CBMs were trained. After initialising the concept predictor and label predictor, the two were trained separately but in the same manner (forward pass, loss calculation, backward pass and parameters update).

4.3.2 Sequential Training

This training method was similar to independent training. The neural network used for both the concept predictor (Equation 2.3) and the label predictor (Equation 2.4) was identical to that used in the independent CBM. Consequently, the optimiser, activation and loss functions were also the same. The key difference is that, instead of training the concept and label predictors independently, the output of the concept predictor $\hat{c}$ was fed into the label predictor during training, Algorithm 2.

4.3.3 Joint Training

In independent and sequential training, the same neural network class was instantiated twice: one for concept predictor $\hat{f}$ and one for label predictor $\hat{g}$. For joint training, on the other hand, a single neural network was used to implement both predictors. This neural network contained six distinct layers for predicting concepts and labels. The first three layers predicted concepts and the subsequent three layers predicted the labels using the output from the concept prediction layers. The same optimiser, activation and loss functions were used in this training method, tailored to the task at hand (classification or regression).

The rationale behind using a single neural network in joint training is that both predictors are trained simultaneously using a combined loss function $\mathcal{L}_{\text{total}} = \mathcal{L}_C + \lambda \mathcal{L}_Y$. This combined loss ensures that both the concept predictions and label predictions are optimised together, improving the overall model performance. The training method for joint training is detailed in Algorithm 3.

4.4 Investigations

4.4.1 Fine-tuning

The training loop for each CBM consists of two nested loops: one for epochs and one for batches within the training data. For each batch, the loss was calculated and used to update the optimiser. To monitor for overfitting, batch losses were accumulated and averaged to yield an overall loss per epoch, which was then plotted. A similar validation loop was also executed without backpropagation, performing a forward pass through validation batches to compute loss on unseen data. These validation losses were also accumulated and averaged per epoch. The CBMs were trained over 60

epochs, and the loss plots for the classifier CBMs can be observed in Figure 4.2. The joint CBM had both concept predictor and label predictor losses similar to the independent and sequential CBMs, but it also included a combined loss (referred to as "Full Model Loss" in the figure) used for training. Using the plots, it was apparent that the initial code showed overfitting because there were fluctuating validation losses. To address this overfitting, modifications were made to the model architecture and training process.

Changes to the model architecture: firstly, the layer sizes and number of layers in the neural network was reduced. The old model had layers with 64 and 128 units. These were reduced to 32 and 64 units. Furthermore, the old model had three layers for each of the concept and label predictors, which was reduced to two layers each. This meant the model had fewer parameters and therefore less capacity to memorise the training data, helping the model learn more generalisable patterns instead of noise in the training set. Figures B.1 and B.2 help visualise the change in architecture. Additionally, a learning rate scheduler was implemented which would help reduce the learning rate after a certain number of epochs, helping the model achieve better convergence.

Changes to the training process: two distinct training modifications were investigated, early stopping and L2 regularisation.

#1 Early Stopping

Early stopping is a technique used to prevent overfitting by halting the training process once the model's performance on a validation set stops improving. To implement this, a patience value was set to 10, meaning the training would be terminated if the validation loss did not improve for 10 consecutive epochs. The loss versus epoch plots can be observed in Figure 4.3. Early stopping resulted in more stable validation losses across all CBMs, reducing the fluctuations that indicated overfitting in the initial runs. Previously, the CBMs had been trained for 60 epochs, but with early stopping, the new CBMs would now stop between epochs 14 and 18. This early termination prevents unnecessary epochs and saves computational resources.

#2 L2 Regularisation

L2 regularisation, also known as weight decay, is another technique used to reduce overfitting. It works by adding a penalty term to the loss function proportional to the square of the magnitude of the model weights. This penalty discourages the model from assigning too much importance to any single feature, promoting smaller and more evenly distributed weights. By regularising the weights, the model generalises better to new data, as evidenced by more stable validation losses in

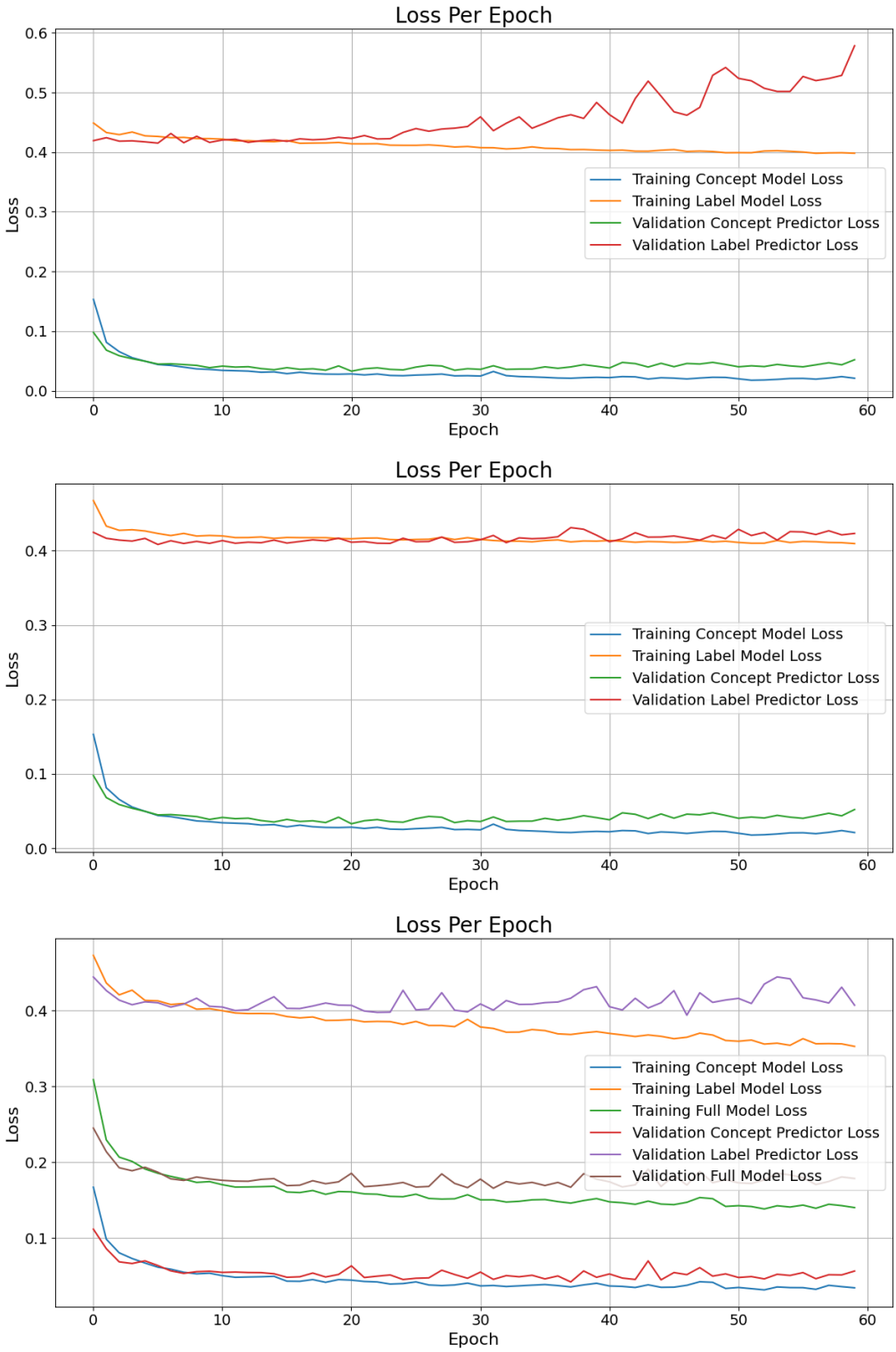

Figure 4.2: Loss plots for independent (top), sequential (middle), and joint (bottom) CBMs with no fine-tuning.

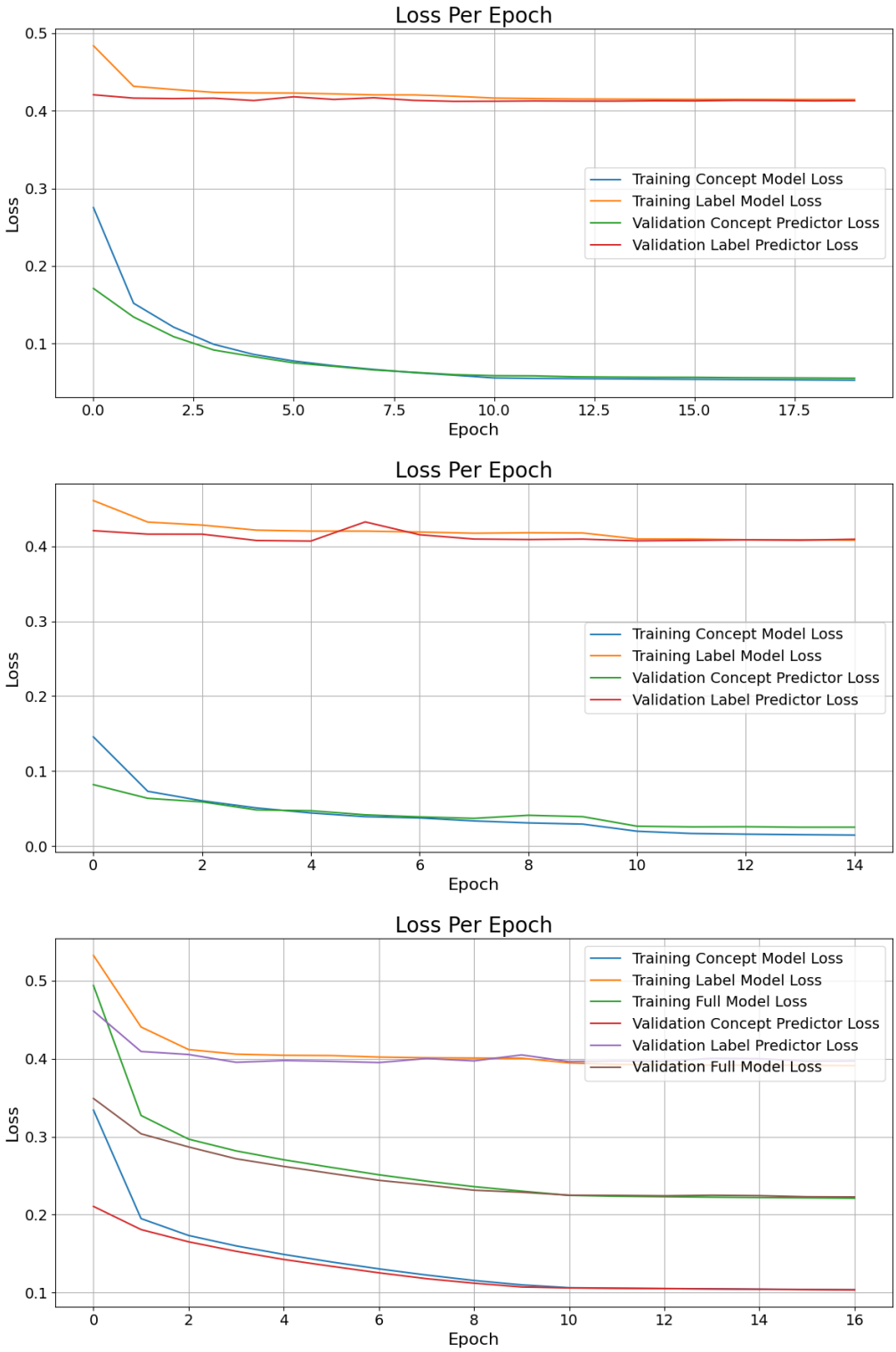

Figure 4.3: Loss plots for independent (top), sequential (middle), and joint (bottom) CBMs with updated architecture and early stopping.

Figure 4.4.

Comparing evaluation metrics: Finally, the original CBMs and fine-tuned CBMs were compared. It was found that the classification CBMs vastly outperformed the regression ones, so only the classification models were used for the remainder of the project. Classification metrics include accuracy, AUC (area under the ROC curve), precision, recall, and F1-score. For the concept predictor, these metrics were calculated for each prediction and then averaged. For the label predictor, the metrics were calculated directly without averaging. The results of these comparisons are presented in Tables 4.4 and 4.5.

Starting with the concept predictor in Table 4.4, the accuracy across the models was consistently high (0.96-0.99), indicating that the model is generally proficient at predicting mortality concepts. The other metrics were more varied. The AUC had a wider range of values (0.75-0.90), with the sequential models tending to perform the best at distinguishing between classes effectively. Precision was also generally high (mostly 0.8+), indicating that when the models predicted a positive concept, it was usually correct. However, the recall values were lower, indicating that the model was only moderately good at identifying all positive concepts and did miss some. The lower recall also brought down the F1 score. In terms of fine-tuning, the sequential models with early stopping and L2 regularisation showed the most significant improvement in metrics for the concept predictor.

Concept Predictor's Model	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1 Score**
Independent	0.99	0.87	0.83	0.74	0.76
Sequential	0.99	0.87	0.83	0.74	0.87
Joint	0.98	0.86	0.72	0.73	0.72
Independent w/ Early Stopping	0.98	0.81	0.78	0.63	0.68
Sequential w/ Early Stopping	0.99	0.90	0.85	0.80	0.82
Joint w/ Early Stopping	0.96	0.75	0.74	0.55	0.60
Independent w/ L2 Regularisation	0.99	0.87	0.85	0.74	0.78
Sequential w/ L2 Regularisation	0.99	0.87	0.85	0.74	0.78
Joint w/ L2 Regularisation	0.98	0.84	0.83	0.69	0.73

Table 4.4: Averaged evaluation metrics for Mortality CBM concept predictors.

Next, the label predictors were considered, Table 4.5. These had lower accuracy (0.82-0.83) and AUC values (0.62-0.66), reflecting the more challenging task of predicting final labels compared to intermediate concepts. Precision, recall, and F1 showed similar trends to the concept predictor

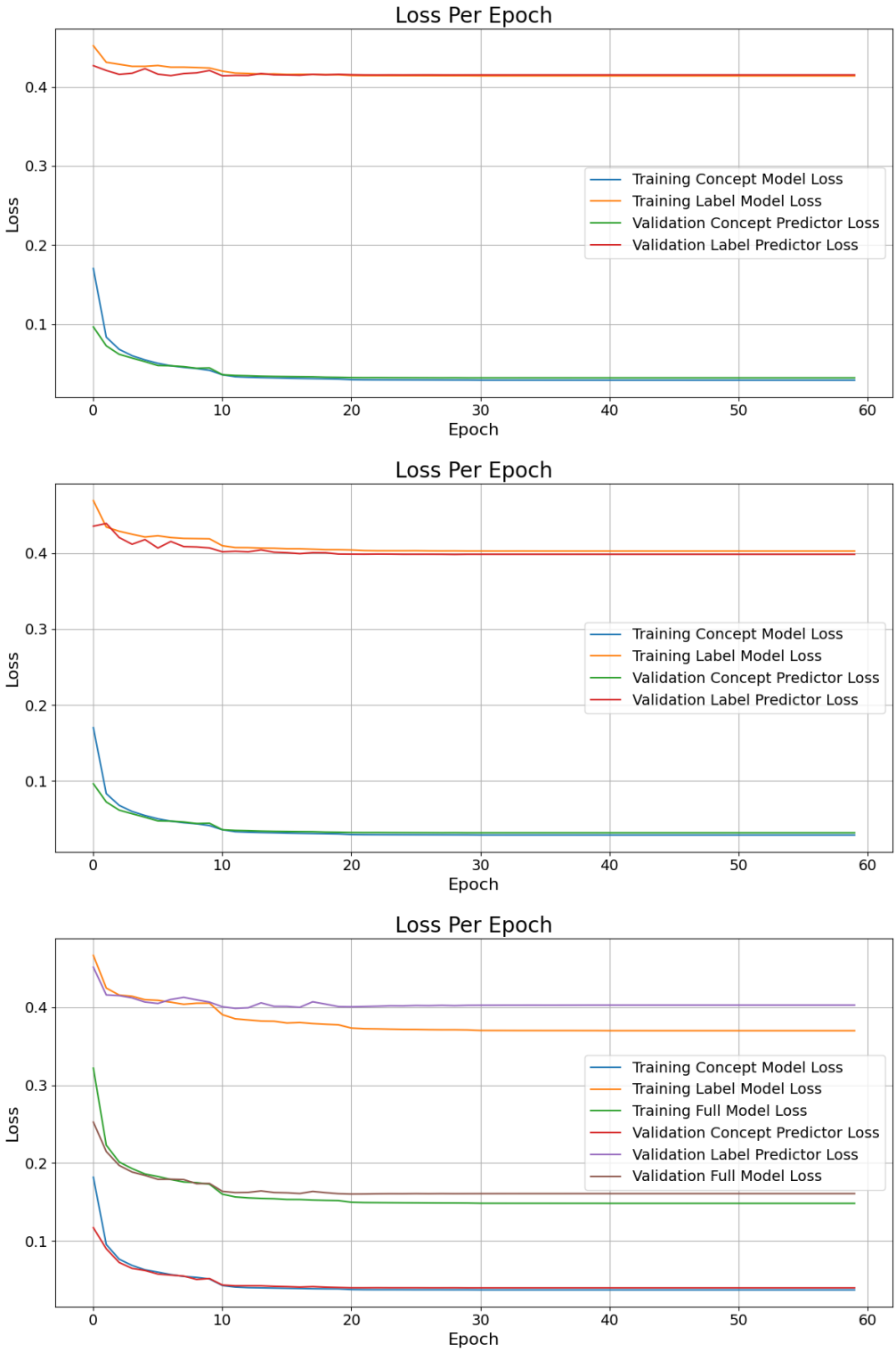

Figure 4.4: Loss plots for independent (top), sequential (middle), and joint (bottom) CBMs with updated architecture and L2 regularisation.

but were lower in most cases. Furthermore, fine-tuning each type of CBM showed very modest performance gains for the label predictor.

Across the fine-tuned CBMs, the final predictor loss was around 0.04, while label loss converged to around 0.4. The overall trends of the evaluation metrics and the tenfold higher label loss compared to concept loss suggest that the models are quite effective in learning and predicting intermediate concepts but struggle more with final label prediction. Overall, the concept predictors perform exceptionally well with high accuracy, AUC, and F1 scores, while the label predictors show weaker performance with room for improvement, particularly in recall. However, the accuracy of the label predictors was still quite strong, especially compared to the accuracy of the ARDS CBMs (see Chapters 5 and 6).

Label Predictor's Model	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1 Score**
Independent	0.82	0.62	0.71	0.27	0.40
Sequential	0.82	0.63	0.75	0.28	0.40
Joint	0.82	0.64	0.70	0.33	0.45
Independent w/ Early Stopping	0.82	0.63	0.77	0.28	0.41
Sequential w/ Early Stopping	0.82	0.63	0.74	0.29	0.42
Joint w/ Early Stopping	0.83	0.65	0.76	0.32	0.45
Independent w/ L2 Regularisation	0.82	0.63	0.79	0.27	0.41
Sequential w/ L2 Regularisation	0.83	0.64	0.78	0.30	0.44
Joint w/ L2 Regularisation	0.83	0.66	0.70	0.37	0.48

Table 4.5: Evaluation metrics for Mortality CBM label predictors.

4.4.2 Comparison Against Logistic Regression Models

To better understand the CBM results, it was useful to compare them with a baseline model, for which logistic regression was chosen. Logistic regression is a statistical model used for binary classification tasks that takes input features x and directly predicts the binary label y. Unlike a concept bottleneck model, which involves two stages, logistic regression provides a direct mapping from input to output. This simplicity makes it less computationally intensive since it does not need to predict intermediate concepts. Two logistic regression models were implemented using the scikit-learn library in Python, and their evaluation metrics were recorded for comparison, as shown in Table 4.6. The X to Y model (referred to as the feature regressor) uses the CBM features as inputs, while the C to Y model (concept regressor) uses the concepts as inputs to predict the

mortality label.

Model	Accuracy	AUC	Precision	Recall	F1 Score
X to Y	0.827	0.73	0.74	0.22	0.34
C to Y	0.832	0.74	0.72	0.28	0.40

Table 4.6: Evaluation metrics for feature regressor (X to Y) and concept regressor (C to Y). Accuracy found to 3 decimal places to make the values more distinguishable.

Interestingly, the features exhibited performance comparable to the concepts themselves. While the concept regressor had higher recall and slightly higher AUC and accuracy, the feature regressor demonstrated slightly higher precision. Even though some of the concepts were meant to group features together to provide additional insights, the overall accuracy did not improve significantly. Both regressor models displayed similar performance to the fine-tuned CBM label predictions, Table 4.5.

This inspired an analysis into how the CBMs makes their final prediction, in an effort to better understand the differences between the models. There are two methods to understand the reasoning behind a concept bottleneck model. One method involves extracting the weights that connect features to concepts and concepts to labels at the neural network level. This investigation was conducted below. The other method takes advantage of the interpretability of CBMs through intermediate concepts, allowing for exploration by feeding in new test data and interpreting the resulting concept predictions and label predictions. This experiment was left as an exercise for future work.

4.4.3 Analysing Model Weights

To start with, the model weights were explored. Two functions were implemented to extract the feature to concept weights and concept to label weights. These were plotted in 3 different heat maps. Heat maps are a data visualisation technique that uses colour to represent the values of a matrix: the more extreme the value is, the higher the intensity of colour. For this study, the 'viridis' heat map scheme was used with the Matplotlib library in Python. In this scheme:

- Yellow: Indicates a high positive contribution, suggesting that the input strongly influences the output in a positive direction.
- Dark Purple: Indicates a high negative contribution, suggesting that the input strongly influences the output in a negative direction.

- Intermediate Colours: Shades of green and lighter purple indicate moderate contributions, with the exact shade corresponding to the value on the colour scale provided on the right side of the heat map.

The following three heat maps were implemented:

1. Feature to Concept Weights Contribution: Shows how each feature influences each concept.
2. Concept to Label Weights Contribution: Shows how each concept influences the final labels.
3. Combined Feature to Concept Contributions Predicting Mortality: Shows the overall influence of each feature on the final label, considering the intermediate concepts.

The weights analysis was implemented on the L2 regularised models rather than the early stopping ones. L2 regularisation is generally better suited for weights analysis in a concept bottleneck model because it directly penalises large weights, leading to a more stable and interpretable model. By ensuring that weights remain small and evenly distributed, L2 regularisation facilitates a clearer understanding of feature contributions. It also reduces overfitting by shrinking weights, providing more reliable insights into the importance of different features. In contrast, while early stopping is useful for preventing overfitting, it does not offer the same level of control over the weight magnitudes, potentially leading to less stable and harder-to-interpret weights. The three heat maps for each of the models can be observed in Figures 4.5, 4.6, and 4.7.

Feature to Concept Weights Contribution

The heat maps for the independent and sequential models were nearly identical, henceforth referred to as the I/S heat maps. Due to the large number of feature-to-concept weight pairs, Tables 4.7 and 4.8 were created to extract useful insights. Table 4.7 highlights the pairs with the strongest weight contributions. There were 72 pairs of interest: 37 were shared between the Joint and I/S heat maps, while 35 were unique (22 in the Joint heat map only and 13 in the I/S heat map only). This indicates that a significant number of relationships identified by the models differ between the Joint and I/S models.

Interestingly, there were more pairs with strong negative contributions (55) than positive contributions (17), suggesting that many features suppress certain concepts in the model. This imbalance implies that the model learns more from features that decrease the likelihood of certain concepts. This is unexpected, as 12 of the 23 concepts were basic SOFA organ component scores directly

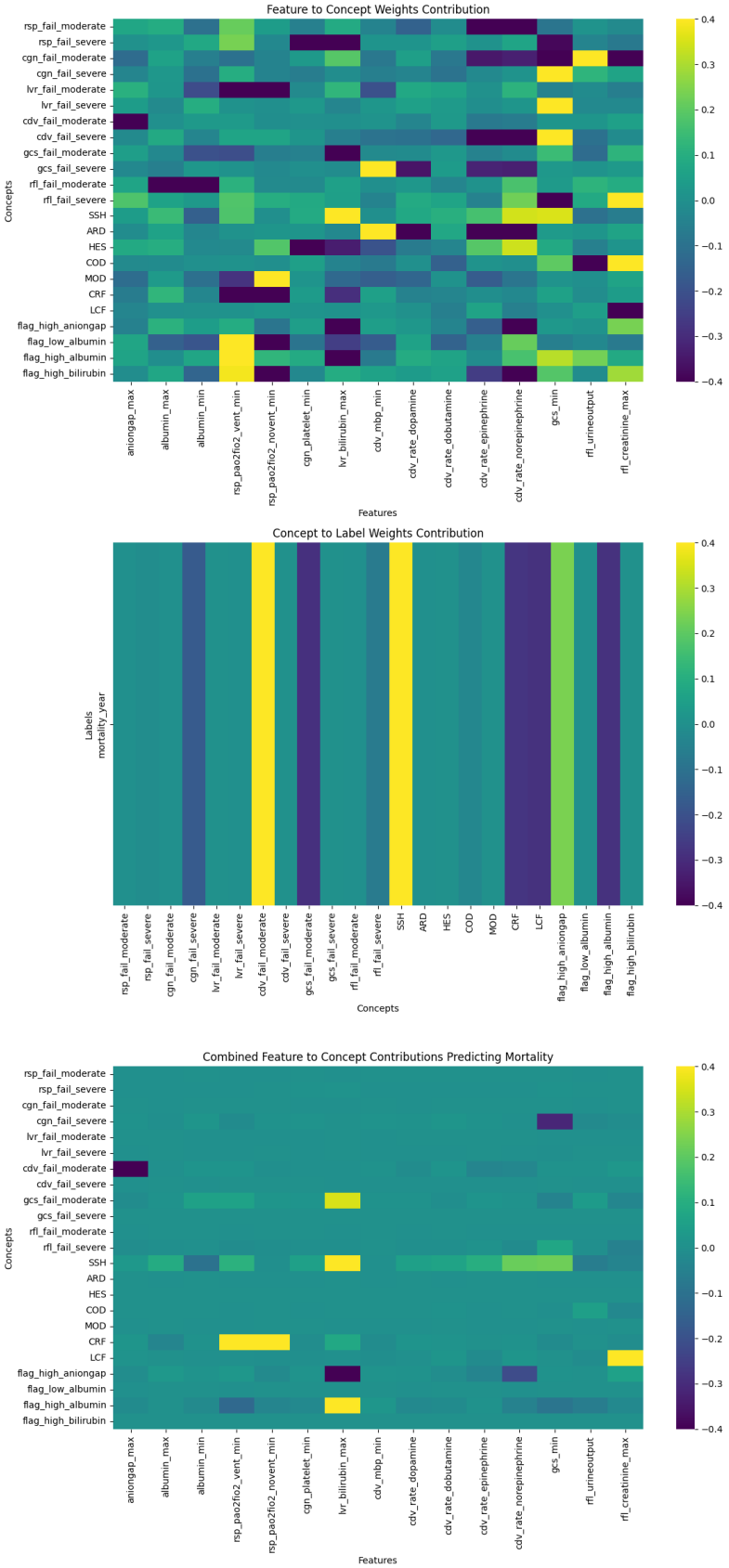

Figure 4.5: Heat maps for feature and concept weights of the L2 regularised, independent concept bottleneck model.

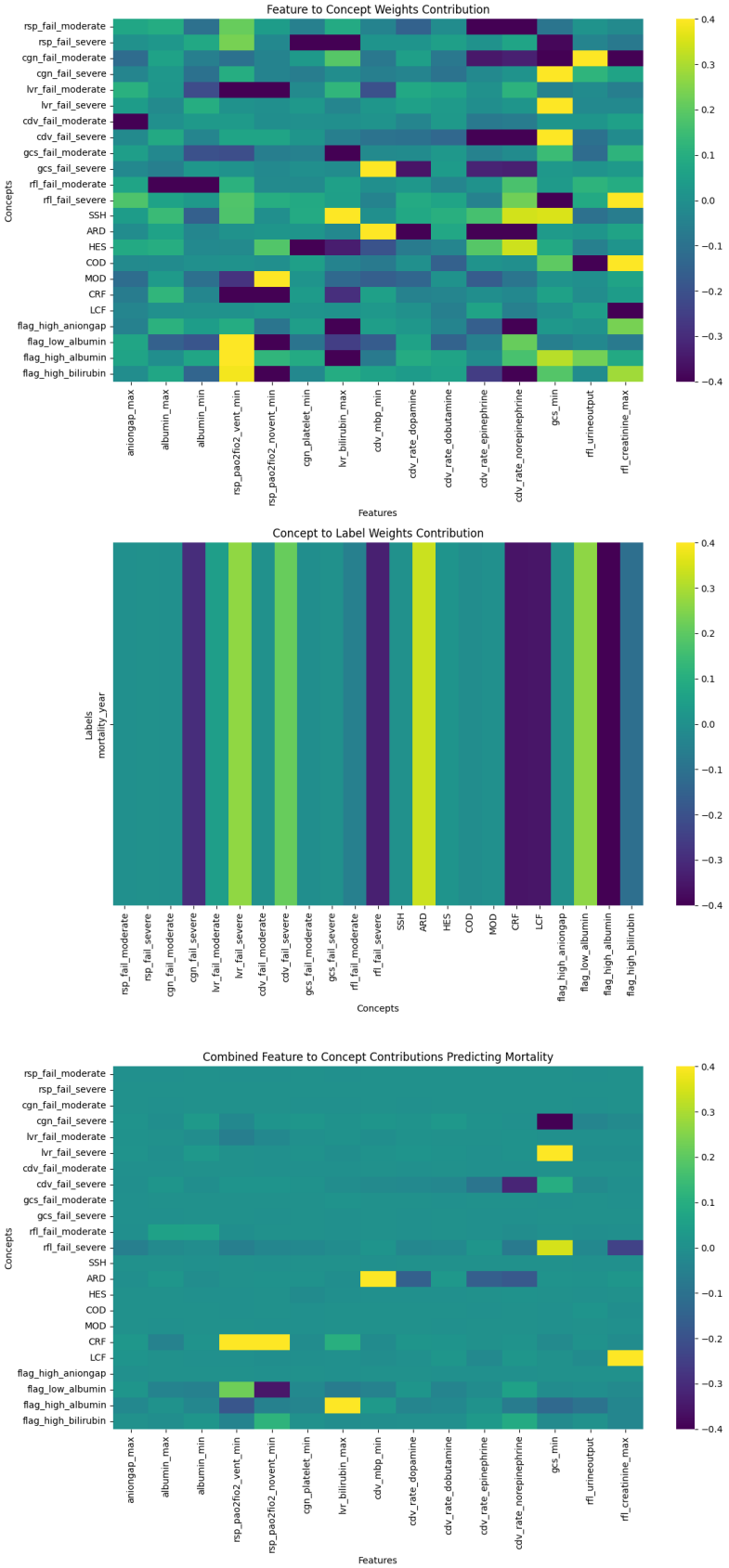

Figure 4.6: Heat maps for feature and concept weights of the L2 regularised, sequential concept bottleneck model.

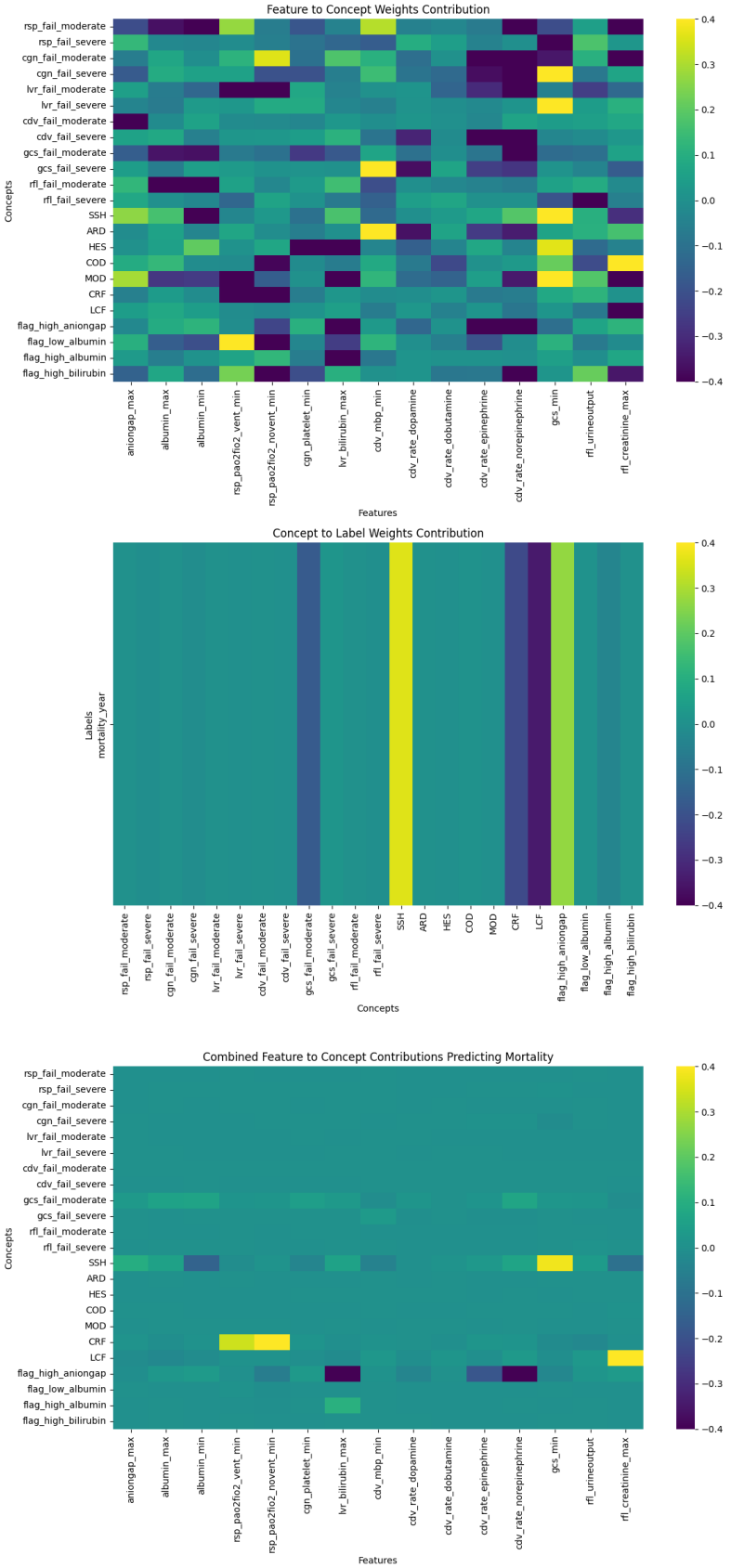

Figure 4.7: Heat maps for feature and concept weights of the L2 regularised, joint concept bottleneck model.

derived from lab measurements. For instance, coagulation failure (moderate/severe) should be directly derived from the platelet count feature. However, the model did not show clinically relevant pairings for any directly derived concepts, prompting an investigation into the clinical relevance of the concept pairings in Table 4.8.

Table 4.8 reveals that only 38 out of 72 total pairs were clinically relevant, most with negative weight contributions. Features like mean blood pressure, GCS score, and ventilated PaO2/FiO2 consistently produced useful pairings with concepts, while other features were inconsistent. These results indicate that although the CBMs are good at predicting concepts, the way they derive these predictions does not align with established clinical knowledge or expectations.

Contribution Heatmap	**Positive Contribution**	**Negative Contribution**	**Total**
Across All 3	7	30	37
Independent and Sequential	7	6	13
Joint	3	19	22
Total	**17**	**55**	**72**

Table 4.7: Table containing the number of feature to concept pairs in the heat maps which had strong contributions to the concept predictor weights. It identifies which pairs are shared, and which are unique to the models. Independent and sequential were virtually identical, hence their row was combined.

Feature to Concept Weights	
Relevant	**Irrelevant**
38	34

Table 4.8: Table that identifies the number of clinically relevant feature to concept pairs that had strong contributions to the concept predictor weights.

Concept to Label Weights Contribution

The distinctions between the 3 models were even more apparent in the concept to label weight contribution plots. None of the models' heat maps were identical; in fact, they were vastly different, as shown in Table 4.9. The independent and sequential models had more extreme weights, while the joint model was more neutral across most concept to label weight pairs. Further investigation is required to understand the reasons behind the differences between the models, which is left for future work. One explanation could be leakage (Section 3.3). Leakage in concept bottleneck models occurs when additional label information is unintentionally encoded in the concept predictions. This can improve label accuracy but reduces interpretability. Sequential and joint models suffer from leakage. These models showed many negative weight contributions, including for concepts that indicate severe health conditions that should have been predicting mortality to be high rather

than low. This suggests that their concepts inadvertently capture direct label information or artifacts, leading to contributions that do not align with clinical intuition.

Contribution Heatmap	**Positive Contribution**	**Negative Contribution**
Across All 3	-	CNS and Renal Failure, Liver and Coagulation Failure
Independent and Sequential	-	High albumin
Independent and Joint	Septic shock	-
Sequential and Joint	-	-
Independent	Moderate cardiovascular failure	Moderate GCS (neurological) failure
Sequential	ARDS	Severe renal failure, Severe coagulation failure
Joint	High anion gap	-

Table 4.9: Table containing the concepts that had strong weight contributions to the final label prediction.

Combined Feature to Concept Contributions Predicting Mortality

There was more consistency in the combined heat map plots. The notable concept and feature combinations have been identified in Table 4.10.

Contribution Type	**Feature**	**Concept**	**Model**
Positive Contribution	Ventilated and non-ventilated PaO2/FiO2 ratios	CNS and Renal Failure	Independent, Sequential, and Joint
	Creatinine value (max)	Liver and Coagulation Failure	Independent, Sequential, and Joint
	Bilirubin (max)	High Albumin level	Independent and Sequential
	Glasgow Coma Scale	Septic Shock	Independent and Joint
Negative Contribution	Bilirubin (max)	High Aniongap	Independent and Joint
	Glasgow Coma Scale	Coagulation Failure (severe)	Independent and Sequential

Table 4.10: Strong combined feature to concept contributions in different models.

The results are clinically relevant as they highlight the interplay between different organ systems in critically ill patients. The positive contributions generally align with clinical expectations: respiratory, renal, and liver dysfunctions are interconnected and can exacerbate each other. For example, renal failure often accompanies liver dysfunction, leading to a complex clinical picture.

The negative contributions are also noteworthy, especially the Glasgow Coma Scale's relationship with severe coagulation failure. This could indicate that neurological deterioration may not always be directly reflected in coagulation parameters, suggesting the need for multi factorial assessment in critically ill patients.

Conclusion

From these results, the following ideas are proposed which require further investigation:

1. The models may be identifying spurious correlations between features and concepts that still result in strong overall mortality predictions, possibly due to the imbalanced mortality dataset favouring positive mortality predictions.
2. The chosen concepts do not encompass a diverse range of information sources. The main information provided by the concepts were components of the SOFA score, known to be strong predictors of mortality. Thus, even though the concept predictors may not be making clinically sound correlations, the overall predictive strength of the score yields higher accuracy.
3. There could be an error in weights analysis method. Further investigation is needed to determine if there are any inconsistencies in the way the weights were extracted and used. Techniques like Layer-wise Relevance Propagation or Integrated Gradients, which were not utilised in this study, could provide more accurate insights.
4. The inconsistency in the outputs of the models for the feature to concept weights and concept to label weights could be due to leakage, which can improve label accuracy but at the cost of reduced interpretability, leading to clinically unintuitive weight contributions.

4.5 Future Work

The conclusion of the mortality prediction section of this project reveals several avenues for additional work and investigation.

Firstly, the initial concept set choice involved several simplifications. Based on the mortality prediction study referenced in [38], key features such as the anion gap, albumin, bilirubin, and SOFA scores were selected due to their high generalisability for large-scale trend analysis across a broad patient cohort. However, to capture the diverse complexities of patient conditions more

comprehensively, future studies should incorporate a wider array of information sources. These could include ventilation status, surgery information, fluid administration, medication, and pre-existing conditions, among others. As reiterated in [34], SOFA organ system scores only evaluate a limited number of organs, and markers of damage to one organ may sometimes represent damage to another. Furthermore, chronic or underlying causes that impact SOFA scores are often not accounted for in many studies used for clinical validation. Therefore, to achieve the 90%+ accuracies reported in current literature, an expanded feature and concept set is essential.

Additionally, the dataset in this study was simplified by limiting the time frame of the patient's stay to only the first day's lab measurements. This approach, while maintaining dataset simplicity, does not fully capture the experiences of patients in the ICU, where many patients have extended stays. To more accurately reflect ICU conditions, future research should process time series data or develop high-level concepts that capture trends throughout a patient's ICU stay. This would provide a more detailed and accurate representation of patient conditions and outcomes.

Moreover, the mortality prediction in this study was set to over a year after ICU discharge. While this provides long-term outcome data, it may not be as clinically useful as predicting mortality within a shorter timeframe. Future work should involve consultation with clinicians to determine more relevant and useful prediction timeframes for clinical settings. This would ensure that the predictions are not only accurate but also practical and beneficial for patient care.

The combined concepts based on SOFA scores, as illustrated in Table 4.2, were derived loosely from medical definitions. However, conditions such as ARDS or Septic Shock have specific nuances that require more informative features for accurate prediction. For instance, ARDS necessitates a comprehensive analysis of multiple medical sources. Despite extensive work in Chapters 5 and 6, the highest accuracy achieved was 72%. Therefore, future efforts should focus on enhancing the robustness of concept predictions by processing a broader range of features before creating these categorisations. This will improve the clinical utility and accuracy of the predictions.

Furthermore, investigation into the concept bottleneck models for predicting mortality reveals several key areas for future enhancement.

When fine-tuning the concept bottleneck models, the search space was highly constrained due to time limitations. In this study, the models were simplified by reducing the size and number of layers, implementing a learning rate scheduler, and exploring the differences between early stopping and L2 regularisation. However, neural networks possess a myriad of hyperparameters that can be fine-tuned, such as batch size, activation functions, dropout rates, and optimiser types. Future

research should explore a broader range of these hyperparameters to optimise model performance more comprehensively.

The evaluation of the concept and label predictors showed high accuracy and precision but lower recall values. This indicates that the model performs better on majority classes compared to minority ones, with the minority class for the mortality label being the occurrence of death. Upon closer inspection, it is observed that in the training cohort of 20,053 patients, there were 4,268 deaths and 15,785 survival cases, illustrating a class imbalance with a skew towards the negative class. The distribution of positive and negative classes for the intermediate concepts was not explored in this project. Future work should examine how class imbalance impacts evaluation metrics, as addressing this issue could improve recall and overall model robustness.

A notable strength of concept bottleneck models is the ability to intervene at the bottleneck layer. Collaborating with a medical professional to understand the concept predictions, alter values, or remove clinically accurate concepts could provide valuable insights. The final label predictions of these modified models could then be compared to the original models to assess the impact of these interventions on model accuracy and clinical relevance.

Additionally, efforts to understand the reasoning behind concept bottleneck models included weights analysis, which, while insightful, also generated further questions for exploration as described in Section 4.4.3. Another approach to leveraging the interpretability of concept bottleneck models involves introducing new test data and interpreting the resulting concept and label predictions in collaboration with a physician. This test data could focus on specific patient profiles where mortality prediction is particularly relevant or examine how altering a single feature affects the overall prediction. Such a study could provide deeper insights into the model's functioning and improve its clinical applicability.

By addressing these areas, future research can significantly improve the performance, interpretability, and clinical relevance of mortality prediction models.

5

ARDS Identification with CBM

Contents

5.1 Cohort Selection **52**
5.1.1 Berlin Definition 52
5.1.2 Screening Process 53
5.2 Dataset Construction **54**
5.2.1 Concepts Derived from SOFA Scores 54
5.2.2 Concepts Derived from Sepsis and Shock 55
5.2.3 Concepts Derived from Ventilation Status 55
5.2.4 Concepts Derived from Pre-existing Respiratory Conditions 56
5.2.5 Data Pre-processing 57
5.3 CBM Training **57**
5.4 Investigations **57**
5.4.1 Fine-tuning 57
5.4.2 Comparison Against Logistic Regression Models 61
5.4.3 Analysing Model Weights 61
5.5 Future Work **62**

After investigating the use of concept bottleneck models for predicting mortality, the project's focus shifted to using CBMs to identify Acute Respiratory Distress Syndrome (ARDS) in patients. Section 3.6 discusses the challenges of identifying ARDS and highlights the use of machine learning, including pre-trained large language models, to improve detection accuracy by integrating structured and unstructured medical data. Initially, a "vanilla" CBM, trained solely on EHR data, was implemented to identify ARDS. To complete this part of the project, collaboration with Dr Dominic Marshall was essential. His expertise was needed for cohort selection, dataset construction, and validation of the model outputs. As before, Google BigQuery was used to construct the dataset, while Google Colab was utilised for running CBM code and conducting experiments.

5.1 Cohort Selection

The following provides a summary of the steps taken by Dr Marshall to select the patient cohort, including the relevant background to understand his choices. It is important to understand this detailed review process because it informed the choice of labels that were extracted from the clinical notes in MIMIC-IV by the LLM in Chapter 6.

5.1.1 Berlin Definition

In 2011, a panel of experts developed the Berlin Definition [42], which has since been the standard criteria used to diagnose patients with ARDS. It includes the following 4 key points.

1. Timing: The symptoms must develop within one week of a known clinical insult (such as pneumonia, sepsis, or trauma) or new/worsening respiratory symptoms.

2. Chest Imaging: Chest X-rays or CT scans must show bilateral opacities. Opacities are white spots that appear on the imaging of the lungs through chest X-rays and CT scans because of fluid, inflammation, infection, or other substances. Bilateral opacities are when both sides of the lungs have such white spots.

3. Origin of edema: Edema is when there is a build up of fluids in the lungs' air sacs (called alveoli) which makes it difficult for oxygen to pass into the blood stream. The Berlin Definition requires the source of edema to not be because of cardiac failure or fluid overload.

4. Oxygenation: The level of oxygen in the blood is used to determine the severity of ARDS. There are two key metrics involved: PEEP and PaO2/FiO2 (P/F) ratio. PEEP or Positive End-Expiratory Pressure is a type of mechanical ventilation provided to patients with ARDS to try and keep their airways and alveoli open to improve oxygenation. P/F ratio is a metric used to determine how low the patient's blood oxygen level is. Using these two units, the Berlin Definition provides the following severity of ARDS:

   - Mild ARDS: P/F ratio between 200 and 300 mmHg with $PEEP \geq 5$ cm H2O.
   - Moderate ARDS: P/F ratio between 100 and 200 mmHg with $PEEP \geq 5$ cm H2O.
   - Severe ARDS: P/F ratio less than 100 mmHg with $PEEP \geq 5$ cm H2O.

Patient Cohort Size: 2021	
ARDS Positive	**ARDS Negative**
1001	1020

Table 5.1: Selected ARDS Cohort.

5.1.2 Screening Process

Dr Marshall's approach to cohort selection involved observing the Berlin Definition. In his literature review (Section 3.6), he noted that existing studies often relied on early ICU admission data (like the first day of admission) and various physiologic criteria around the Berlin definition, such as P/F ratios and PEEP levels, alongside keyword searches in radiology reports. However, the number of times these criteria are met varies a lot, so it is difficult to be sure the sample is accurate. Sometimes, patients might have a short-term drop in oxygen levels but that does not necessarily mean they have ARDS. Also, in the existing studies, it was not always clear if enough work had been done to exclude patients whose respiratory failure was due to cardiac failure or fluid overload (see Origin of edema above).

To address these issues, Dr Marshall applied stricter initial screening criteria, identifying instances where patients had a P/F ratio of less than 300 with a concurrent PEEP of 5. He then tracked the average P/F ratios over the following three days. If a patient maintained a P/F ratio of less than 300 for three days or for two days and then died on the third, the first instance was marked as time zero. The overall oxygenation criteria was that patients needed to have a P/F ratio below 300 for three consecutive days and a PEEP of at least 5 cm H2O, with an additional flag for those who died within 48-72 hours of the initial P/F measurement. Dr Marshall then attempted to refine his selection by screening chest radiographs for bilateral infiltrates using search terms like "ARDS" and "bilateral infiltrates." However, due to inconsistencies in radiograph reporting, this method proved unreliable.

Dr Marshall went on to investigate the cause of respiratory failure in the patients by reviewing their discharge summaries. These summaries provided comprehensive details, including history, radiology, echocardiography, and overall clinical context, allowing for a more accurate assessment of ARDS. The discharge summaries were also examined to identify physiological insults, which are events or conditions that cause damage or stress to the body's systems. The relevant physiological insults for this case included pneumonia, aspiration, and pancreatitis. Furthermore, Dr Marshall reviewed the radiology reports for bilateral infiltrates, taking into account clinical context. He

also checked echocardiography reports for significant heart failure indicators. Dr Marshall used his screening process to categorise cases as ARDS, possible ARDS, or not ARDS based on clinical evidence and expert interpretation. From this dataset, a nearly balanced selection of patients with and without ARDS was chosen for this project, Table 5.1.

5.2 Dataset Construction

In this study, the label was a binary value indicating whether Dr Marshall had diagnosed the patient with ARDS or not. The features and concepts used were derived from useful indicators of ARDS found in the structured data within the ICU and Hosp modules of MIMIC-IV. Ultimately, 26 features and 19 concepts were chosen, Table B.3.

5.2.1 Concepts Derived from SOFA Scores

Sections 3.5 and 4.2.1 provide an overview of the sequential organ failure assessment (SOFA) score. The useful SOFA components for ARDS are the renal, respiratory, and neurological scores. The renal score evaluates kidney function which is often impaired when the patient suffers from severe ARDS. Furthermore, ARDS is characterised by severe lung dysfunction, so the respiratory score helps capture the hallmark feature of ARDS. Finally, one of the common complication of ARDS is an insufficient supply of oxygen (hypoxia) to the brain that can lead to brain dysfunction. The neurological scores help in detecting early signs of hypoxia-induced brain impairment.

The scores were then used as concepts for categorising the severity of organ failure: a score of 1 or 2 indicated moderate failure, while a score of 3 or 4 indicated severe failure. The features were the lab measurements used to calculate the corresponding SOFA component scores. This study focused on identifying ARDS in patients based on their overall ICU stay rather than pinpointing the onset of ARDS, thus no time series processing was required. Instead, the SOFA component scores used for analysis were the maximum score recorded for each patient and the average score during their stay. Similarly, the features included the measurement associated with the first instance of the maximum score and the average of all measurements. Refer to rows 1, 2, and 3 of Table B.3 for the relevant list of concepts and features.

5.2.2 Concepts Derived from Sepsis and Shock

Sepsis and shock were both identified as useful indicators for ARDS. Sepsis (Section 4.2.1) is a common trigger for ARDS because widespread inflammation and infection can damage lung tissues, leading to fluid accumulation and respiratory failure. The features and concepts chosen for sepsis were influenced by the Sepsis-3 clinical definition, established in 2016 [48]. According to this criterion, a patient suffers from sepsis if they have a clinical suspicion of infection (indicated by positive cultures or the administration of antibiotics) and a change in SOFA score of 2 or more points due to the infection. In this study, the features for sepsis included suspicion of infection (a binary flag) and SOFA score (between 48 hours before to 24 hours after the suspected infection). The concept was whether or not the patient had sepsis.

Similarly, shock can cause inflammation and multiple organ dysfunction, further increasing the risk of ARDS. Shock victims suffer from low blood pressure due to factors that impair the body's ability to maintain adequate circulation and perfusion (oxygen transfer) to vital organs. They are commonly given vasopressors, which are medications that narrow blood vessels to increase blood pressure. Examples of vasopressors include Dopamine, Epinephrine, Norepinephrine, Vasopressin, and Phenylephrine. For this study, it was assumed that if a patient in the ARDS cohort was administered any of these drugs, they suffered from shock. The concept chosen was whether or not the patient suffered from shock. The features chosen were flags for whether the patient had been administered one or multiple vasopressors. This distinction was made in the initial data construction to determine if it could provide useful information for predicting other concepts. Refer to rows 4 and 5 of Table B.3 for the relevant list of concepts and features.

5.2.3 Concepts Derived from Ventilation Status

The primary treatment for ARDS involves ventilation to improve blood oxygen levels. In MIMIC, there are five types of ventilation statuses, which can be grouped into three categories.

Ventilation for severe ARDS:

- Tracheostomy: This is the surgical creation of an opening in the trachea (windpipe) through which a tube is inserted to facilitate breathing. It is often considered for patients who require prolonged mechanical ventilation (usually more than 7-10 days). Performing tracheostomy on a patient can indicate a severe and prolonged ARDS course.

- Invasive Ventilation: This is mechanical ventilation via an endotracheal tube (ET tube) or tracheostomy tube inserted into the patient's airway. The need for invasive ventilation indicates a high severity of ARDS and poor spontaneous breathing capacity.

Ventilation for moderate ARDS:

- High-Flow Nasal Cannula (HFNC): This delivers a high flow of heated and humidified oxygen through nasal prongs. It can provide higher concentrations of oxygen and some degree of positive airway pressure, improving oxygenation and reducing work of breathing. It is suitable for patients with moderate ARDS who need more support than standard supplemental oxygen but do not yet require invasive ventilation.
- Non-Invasive Ventilation: Mechanical ventilation using a mask or similar device. It is also useful for patients with moderate ARDS because it can help reduce the work of breathing and improve gas exchange without the need for intubation. Failure to improve with non-invasive ventilation may require transitioning to invasive ventilation.

Ventilation for mild ARDS:

- Supplemental Oxygen: This is the delivery of additional oxygen via nasal cannula, face mask, or other non-invasive methods. It is used in mild cases of ARDS where the patient can still breathe on their own but requires extra oxygen to maintain adequate blood oxygen levels.

For this study, the features were binary flags that indicated which type(s) of ventilation the patients were on during their time in the ICU. The concepts were based on the three categories above: severe ventilation, moderate ventilation, and mild ventilation. If they received ventilation from that category, the value of that concept was set to 1 otherwise it was 0. Refer to row 6 of Table B.3 for the relevant list of concepts and features.

5.2.4 Concepts Derived from Pre-existing Respiratory Conditions

The presence of pre-existing respiratory conditions in patients is also important to consider. This is because such conditions make the patient more susceptible to severe respiratory complications, including ARDS, when exposed to additional stressors like infections or trauma. Information about the patient's existing respiratory health was deduced from the International Classification of Diseases (ICD) codes associated with each patient. Table 5.2 describes the different categories

of relevant pre-existing conditions. The features for this study included whether the patient was flagged for these six conditions. Two concepts were derived from this: moderate (if the patient had 1 or 2 pre-existing conditions) or severe (if the patient had 3 or more conditions). Refer to row 7 of Table B.3 for the relevant list of concepts and features.

5.2.5 Data Pre-processing

This step mirrored the process used for the mortality prediction dataset. All missing features and concepts were replaced with 0. The data was then split into training, validation, and test sets. The validation set was used for evaluation during training, and the test set was used for the final evaluation of the model's generalisation ability.

5.3 CBM Training

The initial training approach for the vanilla ARDS identification CBMs was also virtually identical to that of the mortality prediction CBMs, Section 4.3. The same optimiser, activation functions, and loss functions were used for the corresponding independent, sequential, and joint CBMs, (Algorithms 1, 2, and 3, respectively). The main differences were that the dataset had changed, leading to a different total number of features and concepts, and the layer sizes were reduced to accommodate for a smaller patient cohort.

5.4 Investigations

5.4.1 Fine-tuning

The loss plots from training the independent, sequential, and joint classifiers can be seen in Appendix B.3.2. All three showed overfitting in their validation loss plots, Figure B.3. Consequently, the same strategies discussed in Section 4.4.1 were employed to improve the model architecture and training process. Figure B.4 illustrates the improvements from implementing early stopping, and Figure B.5 shows the benefits of using L2 regularisation. The evaluation metrics of the concept predictor and label predictor for the different CBMS were then recorded for analysis.

Starting with the concept predictor in Table 5.3, the accuracy across the models was consistently

Categories	Description	ICD Codes
Upper Respiratory Infections	These include infections of the nasal passages, pharynx, larynx, and upper trachea. Common conditions include the common cold, pharyngitis, sinusitis, and laryngitis.	Common Cold (ICD-9: 460, ICD-10: J00) Acute Pharyngitis (ICD-9: 462, ICD-10: J02.9) Acute Sinusitis (ICD-9: 461, ICD-10: J01.90)
Influenza Pneumonia	This category encompasses viral infections like influenza and bacterial infections like pneumonia. These conditions affect the lungs and respiratory system, often causing symptoms such as fever, cough, and difficulty breathing.	Influenza with Pneumonia (ICD-9: 487.0, ICD-10: J11.00) Viral Pneumonia (ICD-9: 480, ICD-10: J12.9) Bacterial Pneumonia (ICD-9: 481, ICD-10: J13)
Acute Lower Respiratory Infections	These include infections that affect the lower respiratory tract, such as bronchitis and bronchiolitis. These infections are typically more severe than upper respiratory infections and can lead to significant respiratory distress.	Acute Bronchitis (ICD-9: 466.0, ICD-10: J20.9) Bronchiolitis (ICD-9: 466.1, ICD-10: J21.9)
Chronic Lower Respiratory Diseases	Chronic conditions such as chronic obstructive pulmonary disease (COPD), chronic bronchitis, and emphysema fall under this category. These diseases are characterised by long-term respiratory symptoms and airflow limitation.	Chronic Obstructive Pulmonary Disease (ICD-9: 496, ICD-10: J44.9) Chronic Bronchitis (ICD-9: 491.9, ICD-10: J41.0) Emphysema (ICD-9: 492.8, ICD-10: J43.9)
Lung Diseases Due to External Agents	This category includes diseases caused by inhalation of external agents such as smoke, dust, chemicals, and environmental pollutants. Conditions like pneumoconiosis and other occupational lung diseases are included.	Pneumoconiosis (ICD-9: 500-505, ICD-10: J60-J65) Respiratory conditions due to chemicals, gases, fumes, and vapors (ICD-9: 506, ICD-10: J68)
Other Respiratory Diseases	This is a broader category that includes various other respiratory conditions not classified under the above categories. Examples include respiratory failure, pulmonary embolism, and pleurisy.	Respiratory Failure (ICD-9: 518.81, ICD-10: J96.00) Pulmonary Embolism (ICD-9: 415.19, ICD-10: I26.99) Pleurisy (ICD-9: 511.9, ICD-10: J90)

Table 5.2: Categories of pre-existing respiratory conditions with their corresponding ICD codes.

high (0.94-0.96), indicating that the models are proficient at identifying ARDS concepts. The AUC values were stable at 0.82, except for the joint model with early stopping, which had a significantly lower AUC of 0.62, indicating poorer performance in distinguishing between classes for this specific setup. Precision was generally high (0.79-0.88), suggesting that when the models predicted a positive concept, it was usually correct. Recall values varied, with the lowest being 0.60 for the joint model with early stopping and the highest being 0.81 for other models, indicating moderate to good ability in identifying all positive concepts. Consequently, the F1 scores followed the trends of precision and recall.

Concept Predictor's Model	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1 Score**
Independent	0.96	0.82	0.87	0.80	0.83
Sequential	0.96	0.82	0.87	0.80	0.83
Joint	0.94	0.82	0.79	0.76	0.76
Independent w/ Early Stopping	0.96	0.82	0.88	0.81	0.83
Sequential w/ Early Stopping	0.96	0.82	0.87	0.81	0.83
Joint w/ Early Stopping	0.89	0.62	0.68	0.60	0.62
Independent w/ L2 Regularisation	0.96	0.83	0.88	0.81	0.83
Sequential w/ L2 Regularisation	0.96	0.83	0.88	0.81	0.83
Joint w/ L2 Regularisation	0.96	0.82	0.87	0.80	0.82

Table 5.3: Averaged evaluation metrics for ARDS CBM concept predictors.

Next, the label predictors in Table 5.4 showed lower performance metrics compared to the concept predictors. Accuracy ranged from 0.55 to 0.68, reflecting the more challenging task of predicting final labels compared to intermediate concepts. The AUC values were also lower (0.56-0.68), with the joint model with early stopping showing the highest AUC at 0.68, indicating slightly better performance in class separation for this specific setup. Precision ranged from 0.50 to 0.65, showing that positive label predictions were less consistently correct compared to concept predictions. Recall varied widely, with a high of 0.75 for the independent model and a low of 0.49 for the sequential model with early stopping, indicating inconsistencies in identifying all positive labels. The F1 scores were generally lower as well, ranging from 0.54 to 0.66. Fine-tuning with early stopping and L2 regularisation provided improvements, particularly for the joint model, which showed the most balanced performance across all metrics.

Finally, the label predictors for the Mortality CBMs (Table 4.5) and ARDS CBMs (Table 5.4) were compared. The accuracy of the ARDS label predictor (0.55-0.68) was significantly lower than

Label Predictor's Model	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1 Score**
Independent	0.57	0.58	0.52	0.75	0.61
Sequential	0.55	0.56	0.50	0.71	0.59
Joint	0.61	0.62	0.57	0.65	0.61
Independent w/ Early Stopping	0.62	0.62	0.58	0.63	0.60
Sequential w/ Early Stopping	0.61	0.60	0.59	0.49	0.54
Joint w/ Early Stopping	0.68	0.68	0.65	0.65	0.65
Independent w/ L2 Regularisation	0.63	0.63	0.59	0.63	0.61
Sequential w/ L2 Regularisation	0.60	0.59	0.56	0.57	0.57
Joint w/ L2 Regularisation	0.67	0.67	0.63	0.69	0.66

Table 5.4: Evaluation metrics for ARDS CBM label predictors.

that of the Mortality CBM (0.82-0.83). The AUC values for the ARDS CBMs were also generally lower, indicating that classifying ARDS cases was more challenging than classifying mortality. This is understandable because ARDS identification is inherently more complex. Furthermore, the ARDS cohort consists of patients already screened for a high likelihood of having ARDS, making the task of distinguishing specific diagnoses even more difficult.

Precision in the ARDS CBM label predictors varies from 0.50 to 0.65, whereas it is consistently higher in the Mortality CBM, ranging from 0.70 to 0.79. This suggests that the Mortality CBM is better at making correct positive predictions. Recall values are more variable in the ARDS CBM (0.49-0.75), with the highest recall being significantly better than the Mortality CBM (0.27-0.37), but generally, Mortality CBM models struggle with recall, indicating a tendency to miss positive cases.

Overall, the Mortality CBM label predictors demonstrate higher accuracy and precision, while the ARDS CBM label predictors exhibit better recall in certain cases. Both models benefit from fine-tuning strategies, but the inherent complexity of ARDS prediction presents greater challenges compared to mortality prediction.

Clinically, it is understood that the best way to improve ARDS diagnosis accuracy is by incorporating additional insights from various medical data sources, a task that will be addressed in Chapter 6. However, it was still useful to develop a deeper understanding of the current vanilla ARDS CBM. Comparing it against the Mortality CBM investigations would provide insights and inform the decisions aimed at enhancing the ARDS CBM's performance in Chapter 6. Conse-

quently, the same investigations conducted for the Mortality CBM were replicated for the current ARDS CBM to facilitate this comparison.

5.4.2 Comparison Against Logistic Regression Models

Like before, a feature regressor (X to Y model) and concept regressor (C to Y model) were implemented to predict the ARDS label. Unlike the mortality regressors, these regression models exhibited the expected behaviour, as shown in Table 5.5: the concept regressor performed better than the feature regressor across all evaluation metrics because the concepts capture the patient information in a more meaningful way than the raw features.

Model	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1 Score**
X to Y	0.56	0.61	0.58	0.54	0.56
C to Y	0.60	0.66	0.62	0.56	0.59

Table 5.5: Evaluation metrics for ARDS feature regressor (X to Y) and concept regressor (C to Y).

It was observed from Table 5.4 that generally, the concept regressor model demonstrated slightly higher overall accuracy, AUC, precision, recall, and F1 scores compared to the non fine-tuned ARDS CBM label predictors. However, fine-tuning the CBMs resulted in noticeable improvements relative to the concept regressor. The fine-tuned joint models outperformed the concept regressor, indicating that training a bottleneck model on both features and concepts yielded the best performance when overfitting was reduced. The fine-tuned independent models displayed comparable performance to the concept regressor, while the fine-tuned sequential models performed worse.

5.4.3 Analysing Model Weights

Subsequently, the model weights were examined to determine if the ARDS CBM exhibited the same inconsistencies in feature and concept weighting as the Mortality CBM, and to identify any relevant clinical insights from these weights. As in Section 4.4.3, the L2 regularised models were selected for this investigation, and the same 'viridis' colour scheme from the Matplotlib library in Python was used for the heat maps. The heat maps for 'Feature to Concept Weights Contribution', 'Concept to Label Weights Contribution', and 'Combined Feature to Concept Contributions Predicting ARDS' can be found in Appendix B.3.3.

This investigation was conducted to ensure project completeness. Instead of performing an

in-depth analysis of specific feature and concept weightings, the focus was on observing the general trends in the heat maps. For the Mortality CBMs, the weight scale ranged from 0.4 to -0.4. However, for the ARDS CBM, most weights were neutral within this same scale, particularly in the combined feature-to-concept contributions heat map. Therefore, the scale was reduced to 0.2 to -0.2.

Considering the feature-to-concept weights heat map, the ARDS CBM, like the Mortality CBM, showed nearly identical heat maps for the independent and sequential models, but these differed from the joint model. The ARDS CBM exhibited many more strongly positive and negative weightings because the scale had to be reduced to visualize the lower weights in the other two heat maps. Despite this adjustment, the weights did not exhibit the expected behaviour for more straightforward concept derivations. Furthermore, the concept-to-label weights and combined feature-to-concept contribution heat maps showed varied results across models. Due to this variation and the lower accuracy of the ARDS label prediction, the clinical reasoning behind the models was deemed not useful for evaluation. Overall, these models reaffirmed the presence of potential spurious correlations in the concept prediction level and a strong possibility of leakage, which warranted further investigation as outlined for the Mortality CBMs.

5.5 Future Work

The conclusion of creating the vanilla ARDS CBM for this project reveals several avenues for future work. In this chapter, collaboration with Dr Marshall ensured the selection of robust features and concepts. However, prioritising the most useful features to minimise dataset construction time led to certain omissions. For instance, the model did not consider whether the patient had undergone surgery or suffered from infections (indicated by white blood cells, temperature, etc.) from the EHRs. Additionally, the time series aspect of the data was simplified. The task was framed to diagnose ARDS based on all available information during the patient's stay, rather than pinpointing the onset of ARDS. Future work should aim to replicate ICU conditions where real-time data processing is essential for making a timely ARDS diagnosis. This includes considering SOFA component scores and associated measurements dynamically, rather than using average and maximum values.

Following the initial results from the vanilla models, the project progressed to integrating additional insights from LLMs in Chapter 6. This leaves more work to be done in understanding

and modifying the current data. If more time had been available, it would have been beneficial to conduct an in-depth analysis of concept predictions with Dr Marshall to refine the concept set. While Dr Marshall's expertise was frequently consulted, a more systematic iterative procedure should have been established, involving continuous feedback to frequently incorporate medical expertise into the data selection and evaluation of results.

Furthermore, the approach for training the CBMs, fine-tuning them, and investigating their weights was identical to that of the mortality prediction CBM. Consequently, similar future work applies to the ARDS CBM (refer to Section 4.5). Expanding the search space for fine-tuning could lead to more comprehensive results. Although the label prediction for ARDS was more balanced, the concept predictions were not evaluated for class imbalance. Addressing this gap by investigating the impact of class imbalance on concept predictions is crucial for improving the model's performance. Further investigation is also needed on intervening on concepts, explaining the results of weights analysis, and interpreting the model's predictions on new test data of specific patient profiles.

6

Enhancing ARDS CBM with Labels from LLM

Contents

6.1 Dataset Construction 66
6.1.1 Data Pre-processing 66
6.1.2 LLM Concept Labels 67
6.2 Implementing LLM for Clinical Concept Labelling 69
6.2.1 Evaluating Open Source LLMs for Concept Labelling 70
6.2.2 Scaling Up and Further Investigations 74
6.2.3 Final LLM Implementation 77
6.2.4 Random Sampling Validation 77
6.3 Augmenting Concept Set with LLM Labels 78
6.3.1 Initial Attempt 78
6.3.2 Using Altered Architecture 80
6.3.3 Investigating Completeness Score 82
6.4 Future Work 83

Investigating the vanilla ARDS identification CBMs revealed significant room for improvement. Clinically, this is understandable because, as discussed in Section 3.6, diagnosing ARDS requires processing various types of medical data, including laboratory measurements, radiology reports, and clinical text. In Dr Marshall's approach to diagnosing ARDS, although the EHR data from MIMIC served as a useful screening tool, the insights derived from clinical notes were crucial for making the final diagnosis. Therefore, an attempt was made to integrate these additional insights into the vanilla CBM. To achieve this, an LLM was utilised to process the unstructured notes and generate additional clinical concept labels, aiming to enhance the model's diagnostic capability.

6.1 Dataset Construction

6.1.1 Data Pre-processing

According to Dr Marshall's recommendation, the LLM needed to process the discharge summaries, radiology reports, and echocardiography notes for patients in the ARDS cohort. Since patient selection was already completed in the initial CBM stage (Section 5.1), the next step was to gather all relevant clinical text for each patient using Google BigQuery. For this task, the MIMIC database's Note and ECG modules were utilised. The Note module contains the discharge summaries and radiology reports, while the ECG module the electrocardiogram studies taken for patients.

During the process of compiling this information, the following insights were made about the data stored for the patient cohort:

1. In the ARDS cohort of 2,021 patients, all had a discharge summary. 2,020 patients had at least one radiology report (1 missing), and 1,881 patients had ECG studies (140 missing).
2. Each patient had only one discharge summary, but there were more than 2,021 radiology reports and ECG studies. The ARDS cohort had 50,597 radiology reports, averaging around 25 notes per patient. Similarly, there were 29,442 ECG studies, averaging about 12 studies per patient.
3. The ECG studies included up to 18 echo reports per study, with an average of 3.45 reports per study.
4. Since patients often had multiple radiology reports and ECG studies, these were concatenated together, separated by a special character divider. The corresponding report IDs and ECG study IDs were also recorded for cases where distinguishing between different reports was useful. However, for this project, it was found that this distinction was unnecessary, and considering all data together was crucial for obtaining a complete picture of the patient's health.

Finally, LLMs have token limits. Tokens are the basic units of text that a model processes, which can be as small as characters or as large as entire words or phrases, depending on the tokenization method used. Tokenization is the process of converting a string of text into these tokens, which the model then uses for analysis and understanding.

Statistic	Discharge Text	Radiology Text	ECG Text
Count	2021.00	2020.00	1881.00
Mean	4212.77	5526.75	447.49
Standard Deviation	1960.56	4505.05	740.99
Min	308.00	288.00	5.00
25%	2821.00	2538.25	84.00
50%	3884.00	4309.50	208.00
75%	5161.00	7154.25	488.00
Max	15362.00	60018.00	7967.00

Table 6.1: Token count statistics for discharge, radiology, and ECG texts.

To understand the text in terms of tokens, the discharge summaries, concatenated radiology reports, and ECG studies were run through a standard tokenizer. This process breaks down the text into tokens, allowing the LLM to process and analyse the data effectively. The results of this tokenization can be seen in Table 6.1.

6.1.2 LLM Concept Labels

Now that the data required for the LLM to process had been identified, the next step was to determine which labels the LLM would extract from the clinical text. These labels needed to represent information that was not already provided by the EHR data in MIMIC. They were based on Dr Marshall's manual process of diagnosing ARDS (as discussed in Section 5.1), because the LLM is intended to replicate his method when reviewing the three different types of clinical texts. Dr Marshall's method involved examining discharge summaries for relevant physiological insults such as pneumonia, aspiration, and pancreatitis. He also looked for indicators of heart failure in the ECG studies and mentions of bilateral infiltrates in the radiology reports. After discussions with Dr Marshall, the following labels were identified:

- From Discharge Summaries: Mentions of ARDS, signs of aspiration, pneumonia, pancreatitis, cardiac arrest, and TRALI.
- From Radiology Reports: Mentions of bilateral infiltrates.
- From ECG Studies: Indications of cardiac failure. It was found that combining information from discharge summaries and ECG studies was useful for identifying cardiac failure.

The "mention of ARDS" label is self-explanatory. It should be noted that not all identified ARDS patients had it mentioned in their discharge summary, which supports the claim made in the papers discussed in 3.6 that ARDS often goes undiagnosed. However, there were several instances where it was identified in the text. Including this label could also provide insights into any discrepancies between the discharge summaries and Dr Marshall's diagnoses.

Here is a brief description of the other labels and their relation to ARDS:

- Aspiration occurs when foreign material, such as food or liquid, is inhaled into the lungs. This can lead to lung inflammation and infection, which can trigger ARDS by causing severe lung injury.

- Pneumonia is an infection that inflames the air sacs in one or both lungs. The inflammation and fluid buildup can damage the lung tissue, potentially leading to ARDS.

- Pancreatitis is inflammation of the pancreas. Severe cases can release inflammatory mediators into the bloodstream, which can affect the lungs and contribute to the development of ARDS.

- Cardiac arrest is the sudden loss of heart function. The resulting lack of oxygen to the body's tissues can cause widespread damage, including to the lungs, potentially leading to ARDS.

- TRALI (Transfusion-Related Acute Lung Injury) is a serious respiratory condition that occurs after a blood transfusion. It is characterised by acute lung injury and is one of the leading causes of transfusion-related mortality, often resulting in ARDS.

- Bilateral infiltrates, a specific type of opacity, refer to the presence of fluid or other substances in both lungs, visible on a chest X-ray or CT scan. This finding is a common diagnostic criterion for ARDS, indicating widespread lung involvement, (refer to the Berlin Definition in Section 5.1.1).

- Cardiac failure, or heart failure, is the inability of the heart to pump blood effectively. While it primarily affects the heart, the resulting fluid buildup can lead to pulmonary edema, which can complicate or mimic ARDS. It is important to distinguish between heart failure and ARDS to ensure appropriate treatment.

6.2 Implementing LLM for Clinical Concept Labelling

The LLM component of this project was completed towards the end of its development, so speed and ease of set-up was prioritised. Initially, an attempt was made to deploy an LLM on a free Google Colab account, as Colab had been used for all of the concept bottleneck model code up to this point. However, free Colab accounts typically provide around 12GB of RAM, which is insufficient for most LLMs. The process of loading an LLM into Colab proved arduous, with sessions often timing out and losing progress. Consequently, the decision was made to set up an LLM locally using Ollama.

Ollama is a free platform that simplifies the installation and operation of open-source pre-trained LLM models locally. It efficiently leverages local hardware resources to accelerate large language model inference and optimise performance. Once Ollama was set up, basic prompting experiments were conducted to gain familiarity with working with LLMs. Subsequently, more formal investigations were carried out on the following LLMs (descriptions are from Ollama's library, [49]):

1. stablelm-zephyr:3b - StableLM Zephyr 3B is a lightweight chat model that is "preference tuned on 3 billion parameters for instruction following and Q&A-type tasks."

2. llama3 - For this project, the 8 billion parameter Meta Llama 3 instruction-tuned model was used. It is "optimized for dialogue/chat use cases and outperform(s) many of the available open-source chat models on common benchmarks." According to [49], these are "the most capable openly available LLM to date."

3. phi3:mini - "Phi-3 is a family of open AI models developed by Microsoft. Phi-3 Mini is a 3.8B parameters, lightweight, state-of-the-art open model trained with the Phi-3 datasets."

4. gemma - "Gemma is a family of lightweight, state-of-the-art open models built by Google DeepMind." For this project, the Gemma model which is trained on 7 billion parameters was used.

5. mistral - Mistral AI's 7 billion parameter, instruction fine-tuned model was utilised. It is known to outperform Meta's previous state-of-the art Llama 2 models on most benchmarks.

6.2.1 Evaluating Open Source LLMs for Concept Labelling

The scope for evaluating and fine-tuning parameters is extensive; therefore, the search space was narrowed down to three specific tasks. These tasks were chosen for their relevance in selecting an open-source LLM capable of reading various clinical texts and generating concept labels.

To implement the tasks, the LangChain library was utilised. LangChain is a framework designed to help developers build applications powered by language models. It provides tools and abstractions for integrating language models with various data sources, orchestrating complex workflows, and deploying these applications at scale. For this project, LangChain was primarily used to load the LLM (LangChain supports Ollama) and to perform model inference and chunking.

Model inference involves using a pre-trained machine learning model to make predictions or generate outputs based on new input data. LangChain offers features to easily set up a generic prompt template and use it to prompt any loaded model. Chunking, the process of breaking down large pieces of text into smaller, manageable segments, was necessary because the clinical text used in this project often exceeded the token limits of the models being used, Table 6.1. To perform inference, long documents were divided into chunks, each chunk was fed into the model separately, and the results were then aggregated to decide on a final response. A chunk overlap was also specified. This refers to the practice of including a small portion of text from the end of one chunk at the beginning of the next chunk to ensure continuity and context retention across chunks.

#1 Discharge Text Summarisation Task

For this task, three different discharge texts of varying lengths were selected and fed into each of the five models. Using LangChain's "summarize" library, a summarisation chain was invoked to process text chunks, and the resulting summaries were aggregated into a final output. The summaries were then scored on two criteria: capturing meaningful information and understanding text structure, with each criterion rated between 1 and 5, resulting in a final score out of 10. To produce these scores, the summaries were manually reviewed. Based on Table 6.2, the llama3 model consistently outperforms the other models, achieving the highest average score of 6.7. Models like stablelm-zephyr:3b and mistral struggled to capture the structure of the text in their summaries, failing to clearly distinguish between the different sections of the discharge text. Additionally, the time taken to perform summarisation was recorded, as shown in Figure 6.1. For a smaller text length of 9,219 characters, the summarisation times were approximately the same across models. However, disparities became more apparent with larger text sizes, with the stablelm-zephyr:3b

Model	Trial 1	Trial 2	Trial 3	Average
stablelm-zephyr:3b	4	6	3	4.3
llama3	7	6	7	6.7
phi3:mini	6	7	2	5
gemma	7	5	5	5.7
mistral	3	4	5	4

Table 6.2: Scores (out of 10) for models based on the effectiveness of their summaries in capturing meaningful information and understanding text structure.

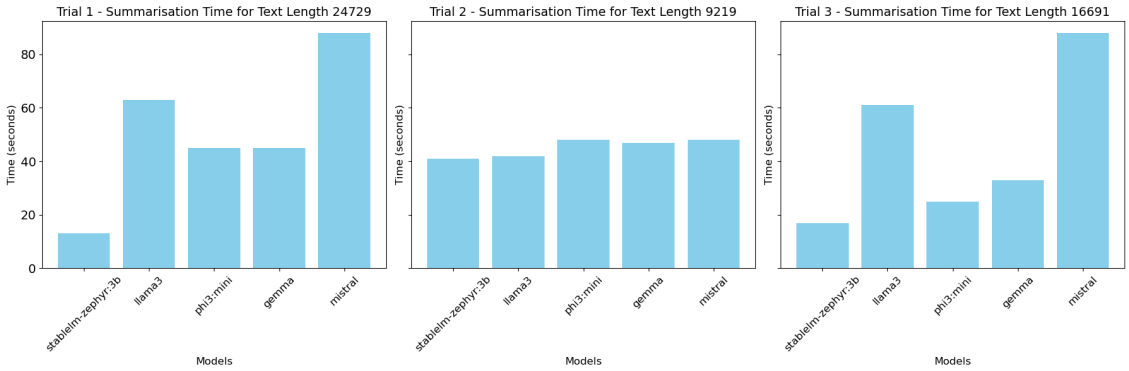

Figure 6.1: Summarisation times for the 5 LLMs under investigation.

model being the fastest and the mistral model being the slowest.

#2 Producing Labels for Pneumonia from Discharge Summary

During the initial investigation, various prompts were tested for identifying meaningful labels in the clinical text. Ultimately, the following prompt was found to be the most effective across the different LLMs used in these experiments:

```
template=(
    "Context: You are a clinician receiving chunks of clinical text for
        patients in an ICU. Please do the reviewing as quickly as possible.\n"
    "Task: Determine if the patient had pneumonia.\n"
    "Instructions: Answer with 'Yes' or 'No'. If there is not enough
        information, answer 'No'.\n"
    "Discharge Text:\n{discharge_text}\n\n"
    "Query: Does the chunk of text suggest that the patient has pneumonia?
        Answer strictly in 'Yes' or 'No'. Then provide a reason for your
        response."
),
input_variables=["discharge_text"]
```

The structure of this prompt was inspired by Named Entity Recognition and Classification strategies found in [26] and [28]. It provides the necessary context for the LLM to process the notes, informs it about chunking, and ensures succinct responses. The prompt also asks for a reason for the overall diagnosis of whether the patient had pneumonia, to evaluate the LLM's reasoning capability. For simpler clinical tasks like identifying pneumonia, which is usually explicitly mentioned in discharge summaries, using more open-ended terms like "suggest" proved useful. However, identifying bilateral infiltrates in radiology reports, which requires clinical reasoning and is not usually explicitly stated, benefited from more direct terms like "mention." On reflection, it would have been beneficial to investigate few-shot prompting and providing additional context in prompts. These aspects are among several areas for further exploration, discussed in Section 6.4.

The pneumonia prompt was applied to chunks of data, leading to varied responses based on the provided chunks. All responses were aggregated and manually reviewed to assign a score to the quality of reasoning provided by the model. The scoring was based on two criteria: how well the model extracted medical evidence and how well it understood the patient's context, with each criterion rated between 1 and 5, resulting in a final score out of 10. The LLMs performed inference on four different discharge summaries of varying lengths, Table 6.3.

Model	**Trial 1**	**Trial 2**	**Trial 3**	**Trial 4**	**Average**
stablelm-zephyr:3b	6	8	7	5*	6.5
llama3	8	7	8	9	8
phi3:mini	8	7	7	5*	6.75
gemma	5	6	6	4*	5.25
mistral	8	9	8	8	8.5

Table 6.3: Evaluating the models' ability to produce labels for pneumonia. The scoring (out of 10) is based on their effectiveness in extracting medical evidence and understanding clinical context. Scores with asterisks (*) are for cases where the LLM's response was incorrect.

Most of the models displayed strong evidence extraction and medical reasoning for the first three trials. Gemma, on the other hand, performed well with extraction but showed weaker reasoning based on the medical evidence. In Trial 4, only the mistral and llama3 models correctly identified that the text mentions that upon discharge, the patient's chest X-ray (CXR) showed a new right lower lobe (RLL) infiltrate, and he was started on "Cefepime" and "Vanco" empirically (which are antibiotics typically used for pneumonia). These models demonstrated an enhanced medical reasoning and understanding of the patient's condition compared to the others.

#3 Producing Labels for Bilateral Infiltrates from Radiology Report

Model	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Average
stablelm-zephyr:3b	5	3	5*	4*	4*	4	4.17
llama3	7	8	7	7	5*	8	7
phi3:mini	7	7	4*	6	7	6	6.17
gemma	6	5	4*	3*	4*	5	4.50
mistral	8	6	5*	7	7	8	6.67

Table 6.4: Evaluating the models' ability to produce labels for bilateral infiltrates. The scoring (out of 10) is based on their effectiveness in extracting medical evidence and understanding clinical context. Scores with asterisks (*) are for cases where the LLM's response was incorrect.

The LLMs were then evaluated on the more challenging task of labelling bilateral infiltrates in radiology reports.

```
template=(
    "Context: You are a clinician receiving chunks of radiology reports for
        patients in an ICU. Please do the reviewing as quickly as possible.\n"
    "Task: Determine if the patient suffered from bilateral infiltrates.\n"
    "Instructions: Answer with 'Yes' or 'No'. If there is not enough
        information, answer 'No'.\n"
    "Discharge Text:\n{radiology_texts}\n\n"
    "Query: Does the chunk of text mention that the patient suffered from
        bilateral infiltrates? Answer strictly in 'Yes' or 'No'.  Then provide
        a reason for your response."
),
input_variables=["radiology_texts"]
```

The prompt structure was nearly identical to that used for pneumonia but adapted for identifying bilateral infiltrates. The context was modified to inform the model that it was processing radiology reports instead of discharge summaries, as the structures of the two types of documents differ significantly. Additionally, the keyword "mention" was used rather than "suggest" to better suit the nature of the task. For this evaluation, the models performed inference on six different reports. They were scored based on the same criteria as before: the quality of medical evidence extraction and the understanding of the patient's context, with each criterion rated between 1 and 5, resulting in a final score out of 10.

The correct answers for whether the patient suffered from bilateral infiltrates are: No, No, Yes, Yes, Yes, No. From the results in Table 6.4, it can be seen that the trials where the models made mistakes were 3, 4, and 5, indicating a tendency for the models to output false negatives for this

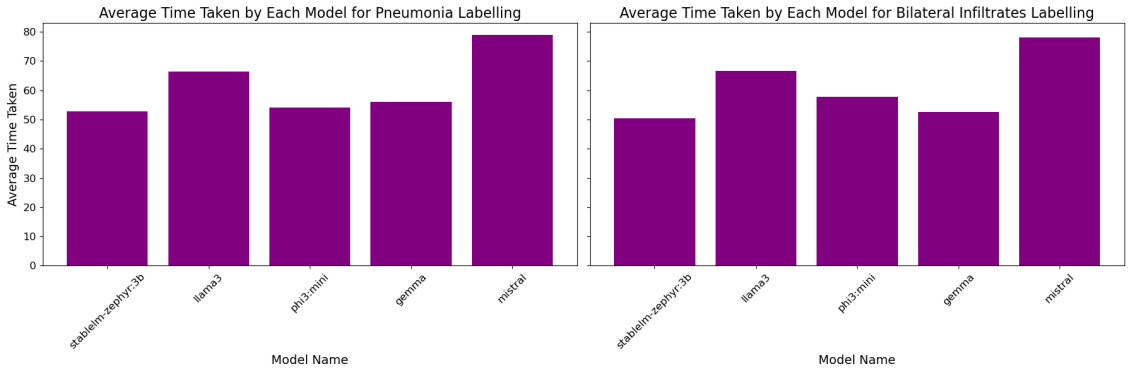

Figure 6.2: Average time taken to produce clinical labels for the 5 LLMs under investigation.

labelling task. Two of the models, stablelm-zephyr:3b and gemma, did not detect the presence of bilateral infiltrates in any of the six trials. Since none of the radiology reports explicitly stated this condition, the models needed to exhibit a certain level of reasoning to identify it. Llama3, phi3:mini, and mistral displayed better accuracy with the task, with llama3 demonstrating the most impressive reasoning behind its labelling decisions.

The average time taken to complete the two clinical labelling tasks was also plotted in Figure 6.2. Although the two experiments used different texts (discharge summaries vs. radiology reports) for different sets of patients and had a different number of trials, the trend for the average time taken by the models was quite consistent.

Having completed these investigations, the clear winner was **llama3**. It exhibited strong medical reasoning across all three tasks and demonstrated the best combination of accuracy and speed.

6.2.2 Scaling Up and Further Investigations

After selecting llama3 for concept labelling, the next task was to scale up the labelling process for a cohort of 2021 patients. Previous investigations provided an indication of the time required to process clinical text locally. This was formalised by running the pneumonia prompt on 15 patients consecutively to visualise the trend (the responses were also validated and found to be correct in all 15 cases). There was a near direct correlation between the length of the text and the processing time, Figure 6.3. To improve performance, an attempt was made to reduce text size as suggested in [50] by removing multiple spaces, stop words and newlines, eliminating unnecessary punctuation, and trimming leading and trailing spaces. However, this approach resulted in minimal performance gains and altered one of the labels from correct to incorrect, so it was ultimately ruled out.

Configuration	**Avg Time Per Patient (secs)**	**For Full Cohort (hrs)**	**For Full Cohort with 8 Different Concepts (hrs)**
M1 MacBook without GPU	81.33	45.66	365.26
M1 MacBook with GPU	32.27	18.12	144.93
M1 MacBook with GPU and Pre-processing	30.07	16.88	135.05
L4 GPU on Colab Pro	3.87	2.17	17.38

Table 6.5: Performance comparison of different configurations.

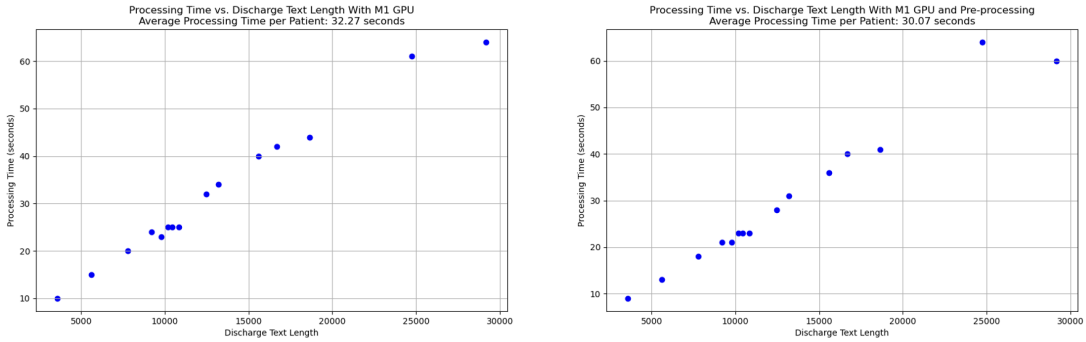

Figure 6.3: Time taken to process 15 discharge summaries with M1 GPU. Pre-processing the data (figure on right) showed a slight improvement in average processing time from 32.27 seconds to 30.07 seconds, but caused an answer to become incorrect.

During testing, some inconsistencies were noticed in the processing time: occasionally, the average time to process each discharge summary nearly tripled. This discrepancy altered the estimated total processing time from around 30 seconds to 80 seconds, Figure 6.4. The code was initially run on an M1 MacBook Pro, and further inspection of the device's activity monitor showed that when the device was not being charged, its 8-core M1 GPU was not utilised, resulting in a 2.6x increase in execution time. This made a significant impact on the overall time required to process the clinical text for the full cohort of patients for 8 different clinical labels, Table 6.5. Consequently, the logical next step was to run the code on a more powerful GPU. The transition was made to Google Colab Pro, which provided access to more powerful GPUs, including the L4 GPU.

The L4 GPU is designed for high-performance computing tasks, offering advanced capabilities in deep learning and AI inference. It features Tensor Cores and multi-instance GPU technology, which significantly accelerate AI and machine learning workloads. This makes it ideal for processing large datasets and complex models efficiently. The Colab Pro account also allowed access to a command line terminal, which was used to quickly install Ollama on the instance, allowing the same code that was run locally to be executed on a more powerful machine remotely. Additionally, running

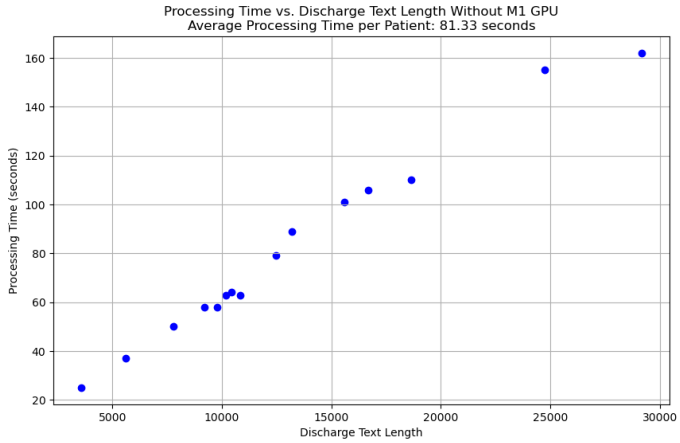

Figure 6.4: Time taken to process 15 discharge summaries without M1 GPU.

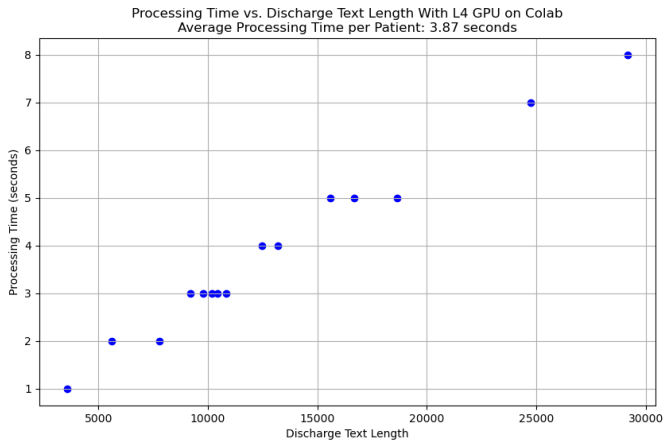

Figure 6.5: Time taken to process 15 discharge summaries with L4 GPU.

the code with a Colab Pro account allowed access to two machines simultaneously, which further halved the overall processing time. As a result, labelling all 8 concepts could be completed within 2 working days.

Having scaled up the processing time to be sustainable for a cohort of 2021 patients, further investigations were conducted to identify the optimal combination of parameters for the LLM. Since llama3 was chosen, the generic prompt template was replaced with the specific template from llama3's documentation. However, this did not impact processing time or model output. Additionally, varying the chunk size (1024, 2048, and 4096) and chunk overlap (100, 200, and 500) had minimal effect on the output. A larger chunk size resulted in faster processing times.

6.2.3 Final LLM Implementation

Following all these investigations, the final implementation of the LLM was achieved with this configuration:

- Model: Llama 3 (trained on 7 billion parameters)
- Chunk size: 4096 tokens
- Chunk overlap: 100 tokens

The prompts were similar to those used in earlier investigations but did not request the reasoning behind the responses. Instead, the model produced only a series of Yes and No answers, which were later processed to provide an overall verdict. If the series of responses contained a "Yes," the concept label was set to 1; otherwise, it was set to 0. The generic prompt was retained for its clarity and reusability, with slight modifications based on the specific labels.

For mentions of ARDS, aspiration, pneumonia, pancreatitis, cardiac arrest, and TRALI, the prompts were identical except for the condition being evaluated, and the input text was the discharge summary. For bilateral infiltrates, "suggest" was changed to "mention," and the input text was the radiology reports. For cardiac failure, ECG studies were provided to the LLM first. If the overall diagnosis of cardiac failure was false, the discharge summary was also provided to check for any mention of a history of cardiac failure, and the label was set accordingly.

6.2.4 Random Sampling Validation

In order to validate the accuracy of the LLM labelling on a larger scale, repeated random sampling validation was carried out. In this procedure, the same 100 patients were processed three times. Each time, 5 patients were selected at random and their corresponding clinical text was manually reviewed. So for each label, the LLM's response for 15 random patients (120 patients in total) were evaluated and the accuracy was recorded, see Table 6.6. Since the labelling was done by one reviewer, this was the scale of the investigation that could be carried out for this project's time frame, but for future work, a more thorough evaluation is required.

ARDS, aspiration, and pneumonia were often explicitly stated in the text so their average accuracy tended to be higher. One of the biggest difficulties with keyword extraction is the inconsistency in how different teams within a clinical setting refer to conditions. Llama3 showed some

Clinical Label	Average Accuracy (%)
Mention of ARDS	93.5
Aspiration	87.4
Bilateral Infiltrates	71.7
Cardiac Arrest	80.3
Cardiac Failure	81.2
Pancreatitis	74.6
Pneumonia	92.7
TRALI	96.3

Table 6.6: Average accuracy of clinical labels from random sampling validation.

promise in this area. For prompting, very little context was provided to llama3, but it understood acronyms like CHF for Congestive Heart Failure. It should also be noted that there were very few instances of TRALI in the data (which was expected), and it was also usually explicitly stated, so its accuracy is quite high. Pancreatitis and bilateral infiltrates were more difficult to discern. They were not explicitly stated as often, and some false negatives were obtained because the LLM had not done a good enough job at understanding the clinical context. Overall, however, the accuracy of the concept labelling from this evaluation was strong, and the concepts were ready for incorporation into the vanilla CBMs.

6.3 Augmenting Concept Set with LLM Labels

6.3.1 Initial Attempt

This study began with the two best-performing vanilla ARDS identification concept bottleneck models: the joint bottleneck model with early stopping and the one with L2 regularisation. The eight additional LLM concepts were added to the existing concept set, and the evaluation metrics for both the concept predictor and label predictor were recorded. Comparing the values in Tables 5.4 and 6.8, it is evident that for the Joint w/ Early Stopping model, the vanilla CBM shows slightly higher accuracy and precision, while the enhanced CBM shows slightly higher recall (Figure B.9). For the Joint w/ L2 Regularisation model, the enhanced CBM shows minimal improvements across all metrics compared to the vanilla CBM.

These results indicate that the additional concepts made little to no improvement for these models. Closer inspection revealed that the concept predictors for the enhanced CBMs (Table

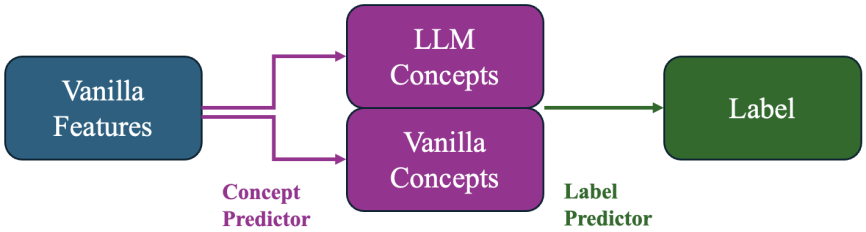

Figure 6.6: Initial architecture for augmenting concept set.

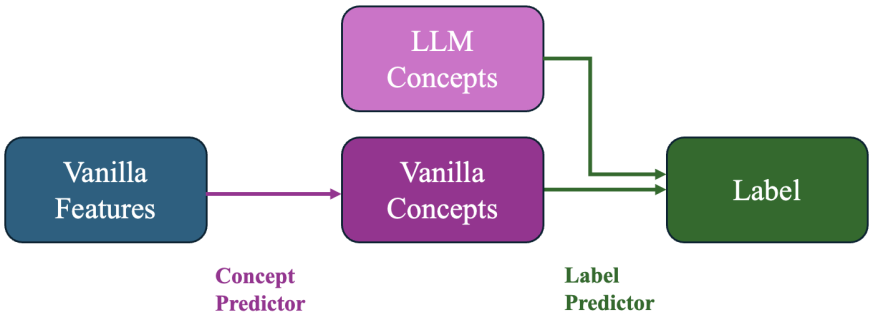

Figure 6.7: Altered architecture for augmenting concept set.

6.7) performed worse than those for the vanilla CBMs (Table 5.3). This suggests that the concept predictor struggled to predict the new LLM concepts, which negatively impacted the overall label predictor performance. Observing the metrics for the Independent w/ Early Stopping model confirmed this: even though the concept predictor was worse for the enhanced model compared to the vanilla model, the label predictor showed slight improvements with the additional concepts (Figure B.10). The independent model demonstrated that the LLM concepts do improve the model's ability to diagnose ARDS because, in an independent model, the label predictor is trained on the ground truth concepts, so it is not limited by a weak concept predictor.

The concept predictor is weaker with additional LLM concepts because the same vanilla features are used to predict these new LLM concepts. However, the lab measurements and vital signs were chosen to predict the vanilla concepts, and these trials confirmed that those features were not effective at predicting the LLM concepts. Hence, a new approach was needed where the concept predictor is trained only to predict the vanilla concepts, but the label predictor is trained on both the vanilla concepts and the LLM concepts. The distinction between the architectures is illustrated in Figures 6.6 and 6.7.

Concept Predictor's Model	Accuracy	AUC	Precision	Recall	F1 Score
Independent w/ Early Stopping	0.90	0.75	0.74	0.69	0.69
Joint w/ Early Stopping	0.84	0.58	0.63	0.56	0.56
Joint w/ L2 Regularisation	0.88	0.73	0.69	0.66	0.66
Independent w/ Altered Architecture	0.47	0.48	0.40	0.43	0.28
Sequential w/ Altered Architecture	0.64	0.50	0.23	0.37	0.26

Table 6.7: Averaged evaluation metrics for Enhanced ARDS CBM concept predictors.

Label Predictor's Model	Accuracy	AUC	Precision	Recall	F1 Score
Independent w/ Early Stopping	0.6535	0.66	0.60	0.70	0.65
Joint w/ Early Stopping	0.6757	0.67	0.64	0.66	0.65
Joint w/ L2 Regularisation	0.6784	0.68	0.67	0.73	0.70
Independent w/ Altered Architecture	0.7234	0.72	0.71	0.79	0.75
Sequential w/ Altered Architecture	0.7209	0.72	0.71	0.78	0.74

Table 6.8: Evaluation metrics for Enhanced ARDS CBM label predictors.

6.3.2 Using Altered Architecture

To implement the architecture depicted in Figure 6.7, the approach began with the code for the independent and sequential concept bottleneck model. These models were designed in such a way that the same model class was instantiated twice: once as a concept predictor and once as a label predictor. These two predictors were subsequently trained according to the conventions of independent and sequential training in the training loop. Given this implementation style, it was straightforward to instantiate two different models for the concept predictor and label predictor. In this case, instead of providing only vanilla concepts to the label predictor, the vanilla concepts were concatenated with the LLM concepts. The augmented concept set was then fed into the label predictor. This was achieved by creating a higher-level neural network class in the code. This class utilised instances of the lower-level predictor class, and trained both predictors with a combined loss. This approach does not strictly follow the conventions of independent and sequential concept bottleneck models, where two separate loss functions are used to train the concept predictor and label predictor. Consequently, this new 'altered' model is referred to as 'Independent w/ Altered Architecture' and 'Sequential w/ Altered Architecture' in figures and tables. The training algorithms for the altered independent and altered sequential models are

provided in Algorithms 4 and 5, respectively.

The altered architecture models were based off of vanilla L2 regularised models, and have been compared against them in Figures B.11 and B.12. For the concept predictors, the vanilla models show strong and consistent performance across both independent and sequential configurations. However, the altered architecture significantly reduces performance, especially in independent models. In contrast, for the label predictors, the vanilla models exhibit moderate performance, with independent models being slightly better. The altered architecture improves performance for both configurations, with independent models again leading slightly.

These results are interesting because, in the initial attempt, the models struggled to predict the new LLM concepts using the old vanilla features. In the new models, the mapping from vanilla features to vanilla concepts might be getting disregarded. One theory to support this is that the new altered architecture uses only one loss function. Unlike conventional sequential and joint models, which train both the concept predictor and label predictor together for the final label prediction, the single loss function may allow the model to focus solely on training the label prediction part. This theory was further investigated by examining regression models that use the new LLM concepts.

Comparing the evaluation of regression models with vanilla concepts in Table 5.5 with the regression models that incorporated the new LLM concepts in Table 6.9 demonstrates that the models with LLM concepts significantly outperform those with only vanilla concepts. The model with just LLM concepts shows higher accuracy (0.7185 vs. 0.60), AUC (0.79 vs. 0.66), precision (0.71 vs. 0.62), recall (0.78 vs. 0.56), and F1 score (0.74 vs. 0.59) compared to the C to Y model. These results indicate that the insights captured from the LLM are far better at diagnosing patients with ARDS as compared to the vanilla concepts constructed from structured data. Additionally, the model combining vanilla and LLM concepts performs even better, with further improvements in accuracy (0.7333), AUC (0.80), precision (0.72), and F1 score (0.75).

Model	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1 Score**
LLM-C to Y	0.7185	0.79	0.71	0.78	0.74
(C + LLM-C) to Y	0.7333	0.80	0.72	0.78	0.75

Table 6.9: Logistic regression metrics for ARDS prediction using LLM concepts. Model (LLM-C to Y) uses only LLM concepts as inputs, while model (C + LLM-C to Y) combines LLM concepts with vanilla concepts.

This information suggests that the altered concept bottleneck models leverage the strength of the LLM concepts. While this achieves the project's aim of improving accuracy with additional

insights from clinical text, it does so without properly implementing the concept predictor part of a concept bottleneck model. It is also unclear whether the concatenation of both sets of concepts truly benefits the model. The superior performance of the altered independent model compared to the altered sequential model could be attributed to the concept predictor in the sequential model not predicting vanilla concepts properly, thus failing to incorporate their insights into the final prediction. In contrast, the independent model trains the predictors separately, ensuring the vanilla concepts are utilised as well, which may explain its higher accuracy. These results open up more questions, which remain unanswered due to time constraints and are addressed in Section 6.4.

6.3.3 Investigating Completeness Score

A completeness score, defined in [9], is a measure which helps determine how well a set of concepts explains the final label in a CBM. It is defined as the ratio of the mutual information between concepts and labels, $I(y; c)$, to the mutual information between the input and label conditioned on the concepts, $I(y; c, x)$:

$$C = \frac{I(y; c)}{I(y; c, x)} \tag{6.1}$$

In simpler terms:

- $I(y; c)$ is the amount of information the concepts provide about the final label.
- $I(y; c, x)$ is the total amount of information available in the input about the final label, upon considering the concepts and any extra information.

A completeness score of 1 means the concepts capture all the information from the input for the final prediction while a score of 0 means the concept do not provide any useful information for the final prediction. A lower completeness score increases the likelihood of leakage (Section 3.3).

The completeness scores for both the vanilla and enhanced CBM models were calculated in Table 6.10, revealing significant differences. The enhanced models, particularly the altered independent and altered sequential CBMs, exhibited higher $I(y; c)$ values compared to their vanilla counterparts. For example, the altered sequential model achieved an $I(y; c)$ of 0.1364, markedly higher than any values seen in the vanilla models. Additionally, the altered sequential model showed a much lower $I(y; c, x)$ value (0.1864) compared to others. This suggests that the altered architecture significantly reduces the dependency on raw input once the concepts are included.

Overall, the enhanced altered architecture models demonstrated a higher completeness score, with the sequential model displaying a notable increase from 0.025 to 0.73. This improvement indicates that the models benefited from an augmented concept set, allowing the label predictor to effectively train on the additional LLM concepts.

Model Type	**Configuration**	**Completeness**	**I(y; c)**	**I(y; c, x)**
Vanilla	Joint w/ Early Stopping	0.057	0.040	0.69
	Joint w/ L2 Regularisation	0.075	0.052	0.684
	Independent w/ L2 Regularisation	0.026	0.017	0.661
	Sequential w/ L2 Regularisation	0.025	0.017	0.682
Enhanced	Joint w/ Early Stopping	0.059	0.041	0.69
	Joint w/ L2 Regularisation	0.075	0.052	0.68
	Independent w/ Altered Architecture	0.18	0.13	0.68
	Sequential w/ Altered Architecture	0.73	0.1364	0.1864

Table 6.10: Model completeness and mutual information for vanilla and enhanced models.

The remarkable, nearly 30-fold increase in the completeness score for the altered sequential model inspired more investigations which were left for future work due to time constraints:

- Investigate the impact of augmenting the concept set with noisy concepts or concepts from a poorly performing LLM. If the completeness score remains high, it may indicate the model is finding spurious correlations in the data.
- Explore the features and their importance.
- Determine whether removing the most important features lowers the completeness score.
- Analyse the weights of the altered sequential model to understand the factors contributing to the significant increase in completeness and how it differs from the vanilla model.

6.4 Future Work

This chapter demonstrates how incorporating LLM-generated concepts into ARDS CBMs resulted in notable improvements. Nonetheless, it also identifies areas for further research to establish a robust concept labelling pipeline and produce enhanced CBMs.

Firstly, the concept labels chosen for the ARDS LLM were considered. The eight labels selected were clinically deemed the strongest contributors to ARDS. However, Dr Marshall used more than

these eight labels during his manual review. Some notable concepts not considered in this study included:

- From Discharge Summaries: Mention of the patient being hypoxaemic, having hypercarbic respiratory failure, and requiring proning.
- From ECG studies: Evidence of reduced ejection fraction, diastolic dysfunction, moderate-severe aortic stenosis/regurgitation, mitral regurgitation, and cardiomyopathy.

Future studies could explore if these additional concept labels improve the accuracy of ARDS diagnosis.

Next, the selection of LLMs for concept labelling is examined. The field of LLMs is constantly evolving, making it crucial to use cutting-edge models. There are extensive evaluation techniques for selecting the correct LLM, as well as the appropriate prompt template, structure, context provided, pre-processing of contextual data, etc. An in-depth evaluation was beyond the scope of this project. For future work, some initial steps could include:

- Calculating precision, recall, F1, specificity, and sensitivity, rather than just considering the accuracy of the prediction.
- Using a large, well-annotated dataset as a reliable ground truth for evaluation.
- Identifying common types of errors and having a clinician review the reasoning of the LLMs.
- Investigating fine-tuning at a prompt level, such as few-shot prompting or providing more context for tasks like Bilateral Infiltrates.

All the prompts for the clinical labels were handled separately, based on the intuition that LLMs would not perform well when looking for multiple conditions simultaneously. However, the same text (discharge summary) was reviewed for six different conditions. The processing time could be significantly reduced if it could label multiple concepts in the same inference, which merits a formal investigation.

The LLMs used were open-source models available for free from Ollama. Future work could consider paid models like GPT-4. Furthermore, the models used were general LLMs. Investigating models fine-tuned on medical data, such as the T5 models trained on MIMIC data, would be useful. Large-scale fine-tuned models like MedPalm2 or Gatortron [17][51], which have demonstrated

impressive reasoning on medical benchmark examinations, could also be considered. Additionally, fine-tuning the LLMs with the approaches outlined in Section 2.4 could be explored.

Next, the attempt at incorporating the labels into the enhanced CBMs is considered. As discussed in Section 6.3.2, the enhanced CBMs with altered architecture showed a worse-performing concept predictor but a stronger-performing label predictor. One theory is that these altered architectures use a single loss function for a combined neural network that incorporates both concept and label predictors. As a result, the neural network can choose which data to prioritise in training. Since the LLM concepts performed better than the vanilla concepts in the regression models, it may be capitalising on connections based on the LLM concepts and disregarding the mapping from vanilla features to vanilla concepts. This could be investigated further by implementing an architecture that trains the models with distinct loss functions for both concept and label predictors. The poorly performing concept predictor also suggests that the initial vanilla features and concepts were weak from the start. Revisiting the vanilla ARDS CBM and exploring potential performance improvements discussed in Section 5.5 would be worthwhile so that the strength of the LLM concepts does not overshadow the vanilla concepts significantly.

Finally, the remarkable increase in completeness scores for the altered architecture sequential model suggests a significant reduction in CBM leakage. As discussed in Section 6.3.3, further investigations are warranted. These could include examining the impact of adding noisy or poorly performing LLM concepts; if the completeness score remains high, this may indicate spurious correlations. Additionally, exploring feature importance by removing key features to see if the completeness score drops, as well as analysing the weights in the altered sequential model, could provide insights into the factors behind the increase in completeness compared to the vanilla model.

Conclusions

In this project, the capabilities of concept bottleneck models (CBMs) and large language models (LLMs) were utilised to produce interpretable clinical prediction models.

In **Chapter 4**, CBMs were developed to predict mortality for patients in the MIMIC database. The Mortality CBMs targeted a broad cohort of patients, which prompted the selection of a concise, informative concept set that captured the general characteristics of ICU admissions. Both classification and regression models were trained using three different CBM training methods. The classification models demonstrated better performance and were used for the remainder of the project.

To refine the classification CBMs and reduce overfitting, techniques such as early stopping and L2 regularisation were employed. The performance of these models was then compared against standard regression models, with CBMs showing comparable results. It was anticipated that CBMs might capture additional information not directly available in the raw input features, potentially leading to superior performance over regression models. However, analysis of the CBMs' concept and label predictor weights revealed that concept predictions often diverged from clinical reasoning. There was also variation between models in label prediction, and yet the combination of concept-to-label prediction still provided usefully overlapping final weight contributions.

It was hypothesised that the inconsistencies were due to spurious correlations in the concept prediction stage and leakage in sequential and joint models. The results underscored the need for more in-depth exploration. Finally, avenues for further investigation and improvement were identified in Section 4.5.

In **Chapter 5**, CBMs were developed to identify Acute Respiratory Distress Syndrome (ARDS), a more complex task due to the strict Berlin Criteria and the need for analysis of various medical sources for an accurate diagnosis. The patient cohort was selected by Dr Dominic Marshall, who was closely involved in this project's ARDS work. Dr Marshall meticulously screened patients showing initial signs of ARDS and manually reviewed extensive medical data for over 2000 patients to ensure robust diagnoses. This increased the challenge of the prediction task, as it required detecting nuances within a cohort already likely to suffer from ARDS.

Using Dr Marshall's recommendations, an initial concept set was created based on electronic health record (EHR) data. Subsequently, 'vanilla' ARDS CBMs were trained and fine-tuned using the same methods as the Mortality CBMs. These ARDS CBMs performed worse than the Mortality CBMs, which was expected, but they outperformed the corresponding regression models. This demonstrated the ARDS CBMs' superiority in both interpretability and predictive performance.

For thoroughness, the weights of these CBMs were analysed, revealing similar inconsistencies to the Mortality CBMs, indicating the presence of spurious correlations and leakage. The areas for further exploration were summarised in Section 5.5.

In **Chapter 6**, the ARDS CBMs developed in the previous chapter were enhanced with additional insights from clinical text using an LLM. Upon observing Dr Marshall's manual review process of discharge summaries, radiology reports, and ECG studies, additional concept labels were selected. These labels indicated whether the patient suffered from eight different conditions identifiable in the text but not explicitly found in the EHR data.

After an evaluation process, Meta's Llama3 model was chosen to review the clinical text and produce labels for each patient. This was executed on L4 GPUs to process text for a cohort of over 2000 patients, eight times for the different conditions. When these additional concepts were integrated directly into the vanilla CBMs, there was only marginal improvement in performance because the vanilla features predicted the new LLM concepts poorly.

This necessitated an alteration of the architecture: the concept predictor would only predict the vanilla concepts, while the label predictor would use the concatenation of the LLM concepts and vanilla concepts to make the label prediction. The altered architecture demonstrated a marked increase in accuracy. Further evaluation of the concept predictions revealed interesting insights, which were left for future exploration. Additionally, the completeness score was calculated to identify the impact of additional concepts on leakage. It was found that the possibility of leakage was dramatically reduced for the sequential model, raising new questions to be addressed in future work, as detailed in Section 6.4.

A

Ethical, Legal, and Safety Plan

The work done in this project involves using real medical data from patients, which is highly sensitive, so complying with all safety guidelines to work with the dataset was a top priority. To obtain the credentials for accessing MIMIC, the "CITI Data or Specimens Only Research" course was completed (A.1). This course covers important aspects of research with human participant data including ethical principles, privacy, and HIPAA (Health Insurance Portability and Accountability Act) compliance.

CITI PROGRAM

Completion Date 15-Nov-2023
Expiration Date 15-Nov-2026
Record ID 59694541

This is to certify that:

Anish Narain

Has completed the following CITI Program course:

Not valid for renewal of certification through CME.

Human Research
(Curriculum Group)
Data or Specimens Only Research
(Course Learner Group)
1 - Basic Course
(Stage)

Under requirements set by:

Massachusetts Institute of Technology Affiliates

CITI
Collaborative Institutional Training Initiative
101 NE 3rd Avenue, Suite 320
Fort Lauderdale, FL 33301 US
www.citiprogram.org

Generated on 26-Apr-2024. Verify at www.citiprogram.org/verify/?w400213a4-c77e-42c1-b0b1-f4195723bc5d-59694541

Figure A.1: Completion certificate for Data or Specimens Only Research Course.

Furthermore, the terms of use for the MIMIC dataset [52] were diligently observed. This includes exercising care to prevent disclosure of identities in any communication, not sharing access

with others, using the data solely for lawful scientific research, contributing code if publishing results, and acknowledging that these obligations persist even after termination of the agreement.

Finally, research has been shown that large language models have the potential to leak sensitive information [53]. Hence, following the example from [29], any models released to the public will be done so under PhysioNet credentialed access.

B

Additional CBM Information

B.1 CBM Training Algorithms

Algorithm 1 Independent CBM Training

Require: Training data (X, C, Y), learning rates η_g, η_f, number of epochs E
Ensure: Trained concept predictor $\hat{g}$ and label predictor $\hat{f}$

```
Initialise concept predictor ĝ and label predictor f̂ (Using the same neural network class)
Initialise Adam optimiser for ĝ with learning rate η_g
Initialise Adam optimiser for f̂ with learning rate η_f
Define loss function L (BCE Loss for Classification and MSE Loss for Regression)

Training ĝ (Concept Predictor)
for epoch = 1 to E do
    for each mini-batch (x, c) in training data (X, C) do
        Forward pass: Compute predicted concepts ĉ = ĝ(x)
        Compute loss: L_C = Σ_{i,j} L(g_j(x^(i)), c_j^(i))
        Backward pass: Compute gradients of L_C with respect to ĝ's parameters
        Update parameters: Update ĝ's parameters using the optimiser
    end for
end for

Training f̂ (Label Predictor)
for epoch = 1 to E do
    for each mini-batch (c, y) in training data (C, Y) do
        Forward pass: Compute predicted labels ŷ = f̂(c)
        Compute loss: L_Y = Σ_i L(f(c^(i)), y^(i))
        Backward pass: Compute gradients of L_Y with respect to f̂'s parameters
        Update parameters: Update f̂'s parameters using the optimiser
    end for
end for
return ĝ, f̂
```

Algorithm 2 Sequential CBM Training

Require: Training data (X, C, Y), learning rates η_g, η_f, number of epochs E
Ensure: Trained concept predictor $\hat{g}$ and label predictor $\hat{f}$

```
Initialise concept predictor ĝ and label predictor f̂ (Using the same neural network class)
Initialise Adam optimiser for ĝ with learning rate η_g
Initialise Adam optimiser for f̂ with learning rate η_f
Define loss function L (BCE Loss for Classification and MSE Loss for Regression)

Training ĝ (Concept Predictor)
for epoch = 1 to E do
    for each mini-batch (x, c) in training data (X, C) do
        Forward pass: Compute predicted concepts ĉ = ĝ(x)
        Compute loss: L_C = Σ_{i,j} L(g_j(x^(i)), c_j^(i))
        Backward pass: Compute gradients of L_C with respect to ĝ's parameters
        Update parameters: Update ĝ's parameters using the optimiser
    end for
end for

Training f̂ (Label Predictor)
for epoch = 1 to E do
    for each mini-batch (x, y) in training data (X, Y) do
        Forward pass: Compute predicted concepts ĉ = ĝ(x) (detach gradients)
        Compute predicted labels: ŷ = f̂(ĉ)
        Compute loss: L_Y = Σ_i L(f(ĉ^(i)), y^(i))
        Backward pass: Compute gradients of L_Y with respect to f̂'s parameters
        Update parameters: Update f̂'s parameters using the optimiser
    end for
end for
return ĝ, f̂
```

Algorithm 3 Joint CBM Training

Require: Training data (X, C, Y), learning rate η, number of epochs E, weight λ
Ensure: Trained joint model $(\hat{g}, \hat{f})$

```
Initialise joint model (ĝ, f̂)
Initialise Adam optimiser for joint model with learning rate η
Define loss function L (BCE Loss for Classification and MSE Loss for Regression)

Training Joint Model
for epoch = 1 to E do
    for each mini-batch (x, c, y) in training data (X, C, Y) do
        Forward pass: Compute predicted concepts ĉ = ĝ(x)
        Compute predicted labels: ŷ = f̂(ĉ)
        Compute concept loss: L_C = Σ_{i,j} L(g_j(x^(i)), c_j^(i))
        Compute label loss: L_Y = Σ_i L(f(ĉ^(i)), y^(i))
        Compute total loss: L_total = L_C + λL_Y
        Backward pass: Compute gradients of L_total with respect to joint model's parameters
        Update parameters: Update joint model's parameters using the optimiser
    end for
end for
return joint model (ĝ, f̂)
```

Algorithm 4 Altered Independent CBM Training

Require: Training data (X, v_c, l_c, Y), where X represents vanilla features, v_c represents vanilla concepts, l_c represents LLM concepts, and Y represents label. Learning rate η, number of epochs E, weight decay λ

Ensure: Trained combined model $\hat{m}$

Initialise combined model $\hat{m}$ with parameters $(\hat{g}, \hat{f})$
Initialise Adam optimiser for $\hat{m}$ with learning rate η and weight decay λ
Define loss function $\mathcal{L}$ (BCELoss for multi-label classification)

Training the Combined Model
for epoch = 1 to E **do**
    Set model to training mode
    Initialise running loss to 0
    for each mini-batch (x, v_c, l_c, y) in training data **do**
        Forward pass:
            Compute predicted vanilla concepts $\hat{v_c} = \hat{g}(x)$
            Concatenate v_c with l_c to form combined concepts
            Compute predicted labels $\hat{y} = \hat{f}$(combined concepts)
        Compute loss: $\mathcal{L}_Y = \mathcal{L}(\hat{y}, y)$
        Backward pass: Compute gradients of $\mathcal{L}_Y$ with respect to $\hat{m}$'s parameters
        Update parameters: Update $\hat{m}$'s parameters using the optimiser
        Accumulate running loss
    end for
end for
return $\hat{m}$

Algorithm 5 Altered Sequential CBM Training

Require: Training data (X, v_c, l_c, Y), where X represents vanilla features, v_c represents vanilla concepts, l_c represents LLM concepts, and Y represents label. Learning rate η, number of epochs E, weight decay λ

Ensure: Trained combined model $\hat{m}$

Initialise combined model $\hat{m}$ with parameters $(\hat{g}, \hat{f})$
Initialise Adam optimiser for $\hat{m}$ with learning rate η and weight decay λ
Define loss function $\mathcal{L}$ (BCELoss for multi-label classification)

Training the Combined Model
for epoch = 1 to E **do**
    Set model to training mode
    Initialise running loss to 0
    for each mini-batch (x, v_c, l_c, y) in training data **do**
        Forward pass:
            Compute predicted vanilla concepts $\hat{v_c} = \hat{g}(x)$
            Concatenate $\hat{v_c}$ with l_c to form combined concepts
            Compute predicted labels $\hat{y} = \hat{f}$(combined concepts)
        Compute loss: $\mathcal{L}_Y = \mathcal{L}(\hat{y}, y)$
        Backward pass: Compute gradients of $\mathcal{L}_Y$ with respect to $\hat{m}$'s parameters
        Update parameters: Update $\hat{m}$'s parameters using the optimiser
        Accumulate running loss
    end for
end for
return $\hat{m}$

B.2 Mortality Prediction CBM

B.2.1 Full List of Features and Concepts

Ultimately, 15 features and 23 concepts were identified for the Mortality CBMs, Tables B.1 and B.2.

Category	**Features**
Respiratory SOFA component	PaO2/FiO2 ratio for ventilated and non-ventilated patients
Liver SOFA component	Max bilirubin
Cardiovascular SOFA component	Mean Blood Pressure and Dopamine, Dobutamine, Epinephrine and Norepinephrine rate
Coagulation SOFA component	Platelet count
Central Nervous System SOFA component	Glasgow Coma Scale score
Renal system SOFA component	Urine output and creatinine level
Additional lab measurements	Anion gap, albumin and total bilirubin

Table B.1: Table of 15 features for Mortality CBM.

Category	**Concepts**
Moderate/severe SOFA components	Moderate/serious for respiratory failure, liver failure, cardiovascular failure, coagulation failure, Glasgow Coma Scale failure, renal failure
Concepts from potential conditions based on SOFA scores	Septic Shock, ARDS, Hepatorenal Syndrome, Coagulopathy-Related Organ Dysfunction, Multiple Organ Dysfunction Syndrome, CNS and Renal Failure, Liver and Coagulation Failure
Concept based on additional lab measurements	High anion gap, low albumin, high albumin, high total bilirubin

Table B.2: Table of 23 concepts for Mortality CBM.

B.2.2 Architecture

Figures B.1 and B.2 were constructed using the 'torchviz' library for PyTorch which helps visualise neural networks. Figure B.1 is the visualisation of the Joint Classifier model which contains the concept predictor and label predictor. The neural network is designed to predict mortality based

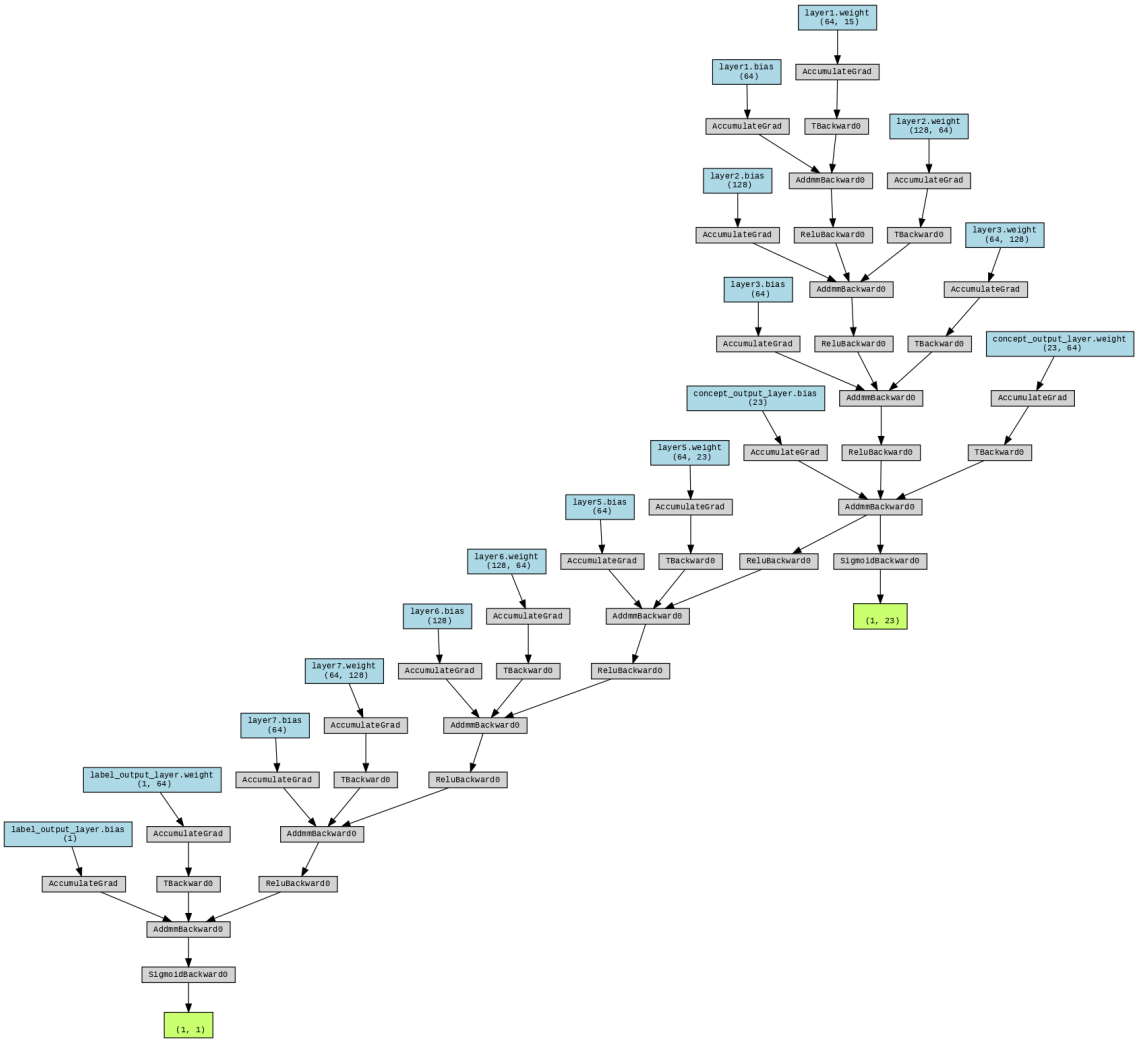

Figure B.1: Neural network visualisation for joint CBM before change in architecture. Has 3 hidden layers and an output layer for the concept predictor and then the same for the label predictor. The layer sizes are 64 and 128 units. The green boxes signify the concept predictor output (there are 23 concepts) and label predictor output.

on 15 input features. The input layer, with a shape of (1, 15), feeds into the concept prediction pathway comprising three hidden layers (Layer1, Layer2, Layer3) with intermediate shapes (1, 64), (1, 128), and (1, 64) respectively, using ReLU activations. The concept_output_layer generates 23 concept predictions, processed with a sigmoid activation. These concepts serve as input for the label prediction pathway, which includes three additional hidden layers (Layer5, Layer6, Layer7) with the same intermediate shapes, followed by the label_output_layer that predicts the mortality label as a probability using a sigmoid activation. Each layer's parameters (weights and biases) are depicted, showing how data flows and transforms through the network to achieve the final prediction.

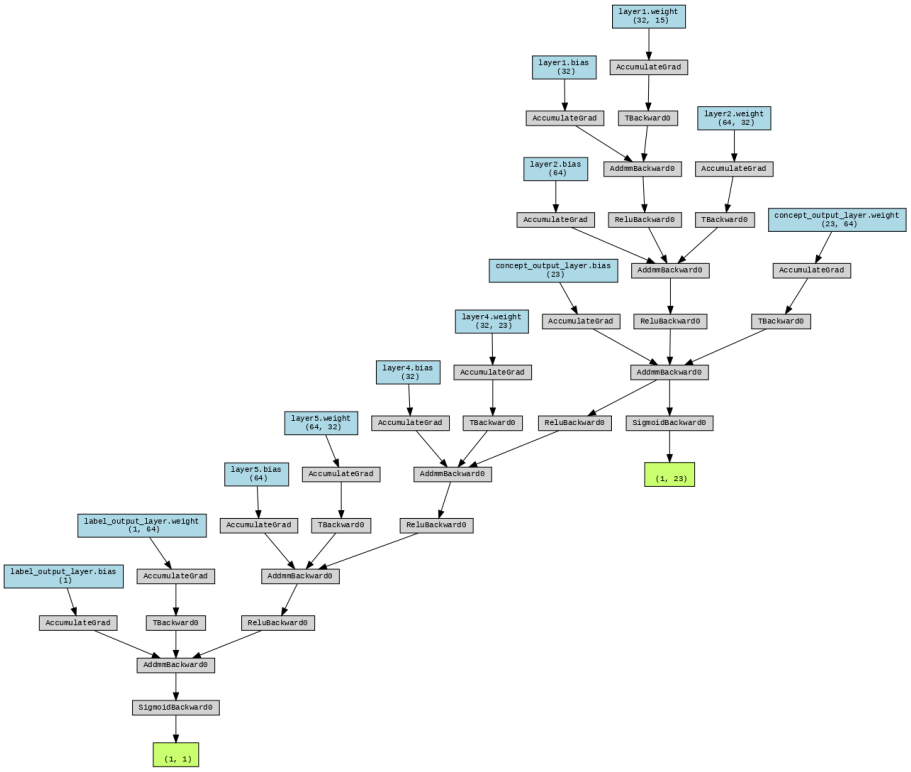

Figure B.2: Neural network visualisation for joint CBM after change in architecture. Has one fewer hidden layer for the concept and label predictor. The layer sizes are now 32 and 64 units.

B.3 ARDS Identification CBM

B.3.1 Full List of Features and Concepts for Vanilla CBM

Ultimately, 26 features and 19 concepts were chosen for the ARDS CBMs, Table B.3.

B.3.2 Vanilla CBM Loss Plots

B.3.3 Vanilla CBM Heat Maps

B.3.4 Augmented Concept Set Comparison

Category	Features	Concepts
Renal failure	1. Average urine output (24hr) 2. Average creatinine maximum 3. Maximum renal urine output (24hr) 4. Maximum renal creatinine maximum	1. Moderate renal failure (max) 2. Severe renal failure (max) 3. Moderate renal failure (avg) 4. Severe renal failure (avg)
Respiratory failure	1. Average PaO2/FiO2 ratio (no vent) 2. Average PaO2/FiO2 ratio (vent) 3. Maximum respiration PaO2/FiO2 ratio (no vent) 4. Maximum respiration PaO2/FiO2 ratio (vent)	1. Moderate respiratory failure (max) 2. Severe respiratory failure (max) 3. Moderate respiratory failure (avg) 4. Severe respiratory failure (avg)
Neurological failure	1. Average Glasgow Coma Scale (min) 2. Maximum CNS Glasgow Coma Scale (min)	1. Moderate CNS failure (max) 2. Severe CNS failure (max) 3. Moderate CNS failure (avg) 4. Severe CNS failure (avg)
Sepsis	1. SOFA score 2. Suspected infection	Sepsis
Shock	1. Single vasopressor 2. Multiple vasopressors	Shock
Ventilation status	1. Invasive Ventilation 2. Supplemental Oxygen 3. High-Flow Nasal Cannula 4. Non-Invasive Ventilation 5. Tracheostomy 6. None	1. Low ventilation 2. Moderate ventilation 3. High ventilation
Pre-existing respiratory conditions	1. Upper respiratory infections 2. Influenza pneumonia 3. Acute lower respiratory infections 4. Chronic lower respiratory diseases 5. Lung diseases due to external agents 6. Other respiratory diseases	1. Moderate respiratory condition 2. Severe respiratory condition

Table B.3: ARDS features and concepts table.

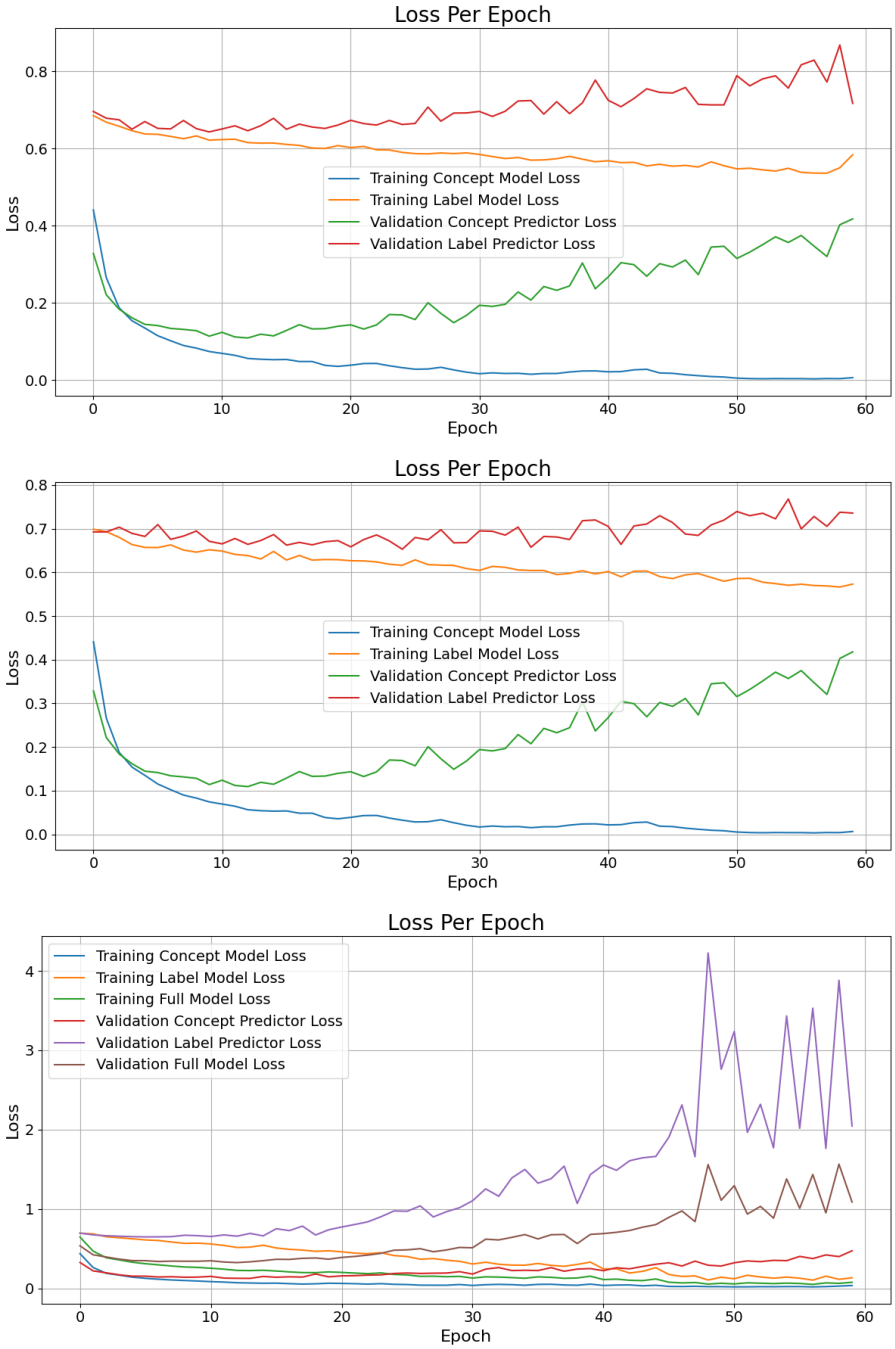

Figure B.3: Loss plots for independent (top), sequential (middle), and joint (bottom) ARDS CBMs with no fine-tuning.

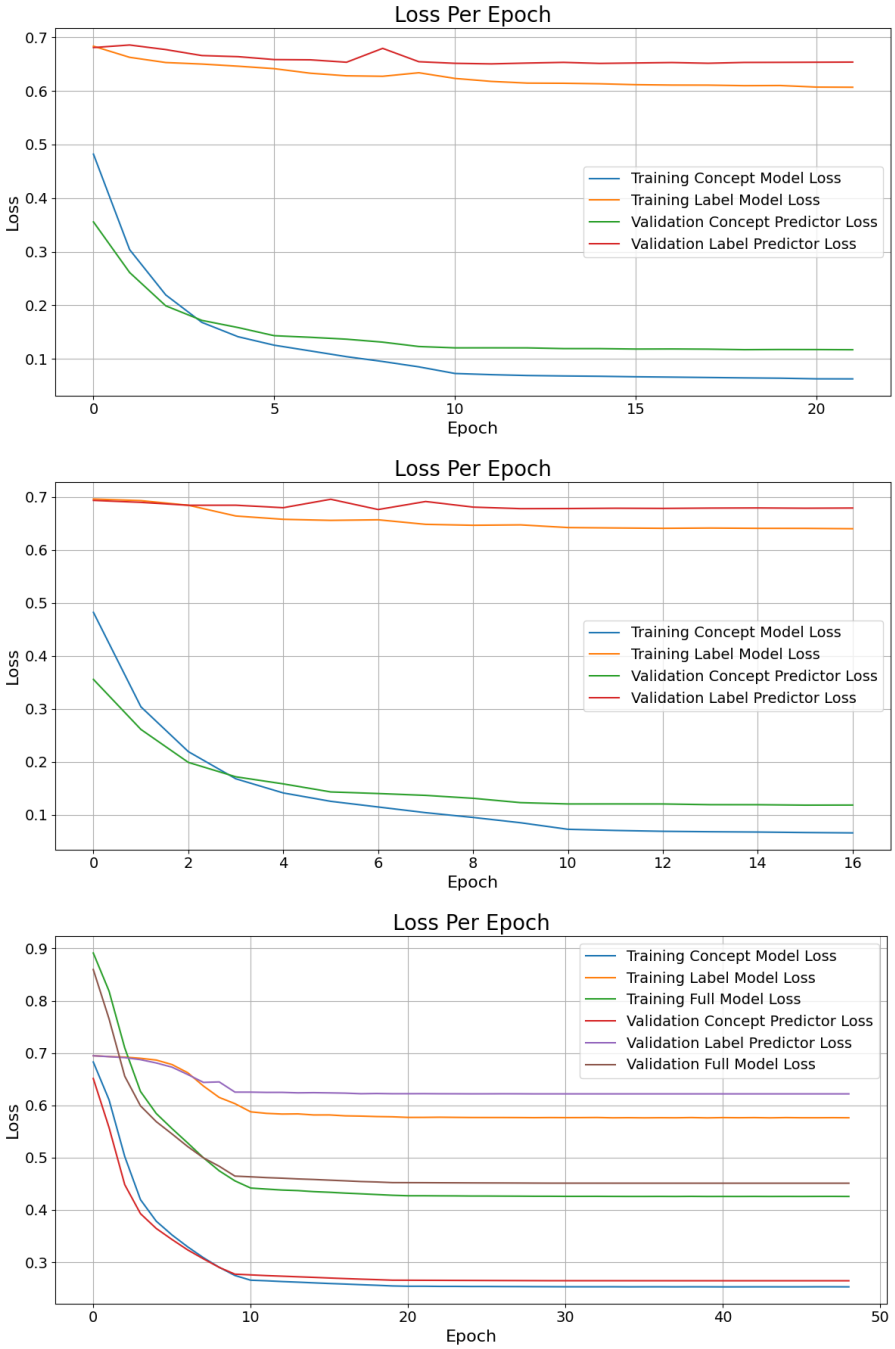

Figure B.4: Loss plots for independent (top), sequential (middle), and joint (bottom) ARDS CBMs with updated architecture and early stopping.

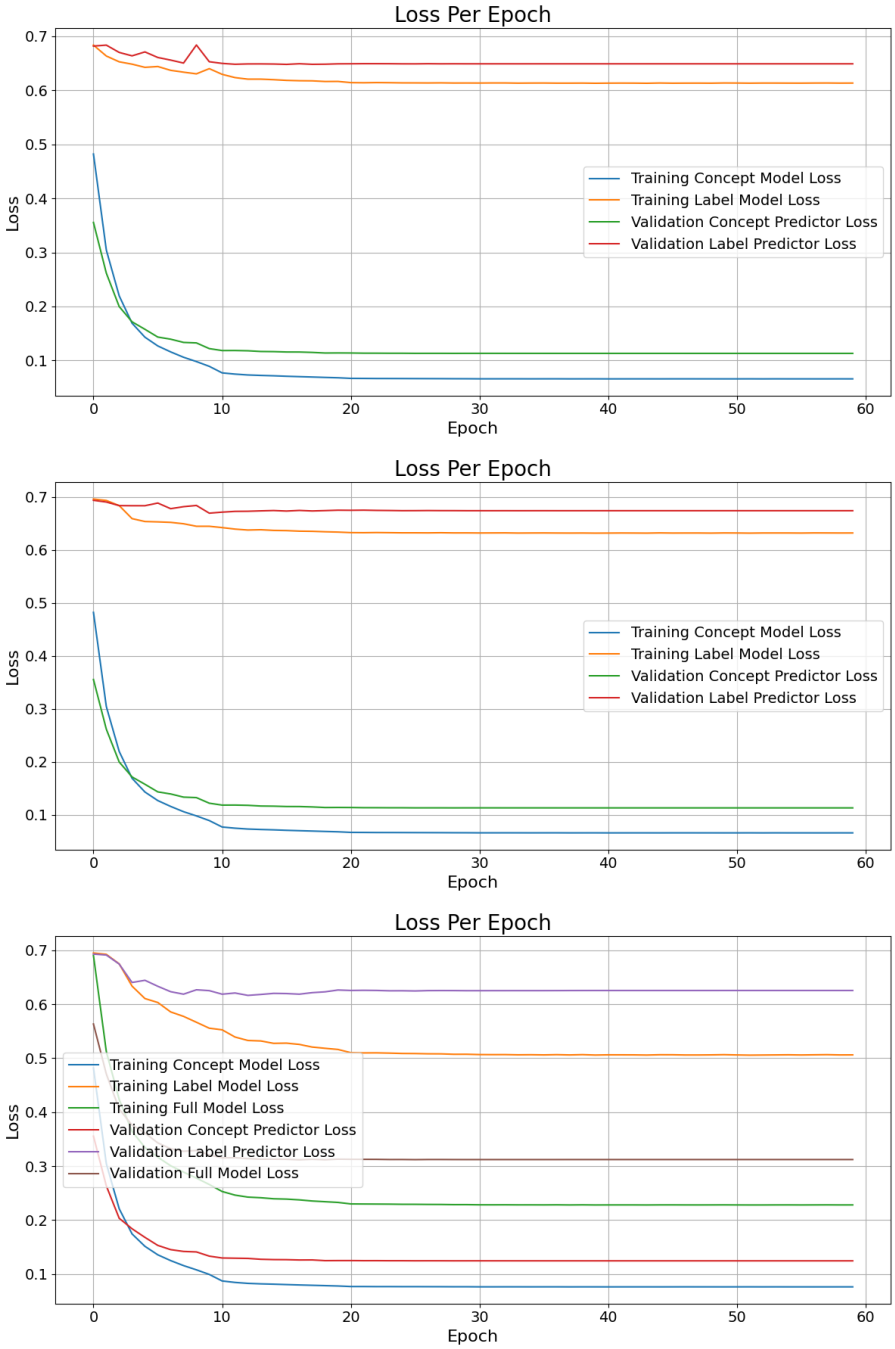

Figure B.5: Loss plots for independent (top), sequential (middle), and joint (bottom) ARDS CBMs with updated architecture and L2 regularisation.

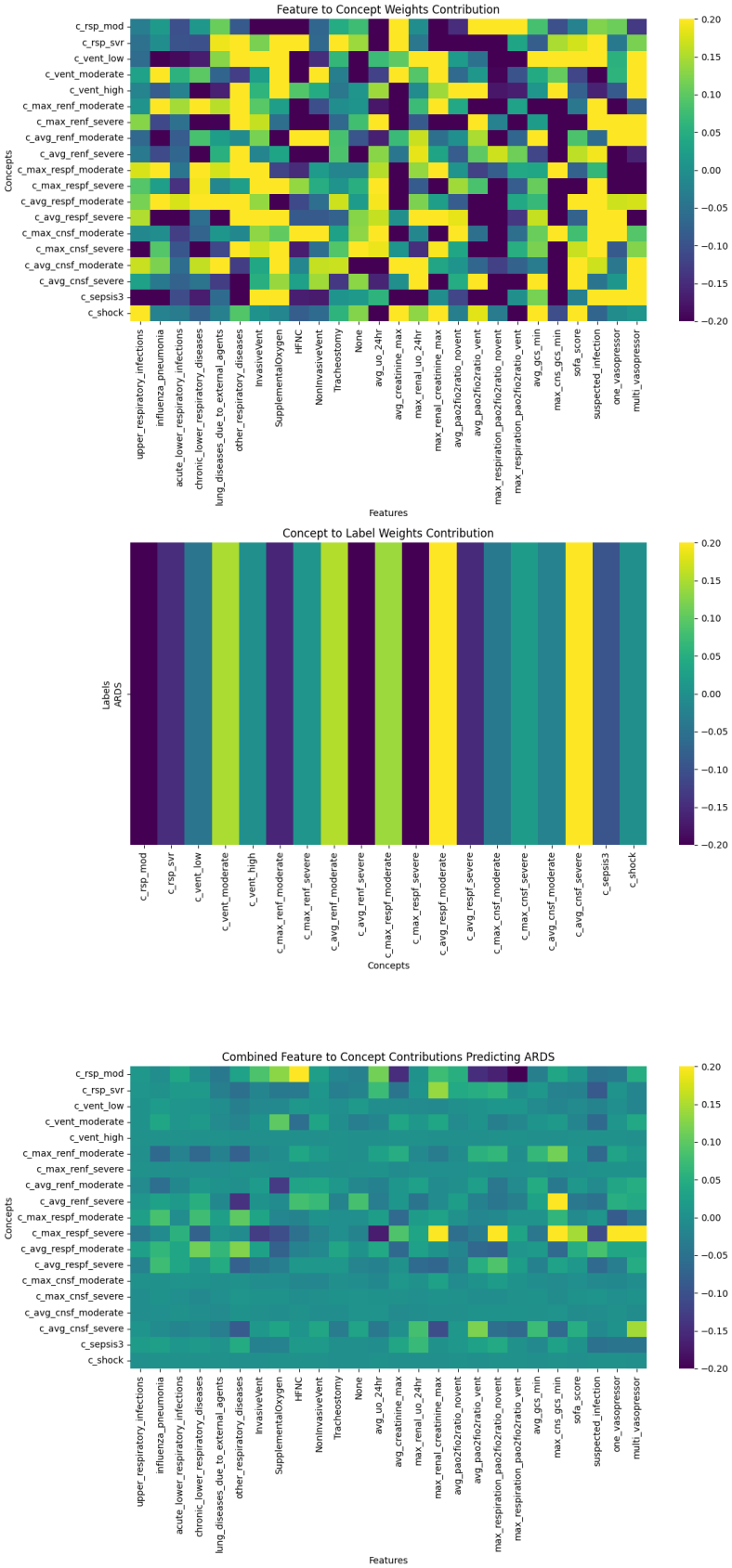

Figure B.6: Heat maps for feature and concept weights of the L2 regularised, independent concept bottleneck model.

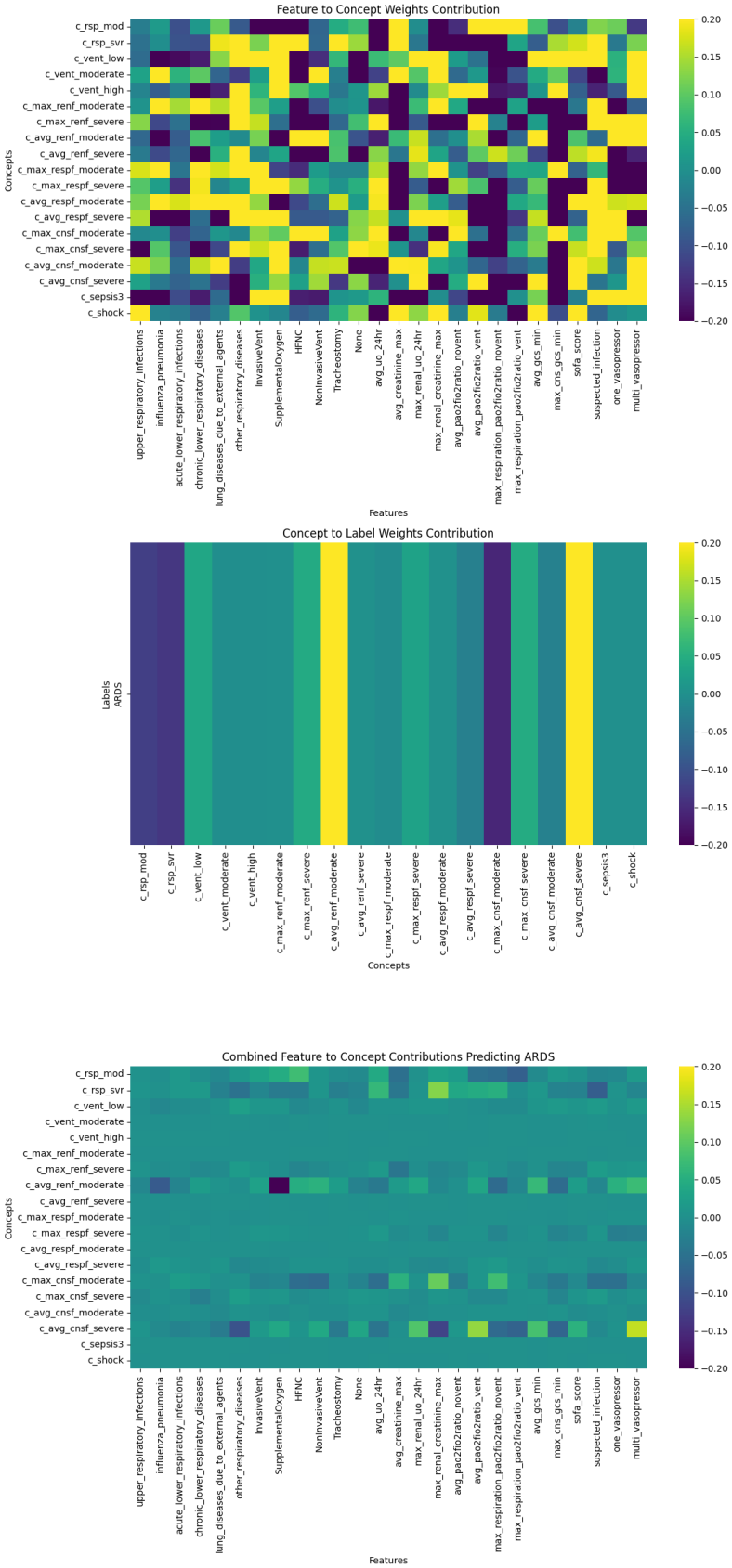

Figure B.7: Heat maps for feature and concept weights of the L2 regularised, sequential concept bottleneck model.

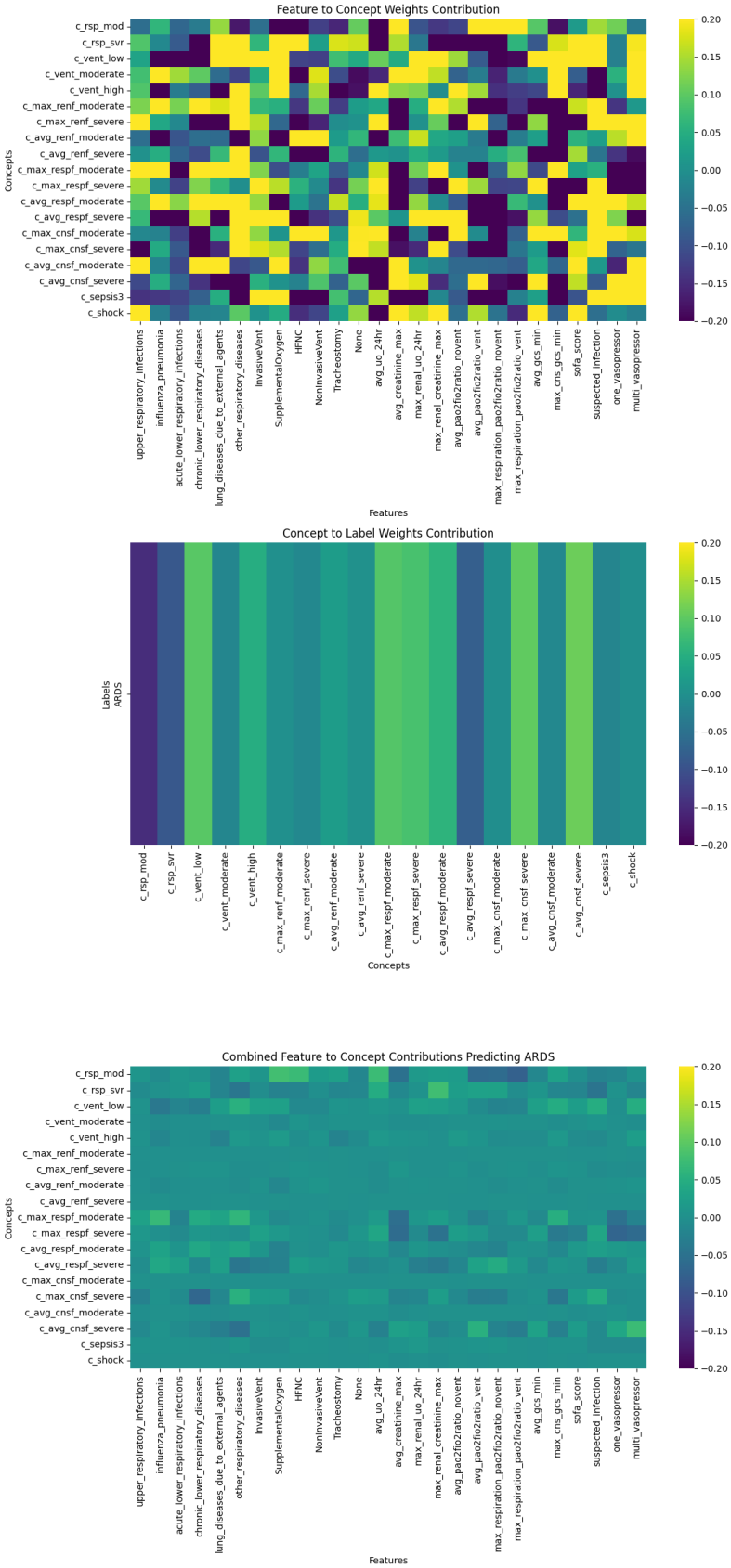

Figure B.8: Heat maps for feature and concept weights of the L2 regularised, joint concept bottleneck model.

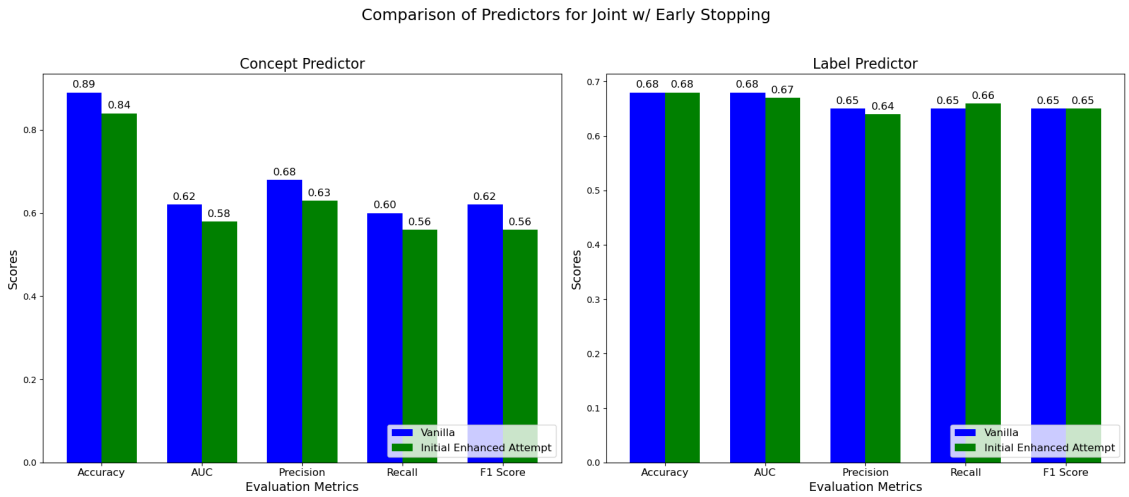

Figure B.9: Comparing the evaluation metrics for the predictors of Joint w/ Early Stopping CBMs. The graph contains side-by-side values for the original CBM and enhanced CBM (after its concept set was augmented).

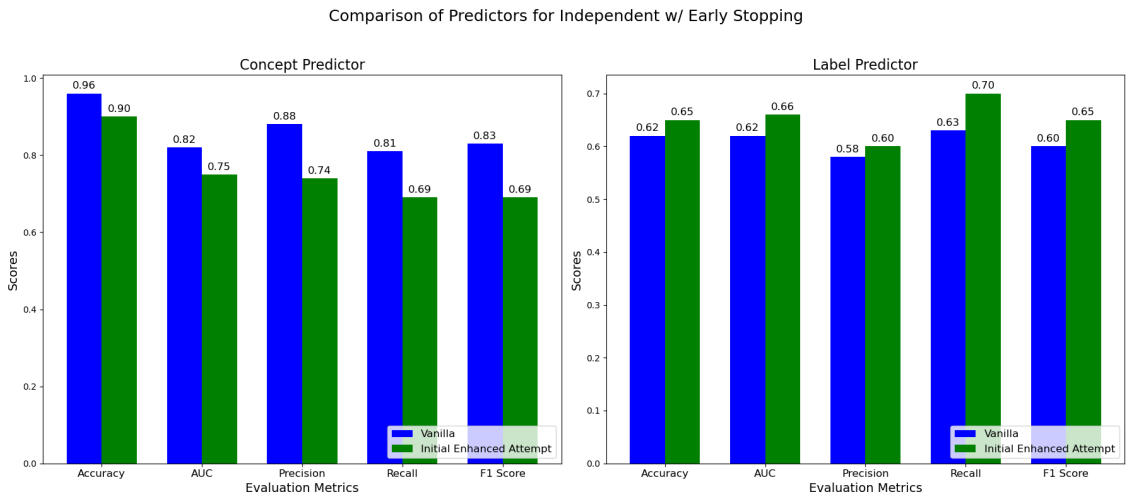

Figure B.10: Comparing the evaluation metrics for the predictors of Independent w/ Early Stopping CBMs. The graph contains side-by-side values for the original CBM and enhanced CBM (after its concept set was augmented).

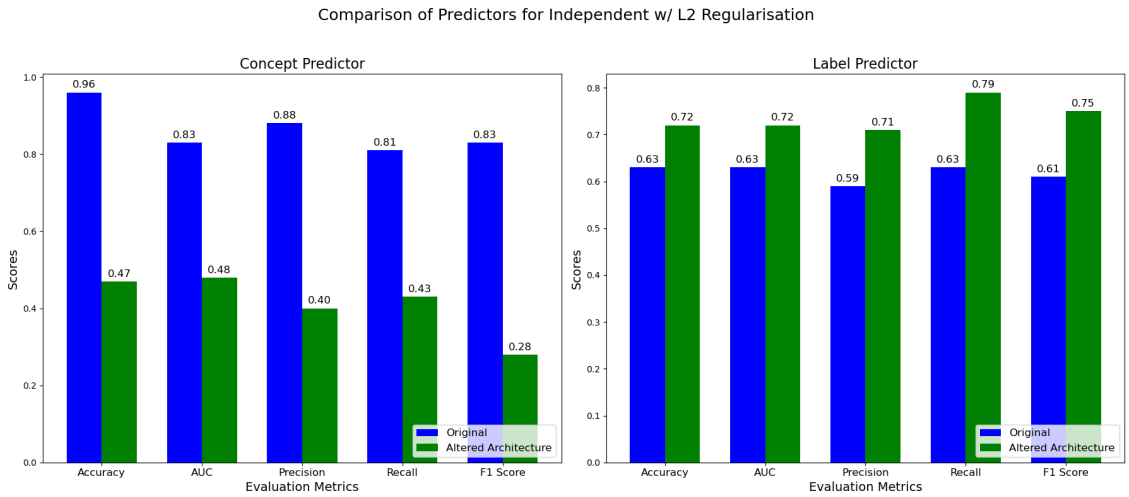

Figure B.11: Comparing the evaluation metrics for the predictors of Independent w/ L2 Regularisation CBMs. The graph contains side-by-side values for the original CBM and altered architecture CBM (after its concept set was augmented and architecture was changed).

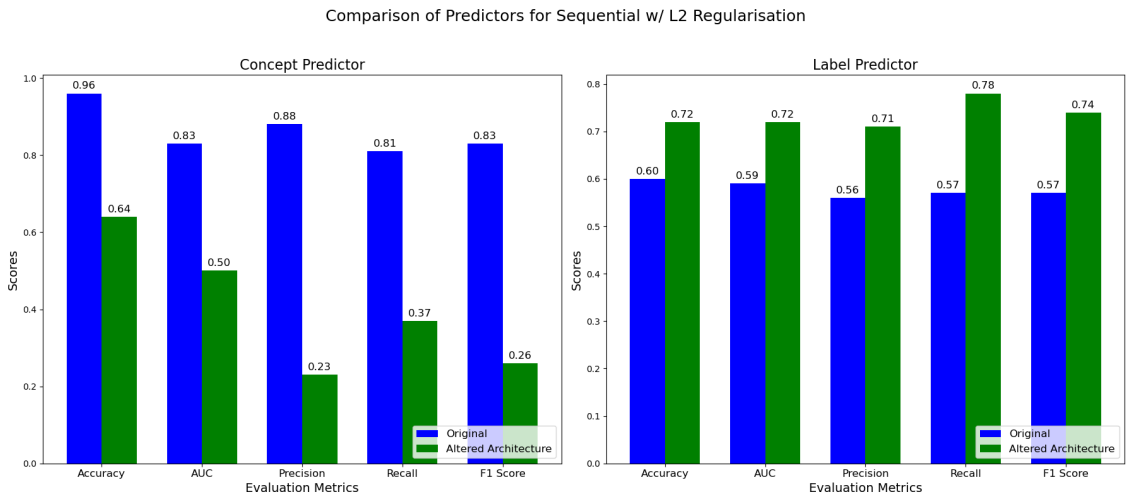

Figure B.12: Comparing the evaluation metrics for the predictors of Sequential w/ L2 Regularisation CBMs. The graph contains side-by-side values for the original CBM and altered architecture CBM (after its concept set was augmented and architecture was changed).

C

Project Code

The code for this project can be found in the following GitHub repository: `https://github.com/anish-narain/enhancing-clinical-decision-making`.

Bibliography

[1] K. Singhal, S. Azizi, T. Tu, *et al.*, "Large language models encode clinical knowledge," *arXiv preprint arXiv:2212.13138*, 2022.

[2] A. Johnson, L. Bulgarelli, T. J. Pollard, S. Horng, L. A. Celi, and R. G. Mark, *MIMIC-IV (version 2.2)*, `https://doi.org/10.13026/6mm1-ek67`, 2023.

[3] A. Johnson, T. J. Pollard, S. Horng, L. A. Celi, and R. G. Mark, *MIMIC-IV-Note: Deidentified free-text clinical notes (version 2.2)*, `https://doi.org/10.13026/1n74-ne17`, 2023.

[4] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[5] Google Cloud, *What is supervised learning?* Accessed: 2024-04-21, 2024. [Online]. Available: `https://cloud.google.com/discover/what-is-supervised-learning`.

[6] A. Margeloiu, M. Ashman, U. Bhatt, Y. Chen, M. Jamnik, and A. Weller, "Do concept bottleneck models learn as intended?" *arXiv preprint arXiv:2105.04289*, 2021.

[7] P. W. Koh, T. Nguyen, Y. S. Tang, *et al.*, "Concept bottleneck models," in *International conference on machine learning*, PMLR, 2020, pp. 5338–5348.

[8] C. Wu, S. Parbhoo, M. Havasi, and F. Doshi-Velez, "Learning optimal summaries of clinical time-series with concept bottleneck models," in *Machine Learning for Healthcare Conference*, PMLR, 2022, pp. 648–672.

[9] M. Havasi, S. Parbhoo, and F. Doshi-Velez, "Addressing leakage in concept bottleneck models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 386–23 397, 2022.

[10] "What is a large language model?" Cloudflare. (2024), [Online]. Available: `https://www.cloudflare.com/en-gb/learning/ai/what-is-large-language-model/`.

[11] "Large language models - what is a large language model?" Elastic. (2023), [Online]. Available: `https://www.elastic.co/what-is/large-language-models`.

[12] H. Zhou, B. Gu, X. Zou, *et al.*, "A survey of large language models in medicine: Progress, application, and challenge," *arXiv preprint arXiv:2311.05112*, 2023.

[13] "Hugging face models," Hugging Face. (2024), [Online]. Available: https://huggingface.co/models.

[14] J. Ferrer. "An introductory guide to fine-tuning llms," DataCamp. (2024), [Online]. Available: https://www.datacamp.com/tutorial/fine-tuning-large-language-models.

[15] H. Naveed, A. U. Khan, S. Qiu, *et al.*, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2023.

[16] Turing, *Fine-tuning large language models: How and why you should do it*, Accessed: 2024-05-20, 2024. [Online]. Available: https://www.turing.com/resources/finetuning-large-language-models.

[17] K. Singhal, T. Tu, J. Gottweis, *et al.*, "Towards expert-level medical question answering with large language models," *arXiv preprint arXiv:2305.09617*, 2023.

[18] H. Chen, Z. Fang, Y. Singla, and M. Dredze, "Benchmarking large language models on answering and explaining challenging medical questions," *arXiv preprint arXiv:2402.18060*, 2024.

[19] J. M. Ludan, Q. Lyu, Y. Yang, L. Dugan, M. Yatskar, and C. Callison-Burch, "Interpretable-by-design text classification with iteratively generated concept bottleneck," *arXiv preprint arXiv:2310.19660*, 2023.

[20] A. Yan, Y. Wang, Y. Zhong, *et al.*, "Robust and interpretable medical image classifiers via concept bottleneck models," *arXiv preprint arXiv:2310.03182*, 2023.

[21] A. Mahinpei, J. Clark, I. Lage, F. Doshi-Velez, and W. Pan, "Promises and pitfalls of black-box concept learning models," *arXiv preprint arXiv:2106.13314*, 2021.

[22] MIT Laboratory for Computational Physiology. "MIMIC documentation." (2021), [Online]. Available: https://mimic.mit.edu/docs/about/.

[23] M. Ghassemi, M. Wu, M. C. Hughes, P. Szolovits, and F. Doshi-Velez, "Predicting intervention onset in the icu with switching state space models," *AMIA Summits on Translational Science Proceedings*, vol. 2017, p. 82, 2017.

[24] M. Gupta, B. Gallamoza, N. Cutrona, P. Dhakal, R. Poulain, and R. Beheshti, "An Extensive Data Processing Pipeline for MIMIC-IV," in *Proceedings of the 2nd Machine Learning for Health symposium*, ser. Proceedings of Machine Learning Research, vol. 193, PMLR, 2022, pp. 311–325. [Online]. Available: https://proceedings.mlr.press/v193/gupta22a.html.

[25] A. E. W. Johnson, D. J. Stone, L. A. Celi, and T. J. Pollard, "The mimic code repository: Enabling reproducibility in critical care research," *Journal of the American Medical Informatics Association*, vol. 25, no. 1, pp. 32–39, 2018.

[26] A. Goel, A. Gueta, O. Gilon, *et al.*, "Llms accelerate annotation for medical information extraction," in *Machine Learning for Health (ML4H)*, PMLR, 2023, pp. 82–100.

[27] A. Goel, A. Gueta, O. Gilon, S. Erell, and A. Feder, *Medication extraction labels for mimic-iv-note clinical database (version 1.0.0)*, PhysioNet, `https://doi.org/10.13026/ps1s-ab29`, 2023.

[28] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag, "Large language models are few-shot clinical information extractors," *arXiv preprint arXiv:2205.12689*, 2022.

[29] E. Lehman and A. Johnson, *Clinical-T5: Large Language Models Built Using MIMIC Clinical Text (version 1.0.0)*, PhysioNet, 2023. DOI: `10.13026/rj8x-v335`. [Online]. Available: `https://doi.org/10.13026/rj8x-v335`.

[30] M. Syed, S. Syed, K. Sexton, *et al.*, "Application of machine learning in intensive care unit (icu) settings using mimic dataset: Systematic review," in *Informatics*, MDPI, vol. 8, 2021, p. 16.

[31] R. Pirracchio, M. L. Petersen, M. Carone, M. R. Rigon, S. Chevret, and M. J. van der Laan, "Mortality prediction in intensive care units with the super icu learner algorithm (sicula): A population-based study," *The Lancet Respiratory Medicine*, vol. 3, no. 1, pp. 42–52, 2015.

[32] W. Caicedo-Torres and J. Gutierrez, "Iseeu: Visually interpretable deep learning for mortality prediction inside the icu," *Journal of biomedical informatics*, vol. 98, p. 103 269, 2019.

[33] Y. Sha and M. D. Wang, "Interpretable predictions of clinical outcomes with an attention-based recurrent neural network," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2017, pp. 233–240.

[34] S. Kumar, S. C. Yu, T. Kannampallil, Z. Abrams, A. Michelson, and P. R. Payne, "Self-explaining neural network with concept-based explanations for icu mortality prediction," in *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2022, pp. 1–9.

[35] O. Nunez Lopez, J. Cambiaso-Daniel, L. K. Branski, W. B. Norbury, and D. N. Herndon, "Predicting and managing sepsis in burn patients: Current perspectives," *Therapeutics and clinical risk management*, pp. 1107–1117, 2017.

[36] Z. Nowroozilarki, A. Pakbin, J. Royalty, D. K. Lee, and B. J. Mortazavi, "Real-time mortality prediction using mimic-iv icu data via boosted nonparametric hazards," in *2021 IEEE EMBS international conference on biomedical and health informatics (BHI)*, IEEE, 2021, pp. 1–4.

[37] K. Pang, L. Li, W. Ouyang, X. Liu, and Y. Tang, "Establishment of icu mortality risk prediction models with machine learning algorithm using mimic-iv database," *Diagnostics*, vol. 12, no. 5, p. 1068, 2022.

[38] T. Huang, D. Le, L. Yuan, S. Xu, and X. Peng, "Machine learning for prediction of in-hospital mortality in lung cancer patients admitted to intensive care unit," *PLOS one*, vol. 18, no. 1, e0280606, 2023.

[39] M. Diamond, H. L. Peniston, D. K. Sanghavi, *et al.*, *Acute Respiratory Distress Syndrome* (StatPearls [Internet]). Treasure Island (FL): StatPearls Publishing, 2024, [Updated 2024 Jan 31]. [Online]. Available: `https://www.ncbi.nlm.nih.gov/books/NBK436002/`.

[40] M. T. Wayne, T. S. Valley, C. R. Cooke, and M. W. Sjoding, "Electronic "sniffer" systems to identify the acute respiratory distress syndrome," *Annals of the American Thoracic Society*, vol. 16, no. 4, pp. 488–495, 2019.

[41] B. J. Clark and M. Moss, "The acute respiratory distress syndrome: Dialing in the evidence?" *JAMA*, vol. 315, no. 8, pp. 759–761, 2016.

[42] V. M. Ranieri, G. D. Rubenfeld, B. Taylor Thompson, *et al.*, "Acute respiratory distress syndrome: The berlin definition.," *JAMA: Journal of the American Medical Association*, vol. 307, no. 23, 2012.

[43] A. C. McKown, R. M. Brown, L. B. Ware, and J. P. Wanderer, "External validity of electronic sniffers for automated recognition of acute respiratory distress syndrome," *Journal of intensive care medicine*, vol. 34, no. 11-12, pp. 946–954, 2019.

[44] M. Afshar, C. Joyce, A. Oakey, *et al.*, "A computable phenotype for acute respiratory distress syndrome using natural language processing and machine learning," in *AMIA Annual Symposium Proceedings*, American Medical Informatics Association, vol. 2018, 2018, p. 157.

[45] A. Gandomi, P. Wu, D. R. Clement, *et al.*, "Ardsflag: An nlp/machine learning algorithm to visualize and detect high-probability ards admissions independent of provider recognition and billing codes," *medRxiv*, pp. 2022–09, 2022.

[46] A. Pathak, R. Kamaleswaran, C. Marshall, C. Davis, and P. Yang, "Respbert: A multi-site validation of a natural language processing algorithm, of radiology notes to identify acute respiratory distress syndrome (ards)," *Authorea Preprints*, 2023.

[47] C. Clinic, *Medical questions & answers*, `https://my.clevelandclinic.org/health`, Accessed: 2024-05-27.

[48] M. Singer, C. S. Deutschman, C. W. Seymour, *et al.*, "The third international consensus definitions for sepsis and septic shock (sepsis-3)," *Jama*, vol. 315, no. 8, pp. 801–810, 2016.

[49] Ollama, *Ollama library*, Accessed: 2024-06-13, 2024. [Online]. Available: `https://ollama.com/library`.

[50] S. Kweon, J. Kim, H. Kwak, *et al.*, "Ehrnoteqa: A patient-specific question answering benchmark for evaluating large language models in clinical settings," *arXiv preprint arXiv:2402.16040*, 2024.

[51] X. Yang, A. Chen, N. PourNejatian, *et al.*, "Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records," *arXiv preprint arXiv:2203.03540*, 2022.

[52] PhysioNet. "MIMIC-IV Data Use Agreement." Supported by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) under NIH grant number R01EB030362., MIT Laboratory for Computational Physiology. (), [Online]. Available: `https://physionet.org/content/mimiciv/view-dua/2.2/`.

[53] J. G. Wang, J. Wang, M. Li, and S. Neel, "Pandora's white-box: Increased training data leakage in open llms," *arXiv preprint arXiv:2402.17012*, 2024.