

Machine Learning Methods for Tabular Electronic Health Record Data

Odhran O'Donoghue

Green Templeton College
University of Oxford

*A thesis submitted for the degree of
Doctor of Philosophy*

Michaelmas 2023

Abstract

The vast majority of the world's medical data exists in the Electronic Health Record (EHR), a document used to record demographic characteristics and discrete events throughout a patient's life. This EHR data is often poorly suited for machine learning, consisting of manually recorded data prone to errors and missing values, multiple time-series at highly varying levels of granularity, and typically high dimensionality and low patient numbers in datasets.

Nevertheless, machine learning on EHR data has a high potential for impact, and it is necessary to evaluate the efficacy of different machine learning methods for medical data. This thesis is an accumulation of theoretical and empirical work on developing machine learning methods appropriate for EHR data. Three work components are described.

The first work component of this thesis is a systematic evaluation of recurrent neural networks and transformer models for machine learning on EHR data. This thesis also introduces novel transformer-based architectures with properties that are desirable for EHR data. Transformers are identified as models with high performance for this task, achieving accuracy scores exceeding 94% in all evaluated tasks.

The second work component of this thesis is an effort to build machine models that can predict COVID-19 vaccine adverse events with a less than 0.1% occurrence rate in training data. An Area Under Receiver Operating Curve (AUROC) of 70.1% was achieved on this task, and feature and subgroup analysis to inform clinical understanding was performed.

The third work component is an effort to build machine learning models to predict stroke deterioration events from small neurointensive care EHR datasets. We evaluate a range of machine learning architectures, and we identify architectures that perform well both on training data and external validation data. We perform further feature and subgroup analysis to help place this work in the wider context of our clinical understanding of ischaemic stroke and malignant cerebral oedema.

This thesis ultimately finds that a diverse range of models are needed for EHR data in different contexts - a 'one size fits all' approach to machine learning is insufficient to address the diverse range of tasks, requirements and data formats within this setting.

Machine Learning Methods for Tabular Electronic Health Record Data



Odhran O'Donoghue
Green Templeton College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas 2023

This thesis is dedicated to Helen Feazey and Michael O'Donoghue.

I know how rare it is to have truly unconditional love, and I will never once take that for granted. Thank you for being lights when all other lights went out.

Thank you for the chances you gave me.

Dad.

Mum.

I love you both more than anything else.

Acknowledgements

I would be nowhere without my supervisor Professor David Clifton. Thank you for being the intellectual backbone upon which my work has rested, and being a constant sounding board for my academic trajectory. These thanks extend to the rest of the people I have collaborated with in the Computational Health Informatics group - Andrew Soltan, Jeny Yang, and in particular Anshul Thakur. Although our closeness may be based on a shared trauma of using Windows remote desktop for academic work, it was phenomenal to conduct research with you.

I would also be nowhere without my collaborations with Professor Agni Orfanoudaki at the Said Business School, Professor Charlene Ong and Professor Stelios Smirnakis from Massachusetts General Hospital, and Panos Tsimpos from MIT. Working with you all has been the highlight of my PhD. I also want to give particular thanks to Yumeng Zhang, who has helped immensely with our work towards publishing part of this research. I similarly want to give my thanks to my OATML collaborators Jannik Kossen, Neil Band and Lars Holdijk. I learned a great deal working with you all about how to be both a better researcher and a better engineer - immensely valuable lessons.

My research has been generously funded and supported by the EPSRC Centre for Doctoral Training in Health Data Science, without which my study would not be possible. I am deeply grateful to the program organizers: Professors Tom Nichols, Jim Davies, Jens Rittscher, and most of all Nina Hallowell - whose passing we all mourn and who will be remembered forever in our hearts.

I would similarly like to give my academic thanks to my PhD cohort: Andres, Henrique, Claudia, Cornelius, Frank, Hang, Jack, Jacob, Kangning, Maxime, Xinchu and Yifan. Thank you so much for the time you gave to help the members of the group learn a little computer science. A particular thank you goes out to Cornelius, who was the best on-the-spot teacher I could ever ask for.

Probably most of all, I'd like to thank the two computer scientists who have been there with me almost every step of the way, Ralph Abboud and Vid Kocijan. Dear friends thank you for helping me every step of the way on my journey from medicine to machine learning.

A similar thanks goes out to the Oxford Artificial Intelligence Society. It was an honour to both lead and serve - In particular Liam Bekirsky, Roxana Akhmetova, Hailey An, Sagar Vaze, Michael Collyer, Jessie Jiang, Tanvi Nayatal and Joy Wang - you will all be missed.

Further thanks go out to my Frontier Development Lab colleagues - Paul Duckworth, Linus Scheibenreif and Giuseppe Ughi. I took great pleasure in working with you all on some truly brilliant things (and the *cool factor* of working for NASA is never really going to fade). A similar thanks goes out to our project lead at NASA, Dr. Lauren Sanders. Thank you for your incredible leadership during our project and your help over the years that came!

I also need to thank the dream team as a whole - Anna Jungbluth, Amy Kao, Robin Van Aeken, and Vid Kocijan. We're now all so very far away, but the time we spent together was truly formative and is no doubt the reason we're on the paths we've taken today. Thanks go to Robin for hosting the only parties that matter, Amy for being a wonderful flatmate, and Ana for introducing me to the amazing world of space science.

Extra thank-yous go out to Runbei Cheng (the greatest chef in recorded history), Eito Miyamura (who foolishly introduced me to Umesu), Eduardo Eizaguirre (for answering my strange physics questions and being a wonderful host), Eric Tong (for being a genuinely wonderful friend) and Alexander Grassam-Rowe (for being the one constant in my academic life). I would also like to give my thanks to Loaie Solyman who did a phenomenal proof-read of my thesis.

Finally, my thanks go out to Kathryn Walsh - the best friend I could ever ask for. I don't think this PhD would have been possible without you in my life. I love you to bits, and it's been wonderful seeing how far we've both come in these past four years. I hope I'll make a good Godfather.

Abstract

The vast majority of the world’s medical data exists in the Electronic Health Record (EHR), a document used to record demographic characteristics and discrete events throughout a patient’s life. This EHR data is often poorly suited for machine learning, consisting of manually recorded data prone to errors and missing values, multiple time-series at highly varying levels of granularity, and typically high dimensionality and low patient numbers in datasets.

Nevertheless, machine learning on EHR data has a high potential for impact, and it is necessary to evaluate the efficacy of different machine learning methods for medical data. This thesis is an accumulation of theoretical and empirical work on developing machine learning methods appropriate for EHR data. Three work components are described.

The first work component of this thesis is a systematic evaluation of recurrent neural networks and transformer models for machine learning on EHR data. This thesis also introduces novel transformer-based architectures with properties that are desirable for EHR data. Transformers are identified as models with high performance for this task, achieving accuracy scores exceeding 94% in all evaluated tasks.

The second work component of this thesis is an effort to build machine models that can predict COVID-19 vaccine adverse events with a less than 0.1% occurrence rate in training data. An Area Under Receiver Operating Curve (AUROC) of

70.1% was achieved on this task, and feature and subgroup analysis to inform clinical understanding was performed.

The third work component is an effort to build machine learning models to predict stroke deterioration events from small neurointensive care EHR datasets. We evaluate a range of machine learning architectures, and we identify architectures that perform well both on training data and external validation data. We perform further feature and subgroup analysis to help place this work in the wider context of our clinical understanding of ischaemic stroke and malignant cerebral oedema.

This thesis ultimately finds that a diverse range of models are needed for EHR data in different contexts - a 'one size fits all' approach to machine learning is insufficient to address the diverse range of tasks, requirements and data formats within this setting.

Contents

List of Figures	xxiii
------------------------	--------------

List of Abbreviations	xxv
------------------------------	------------

1 An Introduction to Electronic Health Records	1
1.1 Introduction	1
1.2 What does Electronic Health Record Data look like?	2
1.2.1 Data Encoding	2
SNOMED-CT	3
ICD	4
1.2.2 Static Data	4
1.2.3 Time-Series Data	5
Vital-Signs	5
Imaging Data	7
Arterial Blood Gas Analysis	8
1.3 Machine Learning on EHR	9
1.3.1 How suitable is EHR Data for Machine Learning?	10
1.3.2 Data Access	10
1.3.3 Small Dataset Sizes	11
The Completeness of the EHR as a Medical Record	12
Time-Series Spacing	13
Data Errors	13

Domain Shift	13
Difficulties in Defining Outcomes of Interest	14
Class Imbalance	16
Multi-Modal Data	16
1.3.4 High-Dimensional Datasets	16
Metrics of Interest	17
1.4 The Focus of this DPhil	18
1.4.1 Research Hypothesis	18
1.4.2 How I Will Test my Research Hypothesis	18
1.4.3 Secondary Objectives	18
1.4.4 DPhil Structure	19
2 Machine Learning Methods for EHR Data	21
2.1 Introduction	21
2.2 Visualisation and Feature Selection	21
2.2.1 Principal Component Analysis	21
2.2.2 t-Distributed Stochastic Neighbour Embedding	22
2.2.3 Uniform Manifold Approximation and Projection	24
2.3 Resampling Methods	26
2.3.1 Linear Interpolation	26
2.3.2 Undersampling and Oversampling	26
SMOTE	27
2.3.3 Gaussian Processes	27
2.4 Genetic Programming	29
2.5 Tree-Based Models	30
2.5.1 Decision Trees	30
2.5.2 Random Forests	30

2.5.3	Gradient Boosting Trees	31
2.5.4	XGBoost	31
	XGBoost Feature Importance Explanations	32
2.6	Multinomial Naive Bayes	34
2.7	Shapley Additive Explanations	34
2.8	Hyperparameter Search Methods	35
2.8.1	Grid Search	35
2.8.2	Bayesian Hyperparameter Optimisation	36
	Surrogate Models	36
	Acquisition Functions	36
3	Deep Learning Methods for EHR Data	39
3.1	Introduction	39
3.1.1	Limitations of Traditional Machine Learning for EHR Data	39
3.1.2	The Goals of this Chapter	41
3.2	An Overview of Deep Learning	41
3.2.1	The Basic Formulation of Deep Learning	41
	Layers	41
	Activation Functions	42
3.2.2	Loss Functions	43
	Cross-Entropy Loss	43
	Hinge Loss	44
	Mean Squared Error	44
	Mean Absolute Error	44
	Weighted Losses	44
3.2.3	Backpropagation	45
3.2.4	Gradient Descent	46

3.2.5	Optimisation	46
	Stochastic Gradient Descent	47
	Momentum	48
	Adaptive Gradient Algorithm	49
	RMSProp	50
	Adadelata	50
	Adaptive Moment Estimation	51
	Layer-wise Adaptive Rate Scaling	52
	Lookahead	53
3.2.6	Challenges in Neural Network Architectures	53
	Internal Covariate Shift	53
	Gradient Explosion and Vanishing Gradients	54
	Overfitting and Underfitting	54
3.2.7	Partial Solutions to Challenges in Neural Network Architec- tures	54
	Batch Normalisation	54
	Dropout	55
	Early Stopping	55
3.3	Neural Network Architectures	56
3.3.1	RNNs	56
	Long-Short-Term Memory Network	57
	Gated Recurrent Unit	59
	Training RNNs	60
	Limitations of RNNs	60
3.3.2	Transformers and Attention	61
	Positional Encoding	63

4 Literature Review	65
4.1 Introduction	65
4.2 Feature Extraction Techniques	66
4.2.1 Hand-Crafted Features	66
4.2.2 Feature Transforms	67
Feaure Scaling	67
Statistical Features	68
Correlation Analysis	68
Signal Transforms	68
Hidden Markov Models (HMMs)	69
Time Series Standardisation	69
Gaussian Process Regression (GPR)	69
Clustering and Novelty Identification	70
4.3 Dimensionality Reduction	70
4.3.1 PCA	71
4.3.2 Embedding Layers	71
4.3.3 Autoencoders	72
4.4 Classification and Prediction	72
4.4.1 Regression Models	73
4.4.2 Tree Models	74
4.4.3 Deep Learning	75
Convolutional Neural Networks	75
Recurring Neural Networks	76
4.4.4 Reinforcement Learning	77
4.4.5 Other Interesting Models	77
4.5 Successful Models in Non-Medical Domains	78

4.5.1	Convolutional Neural Networks	78
4.5.2	Transformers	78
4.6	Conclusion	80
5	Developing Novel Deep Learning Models for Binary Classification on Electronic Health Record Data	81
5.1	Introduction: Variations of Attention-based Models	82
	Non-Parametric Transformers	82
	Limitations of Transformers	82
5.1.1	Inducing Point Models	83
	Perceiver Models	85
5.1.2	Novel Contributions	86
	Non-Parametric Perceiver	86
	Hierarchical Perceiver	86
5.2	Research Objective	87
5.3	Dataset Description	88
5.3.1	MIMIC-III	89
5.3.2	FlexIble Data-Driven pipeLinE for MIMIC-III Pre-processing	90
5.4	Data and Task Validation	90
5.5	Experimental Setup	94
5.5.1	Computational Resources	94
5.5.2	Model Evaluation	95
5.5.3	Model Pipeline	95
5.5.4	Hyperparameter Optimisation	95
5.6	Results	96
5.6.1	Sweep 1: Optimisers and Loss Functions	97
5.6.2	Sweep 2: Model Dimensions	99

5.6.3	Sweep 3: Learning Rates	100
5.6.4	Cross-Validation Results	102
5.7	Discussion	104
5.7.1	Cross-Validation Results	104
5.7.2	Failures in Training the Hierarchical Perceiver	105
5.8	Conclusion	106
6	Working with High-Class Imbalance in a Low-Compute Setting	109
6.1	Introduction	109
6.2	Research Justification	110
6.3	Related Models	112
6.4	Dataset Description	115
6.4.1	Dataset Source and Access	115
6.4.2	Target Construction	115
6.4.3	Data Filtering	116
6.4.4	Variable Selection	116
6.5	Feature Transformation and Imputation	117
6.6	Methods	118
6.6.1	Visualisation	118
6.6.2	Baseline Creation	118
6.6.3	Model Selection and Implementation	118
6.7	Results	119
6.7.1	Main Results: AUROC	119
6.7.2	Main Results: Interpretability	121
6.7.3	Secondary Analysis	122
	Training Models on Age-Specific Splits of Data	124
6.7.4	Training Models on a Data Subset	127

6.8	Discussion	129
6.9	Limitations	131
7	Working with High-Dimensional Small Sparse Data Sets	133
7.1	Introduction	133
7.2	Research Justification	134
7.3	Dataset Description	137
7.3.1	Study Population	137
7.3.2	Data Curation and Outcomes of Interest	139
	Feature Selection	139
7.3.3	Target Creation	141
7.3.4	Data Processing	142
7.3.5	Data Augmentation	143
	Encoding	143
	Censoring	143
	Imputation	143
7.3.6	Separability Assessment	144
7.3.7	Training, Testing and Evaluation	144
7.3.8	Models Evaluated	145
	Baseline Models	146
	Ensemble Tree Models	146
	Deep Learning Models	146
7.4	Results and Analysis	147
7.4.1	MGB Cross-Validation Results	147
7.4.2	BMC External Validation Results	147
7.4.3	Feature Importances and Clinical Interpretation	149
7.5	Conclusion	152

7.5.1	Assessment of Model Performance	153
7.6	Limitations	154
8	Conclusion	159
8.1	Novel Contributions of this thesis	159
8.2	Summary of Results	160
8.2.1	Developing Novel Deep Learning Models for Binary Classifi- cation on Electronic Health Record Data	160
8.2.2	Working with High-Class Imbalance in a Low-Computing Settings	160
8.2.3	Working with High-Dimensional Small Sparse Data Sets	161
8.3	Addressing the Research Question and Objectives of the Thesis	162
8.4	Future work	163
	Further Epidemiological Research	163
8.4.1	Causal Inference Research	163
8.4.2	Further Research from Chapter 7	164
8.4.3	Human Benchmarking	164
8.4.4	Integration into Markov Decision Processes	165
	References	167
	Appendices	
A	Appendices to Chapter 5	189
A.1	Sweep Configuration Tables	189

B Appendices to Chapter 7	199
B.1 Ensemble Tree Model Hyperparameters Optimised Over	199
B.2 Deep Learning Hyperparameters Optimised Over	200
B.3 Model Features and Feature Construction	201

List of Figures

3.1	A Recurrent Neural Network	56
3.2	A Long-Short-Term Memory network	58
3.3	A diagram of the attention operation	63
5.1	A Non-Parametric Transformer	83
5.2	An Induced Set Attention block	84
5.3	Perceiver Network structure	85
5.4	Hierarchical Perceiver structure	88
5.5	PCA of Atrial Fibrillation data	91
5.6	High coefficients from PCA of the AF data	92
5.7	Heatmaps of feature trajectories over time	94
5.8	Loss curves for the Hierarchical Perceiver on the Shock 12h Task .	106
5.9	Loss curves for the Non-Parametric Perceiver on the Shock 12h Task	106
6.1	PCA analysis	120
6.2	Receiver Operating Characteristic Curve for the best Overall model	123
6.3	SHAP Beeswarm plot of adverse event data points for the best Overall model	123
6.4	SHAP Beeswarm plot of adverse event data points for a Naive Bayes Classifier trained on people aged under 40	126
6.5	SHAP Beeswarm plot of adverse event data points for a Naive Bayes Classifier trained on people aged between 40 to 65	126

6.6	SHAP Beeswarm plot of adverse event data points for a Naive Bayes Classifier trained on people aged over 65	127
6.7	PCA subgroup analysis	129
6.8	SHAP Beeswarm plot of adverse event data points	130
7.1	Summary of data distributions	140
7.2	t-SNE visualisations of data in the four-class setting	145
7.3	ROC curves per MLS scan value	149
7.4	PR-curves per MLS scan value	150
7.5	XGBoost SHAP values per MLS scan value	151
7.6	The Double-Descent phenomenon	156
7.7	An illustration of the grokking phenomenon	157

List of Abbreviations

ADAM		Adaptive Moment Estimation optimizer
AI		Artificial Intelligence
AF		Atrial Fibrillation
ASPECTS	. . .	The Alberta Stroke Programme Early CT Score
AUROC		Area Under Receiver Operating Characteristic Curve
BCE		Binary Cross-Entropy
BPTT		Back-Propagation Through Time
CNN		Convolutional Neural Network
CT		Computer Tomography
DNAR		Do Not Attempt Resuscitation
EHR		Electronic Health Record
EMR		Electronic Medical Record
ECT		Electrocardiogram
FIDDLE		FlexIble Data-Driven pipeLinE for MIMIC-III
FPR		False Positive Rate
GRU		Gated Recurrent Unit
ISAB		Induced Set Attention Block
ICP		Intracranial Pressure

KNN		K-Nearest Neighbours
LARS		Layer-wise Adaptive Rate Scaling
LSTM		Long Short-Term Memory Network
LTME		Life-Threatening Mass Effect
MAB		Multi-Head Attention Block
MAE		Mean Absolute Error
MCA		Middle Cerebral Artery
MIMIC-III	. .	The Medical Information Mart for Intensive Care III
MLS		Midline Shift
MSE		Mean Squared Error
NPP		Non-Parametric Perceiver
NPT		Non-Parametric Transformer
ORCHID	. . .	Oxford-Royal College of General Practitioners Clinical Informatics Digital Hub
PCA		Principal Component Analysis
PHR		Personal Health Record
RMSprop	. . .	Root Mean Squared Propagation
RNN		Recurrent Neural Network
SHAP		SHapley Additive exPlanations
SMOTE		Synthetic Minority Over-sampling Technique
SNOMED-CT		SNOMED Clinical Terms
SVM		Support Vector Machine

t-SNE		t-Distributed Stochastic Neighbor Embedding
TADDS		Themes Access and Dynamic Data Services variable
TPOT		Tree-based Pipeline Optimization Tool for Automating Machine Learning
TPR		True Positive Rate
TRE		Trusted Research Environment
TTS		Thrombotic Thrombocytopenia Syndrome
ReLU		Rectified Linear Unit
RL		Reinforcement Learning
ROC		Receiver Operating Characteristic
UMAP		Uniform Manifold Approximation and Projection

1

An Introduction to Electronic Health Records

1.1 Introduction

Healthcare has in the last 20 years undergone a period of rapid digitisation. Individual patient demographic details are recorded on databases, along with information pertaining to their history of General Practice (GP) meetings, hospital interventions, medications, medical device scans, and diagnostic tests. These data are called Electronic Health Records (EHR) [1]. The EHR can take many different forms, but it is typically characterised as a relational database of many different data formats that correspond to different timescales, ranging from per-second readings in intensive care to sparse records in primary care that can have multi-year data gaps [2, 3].

Within healthcare, there is a value to predicting future outcomes and identifying conditions such that understanding the current and future risks of morbidity and mortality can help guide clinicians on how to prioritise care provision [4]. There is also value in identifying useful clinical interventions. As such, there has been

a great demand for the use of machine learning for prediction and classification tasks using EHR data [5]. This is by no means a trivial task, with EHR data being less than ideal for machine learning use. This section provides a primer on the EHR and its suitability as a data source for machine learning.

1.2 What does Electronic Health Record Data look like?

An EHR, also referred to as Electronic Medical Record (EMR) or Personal Health Record (PHR) [6] is a digital document that aims to capture all medical and clinical data pertaining to a patient. Types of data in this record include but are not limited to, demographic data, diagnostic data, family history data, allergies, immunisation, medications, procedures, lab values, vital-signs, reports, scan results, bio-sample data, genetic information, social data, patient-generated data, and surveys [7]. These data types can be roughly divided along the axes of temporal resolution and dimensionality. A breakdown of common hospital data forms along the axes of temporal resolution and dimensionality is reported in table 1.1.

Data are also available in different modalities, including discrete-valued numerical data (such as the number of infections), continuous-valued numerical data (such as blood pressure), 2D/3D image data (such as an X-ray or computer tomography (CT) scan), waveform data (such as ElectroCardioGram (ECG) data), text/text-like data (such as free-text data from medical notes and genomic data) and 2D/3D video data [8]. This data varies in temporal resolution and dimensionality.

1.2.1 Data Encoding

Given the breadth of conditions and events encountered in medicine, sorting data and identifying outcomes of interest can be challenging. Efforts have been made

Table 1.1: Medical datasets and their positions along axes of dimensionality and temporal resolution

	Low-dimensionality	High-dimensionality
Static in time	Demographic data, family history data, allergies, some elements of social data, immunisations	Genomic sequencing results
Low-temporal resolution	Vital-sign readings, pathology test results, interventional data	X-ray, computer tomography, magnetic resonance imaging, free-text data
High-temporal resolution	ECG data, vital-sign readings, implant data, accelerometer data	Functional magnetic resonance imaging, positron emission tomography, ultrasound

to resolve this by sorting data into ontologies, sets of concepts and categories that show properties and the relations between them. This thesis makes use of two ontologies for data retrieval, SNOMED Clinical Terms (SNOMED-CT) [9], and the International Classification of Diseases (ICD) systems [10].

SNOMED-CT

SNOMED-CT is an ontology of clinical terms developed in the United Kingdom but used worldwide. SNOMED-CT has four principal components:

1. **Concept Codes:** these are numerical identifiers for clinical terms, whether primitive or fully defined, arranged in a hierarchical structure.
2. **Descriptions:** these provide textual explanations of the Concept Codes.
3. **Relationships:** this component details the connections between Concept Codes that share a related significance.

4. **Reference Sets:** these are utilised for organising Concepts or Descriptions into collections, encompassing reference groups and mappings to other classifications and standards.

Critically, SNOMED-CT includes codes for diseases, outcomes, interventions and clinical variables. This thesis makes use of these codes to filter data and identify target groups and variables. It is of particular use in chapter 6, where SNOMED codes are used to harmonise data from many different GP clinics across the United Kingdom for combined analysis. SNOMED codes are also used to identify the target patient population for analysis in this section.

ICD

The ICD system is a framework for disease classification used by the World Health Organisation (WHO). It is currently in its 11th iteration (ICD-11). In comparison to SNOMED-CT, its remit is narrower (only covering diseases), but it is used more widely throughout the world. Like SNOMED-CT, it uses a hierarchical system for identifying conditions. Chapter 7 uses data from the United States, and makes use of ICD codes to identify outcomes of interest to form the target patient cohorts.

1.2.2 Static Data

Static data are defined as data that does not change after being recorded [11]. Low-dimensional forms of this data include demographic data, family history data, allergies, some elements of social data and, typically, immunisations [7]. A commonly encountered high-dimensional form of static data is genomic data [12]. High-dimensional data typically requires different pre-processing procedures in comparison to low-dimensional data before it can be used for machine learning in healthcare problems [13]. In total, static data makes up a small amount of the

data used in this thesis (it is in particular used quite widely in chapter 6). Instead, the majority of data used in this thesis is time-series data.

1.2.3 Time-Series Data

Time-series data are defined as data that is collected at different points in time [14]. Time-series data ARE one of the most commonly collected forms of data in hospital settings.

Some time-series data are collected at very high-temporal resolutions (i.e. multiple times a second). This data can include ECG data [15], accelerometer [16], integrated sensor data (e.g. from neural implants [17]), and vital-sign data such as heart rate and oxygen saturation [18]. Other data are collected at moderately high-temporal resolutions (i.e. multiple times an hour to multiple times a day). This data can include vital-sign data such as respiratory rate and blood pressure [8, 18], and laboratory test values [19]. Other time-series data are collected at low-temporal resolutions (i.e. multiple times a week to multiple times a year or fewer) and typically represents discrete events in an individual's life. This data includes, amongst other readings, a record of hospitalisations, procedures, reports, medications prescribed and taken, diagnostic results, lab values and new immunisations amongst other possible readings [20].

It is relevant to discuss some of the most common examples of time-series data encountered in this thesis in detail, as this informs the potential issues with data collection, data quality, and distributional shifts in data between various clinical settings.

Vital-Signs

- **Heart Rate** is simply the number of Beats Per Minute (BPM) produced by the heart. A typical healthy range is within 60 to 100 BPM. In intensive

care settings in hospitals, heart rate will typically be monitored continuously through a pulse oximeter attached to a patient's finger. Outside of intensive care settings heart rate may be monitored either via a pulse oximeter at intervals or manually by taking a patient's pulse. Taking a patient's pulse is often subject to human recording error, and standard practice can involve only taking a pulse for 15 seconds and multiplying the rate by 4 to get a BPM [21]. As such, manually recorded pulse rates are typically quite inaccurate and subject to human rounding errors.

- **Blood Pressure** is the pressure exerted by circulating blood on arterial walls. It is divided into two pressures: systolic pressure (the pressure on the system when the ventricles of the heart contract) and diastolic pressure (the pressure on the system when ventricles are relaxed). Blood Pressure is typically measured in its historic unit of millimetres of mercury (mmHg), which is derived from pre-electronic measuring devices. Healthy systolic blood pressure is typically around 120 mmHg, while healthy diastolic blood pressure is typically around 80 mmHg. While Blood Pressure is one of the most commonly recorded clinical time-series variables, measuring systolic blood pressure requires the temporary restriction of blood vessels, meaning constant monitoring is not possible. Modern measurement devices use a self-inflating cuff to restrict blood pressure and automatically record systolic and diastolic variables. However, different brands of devices and different procedures for monitoring (such as cuff size and cuff placement) [22] can introduce systematic errors in measurement. Critically, these systematic errors can be significantly different between hospitals and care settings, leading to a shift in the underlying distribution of data. Electronic blood pressure monitoring devices are typically available in all care settings in

the United States and the United Kingdom. There may, however, be some edge cases in GP clinics where pre-electronic devices are used. This requires humans to listen to circulating blood under pressure with a stethoscope and as such is subject to measurement and rounding errors.

Imaging Data

- **CT scans** are three-dimensional scans of the body's internals produced by X-ray beams being passed around the body in a circular arc. Scans such as these are required for imaging internal soft tissues in high detail. CT scans are used relatively frequently in clinical settings, but the ionising radiation they produce limits their constant use. Radiologists are trained to read CT scans and identify radiographic features, and quantitative and qualitative data points that can be used to make diagnoses and inform treatment decisions. CT scans are used regularly in neurology to image the brain, given the inherent issues with other forms of monitoring. Radiographic features from the brain can include the degree of fluid in the brain (edema), and the placement of key structures in the brain relative to their usual position. Through advances in segmentation and machine learning, the recording of radiographic features is progressively becoming more automated [23]. However, radiographic features have historically been recorded by hand, which once again can lead to significant measurement errors.
- **MRI scans** are three-dimensional scans of the body's internals produced by radio waves and magnetic fields. Unlike their CT counterparts, MRI scanners do not use ionizing radiation for imaging. This makes them suitable for repeated and sustained use. When compared to CT scanners, MRI scanners can have higher fidelity. This may be particularly useful when imaging soft

tissues like the brain, muscles and nerves. MRI scanners can also be used to image the use of oxygen in tissues over time, which can be used to produce functional MRI (fMRI) 3D video files. This is often used in neuroscience and psychology to interrogate potential functions of different brain regions [24]. As with CT, machine learning techniques are being increasingly used to automate the analysis of MRI data [25]. Nevertheless, manual interpretation remains a critical part of clinical practice, especially for complex cases where machine learning models are still evolving.

Arterial Blood Gas Analysis

Most hospitals have access to blood gas analysers. These devices can be used to analyse blood withdrawn from patients for levels of gases such as oxygen or carbon dioxide. Modern machines are also used to record the concentration of dissolved ions and molecules in blood. Key readings from Arterial Blood Gas Analysis (ABGA) relevant to this thesis include the following:

- **Sodium** is the main ion present in extracellular fluids. As an ion, sodium levels are critical in maintaining the osmotic balance between extracellular fluid and cellular tissue. Hyponatremia (low sodium levels) can lead to edema (tissue swelling) [26]. Sodium also has a key role in the proper functioning of nerve impulse transmission and muscle contraction.
- **Creatinine** is a byproduct produced as part of normal metabolism, which is normally cleared by the kidneys. Increased levels of Creatinine can be used to indicate a reduction in kidney function, which is highly relevant to the management of care in intensive care settings.

- **Glucose** is the standard energy substrate used by the human body. Chronically raised glucose levels can be indicative of conditions such as diabetes, while acute fluctuations can be indicative of stress response or in extreme cases damage to glucose production processes [27].

1.3 Machine Learning on EHR

Predictions and classifications are made routinely by clinicians, who synthesise their findings from all available data [8]. Outcomes of interest can be quantitatively measured (i.e. blood pressure) or qualitatively assessed (i.e. anxiety level). In some situations, probabilistic predictions and classifications may be useful (i.e. the probability of benefiting from one treatment over another treatment).

Classifications are perhaps most relevant in medicine when considering a diagnosis: clinicians will identify diseases and conditions from the results of scans, tests and physical investigations [8]. Predictions, however, are relevant in many different settings: in primary care settings, common predictions include predicting the risk of developing specific diseases such as diabetes [28]. In hospital care, common predictions include predicting the risk of mortality [29] and predicting the risk of Intensive Care Unit (ICU) admission [30].

Given the recent advances in deep learning and its success in domains such as image recognition, translation, time-series prediction, and biochemical and physical properties prediction [31, 32], deep learning has been of particular interest to medical researchers for prediction and classification tasks. Despite this, there are difficulties in implementing classical machine learning methods and deep learning methods for healthcare prediction and classification tasks. Understanding these difficulties is key to understanding the developments that would be necessary for machine learning methods to improve suitability for EHR data.

1.3.1 How suitable is EHR Data for Machine Learning?

EHR data is far from being an ideal dataset for performing machine learning. Despite this, the many limitations and issues with the data are not commonly acknowledged. Understanding such limitations is essential to building machine learning methods that are suitable for EHR data to allow for genuine societal benefit from medical AI to be realised. To this end, the remainder of this section deals with the limitations of using EHR data in its default form for machine learning, which include the following:

1.3.2 Data Access

Access to medical data is, quite appropriately, restricted to protect the confidentiality of patients. Medical data can be shared when it is anonymised [33], but this anonymisation process can be time-consuming and in certain cases infeasible, and as such only small amounts of medical data are available publicly [34]. Individual researchers may have access to private medical datasets, but they may be unable to share these private medical datasets with the wider research community. This prevents reproducibility and slows down the speed of research [35].

Recent work in the United Kingdom has proposed a new method to access data through Trusted Research Environments (TREs), these serve as on-site computing environments that can be remote-accessed, preventing data from leaving a source location [36]. While well intended, this approach is often ill-suited for the current state-of-the-art in machine learning. The rapidly moving nature of the field means that the computational resources available in TREs are often insufficient to run the field’s most commonly used effective models [37]. These considerations mean that more efficient models than what is used in mainstream settings often have to be considered.

Table 1.2: Commonly used medical and non-medical datasets and their size

Non-Medical Dataset	Type	Size
ImageNet [39]	Image classification	14,197,122 images
AudioSet [40]	Audio and video classification	2,084,320 recordings
Microsoft Common Objects in Context (COCO) [41]	Image segmentation and classification	2,500,000 instances
CASIA-HWDB [42]	Character recognition	1,172,907 instances
Amazon reviews [43]	Text and image classification	~82,000,000 instances
Medical Dataset	Type	Size
MIMIC-III ¹ [44]	EMR	53,423 hospital admissions
MIMIC-III waveform [44]	EMR	67,830 record ECG traces
eICU [45]	EMR	200,000 admissions
CheXpert [46]	X-Rays	224,316 chest radiographs

1.3.3 Small Dataset Sizes

Medical datasets are typically smaller in terms of their number of examples than equivalently available non-medical datasets. Examples of medical and non-medical datasets and their sizes are shown in table 1.2. A smaller number of examples in medical datasets can result in an increased difficulty in training deeper models in comparison to non-medical datasets, as deeper models typically require a larger number of training examples to update their larger number of parameters [38].

The Completeness of the EHR as a Medical Record

The medical record of a patient is typically never complete. There are multiple reasons for this, which include the following:

- *Failure to Record Information:* there are various reasons as to why a clinician may not log information in an EHR. One is simply forgetting: when assessed, patients only recall 20-60% of information in a primary care meeting [48], and as such information may not be effectively passed from one care setting to another. Another may be that the information is simply not seen as important.
- *Time Constraints:* healthcare providers often have tight schedules, seeing numerous patients in a single day. This time pressure can lead to brief documentation or, in some cases, omission of certain data points. As the EHR systems have grown in complexity, the time required to input data accurately has also increased, further exacerbating this issue [49]. This is in turn compounded by staffing issues for clinicians in most nations, with the global healthcare workforce being understaffed by almost 15 million in 2020 [50].
- *EHR Restriction:* EHR systems often expect data in certain formats, such as time-stamped events with SNOMED-CT codes [9]. Such codes are useful for curating datasets, but such restrictive input formats can sometimes result in EHRs not giving a full picture of care.

¹MIMIC-III - The Medical Information Mart for Intensive Care III (MIMIC-III) dataset is a large, de-identified and publicly-available collection of ICU medical records [47]. Its structure is detailed in-depth in chapter 5.

- *Interoperability Issues:* different healthcare facilities might use different EHR systems. The lack of interoperability between these systems can lead to fragmented or missing data when a patient receives care at multiple locations [51].
- *Intentional Omissions:* at times, clinicians might deliberately omit sensitive information to protect a patient's privacy or avoid potential liabilities [52, 53].

Time-Series Spacing

The spacing between time points is unlikely to be constant in most clinical time-series data. Such unevenly spaced time-series are not homogeneous and pose challenges for traditional time-series analysis methods that assume constant intervals [54]. The non-uniformity of these intervals can arise from various reasons, such as irregular sampling, missing data, or variable reporting intervals.

Data Errors

As medical data are typically recorded by hand, mistakes and omissions can be made. Common mistakes are entering incorrect volumes of provided drugs or incorrect numerical test results [55]. Different clinicians may also choose to report different observations using different units of different scales: this can result in the distribution of data being artificially warped and spotting these different units or scales can be difficult without expert guidance from clinicians [47].

Domain Shift

Distribution shift is a change in the distribution of data that can affect inputs or labels [56]. At its core, domain shift implies that the distribution $P(X, Y)$ that the model is trained on (source domain) is not the same as the distribution $Q(X, Y)$,

it will be tested on (target domain). Medical datasets, in particular, are highly susceptible to domain shifts. Variations arise due to numerous reasons:

- **Temporal Variations:** over time, treatment protocols, technology, and disease dynamics can evolve, leading to different distributions in newer data compared to older datasets [57, 58]. Of particular note in healthcare are epidemics and pandemics, which are projected to become more frequent occurrences [59]. These events significantly change global disease burdens and the resulting distribution of health data.
- **Population-Specific Factors:** in different populations, different risk factors may drive the prevalence of the same condition. An example of this is high blood pressure, where blood pressure sensitivity to salt intake and body mass varies between ethnic groups [60].
- **Geographical Factors:** different regions may have unique environmental conditions, healthcare practices, and patient demographics that can significantly influence disease prevalence and presentation [61]. Variability in diagnostic tools, imaging devices, or data annotation protocols between institutions can similarly introduce shifts in data distributions [62]. Shifts can also arise from the underlying EHR system used to record data which can enforce and restrict certain recording practices [51].

Difficulties in Defining Outcomes of Interest

Most standard forms of machine learning require enforced boundaries between classes for the purpose of classification. The performance of machine learning is also highly dependent on the accuracy of labelled data. In medical settings, it can be difficult to accurately collect data on outcomes of interest. This can be due to

difficulty in defining and diagnosing certain conditions, incomplete or unavailable EMR data, and difficulties with identifying attribution. To explain in more detail:

- Some conditions are hard to diagnose, and may be misdiagnosed or go undiagnosed. One example is Systemic Lupus Erythematosus, which is commonly misdiagnosed [63]. Another example is Parkinson’s Disease, where approximately 20% of patients are not diagnosed [64]. Creating a training set of these conditions is therefore difficult, as there may be many incorrectly labelled examples in a dataset. This also is particularly true at times when new conditions emerge, a recent example being the rare cases of vaccine-induced thrombocytopenia following Sars-CoV-2. If consensus has not been reached in the medical community on how to label a new condition, past data can be difficult to find and re-label once consensus is reached.
- Small amounts of EMR data can significantly change the interpretation of events, particularly if there has been a mistake in the record or if a finding is concealed. As an example, If a patient has a Do Not Attempt Resuscitation (DNAR) order, no intervention will be attempted in cardiac arrest and death will be inevitable [65]. If the DNAR is unavailable in the EMR dataset (either due to confidentiality, due to no pathway to process the data, or due to an error), the interpretation of the causes of death following cardiac arrest will be faulty.
- Another issue is in attribution: an EMR may not associate an adverse event with a specific cause. As such, it can be hard to identify adverse events associated with specific precursors in comparison to adverse events that may happen due to chance. An example would be identifying if a blood clot is attributable to a surgical intervention or was just a random event. These

issues introduce noise and/or potentially systematic bias into the labels of training datasets.

Class Imbalance

Many medical conditions of interest are rare: in the United Kingdom, 3.5 million people have a disease that occurs at a rate of less than 1 in 2,000 people [66]. Adverse events relating to interventions or medications can also be rare: as an example, adverse events relating to blood clots following vaccination with the Oxford-AstraZeneca COVID-19 vaccine are thought to occur at a rate of around 11 excess events per 100,000 [67]. When attempting to predict or identify such conditions or adverse events, traditional machine learning methods that are designed for more balanced datasets may fail in these circumstances.

Multi-Modal Data

Making a diagnostic decision in medicine relies on the use of many information sources together (e.g. the results of a CT scan combined with a blood test and patient-reported history may together be used to diagnose appendicitis) [8]. Most machine learning methods are narrow in scope, in that they are designed to interpret just one data type [68]. This prevents most machine learning methods from being able to emulate the decision-making of clinicians and act with all available information in medical domains.

1.3.4 High-Dimensional Datasets

Modern medical data routinely consists of high-dimensional data, with key examples being genomic data and three-dimensional scan data [12]. When combined with the issue of small dataset sizes, this can result in difficulties in training models due to parameter count exceeding dataset size [69]. Many dimensionality reduction

protocols have been created to pre-process specific high-dimensional medical datasets for downstream machine learning tasks [13]. However, as these specific dimensionality reduction protocols are not used by the non-medical research community, the reduced datasets may be less suitable for state-of-the-art methods developed outside of the medical field. Furthermore, because of the necessity of pre-processing these high-dimensional datasets for machine learning tasks, there is a huge degree of variance in the different pre-processing pipelines used by researchers [47]. This can result in a high error rate and difficulties in replicating the findings of studies in medical machine learning, which in turn can slow down the speed of research in the field.

Metrics of Interest

In many non-medical settings, machine learning models are trained with the goal of maximising overall accuracy on a test dataset. Common metrics for accuracy include top-1 accuracy, calculated as the fraction of results that are labelled correctly, and top-5 accuracy, calculated as the fraction of results that include the correct label in their top-5 predictions [70]. The machine learning research community thus typically chooses model architectures, optimisers and loss functions that aim solely to maximise accuracy. In medical settings, however, accuracy alone is often seen as an insufficient metric. This is because accuracy does not fairly assess the performance of a model on rare conditions or events. Furthermore, accuracy does not fairly assess the rate of false positives to true positives and false negatives to true negatives. False positives can be costly in medicine due to unnecessary interventions being provided [71], and false negatives can be dangerous in medicine as potentially life-threatening events may go unobserved [72]. Machine learning models designed to optimise only accuracy do not necessarily perform well on these metrics [73].

1.4 The Focus of this DPhil

1.4.1 Research Hypothesis

Given how variable current machine learning methods are for processing EHR data, there are many issues with clinical implementation. When clinicians do not have expertise in implementing models the significant varying options of choice mean that many mistakes can occur in the implementation stage. A single unified modeling framework for dealing with the different tasks in EHR data would be incredibly beneficial for the field, as it would allow for a standardized approach to processing data. This could in turn lead to improved reproducibility, increased understanding of methods, and ideally improved medical outcomes. My research hypothesis is thus the following: **is it possible to build a single unified machine learning architecture appropriate for all tasks in EHR-based machine learning?**

1.4.2 How I Will Test my Research Hypothesis

To test my research hypothesis, I aim to develop novel machine learning architectures, and I aim to assess these architectures on different real-world EHR prediction tasks. I will also evaluate the appropriateness and efficacy of different machine existing learning methods for novel EHR data tasks, as part of my goal of identifying the most appropriate architectures.

1.4.3 Secondary Objectives

A secondary, but equally important objective, of my work will be to generate results that could potentially improve clinical practice. I consider this necessary to maximise the impact of this DPhil work. My work in real-world EHR data settings will focus thus not only on machine learning model efficacy, but also on using results to improve clinical understanding of conditions addressed where

possible. To this end, significant emphasis will be placed on explainability, and clinical interpretation of results

1.4.4 DPhil Structure

The structure of the remaining chapters in this thesis is as follows:

- Chapters 2 and 3 detail the machine learning techniques and methods used in this thesis. Chapter 2 details non-deep learning methods, and chapter 3 details deep learning methods.
- Chapter 4 is a literature review of machine learning methods, analysing their successes both generally and in the specific domain of healthcare.
- Chapters 5, 6 and 7 detail the bulk of the thesis work and describe the individual research projects. Chapter 5 is a systematic evaluation of state-of-the-art machine learning approaches on the Medical Information Mart for Intensive Care III (MIMIC-III), a gold-standard idealised dataset of EMR data. In this section, my novel machine learning architectures are also proposed, and their efficacy is evaluated. Chapters 6 and 7 detail real-world implementation projects of machine learning models in non-ideal data settings. Chapter 6 details building a machine learning model in a data setting with high class imbalance and a compute-restricted environment. The goal of the project is to make accurate predictions and to understand what factors may be contributing to predictions. Chapter 7 details building a machine learning model for a high-dimensional small dataset with high levels of missing and noisy data. The goals of this project are to make accurate predictions, to understand what factors may contribute to predictions, and to consider how models can be used to guide clinical decision-making.

- Chapter 8 delivers the concluding remarks of the thesis and discusses possible future directions for subsequent research.

2

Machine Learning Methods for EHR Data

2.1 Introduction

This section is used as a primer to explain the machine learning methods and tools utilised in this thesis. It includes all techniques used excluding deep learning, which is discussed in the next chapter in more depth.

2.2 Visualisation and Feature Selection

2.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique that finds a linear projection of data such that the variance of the projected data is maximised. PCA is suitable for dimensionality reduction, data compression, and visualisation. It is formulated as follows:

Mathematically, PCA seeks to find a set of orthogonal axes (principal components), that maximise dataset variance. Given a dataset X with n observations, where each observation is a d -dimensional random vector, $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$.

Following standardisation and normalisation of data, the steps for performing PCA are as follows:

Firstly, the covariance matrix Σ of the standardised data is computed:

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}'_i - \bar{\mathbf{x}}')(\mathbf{x}'_i - \bar{\mathbf{x}}')^T \quad (2.1)$$

Secondly, the eigenvalues λ and eigenvectors $\mathbf{v}$ of the covariance matrix Σ are computed:

$$\Sigma \mathbf{v} = \lambda \mathbf{v} \quad (2.2)$$

Thirdly, the eigenvalues are ordered in decreasing order. For dimensionality reduction into k dimensions, only the eigenvectors corresponding to the top k eigenvalues are selected.

Finally, a $d \times k$ dimension projection matrix W is created by concatenating the k eigenvectors. This matrix is used to project the original data X onto the lower-dimensional subspace Y :

$$Y = XW \quad (2.3)$$

The PCA typically preserves large pairwise distances from the original dimensions in the projected dimensions because it makes use of a squared error term. This can result in some unfaithful representations if data manifolds have a particularly pronounced local structure.

2.2.2 t-Distributed Stochastic Neighbour Embedding

Manifold techniques are techniques used for dimensionality reduction, which rely on the assumption that points can be embedded on a low-dimensional manifold representation of a high-dimensional space. Methods can be used to discover these potential manifolds, and the manifolds can then be used to represent data.

The t-Distributed Stochastic Neighbour Embedding (t-SNE) is one such manifold technique that is commonly used visualisation technique. t-SNE is used to embed higher-dimensional data into a lower-dimensional space. t-SNE measures similarity between points in high dimensions by computing a matrix $n \times p$ of similarity probabilities, where n is the number of data points and p refers to the probabilities associated with each point pair. Probabilities p_{ij} are based on the density of points under a Gaussian distribution, and these values are calculated as the normalised sum of:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)} \quad (2.4)$$

For p_{ij} and p_{ji} , perplexity is used as a factor to scale the Gaussian kernels σ_i such that a fixed number of points fall within the mode of each distribution, this allows t-SNE to adapt to different densities of local clusters. Perplexity is a measure based on the Shannon entropy of the conditional probability distribution $P(j|i)$, which defines the probability of selecting point j as a neighbor of point i . It is defined as follows:

$$\text{Perplexity} = 2^{H(P_i)}$$

where $H(P_i)$ is the Shannon entropy of the conditional distribution $P(j|i)$ over all points j , given by:

$$H(P_i) = - \sum_j P(j|i) \log_2 P(j|i)$$

In the low-dimensional space, t-SNE moves points such that the similarity matrix based on the density of points under a distribution is as similar as possible to the Gaussian distribution in the high-dimensional representation. However in this case the low-dimensional space uses the Student's t-distribution formulated as:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}} \quad (2.5)$$

This minimisation of dissimilarity is achieved through a gradient descent formulation of Kullback–Leibler divergence [74], i.e.:

$$\text{KL}(P \parallel Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (2.6)$$

t-SNE’s computational complexity scales quadratically with the number of data points. As such, this method can only be used on smaller datasets or subsets of larger datasets.

t-SNE is also typically seen as unsuitable for dimensionality reduction. Aside from its scaling issues, t-SNE does not provide a clear mechanism for transforming new, unseen data, and so cannot be applied to out-of-sample extension. Furthermore, t-SNE’s cost function is non-convex, leading to different embeddings on repeat runs.

2.2.3 Uniform Manifold Approximation and Projection

Uniform Manifold Approximation and Projection (UMAP) is a second manifold learning-based dimensionality reduction technique that is effective for visualising high-dimensional datasets. UMAP is more scalable and computationally efficient than t-SNE, and allows for out-of-sample extension making. UMAP relies on approximating a data manifold in a high-dimensional space and then projecting this approximation to a lower-dimensional space.

Given a dataset $X = \{x_1, x_2, \dots, x_n\}$ in high-dimensional space $\mathbb{R}^D$, the steps for performing UMAP are as follows.

First, a weighted k -nearest neighbours graph for each point x_i in the dataset is constructed. Distance is typically calculated using Euclidean distance. A local

connectivity parameter is used to determine the weight $w(x_i, x_j)$ of the edge between x_i and x_j . The weighted graph represents the high-dimensional data as a discrete topological structure. UMAP uses this structure to approximate the geometric structure of the data manifold.

Secondly, this graph is used to construct a fuzzy set of simplexes. A simplex is a generalisation of triangles and tetrahedrons to arbitrary dimensions (e.g. a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron). In the context of UMAP, each simplex contains a set of data points, and data points can be involved in multiple simplex sets leading to complexes. The degree of membership of each simplex reflects the likelihood that the corresponding data points are connected. This fuzzy simplicial set captures the topological structure of the data.

Thirdly, a low-dimensional representation of the data that is as consistent as possible with the fuzzy simplicial set of the high-dimensional data is iteratively optimised. This is formulated as follows:

Let X be the high-dimensional data and Y be the low-dimensional representation. Let $\mu_X : S_X \rightarrow [0, 1]$ and $\mu_Y : S_Y \rightarrow [0, 1]$ be the membership functions for the fuzzy simplicial sets S_X and S_Y derived from X and Y , respectively.

The cross-entropy between S_X and S_Y is defined as:

$$C(Y) = \sum_{s \in S_X \cup S_Y} -\mu_X(s) \log(\mu_Y(s)) \quad (2.7)$$

Where s is a simplex in S_X or S_Y .

The optimisation process seeks to find Y that minimises $C(Y)$, updating Y iteratively using stochastic gradient descent or a similar optimisation technique.

2.3 Resampling Methods

Given the high quantities of missing data present in medical datasets and levels of class imbalance, methods to interpolate between missing data points and methods to generate new synthetic samples are often explored as possible ways to improve model performance. This thesis uses multiple resampling methods for this purpose, which are described in this section.

2.3.1 Linear Interpolation

Linear interpolation is a simple method for resampling from continuous signals with discrete data points. Given two points, (x_1, y_1) and (x_2, y_2) , on a Cartesian plane, linear interpolation calculates the y value for a new x value within the range $[x_1, x_2]$ by assuming a linear relationship between the points. The formula for linear interpolation is given by:

$$y = y_1 + \left(\frac{x - x_1}{x_2 - x_1} \right) \times (y_2 - y_1) \quad (2.8)$$

Here, x is the point at which the value y is estimated, by considering the slope between (x_1, y_1) and (x_2, y_2) .

2.3.2 Undersampling and Oversampling

Oversampling is used in settings with high levels of class imbalance: machine learning methods typically show a bias towards the majority class in datasets, resulting in poor performance on the minority class. This is particularly pertinent in medical tasks, where the rare minority class may be the important event in question. Methods to address this can include re-weighting of loss functions, undersampling (where data from the majority class is removed) and oversampling (where duplicate data from the minority class or synthetic data based on the minority class is added).

SMOTE

Synthetic Minority Over-sampling (SMOTE) is a commonly used oversampling method. The procedure can be summarised as follows:

- **Selection:** choose a minority class instance a randomly.
- **Neighbourhood Identification:** determine the k nearest neighbours of a within the minority class.
- **Synthetic Instance Creation:** for each neighbour, calculate the difference between the neighbour and the selected instance. Multiply this difference by a random number between 0 and 1, and add the result to the selected instance to create a synthetic instance.

Mathematically, the synthetic sample s is generated using:

$$s = a + \lambda \times (\text{neighbour} - a) \quad (2.9)$$

Where λ is a random number between 0 and 1.

2.3.3 Gaussian Processes

Gaussian Processes are a family of methods that utilise the conditioning and marginalisation of Gaussian distributions to make predictions. Gaussian Processes can be used for regression, classification, clustering, and interpolation, amongst other uses.

A Gaussian Process is defined as a collection of random variables, any finite number which have a joint Gaussian distribution. It typically is defined by a mean function $m(\mathbf{x})$ and a kernel function $k(\mathbf{x}, \mathbf{x}')$ used to compute covariance. It is denoted as:

$$\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (2.10)$$

Where $\mathbf{x}$ and $\mathbf{x}'$ are inputs.

Kernel functions, $K(\mathbf{x}, \mathbf{x}')$, are functions which compute similarity between two inputs. In the context of Gaussian Processes they are used to compute the covariance between the inputs $\mathbf{x}$ and $\mathbf{x}'$. A popular kernel used is the Radial Basis Function (RBF) kernel, which is defined as:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right) \quad (2.11)$$

Where l is the length scale parameter.

For the use of Gaussian Processes in inference, consider a regression problem with inputs X and observations $\mathbf{y}$. Assuming a Gaussian Process prior over the functions f , the prior is expressed as:

$$f \sim GP(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}')) \quad (2.12)$$

Given observed data $D = \{\mathbf{X}, \mathbf{y}\}$, the posterior over the function values f_* at new points X_* is a Gaussian distribution with mean and covariance given by:

$$\bar{f}_* = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y} \quad (2.13)$$

$$\text{cov}(f_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \quad (2.14)$$

Where $K(X, X')$ is the covariance matrix computed using a kernel function, and σ_n^2 is the noise term.

While Gaussian Processes provide flexible and robust modelling, a notable challenge lies in their computational complexity, $\mathcal{O}(n^3)$, due to the inversion of the $n \times n$ covariance matrix K .

2.4 Genetic Programming

Genetic Programming (GP) is an evolutionary computation technique that is inspired by biological evolution. While it can be considered dated by modern methods, it can be useful for exploring possible search spaces in low resource settings, and it is used in chapter 6 of this thesis. It involves the generation of many candidate programs, and the progressive selection, combination and mutation of these programs to improve performance on a specific goal. The basic steps for this are as follows:

- Initialisation: create an initial population of candidate solutions, represented as programs.
- Evaluation: evaluate the fitness of each program in the population based on how well it solves the problem.
- Selection: select programs to act as parents based on their fitness scores. Programs with higher fitness have a higher chance of being selected.
- Crossover (Recombination): combine the information of two parent programs to create one or more offspring programs.
- Mutation: apply small, random changes to some programs in the population.
- Replacement: form a new population for the next generation by selecting programs from the current population and the generated offspring.
- Termination: repeat the evaluation, selection, crossover, mutation, and replacement steps for a number of generations, or until some termination condition is met (e.g., a solution is found that satisfies minimum criteria).

GP is well-suited to automatically constructing machine learning pipelines, as a pipeline can include multiple steps that can be optimised effectively with crossover, and hyperparameters can be optimised effectively with mutation.

2.5 Tree-Based Models

2.5.1 Decision Trees

A Decision Tree is a flowchart-like tree structure where an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The root node is the topmost node in a Decision Tree. Mathematically, the process of splitting the nodes for classification purposes can be expressed as minimising some cost function. This is typically entropy, calculated as:

$$H(T) = - \sum_{i=1}^c p_i \log_2 p_i \quad (2.15)$$

Where p_i represents the probability of class i belonging to a particular node of the tree, and c is the number of classes present in the classification task.

2.5.2 Random Forests

Random Forests are an ensemble machine learning method using Decision Trees. This method uses the bagging (Bootstrap Aggregating) technique, where multiple Decision Trees are trained on different subsets of the training dataset, and sampled with replacement. Mathematically, given a dataset D of size N , bagging generates m , new training sets D_i , each of size N' , by sampling from D uniformly and with replacement. The Random Forests algorithm can be summarised as follows:

- Draw n bootstrap samples from the original data.

- Grow n trees by randomly selecting k variables at each node and splitting the node using the variable that provides the best split, according to some objective function.
- Aggregate the predictions. For regression, compute the average of the n predictions. For classification, take a majority vote of the n predictions.

Variable importance in a Random Forest can be calculated by permuting the values of each variable and observing the increase in the model's error rate. The larger the increase, the more important the variable is considered to be.

2.5.3 Gradient Boosting Trees

Gradient Boosting Trees are based on the principle of optimising a loss function by iteratively adding trees to an ensemble, where each new tree is a weak learner that corrects the errors of the whole ensemble learned so far. This requires the iterative computation of the gradient of the loss function with respect to the prediction of the model at each observation (in this context known as pseudo-residuals). The algorithm for this is defined in algorithm 1.

2.5.4 XGBoost

XGBoost [75] is a commonly used software library in for gradient boosted trees. In comparison to standard gradient boosted trees, XGBoost has several improvements which can often lead to empirical performance improvements. These improvements include the use of optimisation via Newton's method using second-order Taylor approximations, and new regularisation terms. A generic unregularised XGBoost algorithm using an input training set $\{(x_i, y_i)\}_{i=1}^n$, a differentiable loss function $L(y, F(x))$, and number of weak learners M is formulated in algorithm 2.

Algorithm 1 Gradient Boosting Tree Algorithm

-
- 1: **Input:** Training data $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$
 - 2: **Initialise:** $F_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$, where $F_0(x)$ is the initial model γ represents model parameters, and $L(y, \gamma)$ is the loss function.
 - 3: **for** $m = 1$ to M boosting operations **do**
 - 4: Compute the pseudo-residuals for each observation:

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)},$$
 where r_{im} is the pseudo-residual for the i -th instance at the m -th iteration
 - 5: **for** $i = 1, 2, \dots, N$ **do**
 - 6: Fit a weak learner $h_m(x)$ (typically a shallow Decision Tree) to the pseudo-residuals:

$$h_m(x) = \arg \min_h \sum_{i=1}^N (r_{im} - h(x_i))^2,$$
 - 7: Compute the multiplier γ_m by solving:

$$\gamma_m = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)),$$
 where γ_m is the amount by which the m -th weak learner is scaled to minimise the overall loss when added to the current model.
 - 8: **end for**
 - 9: Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x),$$
 - 10: **end for**
 - 11: **Output:** Final model $F_M(x) = F_0(x) + \sum_{m=1}^M \gamma_m h_m(x)$, which is the sum of the initial model and all the scaled weak learners.
-

XGBoost Feature Importance Explanations

In XGBoost, Gain is used as a metric to measure the quality of a split in the Decision Tree. It quantifies how much each feature contributes to improving the model's prediction, and it is used to rank the importance of each feature. Gain measures the improvement in accuracy brought by a feature to the branches it is on, and can be used as a feature importance score. To formally define Gain, suppose a Decision Tree function is defined as:

$$f(q) = w_{q(x)}, \text{ where } q : \mathbb{R}^m \rightarrow T \quad (2.16)$$

Where q is a function mapping an instance x to the index of a leaf node, w is a vector of leaf weights, and T is the set of leaves. The Gain, G , from

Algorithm 2 XGBoost Implementation

Input: Training data $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, where x_i represents the features and y_i the target value of the i -th instance.

Initialise:

$m = 0$

$\hat{f}_{(0)}(x) = \arg \min_{\theta} \sum_{i=1}^N L(y_i, \theta)$, where $\hat{f}_{(0)}(x)$ is the initial model and θ are parameters minimising the loss function $L(y, \theta)$.

while $m < M$ number of boosting iterations **do**

 Compute the gradient statistics for each instance:

$\hat{g}_m(x_i) = \left[\frac{\partial L(y_i, \hat{f}(x_i))}{\partial \hat{f}(x_i)} \right]_{\hat{f}(x)=\hat{f}_{(m-1)}(x)}$, the gradient of the loss w.r.t. the prediction of instance i .

$\hat{h}_m(x_i) = \left[\frac{\partial^2 L(y_i, \hat{f}(x_i))}{\partial \hat{f}(x_i)^2} \right]_{\hat{f}(x)=\hat{f}_{(m-1)}(x)}$, the hessian of the loss w.r.t. the prediction of instance i .

 Fit the weak learner $\hat{\phi}_m$ to the gradient statistics:

$$\hat{\phi}_m = \arg \min_{\phi \in \Phi} \sum_{i=1}^N \frac{1}{2} \hat{h}_m(x_i) \left[-\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} - \phi(x_i) \right]^2$$

 Scale and add the weak learner to the ensemble:

$\hat{f}_m(x) = \alpha \hat{\phi}_m(x)$, where α is the learning rate.

 Update the model:

$$\hat{f}_{(m)}(x) = \hat{f}_{(m-1)}(x) + \hat{f}_m(x)$$

$m = m + 1$

end while

Output: The final model $\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^M \hat{f}_m(x)$.

splitting a leaf is given by:

$$G = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \quad (2.17)$$

Where I_L and I_R are the instance sets of left and right children after the split, I is the instance set of the current node, g_i is the first order gradient for instance i , h_i is the second order gradient for instance i , λ is the regularisation term on the leaf weights, and γ is the regularisation term on the number of leaves.

In practice, for calculating feature importance scores, the Gain of each feature across all trees in the ensemble is summed up, and the scores are normalised

such that they sum to 1:

$$\text{Feature Importance, } FI(f) = \frac{\sum_{t=1}^T G_t(f)}{\sum_{t=1}^T G_t} \quad (2.18)$$

Where $G_t(f)$ is the Gain of feature f in tree t and T is the total number of trees in the ensemble.

2.6 Multinomial Naive Bayes

Naive Bayes models can be used for classification when a dataset consists only of binary variables. In a multinomial Naive Bayes model, feature vectors are considered to have been generated from a multinomial distribution $p_1, \dots, p_n$ where p_i is the probability that event i occurs. The likelihood of feature vector x given class k has been observed (C_k) is given by:

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_{i=1}^n x_i)!}{\prod_{i=1}^n x_i!} \prod_{i=1}^n p_{ki}^{x_i} \quad (2.19)$$

This is converted into a linear classifier when expressed in log-space:

$$\begin{aligned} \log p(C_k \mid \mathbf{x}) &\propto \log \left(p(C_k) \prod_{i=1}^n p_{ki}^{x_i} \right) \\ &= \log p(C_k) + \sum_{i=1}^n x_i \cdot \log p_{ki} \\ &= b + \mathbf{w}_k^\top \mathbf{x} \end{aligned} \quad (2.20)$$

Where $b = \log p(C_k)$ and $w_{ki} = \log p_{ki}$.

2.7 Shapley Additive Explanations

Shapley Additive Explanations [76], or SHAP, is a model-agnostic machine learning interpretability tool that evaluates the additive feature importance of individual features to predictions agnostic of feature order. Formally, SHAP values are

computed as a weighted average of all possible differences in scores are computed on all feature subsets $S \subseteq F$, where F is the set of all features. This is formulated as:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \quad (2.21)$$

Where $f_{S \cup \{i\}}$ is a model trained with that feature present, and f_S is a model trained with the feature withheld. Not all these values can be computed, so a game-theoretic approach is used to sample the SHAP value. SHAP values are not causal predictors as they use observational conditional distributions rather than interventional distributions [77]. However, they do provide a useful subset of variables for consideration that can be clinically studied and evaluated, thus informing the scientific discovery of causal relationships.

2.8 Hyperparameter Search Methods

2.8.1 Grid Search

Grid Search is a basic method for finding the hyperparameters for a model, which involves exhaustively searching through a manually specified subset of the hyperparameter space of a learning algorithm. In comparison to random searches, grid searches will have a defined number of searches to perform. While grid searches can be computationally intensive if many hyperparameter options are explored, they allow for more systematic exploration of behaviour than random or Bayesian searches, which can be useful in interrogating the properties of machine learning models under different settings. Indeed, we make use of grid search for this purpose in chapter 5 to interrogate new models.

2.8.2 Bayesian Hyperparameter Optimisation

Bayesian Hyperparameter optimisation uses Bayes' Theorem to iteratively explore a hyperparameter search space and improve guesses for hyperparameters as more information becomes available through testing. It relies on two components, a surrogate model and an acquisition function. The surrogate model models the relationship between hyperparameters and the objective, and the acquisition function is used to pick new hyperparameters to test. Each experimental run is used to improve the surrogate model, which in turn guides the acquisition function to make more informed selections of hyperparameters to test.

Surrogate Models

Surrogate Models are used to model the objective function, i.e. a surrogate model of the actual objective we are optimising. An example would be a surrogate model that models loss function of a deep learning model, or the maximum Area Under Receiver Operating Curve (AUROC) of an ensemble tree model. The surrogate models used in this thesis for Bayesian hyperparameter optimisation in chapter 7 are Gaussian Processes. In this Gaussian Process, points in the input space are the hyperparameters chosen. As mentioned before, the Gaussian Process posterior can be updated with Bayes' theorem as new data points are observed:

$$p(f|x, D) = \frac{p(x|f, D)p(f|D)}{p(x|D)}$$

Where D represents the observed data.

Acquisition Functions

Acquisition Functions are used to select which next point to evaluate in hyperparameter space. Such functions make a trade-off between exploitation (assessing

current areas of hyperparameter space modelled by the surrogate model where performance is good), and exploration (assessing areas of hyperparameter space where the surrogate model has assigned low probability to its estimates). The optimisation used in chapter 7 makes use of the Expected Improvement (EI) acquisition function. The EI at a point x can be calculated as:

$$EI(x) = \mathbb{E} \left[\max(0, f(x) - f(x^+)) \right]$$

Where $f(x^+)$ is the objective function value at the current best point. The model then samples a hyperparameter point from the acquisition function probabilistically, with greater probability weight being placed on hyperparameter combinations with higher EI scores.

3

Deep Learning Methods for EHR Data

3.1 Introduction

3.1.1 Limitations of Traditional Machine Learning for EHR Data

In the previous chapter of this thesis, we detailed non-deep learning-based machine learning methods applicable to data from EHRs for use in prediction and classification tasks. While these models have many strengths, there are some key weaknesses that affect their suitability for medical data. The majority of machine learning methods we have assessed so far are typically well-suited for data with only one feature dimension: traditional machine learning models typically map a fixed-length feature vector X per patient to outcome y , where each feature vector structure is consistent between the same versions of the task. EHR data does not by default conform to this structure.

While medical records include static records (ethnicity, biological sex etc.), the majority of medical records are longitudinal records, which contain values that can change over time [78]. The timescale of these events vary greatly: for

example, vaccinations are typically events that occur infrequently throughout a patient’s life [79], whereas time in an ICU can result in a brief period of records of blood pressure and pulse rate to be recorded multiple times an hour. Typically, to use time-series data in traditional machine learning, these time-series features are distilled into a vector, requiring explicit feature construction. Examples of possible constructions include mean values or mode values for specified time periods. However, choosing what features to construct and ultimately use is an undefined problem. Testing every possible combination of features is infeasible. As such expert guidance is typically used to guide feature selection: a clinician with awareness of the condition may use their understanding of time-courses of conditions and clinically relevant states (e.g. low blood pressure being more relevant than high blood pressure for a specific condition [80]) to guide how features should be selected. When such experts are not available, correct feature selection may be significantly more difficult. Furthermore, for some patients, features may be missing for periods of time. Methods expecting feature vectors of fixed-lengths may poorly tolerate such missing data. Missing data can, however, be addressed by using imputation techniques [81]. These imputation techniques are not without their own problems: they may introduce bias into the data or may not be able to handle missing data in a principled way [82].

Deep learning methods address many of the apparent limitations of non-deep learning-based methods. These models are more flexible in their structure and are well-suited to data with more than two dimensions and data of uneven lengths. These models can also incorporate more complex representations of features, space, and time, allowing for uneven spacing and in some cases removing the requirement for hand-crafted features.

3.1.2 The Goals of this Chapter

The goal of this chapter is to introduce deep learning as a method for performing machine learning on EMRs. The basic formulation of deep learning is outlined, along with common challenges that are encountered in the field. This chapter will then go on to describe Recurrent Neural Networks (RNNs), a family of models that are particularly well suited to the task of learning from sequential data with large numbers of features. The advantages, limitations and methods for training these models will be described. Following this, Transformers will be described, an architecture which addresses some of the shortcomings of RNNs.

3.2 An Overview of Deep Learning

Deep learning refers to a family of models that are characterised by multiple layers of operations. These layers are typically, but not always [83], trained together, and these models use a large number of parameters to learn complex functions from data. The canonical text on deep learning is *Deep Learning* by Goodfellow, Bengio, and Courville [84], which delves into these models in more depth. Deep learning has been used effectively in many settings, including computer vision, speech recognition, natural language processing, and bio-informatics [85]. In the field of clinical prediction, deep learning has been applied to a variety of problems, including cancer outcomes [86], predicting neurodegeneration [87], and the prediction of mortality in patients with COVID-19 [88].

3.2.1 The Basic Formulation of Deep Learning Layers

Let us define f_L an L -layer neural network, for $l = 0, 1, \dots, L$. Layer 0 and layer L represent the input and the output layers, respectively. The output of layer

l is denoted by x_l . The input to layer l is the output of layer $l - 1$, i.e., x_{l-1} . The output of layer l is computed as:

$$x_l = f_l \times (W_l \times x_{l-1} + b_l), \quad (3.1)$$

Where W_l is the weight matrix, b_l is the bias vector, and f_l is the activation function.

The output of the neural network is given by:

$$x_L = f_L \times (W_L \times x_{L-1} + b_L), \quad (3.2)$$

The training of the neural network is to find the optimal values of the weight matrices and bias vectors, i.e., W_l and b_l , for $l = 1, 2, \dots, L$, such that the output of the neural network is close to the desired output. These layers are known as “hidden” layers because they are layers in between the input and output steps.

Activation Functions

Without the use of activation functions, the combined multi-layer operations of neural networks would collapse to an operation that is equivalent to a single linear layer, preventing non-linear relationships from being learned [89]. Activation functions are thus used between each layer to introduce non-linearity into the network. Activation functions include sigmoid, tanh, ReLU, and Leaky ReLU [84].

The sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.3)$$

The tanh function is defined as:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.4)$$

The ReLU function is defined as:

$$\sigma(x) = \max(0, x) \quad (3.5)$$

The Leaky ReLU function is defined as:

$$\sigma(x) = \begin{cases} x, & x > 0 \\ \alpha x, & x \leq 0 \end{cases} \quad (3.6)$$

Where α is a constant.

The most used activation function is the ReLU function because it is easy to implement and has a low-computational cost [84].

3.2.2 Loss Functions

In deep learning, loss functions are used to measure the difference between the predicted and actual values. The resulting loss scores are used to update the weights of the network during the training process. Loss functions may also be used to evaluate the performance of a trained network. Different loss functions are used for different tasks. For binary classification tasks, commonly used loss functions include cross-entropy loss and hinge loss. For regression tasks, commonly used loss functions include Mean Squared Error (MSE) and Mean Absolute Error (MAE) [84]. For the following loss function definitions, y_i is the actual value of a target, $\hat{y}_i$ is the predicted value of a target, and N is the number of samples.

Cross-Entropy Loss

Cross-entropy loss is a commonly used loss function for binary classification tasks [90]. It is also known as the log loss function. The cross-entropy loss function is defined as follows:

$$L_{CE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (3.7)$$

Hinge Loss

Hinge loss is a commonly used loss function for binary classification tasks [91]. It is also known as the max-margin loss function. The hinge loss function is defined as follows:

$$L_{Hinge} = \frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i \hat{y}_i) \quad (3.8)$$

Mean Squared Error

MSE is defined as:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3.9)$$

Mean Absolute Error

MAE is defined as:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (3.10)$$

Weighted Losses

When there are high levels of class imbalance in data, unweighted loss functions as seen above may lead to poor performance on the minority class, as these minority class examples contribute less to the loss. In many medical settings, adverse events are rare and make up a minority class. As such, re-weighting a loss may be useful. Re-weighting a loss involves multiplying a loss term by a factor that can either be fixed or variable. Two common factors include per-dataset weighting where losses for classes are scaled by the inverse of their proportion across the whole

dataset [92], and per-batch weighting where losses for classes are scaled by the inverse of their proportion in the current training batch [93].

3.2.3 Backpropagation

Initial development of deep learning was slowed by difficulties in assigning error signals to hidden layers of neural networks, preventing the training of deep networks. Error signal assignment now, however, is easily possible through backpropagation. Backpropagation was popularised by the work of Rumelhart, Hinton and Williams in 1986 [94] but can trace its origins back to the 1960s [95, 96].

The backpropagation algorithm is a gradient computation algorithm that uses the chain rule to propagate error signals backwards through the network. The backpropagation algorithm is a powerful tool for training deep networks and has been used to train networks with hundreds of layers [97].

The mechanism of backpropagation is shown below. Let us denote the weight pointing from unit i to unit j in forward propagation as w_{ij} , then the gradient of the n th training example with respect to w_{ij} can be obtained by the chain rule:

$$\frac{\partial C_n}{\partial w_{ij}} = \frac{\partial C_n}{\partial j} \frac{\partial j}{\partial w_{ij}} \quad (3.11)$$

Where C_n is the loss of the n th training example and j is the output of unit j . The first term on the right-hand side of the equation is the error signal at unit j , and the second term is the input to unit j from unit i . By applying this rule repeatedly for each unit in the network, the gradients of all weights in the network can be computed. These gradients can then be used to update the weights using a gradient-based optimisation algorithm.

3.2.4 Gradient Descent

Gradient descent is a first-order optimisation algorithm which aims to find the local minimum of a differentiable function [98]. The idea is to start at some point in the domain of the function and then move in the direction of the negative gradient of the function at that point. The algorithm is described below:

Algorithm 3 First Order Gradient Descent

Input: Initial point $x_0 \in \mathbb{R}^n$, learning rate $\alpha > 0$
for $k = 0, 1, 2, \dots$ **do**
 $x_{k+1} = x_k - \alpha \nabla f(x_k)$
 if $f(x_{k+1}) \leq f(x_k)$ **then**
 $x_{k+1} = x_k$
 end if
end for

The gradient descent algorithm is a simple and effective method for finding local minima of a differentiable function. It is also a special case of the more general class of quasi-Newton methods. Newton's method is a second-order iterative optimisation algorithm for finding a local minima of a twice-differentiable function [99]. The idea is to start at some point in the domain of the function and then move in the direction of the negative Hessian of the function at that point. The algorithm is described below:

Algorithm 4 Second Order Gradient Descent

Input: Initial point $x_0 \in \mathbb{R}^n$, learning rate $\alpha > 0$
for $k = 0, 1, 2, \dots$ **do**
 $x_{k+1} = x_k - \alpha (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$
end for

3.2.5 Optimisation

With the gradients extracted from a loss function using backpropagation, gradient descent can be used to optimise the performance of a neural network by minimising

the loss. Unfortunately, however, the standard formulation of gradient descent can result in a global minima for the loss not being reached, with the loss function instead being trapped in local minima [100]. Furthermore, the correct computation of gradient descent requires the entirety of a dataset to be used in the computation to calculate a gradient. Incorporating such data in one operation may be computationally infeasible, and it may also lead to overfitting on training data and poor generalisability to test data [101]. To this end, alternative formulations of gradient descent are used to reduce loss. These formulations are numerous and include concepts such as momentum, second-order information, and batch computation. Key formulations include Stochastic Gradient Descent (SGD) [84], RMSProp [102], ADAM [103], and LAMB [104], most of which can be combined with momentum [84] and all of which can be combined with lookahead [105].

Stochastic Gradient Descent

SGD involves computing a gradient descent step on a random subset of the dataset. To formulate this, let us consider the general formulation of gradient descent:

$$g_{t,i} = \nabla_{\theta} J(\theta_{t,i}) \quad (3.12)$$

Where g_t is the gradient at time-step t , θ_i is the i th parameter of the network and $g_{t,i}$ is the partial derivative of the loss function with regards to the parameter θ_i at time-step t . SGD involves computing a gradient descent step on a random subset of the dataset. Updates for a set of parameters are performed with the following rule:

$$\theta_{t+1,i} = \theta_{t,i} - \eta \cdot g_{t,i} \quad (3.13)$$

Note here the parameter η , the learning rate. This parameter controls the rate of update to parameters θ .

The batch operation of SGD allows for smaller memory footprints: the need for only a subset of data in memory at any one time allows for larger models to be run on equivalent hardware when compared to full-dataset gradient descent. Furthermore, the stochastic updates in SGD mean that the model has a greater potential to escape shallow local minima in loss surfaces. For this reason, SGD is seen as more robust to the choice of the learning rate than standard gradient descent [106].

Momentum

Momentum is a technique that is used to speed up the convergence of the gradient descent algorithm [84]. Momentum is a term that is added to the gradient descent algorithm to help the algorithm converge faster. The momentum term is added to the SGD algorithm as follows:

$$z^{t+1,i} = \beta z^{t,i} + \nabla_{\theta} J(\theta_{t,i}) \quad (3.14)$$

$$\theta_{t+1,i} = \theta_{t,i} - \alpha z^{t+1,i} \quad (3.15)$$

Where $z^{t+1,i}$ is the momentum term for parameter θ_i at time step $t + 1$, β is a momentum coefficient that controls the influence of the previous momentum term $z^{t,i}$ on the current update, and $\nabla_{\theta_i} J(\theta_{t,i})$ is the gradient of the loss function J with respect to parameter θ_i at time step t , and α is the learning rate.

The introduction of this beta term results in the gradient being calculated as an exponentially weighted sum of previous gradients. Momentum results in an acceleration of gradient descent over periods of high gradient, and a deceleration of gradient descent over periods of low gradient.

Adaptive Gradient Algorithm

Adaptive Gradient Algorithm (Adagrad) [107] is an algorithm that adapts the learning rate for each model parameter individually, based on the historical gradient information for that parameter. This means that the algorithm can automatically adjust the learning rate to be higher for parameters that change infrequently and lower for parameters that change frequently [108], allowing the model to converge more quickly and accurately. This makes Adagrad particularly suited for sparse data, which is relevant to medical settings where some events occur infrequently [109]. In comparison to SGD, the η in Adagrad is modified at every time-step for every parameter θ_i based on the past gradients calculated for θ_i :

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i} \quad (3.16)$$

Where $G_t \in \mathbb{R}^{d \times d}$ is a diagonal matrix where each diagonal element is the sum of squares of the gradients with respect to θ_i up to time-step t . ϵ is a term added for numerical stability. This operation can be vectorised by taking a matrix-vector product, resulting in the following calculation:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t \quad (3.17)$$

The Adagrad removes the need to manually adjust learning rates. However, the accumulation of squared gradients in the above denominator means that learning rates become smaller and smaller, preventing the algorithm from learning additional knowledge beyond a certain point.

RMSProp

RMSProp resolves the vanishing learning rate issue of Adagrad by only using a decaying average of past squared gradients [102]. The running average of gradients $E[g^2]$ at time-step t is formulated as follows:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma)g_t^2 \quad (3.18)$$

Where γ is a constant that can be thought of as a momentum term.

When considering Adagrad, the parameter update vector $\Delta\theta_t$ can be expressed as:

$$\Delta\theta_t = -\frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t \quad (3.19)$$

RMSProp replaces Adagrad's diagonal matrix G_t with

$$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \quad (3.20)$$

Adadelta

Adadelta extends on Adagrad and RMSProp by removing the dependence on learning rate [110]. This is done through the following steps:

The denominator of RMSProp is the Root Mean Squared (RMS) of the error criterion of the gradient, which can be expressed short-hand as:

$$\Delta\theta_t = -\frac{\eta}{RMS[g]_t} g_t \quad (3.21)$$

The Adadelta further defines a weighted sum of squared parameter updates, which is:

$$E[\Delta\theta^2]_t = \gamma E[\Delta\theta^2]_{t-1} + (1 - \gamma)\Delta\theta_t^2 \quad (3.22)$$

The RMS error of these parameter updates is expressed as follows:

$$RMS[\Delta\theta]_t = \sqrt{E[\Delta\theta^2]_t + \epsilon} \quad (3.23)$$

Gradient updates are then formulated with the following rule:

$$\begin{aligned} \Delta\theta_t &= -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t \\ \theta_{t+1} &= \theta_t + \Delta\theta_t \end{aligned} \quad (3.24)$$

Adaptive Moment Estimation

Adaptive Moment Estimation (ADAM) [103] uses moving averages of both squared gradients like Adatelda and RMSProp, and gradients like momentum. These decaying averages of past gradients (m_t) and past squared gradients (v_t) are computed as follows:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \quad (3.25)$$

Where β_1 and β_2 are weighting factors. (m_t) and (v_t) are biased towards zero because of their initialisation. Bias-corrected terms are calculated as follows:

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \end{aligned} \quad (3.26)$$

Updates to parameters are then performed using the following update rule:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (3.27)$$

Layer-wise Adaptive Rate Scaling

Layer-wise Adaptive Rate Scaling, or LARS [111], is an optimisation technique comparable to ADAM and RMSProp in that it adjusts different learning rates in the network differently. However, LARS uses a learning rate for each layer in a network, rather than setting a learning rate for every individual weight. This is formulated in algorithm 5, where γ is a global learning rate, and λ^l is the local learning rate per layer.

Algorithm 5 LARS Algorithm

Parameters: Base LR γ_0 , momentum m , weight decay β ,
LARS coefficient η , number of steps T
Initialise: $t = 0$, $v = 0$. Initialise weight w_0^l for each layer l
while $t < T$ **do**
 for each layer l **do**
 $g_t^l \leftarrow \nabla L(w_t^l)$ $\triangleright$ Obtain a stochastic gradient for the current mini-batch
 $\gamma_t \leftarrow \gamma_0 \cdot \left(1 - \frac{t}{T}\right)^2$ $\triangleright$ Compute the global learning rate
 $\lambda^l \leftarrow \frac{\|w_t^l\|}{\|g_t^l\| + \beta \|w_t^l\|}$ $\triangleright$ Compute the local LR λ^l
 $v_{t+1}^l \leftarrow mv_t^l + \gamma_t \cdot \lambda^l \cdot (g_t^l + \beta w_t^l)$ $\triangleright$ Update the momentum
 $w_{t+1}^l \leftarrow w_t^l - v_{t+1}^l$ $\triangleright$ Update the weights
 end for
 $t \leftarrow t + 1$
end while

The LARS algorithm allows for significantly larger batch sizes when learning, which in turn allows for more rapid training of deep neural networks. However, LARS typically performs poorly on some classes of machine learning models such as attention models (which will be detailed later in this chapter). An extension to LARS known as LAMB uses ADAM as the base algorithm instead of momentum, leading LAMB to have improved performance in Transformer networks [104] when compared to LARS.

Lookahead

Lookahead [105] is an extension that can be applied to any optimiser discussed in this section to make the optimiser less sensitive to the selection of hyperparameters. When lookahead is applied, an optimiser sets two sets of initially identical weights: fast weights θ and slow weights ϕ . A standard optimiser is used to update θ for K steps (generating θ'), at which point the slow weights are updated by a scaling factor based on the difference between fast and slow weights. This is formulated as:

$$\phi' = \phi + \alpha (\theta' - \phi) \quad (3.28)$$

Where α is the slow weights learning rate.

This process is then repeated by resetting the fast weight values as the slow weights. This algorithm is detailed as follows:

Algorithm 6 Lookahead Optimiser

Require: Initial parameters ϕ_0 , objective function L

Require: Synchronisation period k , slow weights step size α , optimiser A

for $t = 1, 2, \dots$ **do**

 Synchronise parameters $\theta_{t,0} \leftarrow \phi_{t-1}$

for $i = 1, 2, \dots, k$ **do**

 Sample mini-batch of data $d \sim \mathcal{D}$

$\theta_{t,i} \leftarrow \theta_{t,i-1} + A(L, \theta_{t,i-1}, d)$

end for

 Perform outer update $\phi_t \leftarrow \phi_{t-1} + \alpha (\theta_{t,k} - \phi_{t-1})$

end for

return parameters ϕ

3.2.6 Challenges in Neural Network Architectures

Internal Covariate Shift

Internal covariate shift is the phenomenon by which the distribution of data going to each layer's inputs changes during training, as the parameters of the

previous layers change. Internal covariate shift occurs in unnormalised neural networks, and can slow training because it forces each layer to continually adapt to a moving target. Activation functions such as ReLU or sigmoid functions, which often have gradients at or close to zero, can in turn exacerbate this problem [112]. This shift in distribution can result in difficulties in training, with lower learning rates being required.

Gradient Explosion and Vanishing Gradients

Without normalisation, iterative transformations applied between layers have a risk of reducing a gradient to zero or causing a gradient to increase exponentially, resulting in a numerical overflow error. These are known as vanishing gradients and gradient explosions respectively [113]. Both events lead to zero gradient information being backpropagated to subsequent layers, preventing deep networks from being trained.

Overfitting and Underfitting

Deep learning models risk being over-parameterised. As such, there is an increased risk of overfitting to data if sufficient regularisation is not used [84]. Deep learning can also be highly sensitive to learning rates, resulting in overfitting and underfitting at higher and lower learning rates, respectively.

3.2.7 Partial Solutions to Challenges in Neural Network Architectures

Batch Normalisation

Batch normalisation [112] is a technique that normalises the input of a layer by subtracting the mean and dividing by the standard deviation of the input. This is done for each mini-batch of data, which is used for the calculation of the mean and standard deviation. This is formulated as:

$$x^* = (x - E[x])\sqrt{\text{var}(x)} \quad (3.29)$$

Where x^* is the new value of a single component, $E[x]$ is its mean within a batch and $\text{var}(x)$ is its variance within a batch.

Batch normalisation further extends this formula to:

$$x^{**} = \gamma \times x^* + \beta \quad (3.30)$$

Where x^{**} is the final normalised value.

γ and β are learnable parameters in every layer. The purpose of these parameters is to allow a layer to perform an identity function if required.

Batch normalisation results in layer outputs being normally distributed. This reduces the chance of exploding or vanishing gradients occurring. This also leads to more stable training due to the elimination of covariate shift.

Dropout

Dropout is a regularisation technique that is used in neural networks to prevent overfitting. It is a technique that randomly drops units (neurons) from the neural network during training. This is done by setting the output of a unit to zero with a probability p [114].

Dropout aims to produce more sparse representations of inputs by preventing networks from relying on single neurons to learn specific examples. This is proposed to lead to improved generalisation capacity.

Early Stopping

Early Stopping involves terminating the training of a neural network once a certain criterion has been met. These criteria can include no significant decreases in

loss between subsequent training epochs, or a certain level of accuracy being met. The objective of this termination is to stop the training process before the model overfits the training data [115].

3.3 Neural Network Architectures

While neural network architectures are highly varied, a subset of architectures focuses on sequential data. Of particular interest are the RNN family and Transformer family of architectures, with a particular emphasis on the inducing point model subset of Transformer architectures.

3.3.1 RNNs

The RNN is a type of neural network that is designed to process sequential data [116, 117]. RNNs are able to learn the temporal dependencies between the input and output sequences. The RNN architecture is shown in figure 3.1.

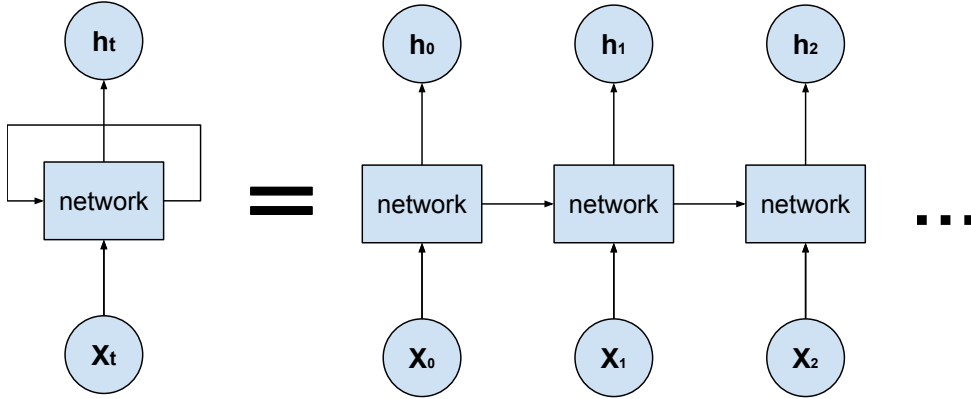


Figure 3.1: An RNN Left: a simplified representation of an RNN, showing a recurrent connection. Right: the unrolled form of the network: at each step, the hidden state h_x from the previous timestep is passed as an input to the next timestep

The RNN architecture consists of an input layer x , an output layer h , and a hidden layer A . The RNN processes a sequential set of data with a structure such as a time-series and outputs a sequence of predictions. The RNN is a type

of neural network that has a feedback loop, which allows it to process sequential data. At every time-step, both the output of the previous time-step and the input of the current time-step are fed to the hidden layer.

The mathematical formulation for this is as follows:

$$h_t = \sigma(W_h \times h_{t-1} + W_x \times x_t + b), \quad (3.31)$$

Where h_t is the hidden state at time t , σ is the activation function, W_x is the weight matrix for the input, W_h is the weight matrix for the previous hidden state, and b is the bias.

The output of the RNN is computed as follows:

$$y_t = W_y \times h_t + b_y, \quad (3.32)$$

Where W_y is the weight matrix for the output and b_y is the bias. The output of the RNN is then fed to a downstream activation function such as a softmax layer for classification tasks.

Long-Short-Term Memory Network

A commonly encountered limitation of a standard RNN is a lack of ability to model long-term dependencies [118]. This is because the gradient of the error with respect to the hidden state decays exponentially with the distance between the current and the previous time-step, which results in a vanishing gradient. To partially address this problem, Hochreiter and Schmidhuber [119] proposed a new type of RNN, called Long-Short-Term Memory (LSTM).

An LSTM is a type of RNN that has a memory cell that can learn long-term dependencies [119]. An LSTM consists of a cell, an input gate, an output gate and a forget gate. The use of these gates regulates the flow of information into

and out of the cell, allowing for memory states to be stored for long periods of time and for long-term dependencies to potentially be modelled. The structure of an LSTM is outlined in figure 3.2.

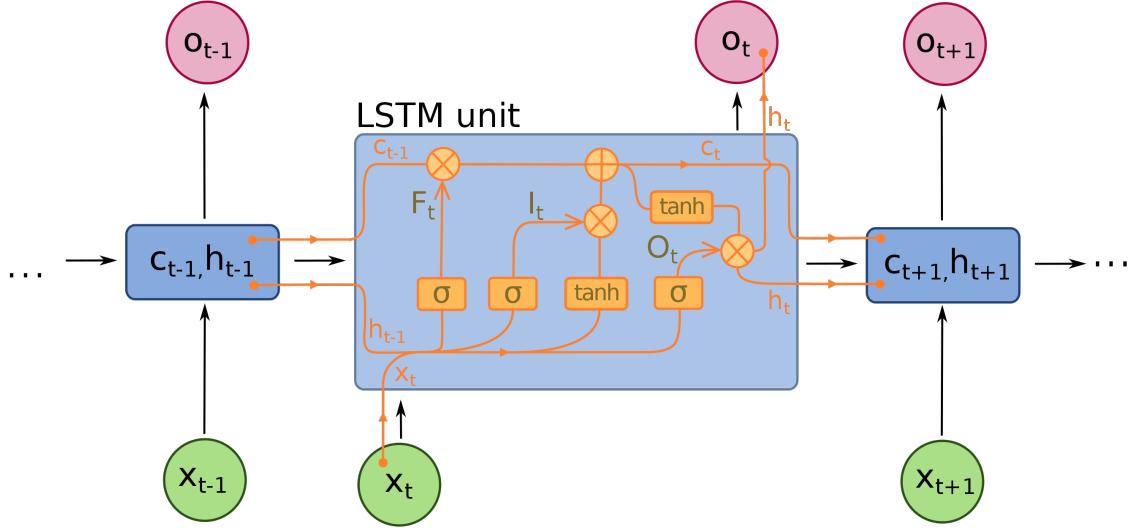


Figure 3.2:

An LSTM network (image reproduced from the Wikimedia Commons under Creative Commons licence). An LSTM contains multiple gates, which are used to potentially retain information between different timesteps. Green items (X_t) represent network inputs, pink items (O_t) represent network outputs, and orange items represent gates and components-wise operations.

The LSTM is formulated mathematically as follows:

$$\begin{aligned}
 f_t &= \sigma_g(W_f \times x_t + U_f \times h_{t-1} + b_f) \\
 i_t &= \sigma_g(W_i \times x_t + U_i \times h_{t-1} + b_i) \\
 o_t &= \sigma_g(W_o \times x_t + U_o \times h_{t-1} + b_o) \\
 \tilde{c}_t &= \sigma_c(W_c \times x_t + U_c \times h_{t-1} + b_c) \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
 h_t &= o_t \odot \sigma_h(c_t)
 \end{aligned} \tag{3.33}$$

Where $x_t \in \mathbb{R}^d$ is the input vector to the LSTM unit, $f_t \in (0, 1)^h$ is the forget gate's activation vector, $i_t \in (0, 1)^h$ is the input/update gate's activation vector,

$o_t \in (0, 1)^h$ is the output gate's activation vector, $h_t \in (-1, 1)^h$ is the hidden state vector, $\tilde{c}_t \in (-1, 1)^h$ is the cell input activation vector, $c_t \in \mathbb{R}^h$ is the cell state vector, and $W \in \mathbb{R}^{h \times d}$, $U \in \mathbb{R}^{h \times h}$, and $b \in \mathbb{R}^h$ are weight matrices and bias vector parameters which need to be learned during training. In a typical LSTM setup, σ_g is the sigmoid activation function (indicated with σ in figure 3.2), and σ_c is the hyperbolic tangent activation function (indicated with $\tanh$).

Gated Recurrent Unit

A Gated Recurrent Unit (GRU) is a type of RNN that can effectively capture dependencies between input sequences [120]. This is achieved using gating mechanisms, which allow the model to selectively learn from, forget, or update its hidden state in response to input. GRUs are similar to LSTMs, but GRUs have a simpler architecture and fewer parameters, making them computationally more efficient and easier to train. This makes them a popular choice for many real-world applications. The GRU is defined mathematically as follows:

$$\begin{aligned}
 z_t &= \sigma_g(W_z \times x_t + U_z \times h_{t-1} + b_z) \\
 r_t &= \sigma_g(W_r \times x_t + U_r \times h_{t-1} + b_r) \\
 \hat{h}_t &= \phi_h(W_h \times x_t + U_h \times (r_t \odot h_{t-1}) + b_h) \\
 h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t
 \end{aligned} \tag{3.34}$$

Where $\odot$ denotes the Hadamard product, x_t is the input vector, h_t is the output vector, $\hat{h}_t$ is the candidate activation vector, z_t is the update gate vector, r_t is reset gate vector, and W , U and b are parameter matrices and vector. σ_g is the sigmoid function, and ϕ_h is the hyperbolic tangent. Initially, for $t = 0$, the output vector is $h_0 = 0$.

Training RNNs

Back-Propagation Through Time (BPTT) is a method used in RNNs to train the network on sequences of inputs [121]. It is an extension of the backpropagation algorithm. The idea behind BPTT is to unroll the RNN in time and treat it as a feedforward neural network, where the weights are shared across time-steps. This allows us to use the standard backpropagation algorithm to compute the gradient of the error with respect to the weights of the network and use this gradient to update the weights and improve the performance of the RNN.

When a sequence is very long, it can be computationally expensive to use BPTT. Furthermore, an increased number of gradient steps through the same layers increases the chance of a gradient explosion or decay. As an alternative, Truncated Back-Propagation Through Time (TBPTT) can be used. TBPTT limits BPTT to n time-steps. This improves computational efficiency, but error signals from earlier time-steps are not used to update the weights of the network [122].

Limitations of RNNs

RNNs are particularly susceptible to exploding and vanishing gradients because of the way they compute the hidden state at each time-step [123]. In an RNN, the hidden state at each time-step is a function of the input at that time-step and the previous hidden state. This means that the error at each time-step depends on the errors at all previous time-steps. If the errors at the earlier time-steps are large, the gradient of the error with respect to the weights of the network can also become large [124]. This can cause the weights of the network to update by a large amount, which can make the network unstable and cause the errors at the later time-steps to become even larger leading to exploding gradients. This is equally true for small error signals, which will cause vanishing gradients.

Exploding gradients can be partly overcome by gradient clipping, which involves setting a threshold on the gradient values [125]. If the magnitude of a gradient is below the threshold, the gradient is left unchanged. However, if the magnitude of a gradient is above the threshold, the gradient is scaled down to have the same magnitude as the threshold. That is if the threshold is set to T , and the gradient of a weight w has a magnitude of $|g| > T$, the gradient of the weight is set to $g' = g \times T/|g|$.

A more difficult limit to overcome in RNNs is the architectural limit of sequential processing [126]: sequences are processed vector by vector, and information in past states is retained only through the hidden state of the network. Even though LSTMs and GRUs allow for information retention via gating mechanisms, the nature of sequential processing means that information from previous time-steps can be lost.

3.3.2 Transformers and Attention

Recent breakthroughs in machine learning on sequences have been driven by attention models [127]. These models are non-sequential in their processing: they process sequences as a whole, rather than token-by-token. This allows Transformers to successfully model long-term dependencies without needing to store information in hidden states. Central to Transformers is the attention mechanism, which is a system for learning relationships between tokens in a sequence.

In attention, three vector embeddings are produced for each token: a query vector Q , a key vector K , and a value vector V . Attention is calculated using a dot product between a query vector and a set of key-value pairs. The query vector is generated by the model and represents the current state of the model's internal representation of the input. The key-value pairs represent the input itself, with the keys representing the different parts of the input and the values representing the

corresponding data for each part. The dot product between the query and each key is used to compute a weight for each key-value pair, indicating the importance of that part of the input to the model's current state. These weights are standardised by a factor $\sqrt{d_k}$ and then normalised via a softmax operation. These weights are then used to compute a weighted sum of the values, which is used as the output of the attention mechanism. Mathematically, this is formulated as follows:

$$\text{Attention, } A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3.35)$$

Where $\sqrt{d_k}$ is the dimension of the key vector k and query vector q .

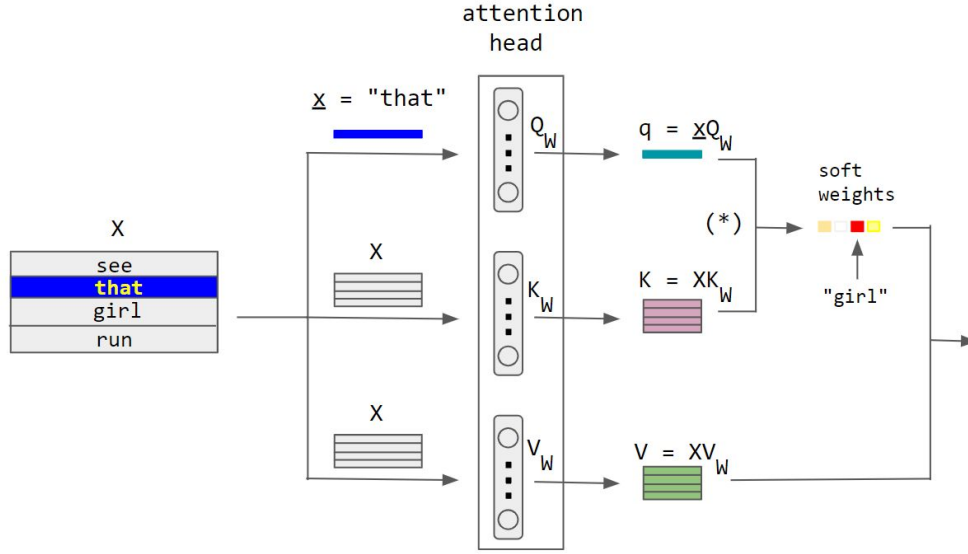
In a Transformer model, multiple attention operations will typically be performed on the same sequence by using MultiHead attention. This involves generating multiple sets of Query, Key and Value vectors per head, allowing for different relationships to be learned in each head. This is formulated as follows:

$$\text{MultiHead, } M(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (3.36)$$

Where

$$\text{head}_i = A(QW^{Q_i}, KW^{K_i}, VW^{V_i}) \quad (3.37)$$

Transformers were first used in sequence-to-sequence translation models [128], but their applications were rapidly expanded to image data [129], time-series data [130] and multi-modal data [131]. The fact that Transformers process a series of tokens also means that they can process variable-length sequences, giving them adaptable functions.

**Figure 3.3:**

A diagram of the attention operation (image reproduced from the Wikimedia Commons under Creative Commons licence). The attention operation involves performing matrix multiplications between three matrices generated from the data: a Query matrix Q (blue), a Key Matrix K (pink), and a Value matrix V (green)

Positional Encoding

Because Transformer models process a series of tokens together, they have no notion of the relative positions of tokens. For this reason, positional information must be added to the embeddings generated from tokens for positional information to be utilised by the model. Embeddings are typically generated using sine and cosine values [128, 129]. These embeddings can be concatenated onto feature tensors or added to existing tensor values.

Sine and cosine positional embedding vectors are constructed as follows: let t be the desired position in an input sentence, $\vec{p}_t \in \mathbb{R}^d$ be its encoding, and d be the encoding dimension. The function that produces this encoding is defined as follows:

$$\vec{p}_t^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k \cdot t), & \text{if } i = 2k \\ \cos(\omega_k \cdot t), & \text{if } i = 2k + 1 \end{cases} \quad (3.38)$$

Where:

$$\omega_k = \frac{1}{10000^{2k/d}} \quad (3.39)$$

An output vector from this function will have the following structure:

$$\vec{p}_t = \begin{bmatrix} \sin(\omega_1.t) \\ \cos(\omega_1.t) \\ \sin(\omega_2.t) \\ \cos(\omega_2.t) \\ \vdots \\ \sin(\omega_{d/2}.t) \\ \cos(\omega_{d/2}.t) \end{bmatrix}_{d \times 1} \quad (3.40)$$

4

Literature Review

4.1 Introduction

As stated in previous chapters, there are difficulties in implementing classical machine learning methods and deep learning methods for healthcare prediction and classification tasks. Difficulties encountered include issues of domain shift between medical datasets [132]; a lower availability of openly-accessible medical data than what is available in other domains [34]; a significant requirement for pre-processing resulting in inconsistent methodologies between researchers [47]; an emphasis on metrics less widely used than in other areas of machine learning (e.g. False Positive Rate (FPR), False Negative Rate (FNR), F1 score) [73]; multiple modalities of data being required to best predict an outcome [68]; class imbalance [38] and fundamental difficulties in predicting and classifying certain medical outcomes and conditions [133]. Such factors mean that the state-of-the-art in machine learning on medical datasets can differ from the state-of-the-art for non-medical datasets.

The main focus of this review is on the different components of machine learning that are relevant to EHR data. It discusses feature extraction techniques,

dimensionality reduction techniques, and possible methods for prediction and classification. This review also discusses some performant models in non-medical domains: this is to both distinguish what makes models performant on medical data and to identify new opportunities to make non-medical models more suitable for medical data.

4.2 Feature Extraction Techniques

The usefulness of EHR data is mostly dependent on the quality of features that can be extracted: EHR data exists in varying time scales and in various linked data formats. It is critical to extract this data into representations that are appropriate for downstream machine learning for prediction and classification tasks [134]. Developing appropriate feature extraction strategies is thus critical to high-quality machine learning in the EHR setting. Key approaches to this include utilizing domain expertise to develop hand-crafted features, statistical feature transforms, applying data standardization techniques, and using representation learning methods where appropriate.

4.2.1 Hand-Crafted Features

A significant proportion of EHR-based machine learning is conducted using hand-crafted features. When performing machine learning for medical tasks, hand-crafted features are often selected by clinicians with a degree of expertise in the field. This was historically done to work effectively with models such as logistic regression models, where typically only small numbers of coefficients can be used to maintain interpretability. There are many disadvantages to using this approach: It requires significant domain knowledge, it can be time-consuming, and it can result in brittleness in pipelines when new datasets are used or data formats change [135,

136]. Furthermore, a bias towards hand-crafted features can make the discovery of previously unknown interactions between factors and outcomes harder. Ultimately, however, hand-crafted features are still a predominant class of features in many medical machine learning settings. Some of their drawbacks can be mitigated with hybrid approaches that combine automatic and hand-crafted features, and the lower dimensional feature representations they produce can be useful for some shallower machine learning tasks. Furthermore, hand-crafted features may be developed that were not originally in the medical record for a specific downstream task. An example might include manual annotation of imaging data from the EHR by clinicians [137].

4.2.2 Feature Transforms

Feature Scaling

Features used in EHR data can exist on a wide range of scales [138]. Multiple tasks in machine learning, such as PCA and gradient descent, function better with equivalently scaled features to prevent disproportionately scaled features from affecting outcomes [84, 139, 140]. As such, it becomes appropriate to scale data.

Multiple normalisation methods can be used [141], but the most commonly encountered normalisation is Z-score normalisation which involves scaling all feature values to have a mean of 0 and a standard deviation of 1. This may not, however, always be appropriate for EHR data as there can be significant skews away from normal distributions [142]. Min-max normalisation can also be used, but this normalisation method is highly sensitive to outliers. This can be a particular issue in medical records, where a value such as haematocrit could be entered with an order of magnitude more zeroes than standard practice based on the preference of individual clinicians [143].

Statistical Features

Common statistical features for continuous data that can be selected on a hand-crafted basis include means, standard deviations, minimum, and maximum values, count data, and percentiles [136, 144]. For categorical data, features may be converted into binary representations (e.g. a feature with three categories becomes three features with one category each) [145], to allow for equivalent statistical features to be generated. These features may typically be chosen on a hand-crafted basis.

Correlation Analysis

In some settings, it can be appropriate to remove correlated features (e.g. for linear and logistic regressions) or identify correlated features for downstream clinical investigation. As such, correlation analysis may be frequently conducted. Correlation analysis can be conducted both over the dimension of features and the dimension of time [146]. It should be noted that such analysis is not equivalent to determining causation, however [147]: causal relationship modelling is an ongoing field in machine learning [148], and methods for this analysis are not yet fully robust [149].

Signal Transforms

Some data in EHR settings can come from devices that produce waveform data, including ECG and electroencephalogram (EEG) data. While machine learning can be conducted on 'raw' unaltered data, the data may first be transformed via wavelet or Fourier transforms, which are used to decompose raw signals into wavelength or frequency components, respectively [150]. These transformations can allow for easier identification of signals relevant to the diagnosis and monitoring of disease states in the spectral space [151]. However, there can remain some

challenges in interpreting results after transforms and choosing the appropriate transform and transform parameters for each clinical task [152].

Hidden Markov Models (HMMs)

HMMs have been occasionally used to capture implicit states in patient data in an unsupervised manner [153]. These models can be used to identify potential underlying states from noisy and partially incomplete signals, which can be useful in the context of EHR data for modelling disease progression stages [154, 155]. However, deciding on appropriate hyperparameters, such as the number of states, can be difficult in the context of using HMMs, and there can be complications in using HMMs for future prediction tasks.

Time Series Standardisation

Multiple methods can be used to standardise time-series data for easier downstream processing [153, 156]. This can include imputation of missing data in settings of irregular sampling or dynamic warping of time series [157]. This is particularly useful in the EHR context because significant amounts of data can be unevenly sampled, despite continuous monitoring being highly valuable for predicting patient deteriorations.

Gaussian Process Regression (GPR)

GPRs use Gaussian Processes to interpolate missing values and extrapolate to new possible values. These models can be very useful in settings where measuring uncertainty is appropriate, and they can also be used to identify potential incorrect outlier data points for censoring [158, 159]. While these models can be useful in certain domains, these models scale poorly to higher numbers of features and there are complications in selecting appropriate kernels for each task [160]. As such,

manual selection or transformation of features is typically performed to allow for tractable space in which to search for an appropriate kernel. This can often lead to GPRs being performed on single patient variables. While potentially accurate, these provide limited visibility into the overall status of a patient, given they are blind to all other variables [159–161]. While significantly more computationally intensive, multi-task GPRs can lead to models with greater predictive capacity for different outcomes [162–164].

Clustering and Novelty Identification

Clustering is an unsupervised technique that can be used to group data into categories based on predefined similarity metrics. This can be useful for creating a smaller set of features for possible downstream tasks. Unsupervised clustering can be particularly useful in settings where there are not known models, such as identifying possible new subtypes of disease or previously unknown patient symptom clusters [165]. Schulam et al. [166] demonstrate that clustering can be used to generate new patient subgroups for conditions that can, in turn, directly lead to the development of new clinical hypotheses and subsequent clinical experiments. More advanced methods, such as Gaussian mixture models, can be used for soft clustering to separate out potentially novel groups in clinical settings [167], but these models scale poorly with data dimensionality, which can limit their use on certain datasets [165].

4.3 Dimensionality Reduction

Given the huge numbers of medical devices and the different environments in which one can obtain medical data, the aggregate combination of data may often contain extraneous, unnecessary or repeated data. As such, it could be

possible to lose some features or combine features in novel ways without a loss of overall information [168]. Without such reduced feature sets, models may be significantly harder to train. This falls broadly under the umbrella of dimensionality reduction, which is often considered appropriate for medical machine learning tasks [169, 170]. Such reduced representations of data can often lead to improved results in medical tasks [171]. Methods can include direct transformations such as PCA, and layer-wise transformations that can be used to produce embeddings [168]. A deeper comparative analysis of dimensionality reduction techniques for unstructured clinical text, including variants of the Linear Discriminant Analysis (LDA) algorithm, is produced by Halpern et al. [172].

4.3.1 PCA

PCA is widely used in dimensionality reduction for clinical features [165]. This technique can also be used for visualising data and interpretability via features that contribute to each component. In its default form, however, PCA is unable to capture non-linear relationships [134]. This can be partially rectified by kernel PCA [173], but this can present issues in selecting the appropriate kernel for the clinical setting.

4.3.2 Embedding Layers

Embedding layers are used to convert data into a space of a fixed dimensionality for a downstream task [174]. In modern iterations, these are typically linear or multi-layer networks. These can include both models that are trained independently [175] and models that are trained as part of a differential pipeline contiguous with downstream models. These layers are used to reduce inputs into a smaller input space for downstream training tasks or to convert non-standard representations,

such as tokens, into a continuous space appropriate for machine learning models. Explicit tokenised embeddings can be used for medical terminology [175], while multi-level embeddings can be used to potentially represent some medical relationships more effectively [176].

4.3.3 Autoencoders

Autoencoders are bottlenecked self-supervised machine learning models that can be used for both dimensionality reduction and feature extraction [177]. A self-supervision signal occurs by forcing data through a bottlenecked portion of a network and forcing the model to reconstruct the original input [178]. Retrieving the activations from the bottleneck layer can be used to generate embeddings. Sadati et al. [179] reviewed various autoencoder models for EHR data, including stacked sparse autoencoders, variational autoencoders, and adversarial autoencoders. Stacked sparse autoencoders were found to perform best on smaller datasets, variational autoencoders performed the best for large medical record datasets. A specific variant of autoencoders that can also be useful in medical settings are Denoising Autoencoders [180]. These encoders work by corrupting the input and tasking a model with recovering the original data. Given the issues with data quality and robustness in medical settings, these autoencoders can be used to both build embedding representations that are robust to noise and to potentially restore missing data. Kim et al. demonstrate this use for handling missing data in medical records [181].

4.4 Classification and Prediction

There have been some difficulties in translating the success of state-of-the-art models in non-medical domains into medical classification and prediction problems,

in part due to the problems mentioned in the introduction of this chapter. As such, many different approaches are being used in the medical field with the aim of finding models that work well on medical data. Some success has been found with the use of RNNs and CNNs for deep learning predictions and classification on EHR data [182–187]. Decision Trees, a more classical method of machine learning, have also proved effective on EHR tasks [5, 188, 189]. Furthermore, some interesting alternative approaches have been used including Reinforcement Learning (RL), graph neural networks, and receiver operating characteristic-specific optimisers. In this section, we describe some models that have achieved particularly high performance on medical datasets.

4.4.1 Regression Models

Regression models are amongst the most basic models usable in machine learning for prediction tasks between dependent variables and independent variables (covariates). However, in their simplicity, these models also allow for a high degree of interpretability. These models also offer robustness and are a key part of the standard analysis pipeline for medical machine learning tasks [190]. Logistic regression is one of the most commonly encountered linear classifier frameworks and is typically the first model applied to problems. It acts as a baseline in medical settings against which more complex models might be compared [191]. The value of such regression comes in part from its interpretability: odds ratios can be calculated directly from the coefficients of the model - assuming the model is a single layer and that coefficients are independent. This can be used for both continuous and categorical data [192]. Further building on logistic regression is linear regression, which is used to predict continuous variables instead of binary categories [190]. These may be less commonly used in medical settings, given the abundance of

categorical classification tasks relative to continuous tasks. Nevertheless, the number of instances of studies of EMR data conducted that use either linear or logistic regressions is too numerous to count. This review highlights a few studies of interest: Churpek et al. compare the impact of outcome selection for logistic regressions in the ICU setting and find that predictive models are of roughly equal quality when predicting between ICU transfer, cardiac arrest CA, and/or death [193]. Cao et al. [144] demonstrate that logistic regression models can be used in ICUs on a minute-per-minute basis for predicting haemodynamic stability, achieving AUROC scores of 0.83 ± 0.03 . Sun et al. [194] conducted a logistic regression of EHR data to find negative descriptions as a function of a patient's ethnicity, demonstrating racial bias in EHR data.

Logistic and linear regressions are among our most used models, but they have some flaws: these models assume both linear relationships between variables and outcomes, and an independence of variables. Non-linear data and interdependence between covariates will break statistical assumptions. While such models can still potentially be used for predictions, their use for generating odds ratios and their clear interpretability is lost under these circumstances.

4.4.2 Tree Models

Tree models such as decision trees and random forests are widely used in medical settings [195–197]. Basic shallow tree models can provide some utility in clinical settings because they can be implemented directly into clinical practice without a computational interface (e.g., in triage) [198]. One example is the laboratory decision tree early warning score (LDT-EWS) developed by Jarvis et al. [199]: a paper-based early warning system based on laboratory test results from 86,472 test results was developed, with an AUROC of 0.755. While interpretable and

easy to use outside of computational settings, single decision trees may have limited representational capacity in some settings due to their parameter count. As such, random forests, gradient-boosted trees, and other ensemble tree methods are regularly used in medicine. While less interpretable than more simple models, such ensemble models often outperform other models in EMR-based tasks: Liu et al. [200] evaluate a set of models including logistic regression, Support Vector Machines (SVMs), random forests and XGBoost (a gradient-boosted tree model) on predicting mortality of patients with acute kidney injury in the eICU Collaborative Research Database - one of the largest ICU databases available consisting of over 200,000 patient visits [45]. The researchers find the best-performing model to be XGBoost, with an AUROC of 79.6%, F1 score of 92.2%, and accuracy of 86.0%. Hyland et al. evaluate the use of machine learning to predict circulatory failure in the HiRID ICU database [188]. Using Gradient Boosted Trees, Hyland et al. achieved an AUROC of 94.0%, and a 90.0% of predicting circulatory failure, with 82% identified more than 2 hours in advance. Hirsch and Gilad-Bachrach propose Set-Trees [189], which are an interesting extension to Decision Trees that combine the models with attention mechanisms. This allows for prediction tasks using sets of variable size, which is otherwise difficult in most machine learning frameworks. Hirsch and Gilad-Bachrach demonstrate the utility of Set-Trees in detecting potential Drug prescription errors in MIMIC-III data.

4.4.3 Deep Learning

Convolutional Neural Networks

CNNs have been used relatively successfully in medical classification and prediction tasks: Pham et al. present a CNN capable of multi-label classification of abdominal chest X-rays [182]. The model is trained in a conditional manner, for example,

training the model to identify Pneumonia when consolidation is present. When evaluated on the CheXpert dataset, Pham et al.'s model achieves a mean AUROC of 94.0% in predicting 5 selected pathologies. While this is amongst the highest accuracies achieved, this is achieved by training the CNN using a known conditional graph that is expert-driven and so may not be scalable to certain scenarios. Oster et al. [184] use an ensemble of a CNN-LSTM network and a SVM to detect subjects with Atrial Fibrillation (AF) in the UK Biobank dataset. A positive AF classification is given only when the SVM classification result agrees with the CNN-LSTM classification result. Using this model, Oster et al. achieve a Positive Predictive Value (PPV) of 97.5% and an F1 score of 84.8%.

Recurring Neural Networks

RNNs have remained dominant in their use for prediction in medical datasets. Some recent examples of interest include DeepSigns, Time-Aware LSTMs, and parallel LSTM-residual networks. Da Silva et al. [185] propose DeepSigns, which is a set of LSTMs used to predict a set of vital signs which are then used to compute the future values of the APACHE II early warning score [186]. On the MIMIC-III dataset, DeepSigns achieves approximately 80% accuracy with predictions 5 minutes into the future. Baytas et al. propose a Time-Aware LSTM (T-LSTM) [187], which handles irregular time intervals in longitudinal patient records: in this model, the elapsed time between events is applied as a discount factor on the previous short-term memory state in the LSTM block. On the Parkinson's Progression Markers Initiative (PPMI) data [201], the T-LSTM sees a marginal improvement in the performance of patient subtyping over LSTM methods that do not have an awareness of the time between events. Miao et al. [202] train a model to predict blood pressure from a one-lead ECG using parallel ResNet-50 and LSTM architectures fed to two fully connected layers. Using this

model, Miao et al. achieve a 0.85 correlation coefficient between predicted and actual values for Mean Arterial Pressure (MAP).

4.4.4 Reinforcement Learning

Reinforcement Learning has been occasionally used in medical prediction tasks and achieves competitive results: Li et al. use a Deep Q Network (DQN) to recommend sequential blood glucose control strategies in patients with Diabetic Ketoacidosis [203]. When trained on MIMIC III, Li et al. demonstrate competitive performance with clinician decision-making regarding survival outcomes. Furthermore, el-Bouri et al. [204] make use of DQNs and Deep Deterministic Policy Gradients (DDPG) for curriculum learning: where data points are fed to a model in a specified order with the aim of improving model performance. When predicting ICU mortality from 11 variables in MIMIC-III, reinforcement learning-driven curriculum learning demonstrates competitive performance.

4.4.5 Other Interesting Models

Other interesting models include the Graph Convolutional Transformer and Deep AUC Maximisation. Choi et al. propose the Graph Convolutional Transformer (GCT) [205], which combines Graph convolutional operations with attention. This model attempts to learn a graph relationship between ICU codes by applying iterative graph convolution operations to a matrix of conditional probabilities between pairs of features in an ICU visit. Compared to a normal graph convolution operation, however, the adjacency matrix used is replaced by a self-attention operation. When evaluated on the eICU database, GCT achieves a 75.3% AUROC on predicting ICU readmission and 90.89% AUROC on predicting mortality. Yuan et al. propose Large-scale Robust Deep AUC Maximisation [206], which provides a new loss function for maximising an AUROC that scales effectively with dataset

size. This loss function is used on deep networks that are pre-trained with standard cross-entropy loss. Yuan et al.’s variant of Deep AUC Maximisation achieves an AUROC of 93.05% on the CheXpert dataset for classifying 14 conditions.

4.5 Successful Models in Non-Medical Domains

4.5.1 Convolutional Neural Networks

In prediction tasks, Liu et al. propose SCINet [207], which makes use of temporal convolution. In comparison to previous temporal convolution networks such as Wavenet [208], SCINet makes use of dilated convolutions and uses future lookahead where appropriate. SCINet achieves an MSE of 31.1% and 18.3% on the ETTh1 and ETTh2 Electricity Transformer Temperature datasets [209] respectively, at a lookahead of 24 time-steps, and a correlation coefficient of 94.16% on the UCI Electricity dataset [210] with a lookahead of 6 time-steps. Work has also been done to combine CNNs with RNNs: Lai et al. [211] propose the Long-and-Short-term time-series network, which conducts convolutions over multivariate time-series and then feeds the convolution layer outputs to RNNs (specifically GRUs). This result is combined with an autoregression to produce a prediction.

4.5.2 Transformers

Transformers have been a predominant architecture for machine learning outside of the space of machine learning, being used in tasks ranging from language generation [212] to robotic movement planning [213]. These networks have quite diverse structures but broadly use weights trained through gradient descent that place more importance on certain sub-sections of input data. This is done via the attention mechanism detailed in chapter 3 of this thesis. Some notable models in different areas of research are discussed below:

In image classification, Dosvitskiy et al. propose the Vision Transformer (ViT) [214]. The ViT overcomes the issue with attention computations scaling quadratically with input size by splitting an input image into patches and feeding linear projections of these flattened patches to the transformer component. ViT achieves 90.45% top-1 accuracy on ImageNet, but this can only be achieved when scaled to two billion parameters [215], making training equivalent networks a difficult task in standard research settings. Attempts have also been made to combine convolutional systems with attention for classification: Wu et al. combine the ViT architecture with convolutional operations, proposing the Convolutional vision Transformer (CvT) [216], which obtains a top-1 accuracy of 83.3% on ImageNet without pre-training on other datasets. Dai et al. alternatively propose CoAtNet [217], which creates a new network block that combines convolution and attention operations into a single operation and achieves an 86.0% top-1 accuracy on ImageNet without pre-training on other datasets. For tabular data, Kossen et al. propose Non-Parametric Transformers (NPTs) [218]. NPTs allow for information from other data points to be attended to when predicting or classifying a data point in a dataset. This is done by applying a per-datapoint attention operation and then a per-attribute attention operation iteratively. Kossen et al. find that NPTs outperform boosted ensemble tree models such as XGBoost [219] and CatBoost [220] on UCI tabular data benchmarks, something which has been previously hard to achieve for deep learning methods [218]. For prediction tasks, Lim et al. propose the Temporal Fusion Transformer (TFT) [221], which processes known past and known future inputs through an LSTM encoder-decoder stack, and then feeds the LSTM outputs to a masked MultiHead attention network. The TFT also makes extensive use of gating mechanisms to skip over unused connections and to flexibly accommodate different datasets and scenarios. The

TFT demonstrates competitive performance on time-series datasets in electricity, traffic, retail sales and stock volatility.

Of particular interest in the attention domain is the Perceiver, proposed by Jaegle et al. [222]. The Perceiver utilises an asymmetric attention mechanism that forces the attention through a tight bottleneck and significantly reduces computational complexity compared to standard attention while making no assumptions about the underlying structure of the data. As such, the Perceiver has demonstrated success in classification tasks across various data modalities, including images, point clouds, audio and video. This is further extended by the Perceiver IO, which can produce a wider variety of outputs than classification, making it suitable for complicated prediction tasks like predicting optical flow [223].

4.6 Conclusion

In summary, a wide range of models are effective for use in machine learning on EHRs. The range of models used also highlights that medical data is not homogeneous in form and that different models can succeed in different settings - even within the restricted domain of EHR data. Furthermore, discoveries of novel performant architectures are being actively made, suggesting that many more effective alternative architectures may be discovered in the future. One notable observation, however, is that transformers are relatively underexplored in the literature for use in medical machine learning tasks. As such, this thesis considers it necessary to conduct a review of transformer architectures in the medical domain to address this gap in the literature. Such analysis will be conducted in the following chapter.

5

Developing Novel Deep Learning Models for Binary Classification on Electronic Health Record Data

This chapter presents an evaluation of RNN-based and Transformer-based deep learning methods for tabular EHR data. The first section describes variations on standard transfer models and the suitability of these models for tabular medical data. Model limitations and non-parametric attention are described as part of this. Inducing point transformers are then described, which are a family of attention models that address some of these limitations. This section details contributions to the field by developing novel architectures using inducing point transformers to perform multi-axis and non-parametric attention.

The second section describes an effort to systematically evaluate different deep learning architectures on medical data. This section starts with an overview of the MIMIC-III ICU dataset [44], which is the dataset chosen for the evaluation tasks, and the justification for its choice. Following this, some experiments are described that apply different models to canonical prediction tasks that use ICU

medical records: predicting mortality, predicting sepsis, and predicting Atrial Fibrillation (AF).

5.1 Introduction: Variations of Attention-based Models

Transformers are most commonly perceived to be the state-of-the-art approach for non-medical machine learning tasks. However, the reason these models are not state-of-the-art in medical machine learning may be in part due to a lack of evaluation of these models and their variants in medical settings. This introduction details both existing variations of attention-based models and novel variations that are proposed as part of this thesis.

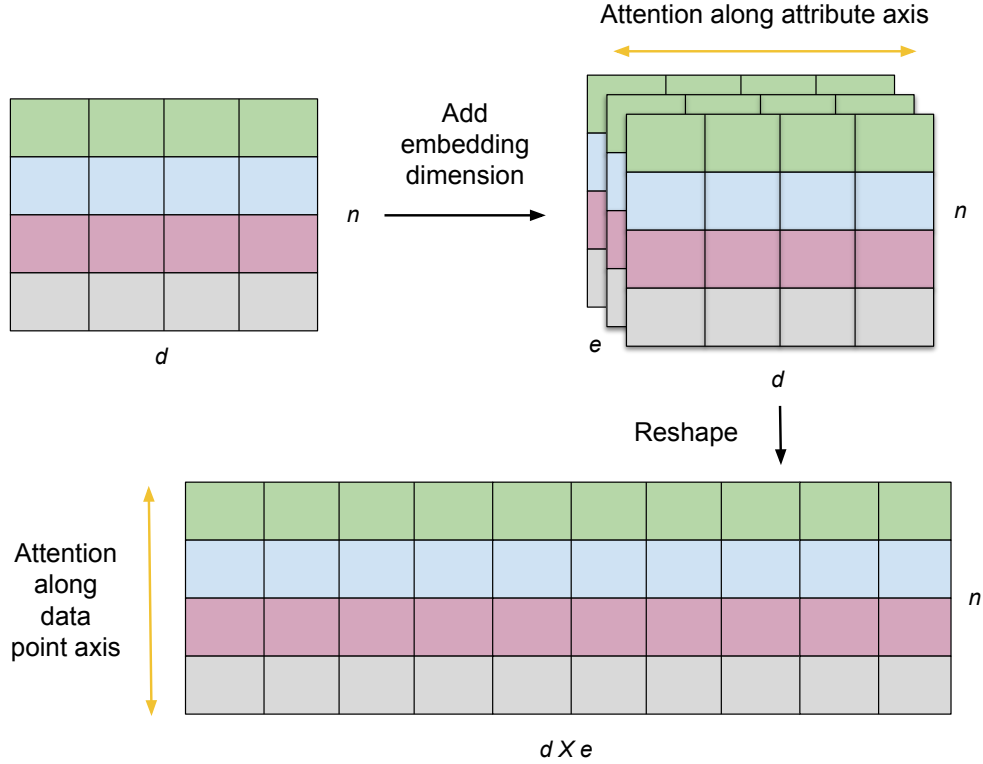
Non-Parametric Transformers

Because Transformers can process information of arbitrary structure, attention can be performed on different tensor axes. While attention is typically performed over feature dimension, Kossen et al. [218], propose NPTs. These NPTs demonstrate high performance on tabular data when performing attention over the batch dimension of tensors. NPTs typically perform attention between data points followed by attention between attributes, this attention over multiple axes is accomplished by iterative reshaping of tensors, as outlined in figure 5.1.

Limitations of Transformers

The most encountered limitation in Transformer models is scalability: most Transformer models perform a self-attention operation, in which query, key and value vectors are generated from the same sequence of n tokens. Comparing queries to key-value pairs results in an n^2 number of operations. This $\mathcal{O}(n^2)$ computational complexity means that performing self-attention on arbitrarily

Figure 5.1: A Non-Parametric Transformer: in an NPT, attention is performed along both attribute axes (d) and data point axes (n), allowing for information to be shared between different data points. This is done via reshaping of data between matrix and tensor forms. Such attention between data points is advantageous for medical data, where high degrees of absence in data points mean that seeing other data points can be informative for prediction



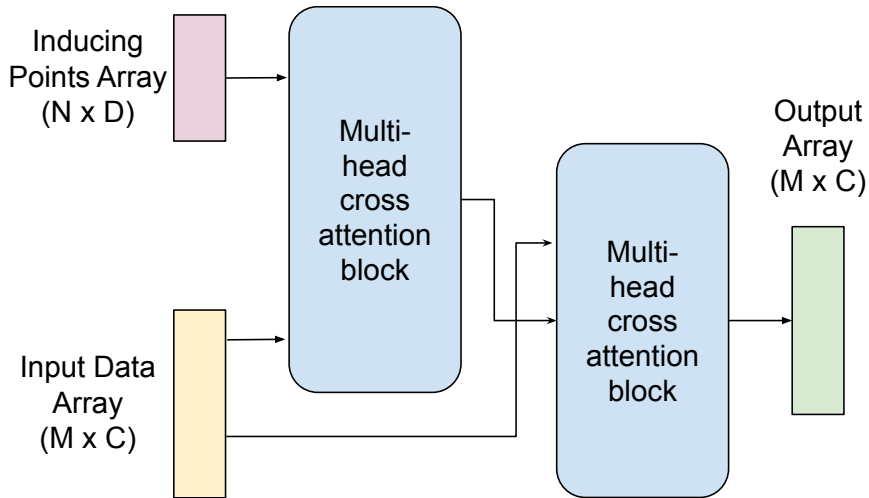
long sequences is infeasible [224]. While some architectures can perform feature selection with masks to direct attention towards certain features [225], the fact that these masks are softmaxes means that information still flows through the network, leading to n^2 computations.

5.1.1 Inducing Point Models

A method to address the limitations of scalability in self-attention is to avoid performing self-attention altogether. Instead, a cross-attention operation is performed between the sequence and a latent vector. These latent vectors are typically called inducing points.

Inducing point models were popularised with the introduction of the Set Transformer [226]. The canonical attention block of the Set Transformer is the Induced Set Attention Block (ISAB), which performs two cross-attention operations on an input. In the first operation, keys and values of length n are generated from input sequence $X^{n \times d}$, whereas queries are generated from a learned vector I of inducing points $I^{m \times d}$. This is fed to a MultiHead Attention Block (MAB). The resulting output $\hat{X}$ has dimensionality $m \times d$. $\hat{X}$ is then used to generate key-value vectors $\hat{K}$ and $\hat{V}$, and a query vector $\hat{Q}$ is generated from the original X input. These are fed to a MAB to produce the final output of the ISAB block. Each of these operations has a complexity of $\mathcal{O}(MN)$ and is thus linear in the input space. The structure of an ISAB block is outlined in figure 5.2.

Figure 5.2: An Induced Set Attention block: the ISAB block consists of two MABs and an array of learned inducing points. Cross-attention is performed first between the inducing points and the input, and then performed between the input and the output of the first cross-attention block. These two actions together allow for attention operations to be performed with linear time complexity that preserve the original input data dimensions



Perceiver Models

Perceiver models [222] extend on Set Transformer architectures by performing self-attention operations in the latent space created by cross-attention operations. This results in self-attention operations of $\mathcal{O}(m^2)$ complexity instead of $\mathcal{O}(n^2)$ complexity. Standard Perceiver architectures do not perform cross-attention to restore the original dimensionality of the input, and are used to perform classification tasks via the use of a neural network classification head. Cross-attention operations to restore the original dimensionality (or any other input dimensionality) can be utilised at the end of the self-attention stack [222]. The Perceiver network structure can be found in figure 5.3.

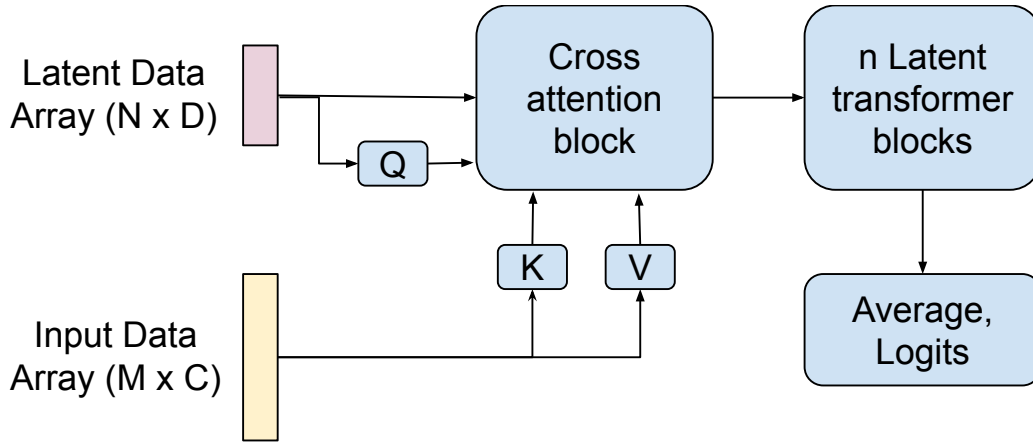


Figure 5.3: Perceiver Network structure: the Perceiver is a system that can handle large, complex inputs like images, videos, audio, point clouds, and multi-modal combinations, without relying on specific knowledge about the domains it is applied to in the first place. It does this by using attention mechanisms to project the high-dimensional input data onto a lower-dimensional latent space, where it can be processed using a series of Transformer-style self-attention blocks. The Perceiver iteratively focuses on different parts of the input data through the use of both cross-attention and latent self-attention blocks

5.1.2 Novel Contributions

While the focus of this research is on evaluating existing deep learning architectures and adapting them for the novel EHR domain, I also wanted to develop my own novel architectures as part of this thesis. These architectures are the 'Non-Parametric Perceiver (NPP)' and the 'Hierarchical Perceiver'. These two models form part of my novel contributions to the field.

Non-Parametric Perceiver

The standard NPT architecture is limited by the computational complexity of the attention operation, as the maximum attention operation that can be conducted is attention over the whole dataset. With the increase in dimensionality from merging the time and feature dimensions, memory overheads are also typically increased. For these reasons, it is believed that the cross-attention mechanisms employed by Perceiver architectures combine well with NPTs. I thus create the 'Non-Parametric Perceiver' (NPP) - a combination of the Perceiver and NPT architectures. The utility of an NPP model is tested on the available data. A normal Perceiver architecture, however, is insufficient to perform non-parametric attention. This is because the cross-attention operation results in the loss of the batch dimension. To resolve this, a cross-attention operation that uses the original input as the Query and the Perceiver output as the Key-Value pair can be used to recover the original input dimensionality [222].

Hierarchical Perceiver

An alternative way to take advantage of the time-complexity advantages of the Perceiver architecture is to attend over multiple axes. As such, I propose a novel perceiver model suitable for 4-dimensional tensors. I name this architecture a *Hierarchical Perceiver* (figure 5.4). Typical tensors in attention have the

dimensionality of $(\text{batch} \times \text{features} \times \text{feature embedding})$. To process time-series, three-dimensional tensors with the structure of $(\text{batch} \times \text{time-steps} \times (\text{features} \times \text{feature embedding}))$ are typically used, which prevents attention operations between features. A Hierarchical Perceiver facilitates attention between features by utilising two stacks of cross-attention layers. The first cross-attention stack is performed on f feature matrices of dimension $(\text{time-steps} \times \text{feature embedding})$ to produce f feature matrices of dimension $(\text{latent dimension} \times \text{feature embedding})$. These are concatenated to produce a vector of dimension $((f \times \text{latent dimension}) \times \text{feature embedding})$, which is fed to a second cross-attention stack. This architecture allows attention to be performed on both feature and time-series axes with linear-time complexity. This system may have particular utility for medical time-series data, where these 4-dimensional tensors emerge. Example 4d medical tensors include time-series data that has an embedding dimension (with the dimensions of batch, timepoint, feature, and feature embedding), and specific time-series medical imaging data (such as functional MRI).

5.2 Research Objective

The objective of this thesis chapter is to evaluate attention-based models on medical data: such models have been underutilised in medical settings, in part perhaps due to their increased computational requirements. Such evaluation can help determine whether these models should be used routinely in medical machine learning, and if such computational requirements can be reduced through non-linear model variations.

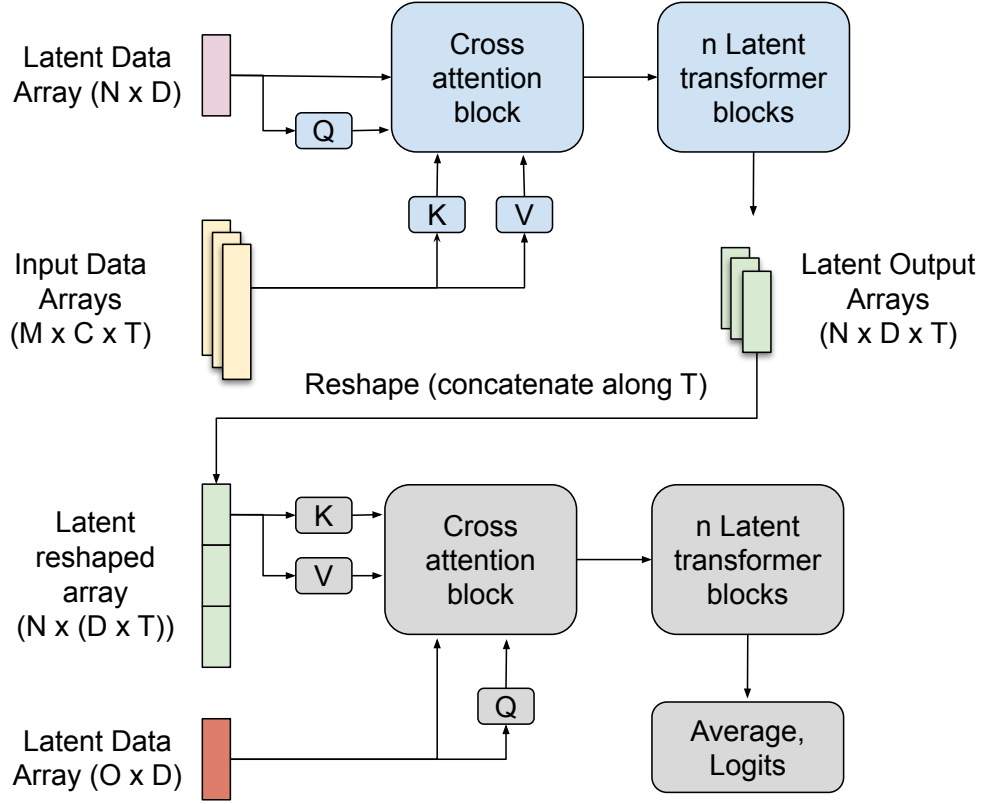


Figure 5.4: Hierarchical Perceiver structure: a Hierarchical Perceiver consists of two separate Perceiver stacks (here indicated in blue and grey). Tensors of input data are passed to the first Perceiver, the latent outputs are concatenated into a vector. This vector is then passed to the second Perceiver to produce a single output

5.3 Dataset Description

Given the objective of this research project is to perform an evaluation of attention-based models on medical data, a widely available dataset should ideally be chosen to allow for both validation and comparison of future attention-based models against baseline scores. However, given the confidentiality issues inherent to medical datasets wide availability is not typically found. There are, however, a small number of de-identified databases maintained by universities that can be accessed by a large number of academics. One of these such datasets, MIMIC-III, was thus selected for this study.

5.3.1 MIMIC-III

MIMIC [44] is a single-centre database from a secondary care hospital in the US consisting of 53,423 distinct hospital admissions from 38,597 patients. The third generation of this database, MIMIC-III, is a widely used database in the medical research community with over 3,000 citations on the original paper. MIMIC-III is a de-identified database: patient-identifiable data has been removed and dates have been shifted into the future by a random offset for each individual patient in a consistent manner (however, time of day, day of week and seasonality have been preserved in shifting). Birth dates for patients aged over 89 have also been censored, (these patients' ages are set to over 300 in the database). Data available in MIMIC-III include:

- Demographics: demographic data available includes admission and discharge dates, date of birth and death, religion, ethnicity and marital status.
- Clinical Measurements and Laboratory Tests: tests taken at bedside (e.g. heart rate, blood pressure and Glasgow Coma Scale) are available along with laboratory test results (e.g. white blood cells count). These datapoints are collected routinely and timestamped.
- Interventions and Pharmacotherapy: a record of interventions in hospital settings (e.g. mechanical ventilation) is provided. Similarly available is pharmacotherapy data which includes a list of medications provided in hospital settings and the method of application.
- Outcomes and Classifications: MIMIC-III is linked to the US deaths registry and thus provides data on mortality outcomes. Other available outcome data are transfer to ICUs and length of stay, along with diagnoses given via

ICD code. These outcomes will be used in prediction tasks to evaluate the performance of machine learning models. Other possible prediction targets include predicting physiological shock and predicting Acute Respiratory Failure. A classification task is identifying ICD code (e.g. phenotyping) from available data.

5.3.2 FlexIble Data-Driven pipeLinE for MIMIC-III Pre-processing

A common issue in medical machine learning is reproducibility: few datasets are repeatedly used in EHR literature, and each individual researcher may use a complex pre-processing pipeline. These pipelines are typically poorly documented, leading to variations between researchers that affect outcome results. This is as true of the MIMIC-III dataset as any other. To avoid this, it was opted to use a set of pre-processed MIMIC-III-based features produced by Tang et al. [227] known as FIDDLE - “the FlexIble Data-Driven pipeLinE”. The standard formulation of FIDDLE provides a pre-processed data matrix of binary features from the MIMIC-III ICU medical record dataset [44] for the following tasks: predicting AF over 4 hours and over 12 hours, predicting shock over 4 hours and over 12 hours, and predicting mortality over 48 hours. This data structure is summarised in table 5.1.

5.4 Data and Task Validation

To illustrate this data and to guarantee the utility of the FIDDLE pipeline, an in-depth look at the four-hour AF prediction task - one of the smallest tasks in the task set - is provided. We can visualise the 4,045 temporal features across each of the four hours to provide some insight into this problem. We conduct this visualisation by performing PCA: we perform one visualisation by flattening the feature/time dimensions and performing a PCA analysis, and a second visualisation

Table 5.1: Dataset Properties

Dataset	N	Time-steps	Static features	Temporal features	Outcome	Class imbalance
AF 4 hours	15,873	4	98	4,045	Binary	18.27% positive
AF 12 hours	14,174	12	96	4,816	Binary	9.65% positive
Shock 4 hours	19,342	4	98	4,522	Binary	14.94% positive
Shock 12 hours	17,588	12	97	5,500	Binary	7.76% positive
Mortality 48 hours	8,577	48	96	7,307	Binary	12.02% positive

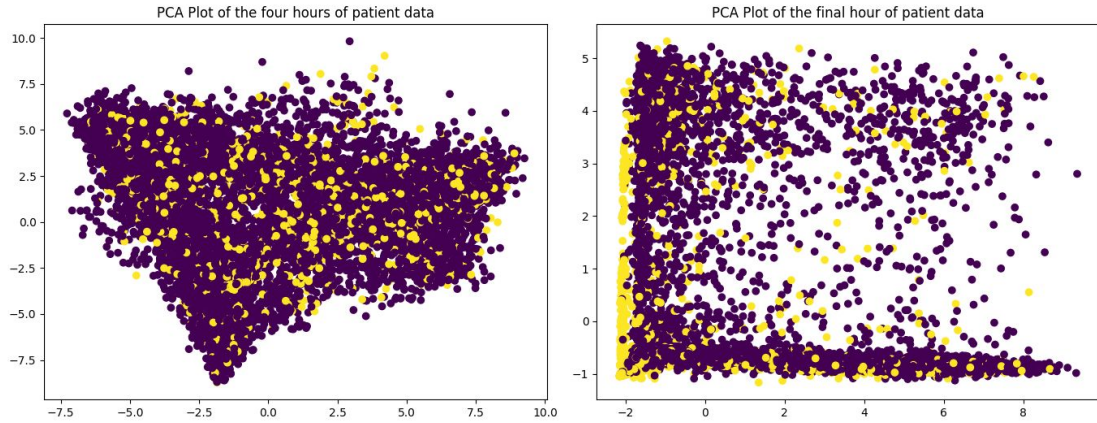


Figure 5.5: PCA of AF data: we observe partial visual separability in the separation of the positive class (yellow) from the negative class (purple) in the last hour of patient data from the 4-hour AF dataset (right). Datapoints are less immediately visually separable when considering all four hours of data in a two-dimensional plot (left)

using only the last hour of data (fig. 5.5). It is found that datapoints are not easily visually separable in the first two principal components of PCA space when all four hours are considered together, but when we consider only the last hour of data we can observe a degree of separation.

To verify that this apparent separation is reasonable and grounded in justifiable medical inference, we can inspect the ‘loadings’ of the first principle component in this last hour of data. The loadings are the contributions of each feature to the

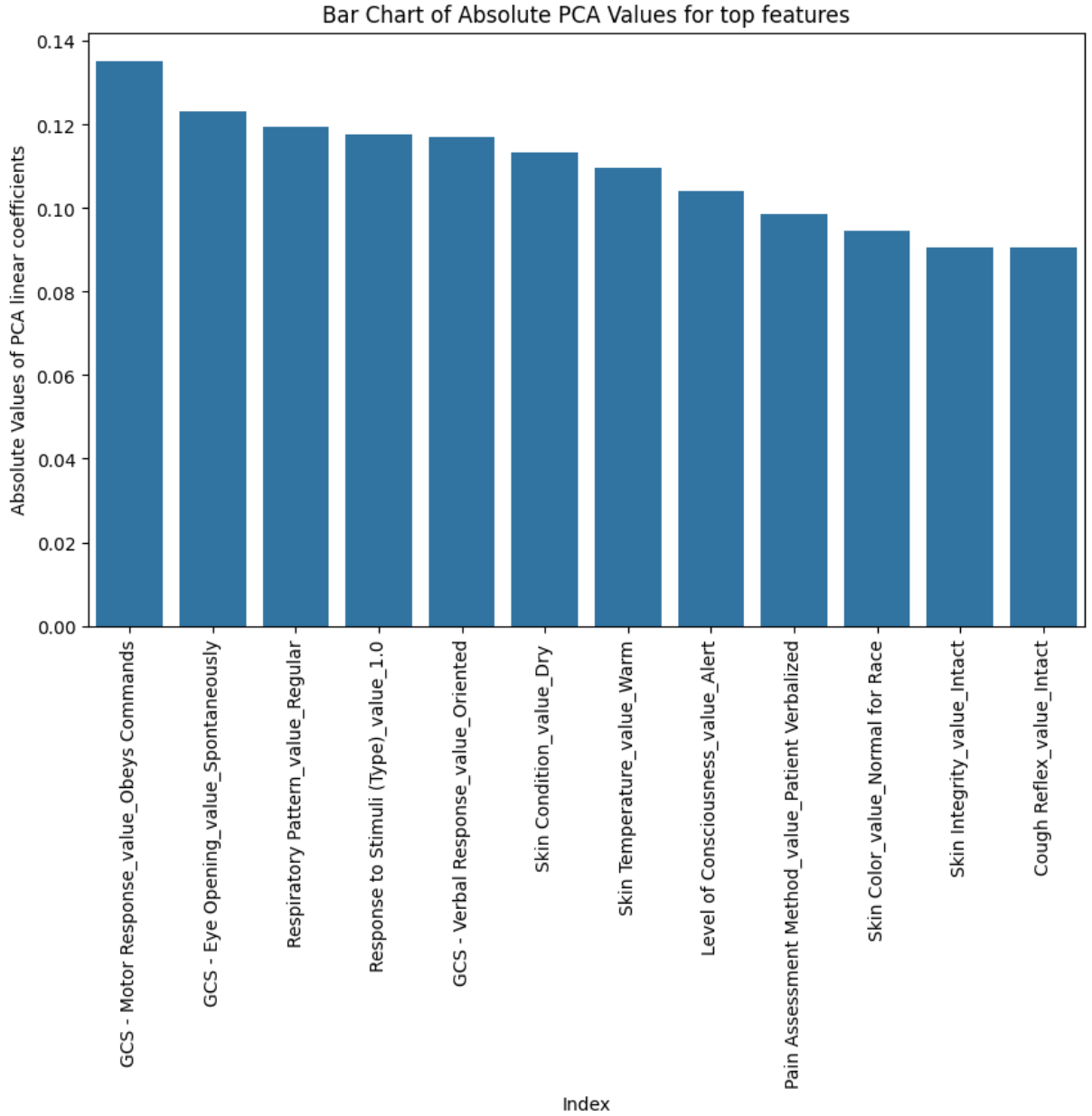


Figure 5.6: A bar chart of high absolute values of coefficients from the first component of the PCA of the last hour of atrial fibrillation data

principle component (i.e. the coefficients of the linear combination of the original variables). Taking the absolute value of these, we can assess which features have the highest contribution to the variance of data. We visualise features with high absolute values of their coefficients in figure 5.6.

In the loadings with the highest coefficients, we find many features corresponding to different components of the Glasgow Coma Scale (GCS), a scale used to measure a patient’s level of consciousness. Altered states of consciousness can be indicative of serious clinical deterioration, and such features could be used to predict AF. To interrogate the top coefficients from fig. 5.6 in more depth, we can produce heatmaps of the average per-hour trajectories of patients who had AF and compare this to the average trajectory of patients who did not have AF in fig. 5.7. Given these features are binary, the fractions in these heatmaps represent the proportion of patients who had each feature on a per-hour basis.

When comparing the two heatmaps of coefficients, we see a clear difference in values across the columns. We see many coefficients corresponding to consciousness (e.g. GCS and verbalised response) as having lower scores in the AF group. Such a difference is consistent with what would be expected in an ICU setting, with unconscious patients typically being more unwell and more at risk of AF. Critically, we also observe a difference in temporal trajectories between the two patient groups, with patients with AF going from 26% responding to commands in hour 1 to 18% by hour 4. In patients without AF, there is a drop of only 3% over the same period. From this data, we thus observe that time trajectory can be a useful feature for making classification decisions. This likely justifies our use of time-series models for these prediction tasks.

Having validated that these datasets and pipelines are clinically accurate and suitable for time-series machine learning, we now move on to testing and evaluating models.

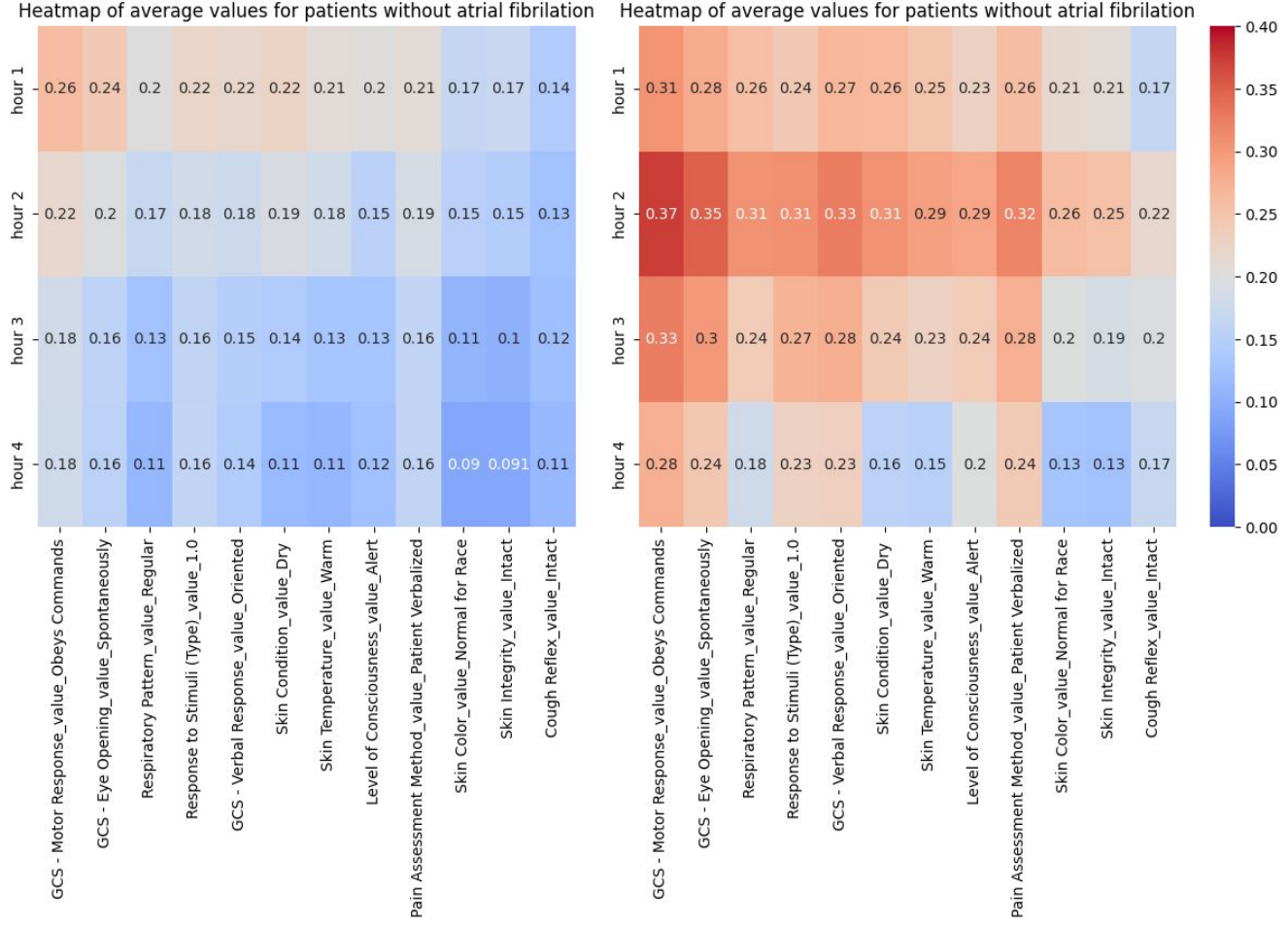


Figure 5.7: Heatmap of average values on a per-hour basis for patients with AF (left) compared to a heatmap of average values on a per-hour basis for patients without AF (right)

5.5 Experimental Setup

We opt to evaluate Attention models (both existing and proposed), baselined against the current routinely used RNNs.

5.5.1 Computational Resources

Models were trained and evaluated on a 32 GB NVIDIA V100 Graphics Processing Unit (GPU).

5.5.2 Model Evaluation

Due to high levels of class imbalance, accuracy is a metric that is insufficient to evaluate model performance. Instead, the F-score (the harmonic mean of precision and recall) and the AUROC are used to evaluate model performance. The Receiver Operating Characteristic (ROC) curve is created by plotting the True Positive Rate (TPR) against the FPR at a range of thresholds. In our experimental setup, a true positive event is a correct identification of a midline shift state.

5.5.3 Model Pipeline

The following standard model architectures were tested on data: RNNs, LSTMs, GRUs, Transformers (attending over the time axis), NPTs (attending over batch axis) and Perceivers (attending over both time axes). RNNs, LSTMs, GRUs and Transformers are described mathematically and visualised in detail in chapter 3 of the thesis. I also tested the performance of the two novel architectures I developed as part of this thesis: NPPs (attending over batch axis), and Hierarchical Perceivers (attending over feature axes then time axes)

All models consisted of an embedding layer followed by a model architecture. Embedding layers were used to convert FIDDLE's binary feature tokens into continuous feature vectors and to reduce the dimensionality of the feature dimension of inputs being fed to the models.

5.5.4 Hyperparameter Optimisation

Before performing cross-validation experiments, a sweep of model hyperparameters was performed to identify the best-performing architectures based on validation AUROC. Different sets of sweeps were performed on RNN class, Transformer class and Perceiver class models as these models have different hyperparameter

sets. In these initial sweeps, model performance was tested over 50 epochs of full data, featuring an 80% to 20% split of training and validation data. Runs were terminated early if AUROC did not increase above 51% after 10 epochs, indicating a failure to train.

It was opted to perform a grid search to determine in what ways parameter selection could influence model performance. This was particularly important for understanding the behaviour of my novel Perceiver-style architectures. Exhaustive grid search sweeps of hyperparameters were performed on the models. However, sweeping over all model hyperparameters simultaneously was infeasible due to computational limitations. As such, three hyperparameter sweep searches were performed per model. In each search, a set of hyperparameters were frozen while another set of hyperparameters was tested. Following each sweep, the top-performing model was used for the next round of sweeps.

For final architectures discovered in hyperparameter grid searches, 5-fold cross-validation over 100 epochs was performed on the models to assess performance. The mean and standard deviations of performance scores are reported in the following section.

Models were evaluated on each of the five datasets. Three sweep configurations were used, the RNN configuration (for RNNs, LSTMs and GRUs), the Transformer configuration (for Transformers and NPTs) and the Perceiver configuration (for the standard perceiver and my two novelly proposed perceiver-style architectures). These sweeps are detailed in appendix A.1.

5.6 Results

Across the initial three sets of sweeps, single runs were performed rather than cross-validation due to computational limitations. A total of 720 model configurations

were tested in the RNN class, with 48 configurations being tested per model per dataset. 640 model configurations were tested in the Transformer class, with 64 configurations being tested per model per dataset. For Perceiver-style architectures, computational limits reduced the scale of configurations that could be tested. For NPPs and the time-axis Perceiver, 88 sweeps per task were performed. For the Hierarchical Perceivers, 20 sweeps per task were performed. Further sweeps were not performed due to the poor performance of the model, leading to the choice to abandon further training. The F1 and AUROC scores of each sweep along with the hyperparameters swept over are reported below.

5.6.1 Sweep 1: Optimisers and Loss Functions

Results for this sweep are reported in tables A.4, A.7 and A.10. The optimiser types swept across were: ADAM, LAMB, RMSProp, Adagrad and Adadelta. The loss functions swept across were: cross-entropy, Binary Cross-Entropy (BCE), globally-weighted cross-entropy, and batch-weighted cross-entropy. Cross-entropy functions were chosen due to their suitability for binary classification problems where probabilistic predictions are preferred [84]. This is relevant when we are attempting to maximise AUROC performance. Fixed parameters are recorded in appendix A.1.

For RNN style models, ADAM and Adagrad performed as the best optimisers, achieving the highest AUROC scores. The utility of Adagrad despite its effect of continually shrinking the learning rate was likely due to its ability to prevent over-fitting, resulting in better AUROC performance on the validation dataset. The success of ADAM is likely due to its incorporation of both first and second-order terms, allowing for optimisation over the complex loss surfaces generated by recurrent architectures. LAMB was ineffective at optimising RNN losses, this

may be attributed to its effect of normalising signals over layers interfering with the propagation of gradients in BPTT.

In comparison to the RNN-class models, LAMB was an effective optimiser on the NPT-class models in terms of AUROC performance. This is to some extent expected given LAMB has been specifically designed with Transformer-class models in mind. The benefits of layer-wise normalisation in LAMB are perhaps more pronounced in the case of NPT models than standard Transformers as NPT models have layer inputs the size of the batch (which in this case is 64) instead of layer inputs the size of the time dimension (which varies from 4 to 48). The performance of Adagrad is likely consistent with the RNN models in that it prevents overfitting, resulting in better AUROC performance on the validation dataset.

For novel architectures, the performance of the Hierarchical Perceiver across runs was exceptionally poor, with many runs failing to surpass an AUROC of 60%. This is potentially due to the hierarchical processing resulting in a complex loss surface, or the embedding models having insufficient capacity to represent data. Regardless, the results of this initial sweep led to the choice to abandon the pursuit of the Hierarchical Perceiver architecture, as these models are the most memory-intensive of all models and took significant computing time. Indeed, the largest Hierarchical Perceiver models for predicting 48-hour mortality all ran into memory errors, resulting in no data being recorded.

In comparison, time-axis Perceivers and NPPs performed well. As with the Transformer models, LAMB and Adagrad were effective optimisers. The justification for the success of these models is also likely equivalent to the justification of the success of base transformer models.

For all models (excluding the Hierarchical Perceiver), weighted loss functions were shown to give a bonus to model performance over non-weighted functions:

these functions allowed for the loss to much more closely align with AUROC, the metric of interest, over accuracy. However, little relationship between the specific loss function and the level of class imbalance was observed.

5.6.2 Sweep 2: Model Dimensions

Results for these feature dimension sweeps are reported in tables A.5, A.8, and A.11.

For all models, the feature embedding dimensions swept across here were 32 and 64. The batch sizes swept across here were 128 and 256. The latent dimensions swept across here were 64, 128 and 256. The neural network depths swept across here were 4, 8 and 12. For Transformer and Perceiver-style models, the number of attention heads used was either set to 8 or 16. Fixed parameters are recorded in appendix A.1.

For recurrent architectures, the performance of the different model classes was comparable after structural architecture selection. There are two key findings in the optimal parameters for each task. Firstly, a latent dimension of 64 was never in the top-performing models. This suggests that a latent dimension of this size is too small to effectively represent the data under the current training regime. Secondly, no architecture with a depth of 12 was in the top-performing architectures. The most likely explanation for this behaviour is the exploding and vanishing gradient phenomena in recurrent networks, which are more prominent in deeper networks [228].

In comparison to RNN class models, Transformer class models had better AUROC performance at deeper model depths. Model depth is important to Transformer-style models as it allows for the mixing of information between heads: at the end of every layer block the output dimension is shuffled, to allow information to be combined between heads. Benefiting from such head-mixing may require

more layers than an RNN, which instead benefits from recurrence as a pseudo-increase to the number of layers. The lack of recurrence means the loss surface of Transformer-style models is also potentially less complex, leading to the conclusion that larger models can be more effectively optimised. The performance of NPT and Transformer-style architectures were very similar. Similar to RNN models, a latent dimension of 64 was never in the top-performing models, which has similar implications for representational capacity.

NPP models required a larger Perceiver cross-head dimension than time-Perceiver models. This is consistent with the fact that in the NPP, the feature dimension is greatly increased as it is the combination of the feature and the time axis. Like the Transformer models, these architectures benefit from a deeper network depth. This is once again likely to be required to support greater head mixing. Having 16 heads instead of the standard 8 heads appeared to result in better AUROC performance. This is understandable given how the cross-attention operation forces information through a bottleneck: each inducing point in a Perceiver model must represent more features than the points in a standard attention block. Having a greater number of attention heads makes this possible.

5.6.3 Sweep 3: Learning Rates

Results for learning rate sweeps are reported in table A.6. For RNN-style models and Transformer models, the learning rates swept over were 10^{-3} , 10^{-4} and 5×10^{-3} . For Perceiver-style models, the learning rates swept over are 10^{-3} and 10^{-4} . For all models, lookahead was either used with the default parameters recommended in the original paper ($K = 5$ and $\alpha = 0.5$) [105] or not used. Fixed parameters are recorded in appendix A.1.

For the task with the largest input matrices (48-hour mortality) using lookahead improved performance across all RNN architecture types. Lookahead is shown to improve learning stability through lowering the variance of the inner optimiser combined with lookahead [105]. This need for stability is likely to be more pronounced when working with longer inputs and computing more complex gradient update steps when backpropagating through time. For NPT models, lookahead also appeared to provide a performance bonus. The benefit of lookahead is likely linked to the large layer-wise operations in NPT models caused by attention across the batch dimension leading to greater variance in gradients. This bonus is not as clearly observed in standard Transformer models, where attention across the time dimension may lead to less variance.

For RNN architectures, models optimised with Adagrad achieved better AUROC performance with slightly higher learning rates, while models optimised with ADAM preferred lower learning rates. For Adagrad, these higher initial learning rates allow for a longer period of learning before learning rates decay to zero. For ADAM, the algorithm is known to be unstable if learning rates parameters exceed the α term, which is set to the default value of 10^{-3} in these experiments [103].

For Perceiver and Transformer models, no particular pattern was observed in model learning rates, nor is there any particular pattern between optimal learning rates and the use of LAMB or Adagrad as an optimiser. Lookahead allows for greater stability across a range of learning rates due to the use of fast and slow weights, which may in part explain this lack of pattern. The majority of Perceiver-style models, however, performed best using the learning rate used on previous sweeps. This may indicate that these models are more sensitive to learning rates when considering their dimensions and optimisers than other models used.

Table 5.2: AF 4-Hour Cross-Validated Overall Results (mean scores ± 1.96 *Standard Error)

Model name	Validation accuracy	Validation AUROC	Validation F1 score
GRU	96.40% $\pm$ 0.82%	92.91% $\pm$ 2.27%	89.67% $\pm$ 2.44%
LSTM	96.53% $\pm$ 0.75%	93.13% $\pm$ 2.11%	90.17% $\pm$ 2.27%
NPP	95.85% $\pm$ 1.23%	93.57% $\pm$ 2.39%	88.51% $\pm$ 3.43%
NPT	95.71% $\pm$ 0.71%	92.76% $\pm$ 1.81%	87.97% $\pm$ 2.59%
RNN	92.46% $\pm$ 16.29%	90.76% $\pm$ 11.81%	82.61% $\pm$ 29.80%
Time-Perceiver	96.18% $\pm$ 0.63%	93.15% $\pm$ 1.34%	89.22% $\pm$ 2.04%
Transformer	92.41% $\pm$ 9.16%	88.74% $\pm$ 8.65%	80.36% $\pm$ 16.72%

Table 5.3: AF 12-Hour Cross-Validated Overall Results (mean scores ± 1.96 *Standard Error)

Model name	Validation accuracy	Validation AUROC	Validation F1 score
GRU	96.51% $\pm$ 0.81%	90.13% $\pm$ 3.08%	81.07% $\pm$ 4.09%
LSTM	96.81% $\pm$ 0.85%	89.19% $\pm$ 3.62%	82.17% $\pm$ 4.62%
NPP	95.97% $\pm$ 6.11%	82.81% $\pm$ 36.00%	67.78% $\pm$ 74.38%
NPT	97.38% $\pm$ 0.75%	90.66% $\pm$ 2.62%	84.88% $\pm$ 5.99%
RNN	96.94% $\pm$ 1.08%	90.95% $\pm$ 1.52%	83.50% $\pm$ 3.55%
Time-Perceiver	96.90% $\pm$ 1.44%	88.39% $\pm$ 9.95%	81.70% $\pm$ 12.32%
Transformer	97.19% $\pm$ 0.60%	89.57% $\pm$ 4.90%	83.15% $\pm$ 4.81%

5.6.4 Cross-Validation Results

Using the best-reported model configurations discovered in the sweep results, 5-fold cross-validation was performed on each section of data. The validation accuracy is reported, validation F1 score, and validation area under the ROC curve. These results are reported in tables 5.2, 5.3, 5.4, 5.5, and 5.6. Each metric reported is the mean of the five runs. Also reported is the standard error for the five runs multiplied by 1.96, which is used to roughly approximate a confidence interval: assuming AUROC is normally distributed, the true value of the mean AUROC performance should fall within 1.96 standard deviations with 95% probability. The highest scores in each table per column are highlighted in bold.

Table 5.4: Shock 4-Hour Cross-Validated Overall Results (mean scores $\pm$ 1.96*Standard Error)

Model name	Validation accuracy	Validation AUROC	Validation F1 score
GRU	91.00% $\pm$ 18.45%	88.49% $\pm$ 16.51%	76.10% $\pm$ 35.39%
LSTM	94.06% $\pm$ 4.15%	91.76% $\pm$ 2.05%	81.65% $\pm$ 10.68%
NPP	96.63% $\pm$ 0.28%	93.00% $\pm$ 0.77%	88.26% $\pm$ 1.48%
NPT	96.11% $\pm$ 0.74%	92.51% $\pm$ 1.33%	85.94% $\pm$ 2.03%
RNN	97.06% $\pm$ 0.54%	93.67% $\pm$ 0.56%	89.92% $\pm$ 0.98%
Time-Perceiver	95.82% $\pm$ 2.14%	91.06% $\pm$ 8.41%	84.68% $\pm$ 9.77%
Transformer	94.32% $\pm$ 7.73%	84.85% $\pm$ 25.20%	75.72% $\pm$ 44.08%

Table 5.5: Shock 12-Hour Cross-Validated Overall Results (mean scores $\pm$ 1.96*Standard Error)

Model name	Validation accuracy	Validation AUROC	Validation F1 score
GRU	97.33% $\pm$ 1.59%	91.74% $\pm$ 1.14%	82.91% $\pm$ 6.85%
LSTM	96.01% $\pm$ 1.67%	92.20% $\pm$ 2.20%	76.25% $\pm$ 8.89%
NPP	94.67% $\pm$ 2.71%	88.85% $\pm$ 4.15%	69.27% $\pm$ 14.02%
NPT	97.41% $\pm$ 1.86%	91.01% $\pm$ 5.09%	81.53% $\pm$ 13.81%
RNN	95.00% $\pm$ 11.38%	90.30% $\pm$ 2.98%	76.52% $\pm$ 33.26%
Time-Perceiver	95.35% $\pm$ 6.00%	90.77% $\pm$ 1.82%	74.20% $\pm$ 21.89%
Transformer	97.24% $\pm$ 0.64%	90.59% $\pm$ 3.98%	80.46% $\pm$ 7.92%

Table 5.6: Mortality 48-Hour Cross-Validated Overall Results (mean scores $\pm$ 1.96*Standard Error)

Model name	Validation accuracy	Validation AUROC	Validation F1 score
GRU	93.92% $\pm$ 9.16%	83.11% $\pm$ 36.45%	66.38% $\pm$ 74.38%
LSTM	95.69% $\pm$ 0.44%	92.16% $\pm$ 1.69%	82.53% $\pm$ 5.86%
NPP	98.06% $\pm$ 0.73%	93.56% $\pm$ 2.69%	90.90% $\pm$ 3.14%
NPT	97.50% $\pm$ 0.73%	92.32% $\pm$ 2.15%	88.94% $\pm$ 2.95%
RNN	97.69% $\pm$ 0.57%	93.86% $\pm$ 0.32%	89.88% $\pm$ 1.97%
Time-Perceiver	97.48% $\pm$ 0.49%	93.08% $\pm$ 2.74%	88.56% $\pm$ 2.89%
Transformer	97.11% $\pm$ 0.90%	94.00% $\pm$ 2.10%	88.15% $\pm$ 3.57%

5.7 Discussion

5.7.1 Cross-Validation Results

Across the five tasks evaluated, only subtle differences were found in the performance of the seven models. None of the best-performing models was found to perform better by a margin greater than their confidence intervals relative to other models.

One possible interpretation of these results is that the tasks established in [227] are relatively easy tasks that are insufficient to demonstrate the capacity of models. However, it is not to be believed that this is the case due to the fairly-wide standard deviation in performance across cross-validation runs: this implies that there is a non-trivial relationship that is data and parameter-sensitive that must be learned to map inputs to correct outputs. The remaining analysis in this discussion section will thus assume that these tasks are sufficient to demonstrate the capabilities of the models.

Across the tasks, RNN models performed well, implying that representational capacity was not a barrier to performance in these tasks. This is likely because the number of time-steps is ultimately small for these tasks (4, 12 and 48 time-steps). Indeed, RNN models performed equivalently to GRU and LSTM models on tasks, indicating a lack of explicit benefit for memory gates.

While Transformer-style models performed equivalently to recurrent models in this setting, the quadratic attention operation gives these models a greater computational overhead and inference time. These considerations are important in medical settings where large machine-learning models may not be locally deployable. As such, recurrent models may be chosen over Transformer-style models. However, a potential advantage that is lost when choosing a recurrent architecture is interpretability: attention weights give an explicit score to features

which can be used to interpret how a model is making a prediction at any one time. This may be of utility to clinicians who wish to interpret which medical features are leading to a prediction result.

The competitive performance of time-domain Perceivers demonstrates they are a ‘best-of-both-worlds’ architecture: these models offer greater interpretability than RNN models due to the use of attention but are still linear in the input space and thus have similar computational overhead. This gives unique utility to Perceiver-style models which do not compromise on overall performance.

The performance of the novel NPPs is roughly equivalent to the performance of parametric Perceivers. This is all while offering a significant speedup in computing time and reduction in computational complexity. This work is considered to be a successful demonstration of the use of NPPs as an alternative to standard parametric Perceivers.

5.7.2 Failures in Training the Hierarchical Perceiver

Hierarchical Perceiver models were not pursued beyond the first sweep due to the poor performance of the models. Examination of the loss training curves indicated that models were failing to effectively learn. These models may benefit from a larger feature embedding size but training such a model is not possible with the computational resources (a single 32GB V100 NVIDIA GPU) available for this DPhil. It is more likely however that a hierarchical network and concatenation operations ultimately result in a complex loss surface that cannot be effectively learned with a standard optimiser. Training losses for the Hierarchical Perceiver over training time are illustrated in figure 5.8, and can be compared to the training of the NPP in figure 5.9.



Figure 5.8: Loss Curves for the Hierarchical Perceiver on the Shock 12h Task: the models fail to decrease their loss beyond initial values

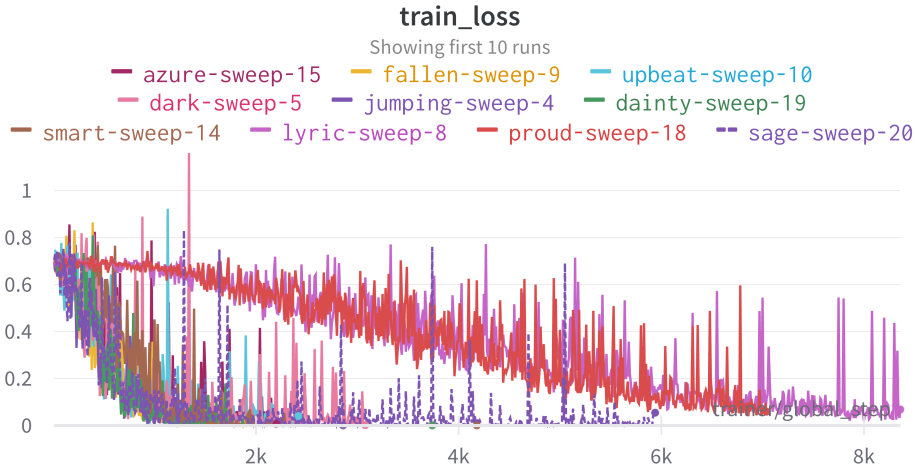


Figure 5.9: Loss Curves for the NPP on the Shock 12h Task: these models learn successfully, with loss values rapidly approaching zero

5.8 Conclusion

This chapter of the thesis presents a evaluation of deep learning models for binary classification on EHR datasets. It provides insight into the performance of both standard RNN-based and Transformer-based model architectures across, and novel architectures like NPPs and Hierarchical Perceivers. This work addresses

a gap in the literature, there has been little systematic evaluation of attention-based models for EMR data tasks, and this work provides this evaluation and comparison. In particular, this work also considers model selection based on computational efficiency, interpretability, and performance. The finding that RNN models can perform comparably to Transformer models while having potentially lower computational overhead, raises considerations for real-world applications where resources may be limited. Thus, standard Transformers may not always have justified costs in the context of EHR data analysis.

The performance of novel Perceiver models, particularly the NPP, is also worth highlighting in this conclusion: these models bridge the gap between the interpretability and computational efficiency of RNNs and standard Transformers, while also retaining performance. This could have significant value in clinical applications where we need to understand the basis of a model's predictions. The linear computational complexity of Perceivers, combined with their competitive performance, makes them an attractive option for future medical machine-learning projects.

However, this chapter also highlights the difficulties in developing novel architectures: the field of deep learning is highly empirical, and difficulties can regularly be encountered in training, as we observed with the complex loss surface of the Hierarchical Perceiver model. While this work did not directly lead to improvement, it does further improve our understanding of training requirements at this scale. Further investigation could go into alternative simpler hierarchical model variants to assess if they can be trained on datasets of the size encountered for these medical machine learning tasks.

6

Working with High-Class Imbalance in a Low-Compute Setting

6.1 Introduction

While it is a great environment for testing novel architectures and training large deep learning models, MIMIC-III is an idealised dataset that was specially curated and quality-controlled for downstream analytical tasks. In most settings, data are unstructured, imperfect and immediate time pressures can mean that there is insufficient time to spend months curating data. Never has this been more true in recent history than COVID-19, where efforts spun up almost overnight to begin modelling the disease and its spread, and the utility of potential treatments. Working on COVID-19 research as part of this thesis was seen as a moral imperative. However, it also had particular utility as a case study of working in a realistic, imperfect data regime for EHR data, based machine learning. Efforts were thus made to identify an appropriate EHR-based project that could assist global efforts in the response to COVID-19. From these efforts, a project was identified that focused on predicting adverse vaccination responses.

At the time of writing, the ChAdOx1 nCoV-19 vaccine is a highly effective vaccine against COVID-19. However, the vaccine has a well-established low risk of adverse side effects related to blood clots. This project describes work towards building a machine learning model that can predictively identify individuals who are at risk of adverse events. This research was conducted as part of the initial efforts to respond to the COVID-19 pandemic in early 2021, in rapid response to the only recently identified low risk of adverse effects at the time. This problem was highly relevant to the focus of the DPhil as a non-ideal dataset: data access was possible only in a compute-restricted remote environment, there were extreme levels of class imbalance in data, and there were difficulties in defining outcomes of interest. This chapter explores different research avenues when working in this restricted setting. While the primary objective of this research was to build effective classifiers, a secondary objective of this work was to identify factors that are associated with the risk of adverse events, which could help inform clinical discovery of the mechanisms by which these adverse events arise.

6.2 Research Justification

ChAdOx1 nCoV-19 (also known by the brand names Covishield and Vaxzevria) is a COVID-19 vaccine produced by Vaccitech and AstraZeneca [229]. It is documented to be a safe and effective vaccine with vaccine efficacy from initial results being documented at approximately 62.1% [230]. Unfortunately, however, in rare cases, side effects of the vaccine have been noted. Epidemiological investigations suggest vaccine exposure results in an increased risk of Cardiovascular and Haemostatic events corresponding to a standardised morbidity ratio of 1.97 (1.50 to 2.54) and 11 (5.6 to 17.0) excess events per 100,000 vaccinations. The standardised

morbidity ratio for any thrombocytopenia or coagulation disorders specifically has been reported as 1.52 (0.97 to 2.25) [67].

Perhaps the most notable side effect reported of these vaccines is thrombosis characterised by thrombocytopenia [231]. While mechanisms for this Thrombotic Thrombocytopenia Syndrome (TTS) have been suggested [232, 233], it is not yet fully understood. Critically, comparisons have been made between the clinical similarity of the presentation of TTS and Heparin-induced thrombocytopenia, which is proposed to be driven by Platelet Factor 4 autoimmunity [234]. As such, current guidance is to manage the condition as equivalent to heparin-induced thrombocytopenia [235].

Given these risk profiles, an alternative vaccine could be provided. An example here would be to provide the Pfizer BNT162b2 vaccine, which is not associated with thrombocytopenic, thromboembolic or haemorrhagic events [236]. However, to replace the entire supply of ChAdOx1 nCoV-19 with BNT162b2 would have many disadvantages. Firstly, ChAdOx1 nCoV-19 makes up one of the largest portions of the supply of COVID-19 vaccines, and any attempt to replace this supply would hugely slow down any efforts to vaccinate populations [237] at global scales. Secondly, the ChAdOx1 nCoV-19 vaccine is a much more practical vaccine for worldwide distribution due to its storage requirements: the vaccine can be stored for 6 months in a refrigerator between 2°C to 8°C, while other COVID-19 vaccines typically require storage at temperatures lower than -20°C [238]. Thirdly, and most importantly, for the vast majority of people, ChAdOx1 nCoV-19 works perfectly as a vaccine without side effects.

Given there is an ample supply of multiple effective vaccines for COVID-19, individuals who are at risk of these adverse events could be given a different

vaccine with a lower risk profile. However, without the ability to accurately identify individuals at risk of developing these clots, such an assignment is impossible.

The principal aim of this project is to develop a machine learning model that can effectively identify individuals who are at highest risk of developing an adverse event, as well as individuals who are at lowest risk of developing adverse events. Such a system, if effective, could be used to inform risk-minimising assignment of vaccine supply. Tied to this, the secondary aim of this project is to use the results of a machine learning model to identify potential risk factors that may predispose individuals to vaccine adverse events. Such results can be used in tandem with epidemiological and experimental findings to better inform our understanding of potential mechanisms of adverse events. This could allow us to better stratify individuals into risk groups, and in the future potentially inform the development of vaccines with a reduced probability of side effects.

6.3 Related Models

To help inform model design, an exploration of the recent space of models being developed to detect COVID-19 related events was first conducted. Multiple machine learning models have been produced that aim to identify the risk of COVID-19 related events, with two examples developed being QCOVID [238] and CURIAL [239]. QCOVID is a set of logistic regression models which are used to predict the risk of hospitalisation or death from COVID-19 infection in male and female populations, while CURIAL is a set of logistic regression and Decision Tree models for triage of COVID-19 using routine clinical data for patients attending hospital. While neither model considers adverse events explicitly, there is a reported increased risk of thrombotic complications in individuals with severe

COVID-19 infection [240, 241]. As the ChAdOx1 nCoV-19 expresses the SARS-CoV-2 spike glycoprotein [242], models that predict severe COVID-19 infection may inadvertently identify adverse effects and risk factors if there is an overlap of mechanisms between the virus and the vaccine. CURIAL and QCOVID are also particularly relevant models because they were developed on UK data, which is also the case for the models outlined in this project. This is useful as distributional shifts between population dynamics, genetics and behaviours may result in different risk factors being important in different countries.

QCOVID is trained on the QResearch database, which consists of 1,205 general practices in England with linkage to COVID-19 test results, Hospital Episode Statistics, and death registry data. 79 variables were considered as potential associated factors, and this list was expert-selected. While QCOVID set a precedent for working with UK general practice data to explore COVID-19 risk dynamics, the research only focuses on simple logistic regression models and a small number of variables. This project aims to explore a much larger space of contributing features at a higher granularity, and a larger search space of possible machine learning models.

The CURIAL model was trained principally on secondary care EHR data for 115,394 patients presenting to the emergency department and admitted across a group of four teaching hospitals in Oxfordshire, UK (Oxford University Hospitals). The models' predictions of COVID-19 infection were subsequently confirmed or denied by in-hospital PCR testing. Variables were selected from presentation blood tests, presentation point-of-care blood gas results, presentation vital-signs and baseline demographic and comorbidity data. Many of the methods explored in CURIAL aim to solve issues of class imbalance, with the study having a ratio of covid-positive to covid-negative cases of 1-to-263. These methods include SMOTE

and SMOTEENN [243] and are of interest due to similar class imbalance issues in attempting to identify adverse events following vaccination. In comparison to CURIAL, this project focuses on predictors that are relevant in a primary care setting (although there is much overlap with secondary care predictors) and considers a larger set of possible relevant features.

Also relevant to this work are the epidemiological studies produced by Simpson et al. [236] and Pottegard et al. [229]. Both studies investigate the epidemiology of thromboembolism, thrombocytopenia, and other vascular adverse events following exposure to the ChAdOx1 nCoV-19 vaccine. These papers are relevant as they provide some insight into risk dynamics and a small set of relevant features.

Simpson et al. report the case-control analysis of thrombocytopenic, thromboembolic and haemorrhagic events in response to first-dose ChAdOx1 and BNT162b2 COVID-19 vaccines in Scotland. A strength of this study is that it reports results with almost complete coverage of Scotland, allowing for systematic evaluation of risk at a population level. The study reports age as a relevant variable that may shift risk profile but does not attempt to analyse any other variables of potential relevance with the exception of sex.

Pottegard et al. report a cohort study analysing arterial events, venous thromboembolism, thrombocytopenia, and bleeding after first-dose vaccination with Oxford-AstraZeneca ChAdOx1-S in Denmark and Norway. This study has coverage of the entire Norwegian and Danish populations, thus allowing for similar representative validity to Simpson et al.. This study also reports a breakdown of events based on age or sex as possible variables of importance.

6.4 Dataset Description

6.4.1 Dataset Source and Access

For this study, the data set was sourced from the Oxford-Royal College of General Practitioners Clinical Informatics Digital Hub (ORCHID) [244]. ORCHID maintains a primary care database of records from 1,858 primary care practices across England and Wales. These records are complete patient primary care records detailing medical conditions, interventions, medications, and test results. ORCHID operates a list of aggregated variables from medical codes known as TADDS (Themes, Access and Dynamic Data Services) variables, and these TADDS variables can be used in classification problems. Data access permissions were granted as part of ORCHID’s existing protocol for COVID-19 research and an ethical assessment for access and utilisation of this data was completed through this protocol. Data was accessed at ORCHID on-site through a secure portal system to ensure data integrity and prevent leakage. While guaranteeing security, this setup meant that computational resources available for machine learning on this dataset were limited. This prevented the use of deep learning methods for this machine learning task.

6.4.2 Target Construction

From the ORCHID database’s population of over 15 million patients, we identified all patients who had received at least one dose of the ChAdOx1 nCoV-19 vaccine. From this data, the goal was to create a binary classification dataset, with the classification target being an adverse thrombotic event following vaccination with ChAdOx1 nCoV-19. Creating this target, however, was non-trivial. This is because, as a newly discovered rare event, there was no agreed event code for the thrombotic events relating to COVID-19 vaccinations. We worked with clinicians

to identify which adverse events would be most linked, and from this we defined adverse thrombotic event cases are patients who were diagnosed with one of the following conditions: Adrenal Haemorrhage, Cerebral Venous Sinus Thrombosis, DaC-VaP Haemorrhage, Deep Venous Thrombosis, Haemorrhage in Optic Nerve, Haemorrhage of Eyelid, Idiopathic Thrombocytopenia, Immune Thrombocytopenia, Intracranial Thrombophlebitis, Intracranial Venous Thrombosis, Intraocular Haemorrhage, OE Throat Haemorrhage, Pulmonary Embolism, Pulmonary Haemorrhage, Renal Artery Haemorrhage, Retrobulbar Haemorrhage, Splanchnic Venous Thrombosis, Thrombocytopenia, Thrombocytopenic Purpura, Thrombosis of Cerebral Veins, Thrombosis of Retinal Artery, Thrombotic Thrombocytopenic Purpura and Trash Foot. Individuals were defined as having an adverse event if they had one or more of the aforementioned events in the 28 days after their first vaccination.

6.4.3 Data Filtering

For this study, interest was focused on individuals presenting with novel adverse events. As such, individuals who has a similar but non-vaccine-related adverse event in the ten years prior to vaccination were removed from the cohort. This case definition was driven by the ORCHID team’s internal work to define adverse events. The final dataset consisted of a population of 1,843,954 people, of whom 797 experienced an adverse event. In subsequent models, this data set was further split into age groups of under 40, 40 to 65 and over 65 to build classification models for individual populations.

6.4.4 Variable Selection

Variables for consideration in the classification problem were selected in the following ways:

- Basic static variable demographic information was collected on individuals which included age, biological sex, and ethnicity.
- Variables used by QCOVID [245] were re-implemented in the model TADDS.
- Variables that were considered to be relevant following clinical guidance were included. From a set of 1,287 variables, 180 were selected. Further variable selection was guided by expert recommendation from the ORCHID clinical team.

In total 215 variables were used as predictors in the binary classification problem. Of these, there were 33 variables reflected from QCOVID and 180 TADDS variables. Two auxiliary variables were also used to augment the data: zero and non-zero feature count.

6.5 Feature Transformation and Imputation

Two classes of variables were available: numerical and non-numerical. Numerical variables included data points such as primary care blood pressure readings and blood test results, while non-numerical data indicate the presence of events such as a specific cancer diagnosis. For both data types, a time-series of events was available. Non-numerical time-series variables were converted into binary static variables, with 1 indicating the presence of an event in the last year, and 0 indicating an absence of the event. Numerical time-series variables were converted into continuous-valued static variables by taking the mean of readings over the last year.

Missing data were imputed using the k -nearest-neighbours algorithm, with the mean of the 3 nearest neighbours measured in Euclidean distance being used as the imputation result.

6.6 Methods

6.6.1 Visualisation

After applied transformations, data separability was visualised using t-SNE and PCA plots, with the goal of determining the separability of classes. Due to the computational complexity of t-SNE scaling quadratically with the number of data points, it was only feasible to use this algorithm on random subsets of data.

6.6.2 Baseline Creation

To identify an initial baseline for model performance, TPOT [246] was used to conduct an initial search of the model space. TPOT, or Tree-based Pipeline Optimisation Tool for automating machine learning, is a genetic programming-based autoML system that constructs a tree of pre-processing steps and multiple models to maximise performance on a classification task. TPOT was configured with a fitness objective of maximising AUROC on binary classification. The TPOT optimisation tool has access to a wide space of models, but deep learning models were not included in its model space due to computational limitations. A TPOT pipeline was trained on the complete dataset for five generations, with a population size of ten.

6.6.3 Model Selection and Implementation

The majority of variables used in this model were binary variables. As such, the model exploration focused on methods suitable for this data. Accordingly, tree methods were explored, including gradient-boosted trees [247] and XGBoost [75]. These methods were chosen principally due to their demonstrated empirical efficacy in relatively low-compute environments. These models also offer a high degree of interpretability, which is useful given the secondary objective of this research

is identifying factors that may predict complications arising from vaccination. Multinomial Naive Bayes [248] was also explored, as the use of this model required the selection of only binary variables as features. Feature importance analysis was conducted using SHAP [76]. Systematic sweeps of parameters were not highly feasible due to the limited computational resources available in the trusted research environment setting.

Given the significant class imbalance in this dataset, it was critical to evaluate the performance of this model on the minority class of adverse events. Thus the TPR, FPR, ROC and AUROC [249] were reported for each model. The final model performance is evaluated on AUROC. Data were split in an 80-20 train-test split, with adverse events being evenly stratified through the train-test split.

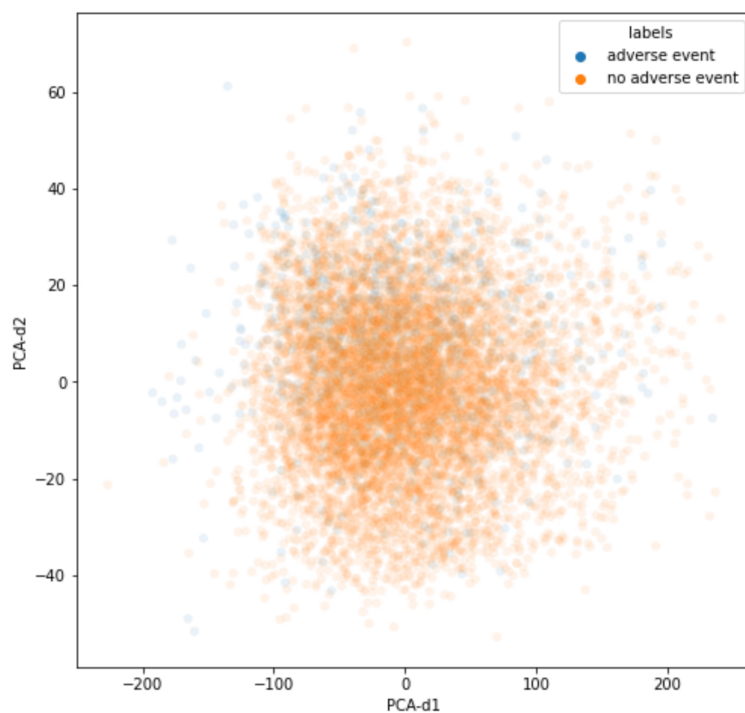
6.7 Results

6.7.1 Main Results: AUROC

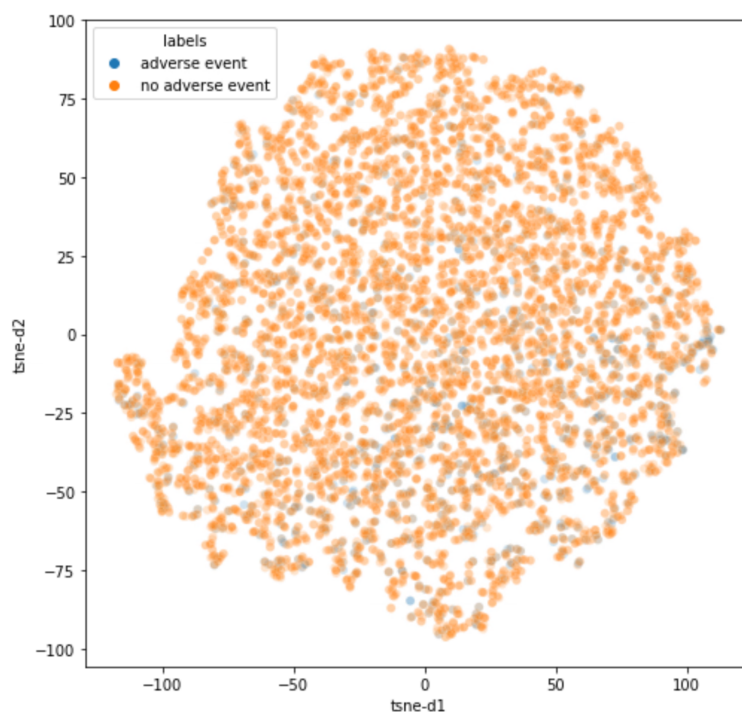
The results of PCA and t-SNE analysis are shown in figure 6.1. In these plots, all available 797 examples of adverse event cases (shown in blue) are compared to a random subset of 7,970 normal population cases (shown in orange) in the complete data set. Data points do not immediately appear separable in the two-dimensional space representations, indicating that individuals with adverse events may be difficult to separate from individuals without adverse events.

The TPOT pipeline found a Gradient Boosting Decision Tree Classifier to be the most effective model (with an AUROC of 62%), which is consistent with the prediction of such models being the most suitable.

The best results for XGBoost models, Naive Bayes models, and basic Gradient Boosting Tree models are reported in table 6.1. AUROC and FPRs are reported at different TPRs, which is informative for real-world use where a calibration



(a)



(b)

Figure 6.1: (a) an example PCA analysis of a random subset of the complete data. (b) an example t-SNE analysis of a random subset of the complete data. Adverse events (blue) are not immediately separable from non-adverse event data (orange) indicating that the classification task is potentially challenging

threshold would be used to decide who to give different types of vaccine to within the population pool.

No model was able to achieve an AUROC that exceeded 70.1%. This is a relatively low baseline, but it is consistent with the expected difficulty of classification in such extremes of class imbalance. The best-performing model was a Naive Bayes model. This is consistent with the fact that the majority of features in the dataset are binary and/or count variables, which are well-suited to be modelled with Naive Bayes methods.

Given the relatively low baseline, we also opted to perform a follow-up study using oversampling via SMOTE [250]. SMOTE was used to balance classes to an equal balance of positive and negative target labels.

AUROC results are reported when using over-sampling via SMOTE to a 50:50 balance of positive and negative target data points in the last line of table 6.1. This results in a performance drop, which is somewhat in-line with expectations. Given the incredibly high degree of class imbalance in the dataset, the majority of generated data for the minority class will be synthetic and it may be a poor representation of the underlying minority class data distribution.

6.7.2 Main Results: Interpretability

Naive Bayes Classifier is explored in greater depth in figures 6.2 and 6.3. Figure 6.2 shows the ROC curve of the Naive Bayes Classifier, while figure 6.3 shows a SHAP Beeswarm plot of the Naive Bayes Classifier. A Beeswarm plot shows how the top features in a dataset impact a model's output. Individual data instances are represented with dots on the plot, with the instance feature values indicated by the colour code. The SHAP value impact of each feature on model output for each data point is shown on the y-axis.

Table 6.1: Performance of individual models on the complete dataset. The best-performing results are highlighted in bold

Metric	Boosted Tree	XGBoost	Naive Bayes
AUROC	69.4%	67.5%	70.1%
FPR at a 50% TPR	24.0%	22.0%	23.2%
FPR at a 75% TPR	44.5%	47.3%	44.5%
FPR at a 90% TPR	73.0%	77.8%	66.9%
AUROC when using SMOTE resampling	62.3%	64.9%	67.2%

Amongst the 20 features that have the biggest impact on classification boundaries in figure 6.3, the top-indicated features are those with significant impacts on blood composition (e.g. disorders of cellular components of blood, Liver Cirrhosis, and toxic Liver Disease). These conditions have the potential to change the underlying immune system and coagulation dynamics which may in turn affect the probabilities of thrombotic events [251–253].

6.7.3 Secondary Analysis

While the first component of this chapter achieved the initial research goals in part, it was deemed necessary to perform significant further analysis to explore ways of improving both AUROC results and interpretability. To this end, further studies were conducted on age-specific splits of data and data subsets, using the same models and feature pipeline described in the methods section.

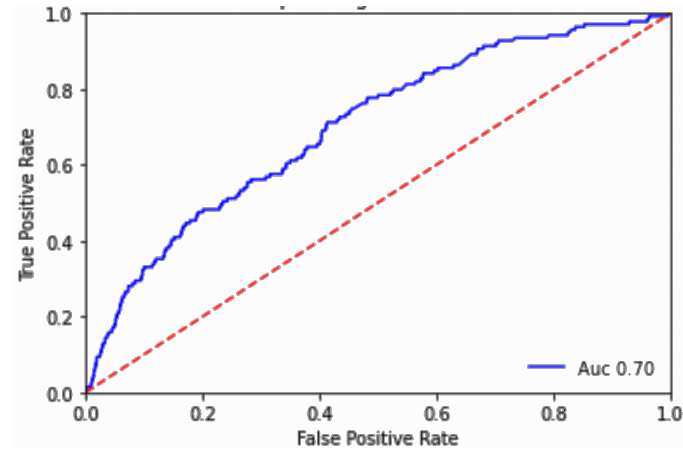


Figure 6.2: Receiver Operating Characteristic Curve for the best overall model: the Naive Bayes Classifier trained on binary categorical data. The blue dotted line indicates true and false positive rates for classification at each possible level of a classification boundary, and the red dotted line indicates the performance of a random classifier

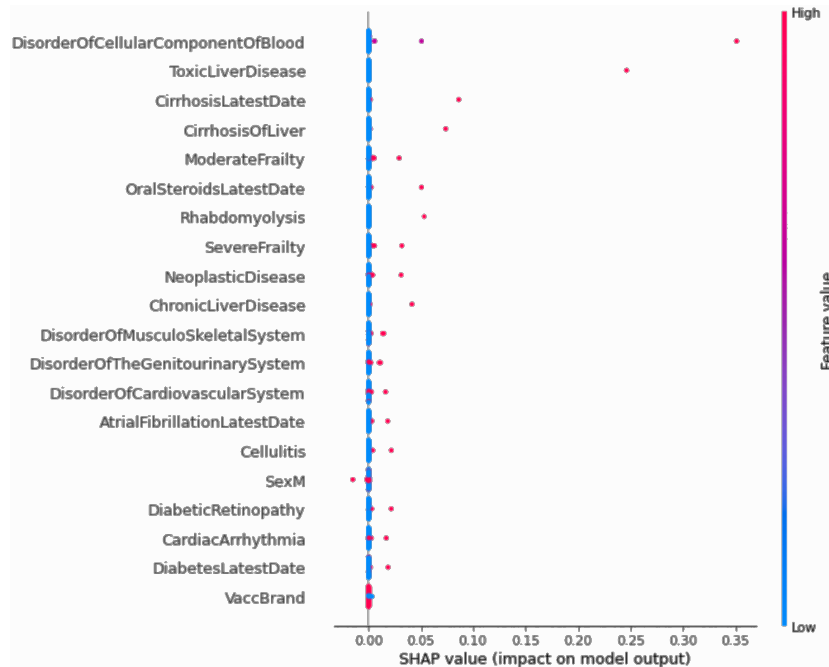


Figure 6.3: SHAP Beeswarm plot of adverse event data points for the best overall model: the Naive Bayes Classifier trained on binary categorical data. This plot shows the 20 features with the most impact on model classification outputs

Training Models on Age-Specific Splits of Data

Other features identified in figure 6.3 include directly age-related variables (e.g. Moderate and Severe Frailty) and other conditions that are directly related to age. This highlights an unfortunate potential limitation with the first models: the adverse clotting events of interest described in this project are likely to happen at an older age due to non-vaccine-related factors. Given the very low baseline probability of thrombotic events occurring in the 30-day window after a vaccination, it is possible that a significant subset (or even the majority) of the events labelled as true occurrences in our data could be false positives.

To attempt to rectify the issue of age as a factor in classification decisions, performance of models was explored on specific age subsets of the data. The aim of this was to partially help disentangle the effects of age on the overall risk of adverse events not related to vaccination. As such, the data set was divided into people aged under 40, people aged between 40 and 65, and people over 65, to analyse the risks in these groups. These results are reported in table 6.2, with the best results for each metric being reported in bold. Classification performance was harder to maintain when working on subsets of age data, with no AUROC score beating the score on the complete dataset (see table 6.1).

SHAP Beeswarm plots for the best-performing models based on AUROC (all Naive Bayes Classifiers) are reported in figures 6.4, 6.5 and 6.6. The SHAP Beeswarm plot for individuals under 40 (figure 6.4) indicates that respiratory conditions and liver status conditions were principally used to inform the classification of predicted adverse events. It is not immediately apparent what may be driving the associations of respiratory and liver conditions with adverse events. As such, there may be value in exploring the pathophysiology of these conditions in more depth. The SHAP Beeswarm plot for individuals aged 40-65 (figure 6.5)

Table 6.2: Performance of individual models on subsets of ages in the dataset

Age band	Model	AUROC	FPR at a 50% TPR	FPR at a 75% TPR	FPR at a 90% TPR
Under 40	Boosted Tree	56.1%	4.7%	81.2%	84.8%
	XGBoost	65.0%	6.0%	66.0%	81.3%
	Naive Bayes	65.5%	33.0%	54.3%	61.7%
40-65	Boosted Tree	56.9%	5.9%	93.2%	93.2%
	XGBoost	64.9%	24.9%	50.4%	71.1%
	Naive Bayes	67.0%	21.2%	56.4%	83.8%
Over 65	Boosted Tree	61.4%	31.7%	59.5%	85.0%
	XGBoost	55.0%	39.8%	68.1%	87.5%
	Naive Bayes	63.3%	30.4%	56.1%	80.4%

also indicates that liver status was used in inference, along with blood disorders and inflammatory status. These latter two variables are consistent with the idea of adverse events being driven by a change in blood composition and/or changing immune dynamics. The SHAP Beeswarm plot for individuals over 65 (figure 6.6) indicates that Neoplastic Disease and Rhabdomyolysis are the most strongly associated variables, which is once again consistent with a change in blood composition being an important factor. Age-associated variables were observed as informing classification decisions in the 40-65 group and the over-65 group.

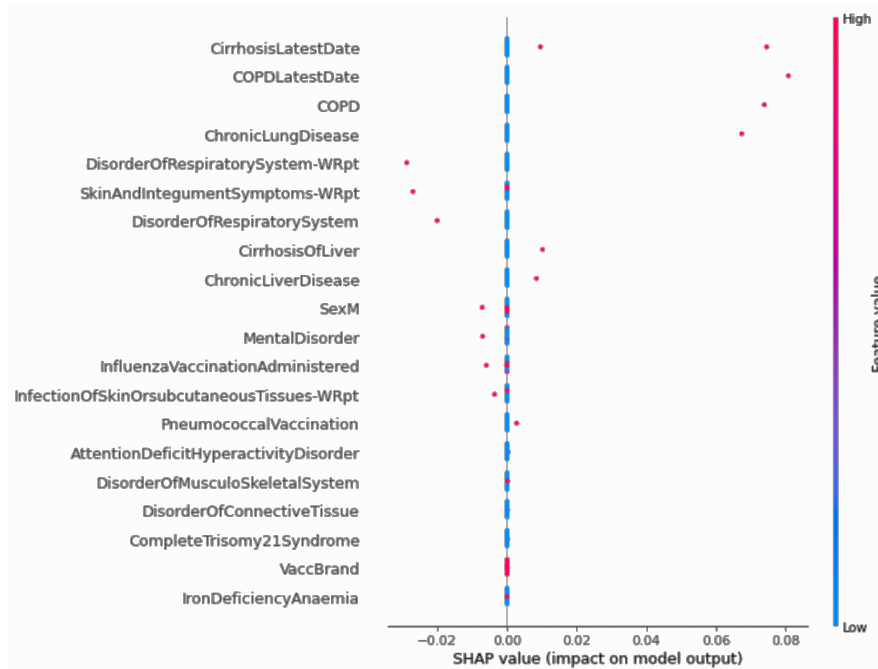


Figure 6.4: SHAP Beeswarm plot of adverse event data points for a Naive Bayes Classifier trained on people aged under 40: this plot shows the 20 features with the most impact on model classification outputs

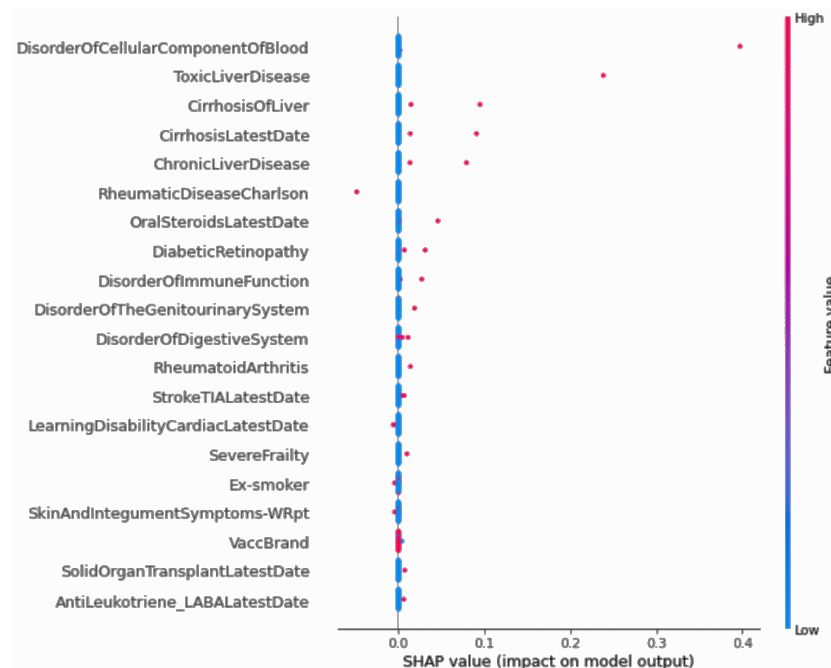


Figure 6.5: SHAP Beeswarm plot of adverse event data points for a Naive Bayes Classifier trained on people aged between 40 to 65: this plot shows the 20 features with the most impact on model classification outputs

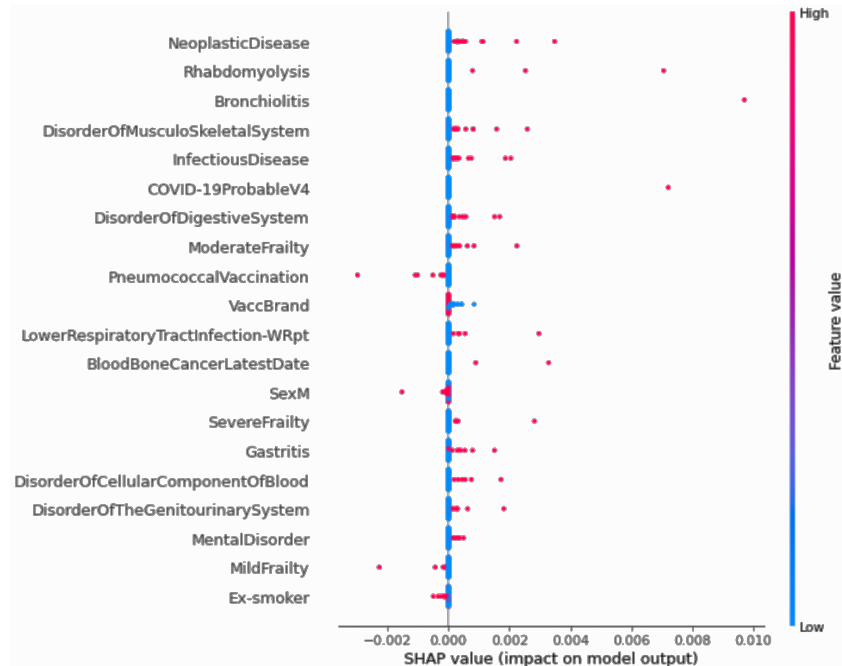


Figure 6.6: SHAP Beeswarm plot of adverse event data points for a Naive Bayes Classifier trained on people aged over 65: this plot shows the 20 features with the most impact on model classification outputs

6.7.4 Training Models on a Data Subset

Given both the visualisation results from manifold techniques (figure 6.1) suggesting that individuals with adverse events may be difficult to separate from individuals without adverse events, and the relatively low maximum AUROC scores achieved, the accuracy of classifiers was evaluated when considering only the easiest individuals to classify. This choice was made for two reasons. Firstly: could we identify if certain sub-groups of high-risk or low-risk individuals could be effectively separated from the data? This could be used to inform vaccine allocation. Secondly, if we performed feature importance analysis on the easiest-to-separate individuals, would this reveal interesting features that could inform subsequent epidemiological analysis?

In order to do this, the results from a Naive Bayes classifier were used and

trained on the whole dataset to obtain model prediction probabilities. A subset was created from the dataset: this subset consisted of the 100 individuals with adverse events that the model predicted had the highest probability of adverse events (i.e. the most certain true positives), and the 1,000 individuals without adverse events that the model predicted had the lowest probability of adverse events (i.e. the most certain true negatives).

From here, PCA and t-SNE analysis of this data was conducted to assess possible separability, as shown in figure 6.7. In these plots, individuals with adverse events are shown in orange and those without adverse events are shown in blue. In both settings, four distinct clusters of patients were observed. Critically, however, in both plots, we observe that the positive class is contained almost entirely within a single cluster. In the case of the PCA plot, it even appears that a linear decision boundary in PCA space would be sufficient to produce an accurate classifier on this subset of data. The visual separability of these clusters along the lines of the categories indicates that the factors relevant for classification in this subset of data should be highly separable. The relatively tight clusters of the positive class in the PCA and t-SNE plots are also of relevance, as they suggest that a small set of specific patient features that can be easily identified from patient records could be used to separate groups for vaccination allocation. These features are discussed in the next section. Such decision boundaries could perhaps be more easily turned into scalable policy decisions. This is particularly true given the features used are binary.

Following this positive finding from the separability plots. A Naive Bayes Classifier model was trained using this subset of the dataset, and this model was then evaluated on 1,000,000 data points not included in this sample of 1,100 used in training. The AUROC of this model was 64.7% (FPR at 50% TPR =

28.8%; at 75% TPR = 51.9%; FPR at 90% TPR = 74.7%). A SHAP plot from this model is shown in figure 6.8. These SHAP Beeswarm plots indicate that the absence of previous vaccines (Diphtheria, Polio, Hepatitis B, Meningitis C, Tetanus, Meningitis ACWY, etc.) are found to be predictive of adverse events in this setting. Given these vaccines all are of different types (toxoid, attenuated, subunits etc.) [254], this finding implies susceptibility to an adverse event may be driven by the underlying dynamics of immune response to a vaccine.

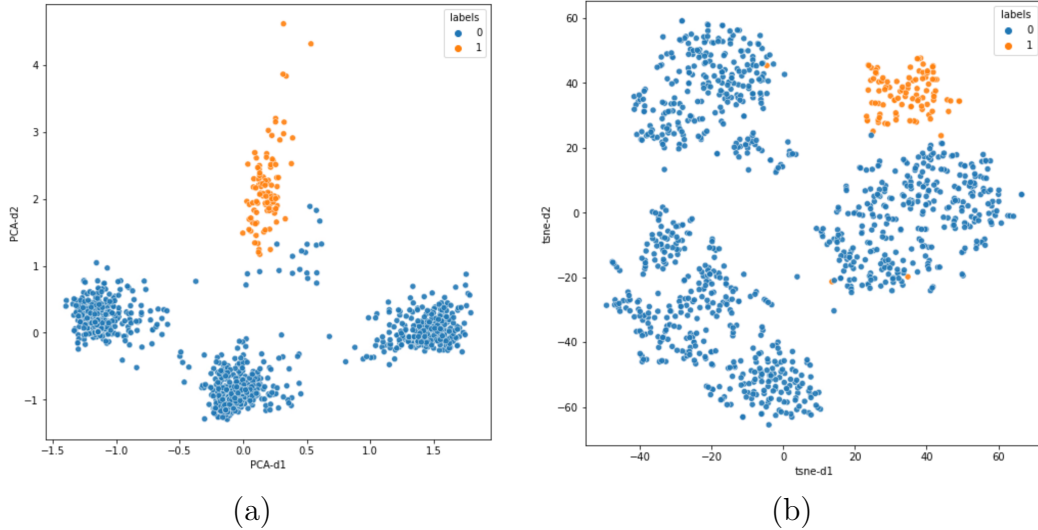


Figure 6.7: (a) PCA of the 100 easiest to successfully classify individuals with adverse events and the 1,000 easiest to successfully classify normal individuals. (b) t-SNE of the 100 easiest to successfully classify individuals with adverse events and the 1,000 easiest to successfully classify normal individuals. Both plots indicate high separability in these subsets of data

6.8 Discussion

The results of this project have been partially successful in achieving both the primary objective of developing a machine learning model that can effectively identify individuals who are at highest risk of developing an adverse event and the

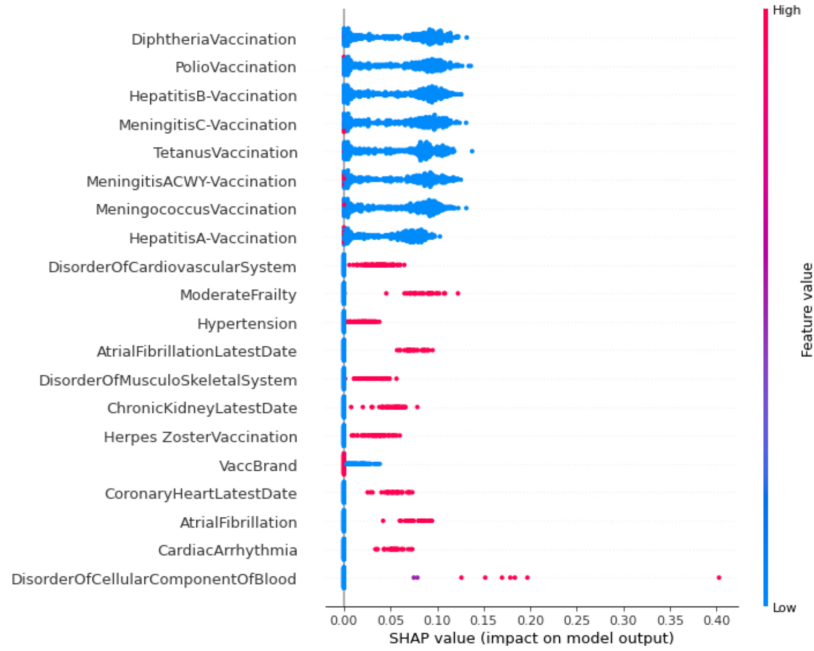


Figure 6.8: SHAP Beeswarm plot of adverse event data points for a Naive Bayes Classifier trained the 100 easiest to successfully classify individuals with adverse events and the 1,000 easiest to successfully classify normal individuals

secondary objective of an exploration of possible risk factors that may predispose individuals to vaccine adverse events.

Regarding the primary objective, the AUROC scores, while low by conventional standards, offer utility in informing the allocation of different types of vaccines. Given that there is not a significant consequence to false positive detection (individuals with false positives can be provided with an alternative vaccine to ChAdOx1 nCoV-19 with a lower risk of adverse thrombotic events), the model can be calibrated to a sufficiently low false negative rate to minimise the risk associated with the ChAdOx1 nCoV-19 vaccine, facilitating a more efficient allocation of healthcare resources.

In terms of the secondary objectives, SHAP Beeswarm plots have identified that across age groups some clinically relevant features are correlated with adverse events. Continuous features of interest in models trained on all data or trained on

certain age groups are Platelet Counts, HBA1C levels, LDL levels and Cholesterol levels. Categorical features of interest include recent Neoplastic Disease, recent Rhabdomyolysis, recent Disorders of Cellular Components of blood, and recent Cirrhosis of the liver. All of these conditions alter the composition of blood, which will in turn affect clotting dynamics. However, all of these variables will be affected by age, which is likely to be contributing to false positive events.

Of further interest, are the results reported in figures 6.7 and 6.8, which describe the data that was easiest to separate between the positive and negative classes. A model trained on these 1,100 data points achieved accuracy relatively comparable to a model trained on 80% of all data, with SHAP analysis suggesting that an absence of previous vaccinations is important to making classification decisions in this reduced data training setting. While these results do not necessary imply that these variables are causal, their use in forming a decision boundary indicates a possible relationship to adverse events. As such, subsequent robust epidemiological analysis should be performed on these variables.

6.9 Limitations

While the aim of this study was to build as performant a model as possible, some key limitations were faced.

Most notably, as documented in this chapter, this project faced some computing limitations due to the requirement to access ORCHID data on its own secure server system. At the time of analysis, graphical processing units were not available on the cluster, which posed some limitations on the ability to train modern deep learning architectures. In particular, this prevented the use of non-compressed representations of patient data and the evaluation of the transformer models explored in the previous chapter, which would have provided useful insight into

the practicalities of these models outside of the idealised setting of MIMIC-III [44]. These limitations also prevented the use of some more computationally intensive tools for hyperparameter optimisation and architecture search.

Complications in this project also arose from the fact that this dataset was provided under special measures in response to the pandemic. Given the accelerated procedure all efforts were made to prevent any patient data leakage, and it was not possible to present any analysis at a per-patient level. The special measures and temporary academic structures that were in place for the pandemic have also made follow-up research on this project difficult to conduct now that these measures and structures are no longer present.

Another significant documented limitation is the interaction of age on the data: given the adverse events described in this study are highly correlated with age, it was difficult to disentangle which adverse events were attributable to vaccination and those which were attributable to age alone (with the latter being potentially more common). Given the interactions of age with a wide range of other recorded health variables, it was also not possible to censor this variable from the data. Consequently, while care was taken to minimise these age-based interactions by producing models for different age groups, there is still a high probability that age-based interaction effects have distorted the reported results.

A third limitation worth mentioning is that the ORCHID dataset used does not have full coverage of the UK population. Approximately 50 million people, around $\frac{3}{4}$ of the UK population, were not included. While this may seem trivial, given how few examples there are of adverse events in this dataset, a lack of complete representativeness may have resulted in some bias in the model predictions and epidemiological results.

7

Working with High-Dimensional Small Sparse Data Sets

7.1 Introduction

While the work in chapter 6 represented a real-world EMR machine learning task with significant clinical need, the dataset and target environment are not indicative of all EMR machine learning tasks. There is value in exploring datasets and environments with radically different properties to assess what machine learning methods can work best in these settings. Critical properties of the data in chapter 6 are its large size and static nature. The computational limits in chapter 6 also affected both method selection and hyperparameter optimisation.

It was thus deemed necessary to produce a third work package using a medical dataset with radically different properties, to provide a greater coverage and understanding of how machine learning methods can be effectively applied to EHR data. In particular, searching for a smaller time-series machine learning dataset was of interest. A dataset and project for this purpose was found in the domain of neuro-intensive care.

This section describes a study to predict MidLine-Shift (MLS) and Life-Threatening Mass-Effect (LTME), from patient EHR, and to use these predictions to inform scanning decisions. Consistent with the goals of this thesis, this dataset was non-ideal to work with due to its real-world properties and is characterised by low patient numbers, high-dimensionality, sparseness, and large amounts of missing data. We explore the utility of both traditional machine learning and the deep learning architectures developed in this thesis for this task.

7.2 Research Justification

The septum pelucidum is a tissue structure that separates the lateral ventricles, two fluid compartments in the centre of the brain [255]. Under normal circumstances, the septum pelucidum runs directly through the middle of the brain and is visible on CT scans. In diseased states, however, its position in the brain is a key marker.

MLS is used as a measurement of displacement of tissue in the brain. It is measured as the number of millimetres the septum pelucidum (and other brain midline structures) have been shifted from the middle of the brain [256]. This shift in tissue is most commonly caused by a mass effect, a growing mass in the brain that displaces other tissue. Examples can include tumours, bleeds in hemorrhagic strokes, and oedema [257–260]. Mass effects have the potential to be particularly deadly in the brain because it is a closed pressurised environment. Increased mass in the brain results in greatly increased IntraCranial Pressure (ICP), which can rapidly lead to further brain tissue death where tissue is squashed [261]. MLS thus acts as a proxy measure of ICP.

Not all mass-effect events a patient may experience during a stroke are life-threatening. Some patients may experience small changes in MLS that recover. Other patients however can experience rapid increases in ICP to potentially lethal

levels over the course of a hospital admission. For patients with ischaemic strokes, this is most commonly driven by oedema, a buildup of fluid in the brain. The precise dynamics of why this oedema occurs and to whom it occurs are not fully understood [259], but understanding who is at risk is essential for informing clinical decision-making and resource allocation.

Each year, more than 7.6 million ischaemic strokes are recorded globally [262]. Large Middle Cerebral Artery (MCA) infarction is one of the most dangerous forms, with a mortality ranging from 2% to 8% among all hospitalised patients with ischaemic stroke and 18% to 31% among those with ischaemic stroke due to unvascularised MCA occlusion [263]. A major driver of worse outcomes is malignant brain oedema, which can result in a further increased risk of 40-80% risk of death [263].

At-risk patients with oedema are often monitored in ICU settings to watch for signs of neurological deterioration. However, clinicians have limited tools to relieve life-threatening ICP. Available treatments include osmotic therapy, which is widely used [264], and surgical intervention in the form of Decompressive HemiCranectomy (DHC), which can reduce mortality in eligible patients from 80% to 20% [265, 266]. But surviving populations are significantly more likely to have moderately severe to severe disability [266]. Given outcomes remain poor following ischaemic strokes, there is a need to build tools that can support clinical decision-making.

Currently, the only way to accurately identify patients at risk of mass effect-driven neurological deterioration is through neuroimaging. This is typically conducted using noncontrast CT scans [263]. Such imaging can show the extent of oedema in the brain, and can be used to measure the extent of MLS of the septum pelucidum [267]. Previous studies indicate that MLS greater than 5mm within the first two days is linked to neurological deterioration and early mortality

[268]. Recent studies have also shown that MLS as low as 3mm is associated with worse long-term outcomes [269].

However, there are key issues with the use of CT scans to detect mass effect: radiation exposure, CT availability and patient transport requirements limit frequent use [270]. Furthermore, there are no firm guidelines on when to scan patients to optimally detect oedema and track potential MLS progression. As such, identifying oedema at a stage when interventions may bring clinical benefits, can be difficult.

Given these issues with CT scans, there is an imperative to predict MLS deterioration at more frequent intervals than what scans allow. Such predictions could be used to optimise treatment decisions and make decisions around how to allocate scanning. Existing predictive models of cerebral oedema risk [271–275] rely on static data, and thus cannot be used to make predictions about deterioration over time. Building a model can make dynamic predictions over time has the potential to significantly improve outcomes, by providing dynamic updates to risk of cerebral oedema as more data becomes available.

The goal of this chapter is to develop machine learning models that can accurately predict future mass effects (via MLS) from clinical variables. These models can be used by clinicians to inform scanning decisions and identify patients at risk. Such predictions can allow for more optimal utilisation of the limited resource of the CT scanner, allowing for effective patient management across a ward. To this end, we explore deep learning approaches (including the approaches created in chapter five), and shallower learning approaches with a longer historical precedent for comparison.

7.3 Dataset Description

7.3.1 Study Population

This project was performed in collaboration with medical teams from the Boston Medical Center (BMC) and the Mass General Brigham (MGB) hospital trust system. We worked collaboratively to identify relevant cohorts from their data that could be used for training machine learning models, and to establish criteria for including patients. Our goal of working with two hospitals was to create both a larger main training dataset, and a smaller external validation dataset for assessing if models could successfully generalise to unseen datasets. Massachusetts General Hospital and Brigham and Women’s Hospital, core members of the MGB healthcare system, were chosen as the primary data cohort. Located in Boston, Massachusetts, these two hospitals host over 1,800 beds and serve over 100,000 inpatient stays annually. The BMC is by comparison a 514-bed academic medical centre. It was opted to use data from this hospital as an external validation population.

MGB data was obtained from the Research Patient Data Registry (RDPR), the centralised clinical data registry for the MGB hospital system. This database contains complete radiology data and demographic data. Clinical record data available in RDPR needs to be completed for patients before 2015 due to a switch between medical record systems. BMC data was obtained from the BMC Clinical Data Warehouse (CDW). This database contains complete radiology, demographic, and clinical information across all hospitalisations taking place within the BMC. Filtering criterion were as follows:

- For MGB data, the RDPR was searched for Unique Patient Encounters from 2006 to 2021 with diagnosis codes of stroke ($N=22,771$). Patients with no acute MCA stroke were excluded, as defined by a natural language

processing patient identification model and manual review ($N=21,511$) [276, 277]. A further 485 patients were excluded for either missing notes ($N=3$) or confounding injuries, including extraparenchymal, subarachnoid, epidural, or subdural haemorrhage ($N=86$); acute contralateral or posterior fossa injury ($N=299$); significant encephalomalacia covering greater than half of a vascular territory ($N=86$); primary intraparenchymal haemorrhage ($N=5$), cerebral venous sinus thrombosis ($N=3$); and Moyamoya disease ($N=3$). The remaining cohort consisted of 892 patients. Patients with admission times of less than 24 hours ($N=211$) were removed, resulting in 675 patients.

- For BMC data, the CDW was searched for patients from 2019 to 2023 diagnosed with acute ischaemic MCA territory stroke ($N=704$). Patients with a stroke less than one-half of the MCA territory were excluded ($N=520$). There was an additional cohort of patients excluded for either uncertain Last Seen Well (LSW) ($N=25$) or coexisting medical conditions, including significant focal encephalomalacia greater than one-half of the MCA, Anterior Cerebral Artery (ACA), or PCA territory ($N=5$); acute contralateral or posterior fossa stroke, haemorrhage or tumour ($N=7$); Withdrawal of Life-Sustaining Therapy (WLST) within twenty-four hours of admission without an interval head CT ($N=5$); history of pre-existing pupil dysfunction ($N=5$); and miscellaneous reasons (Abnormal Npi, non-reactive pupil, blindness, surgical eyes, terminal cancer) ($N=18$). Additional exclusion criteria included instances of a first pupil measurement was taken more than twenty-four hours after LSW ($N=50$); cerebrospinal fluid diversion, DHC, or MLS of five millimeters of greater prior to the first pupil measurement ($N=7$); missing smartguard data ($N=56$); and fewer than three pupil observations ($N=4$).

Some patients met several exclusion criteria resulting in the remaining cohort of 60 patients.

Table 7.1 summarises the basic characteristics of these two patient cohorts. We focused on predictions in the first seven days following the presentation to hospital, and truncated data beyond this point.

Variables	MGB	BMC
Total number	675	60
Mean Age	64.30 ± 5.60	67.70 ± 16.00
% Male	59.3%	40.0%
% Transferred to Hospice/died	36.2%	44.1%
DHC	11.9%	22.0%
Mean first MLS if occurred (mm)	3.77 ± 2.85	2.17 ± 0.82
Mean max MLS if occurred (mm)	6.40 ± 4.30	7.96 ± 4.96
% of patients with an MLS >5mm	34.9%	53.3%
% of patients with a PGS >4mm	24.8%	36.7%
Average Length of Stay	10.54 ± 9.99	40.15 ± 106.62
Average CT scans per patient	4.66 ± 2.70	5.43 ± 1.97

Table 7.1: Basic patient characteristics. Continuous values are reported as the mean $\pm$ standard deviation

Given the known potential issue with partial incompleteness in MGB data, it was opted to visualise the data distribution. It was found that the distribution of events per patient is long-tailed, with many patients having sparse data. This issue is more pronounced in older data in the dataset, where a change of database systems has resulted in incomplete logs. The distribution of data in regards to length of data, number of events, and year of origin of data is shown in figure 7.1.

7.3.2 Data Curation and Outcomes of Interest

Feature Selection

Demographic and clinical variables were used at varying time horizons to construct our models. These variables include time-invariant demographic variables, clinical

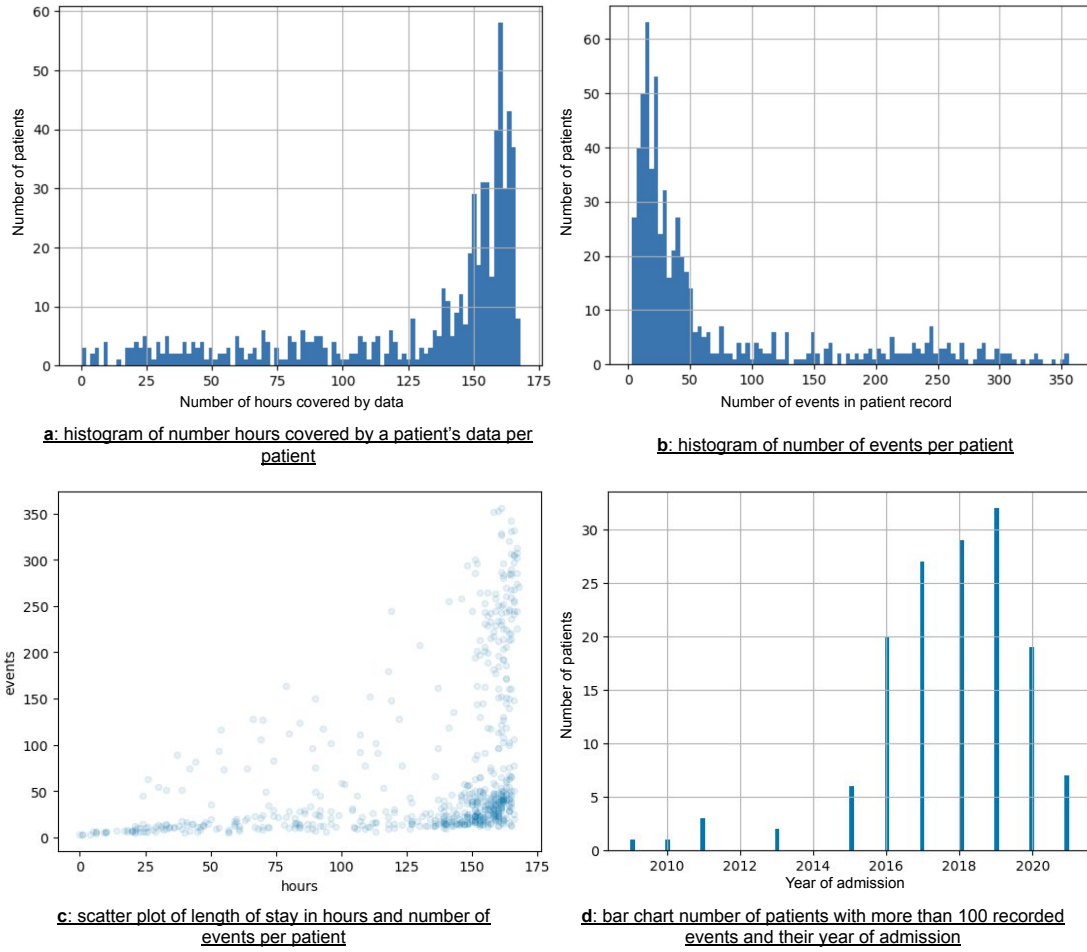


Figure 7.1: Summary of Data Distributions in MGH Patient Data: **a:** number of hours covered by a patient's data per patient. Most patients have a complete record of 7+ days, with data being truncated at the 7-day mark. **b:** number of events per patient: while most patient records contain fewer than 50 events, there is a long tail of patients with more frequent events in their patient records. **c:** length of stay in hours and number of events per patient: the majority of patient data is 7-day duration and populated with sparse events. **d:** number of patients with more than 100 recorded events and their year of admission: it was observed that the vast majority of data before 2016 is sparse and contains few events, which is likely due to the change in logging systems around this time

variables recorded at the time of admission, and time-censored dynamic clinical variables changing throughout admission. Demographic variables include age and sex. Static admission-related variables include the time of admission, time LSW, past medical history variables including prior anticoagulant and antiplatelet use,

vitals and laboratory blood value readings taken on admission, and admission NIHSS. Time-censored variables were variables censored in data until outcome occurrence (MLS category and/or DHC) and included times of surgeries and MLS events above certain thresholds. Dynamic clinical variables included radiographic variables and non-imaging variables (laboratory blood values, vitals readings, ASPECTS scores, and osmolar therapy use). Features were selected based on which features were available both in MGH and BMC data. All overlapping features were included. 65 variables in total were used, consisting of 21 static demographic and admission-related variables, 18 time-censored variables, and 26 dynamic variables. A full data dictionary for these features can be found in appendix B.3. These features were selected due to their possible association with malignant oedema [263, 266, 278, 279]. Non-radiographic variables were extracted directly from the electronic patient record. Radiographic variables were constructed via expert labelling of radiographic images associated with patient scans at the time of dataset construction.

7.3.3 Target Creation

Future MLS was chosen as the outcome of interest for this study due to its key role in determining the severity of ischaemic stroke [271, 274, 280]. The choice of classification categories of 0mm, 0-3mm, 3-8mm and 8+mm was informed by collaborating staff as key decision boundaries in clinical practice, roughly mapping to non, mild, moderate, and severe. While 5mm of MLS has been previously used in early models to define severe oedema, it was found in our dataset that MLS most commonly approached 8mm prior to DHC, which led to our decision to use a 3-8mm boundary. These outcomes were constructed using expert-labeled radiographic data. Hourly patient observations were excluded from the analysis for which a

scan event did not occur in the time window of interest (e.g., if there was a 40-hour gap between scans in the 24-hours ahead prediction task, the first 16 hours of data between scans would be removed as a real target cannot be constructed in this time). Two primary tasks of predicting maximum MLS over a future 8-hour and 24-hour period on a per-hour basis were considered. These time frames were chosen as they represent the relative time windows of most clinically relevant decisions, balancing the addition of new laboratory, imaging, or vital-sign information available.

7.3.4 Data Processing

To increase the size of trainable observations above the number of patients, multiple observations were obtained from single patients. The patient’s data was processed to create hourly-based observations that reflect a patient’s condition based on time-varying and time-invariant features. Data was used within the 8 or 24 hours preceding a scan to make predictions for the 8-hour maximum and 24-hour maximum targets respectively, as these are the only data points with valid prediction targets. This provided at most eight observations per patient scan in the 8-hour maximum prediction task and at most 24 observations in the 24-hour maximum prediction task. For the MGB dataset, our data transformations resulted in 7,216 observations for the 8-hour maximum prediction task and 12,830 observations for the 24-hour maximum prediction task. For the BMC dataset, our transformations resulted in 1,891 observations for the 8-hour maximum prediction task and 3,713 observations for the 24-hour maximum prediction task. Our total observations per class and task are reported in table 7.2.

Data set	Task	Target MLS class			
		0	0-3mm	3-8mm	>8mm
MGB	8-hour maximum prediction	1,891	1,340	2,834	1,151
	24-hour maximum prediction	2,379	2,196	5,620	2,635
BMC	8-hour maximum prediction	523	309	620	439
	24-hour maximum prediction	873	696	1,223	921

Table 7.2: Number of Observations per Task and Class

7.3.5 Data Augmentation

Encoding

Categorical features were converted into one-hot representation to allow for processing. All features were normalised via standard scaling. Time was represented on a per-hour basis starting at the time of admission.

Censoring

Static data in patient records included events that happened over the course of admission (such as DHC surgery) and the time of these events. Including such data in its raw form for per-hour prediction tasks starting at admission would result in the leakage of data from the future into training and testing examples. To alleviate this, temporal censoring of static variables was performed until they occurred in the test dataset (i.e. events such as DHC surgery were removed from the data record until the hour in which they actually occurred).

Imputation

Missing static data was imputed using mean column value. For dynamic data, it is not possible to interpret if data are missing and as such standard imputation is not possible. Instead, it was opted to impute temporal data using data from the past: given some of the models at hand would make predictions using single-hour rows of patient data, information from the past would be missing in patient

observations at the time of model input. To improve performance, it was thus opted to carry forward information from the past in each per-hour row of patient data by transforming the 26 dynamic variables: for each per-hour row of data, features were added indicating the most recent recorded past value (a forward fill), and the maximum recorded value in the past 24 hours (a rolling maximum), for each time-series feature. This resulted in models being trained on a total of 91 features. The approach allowed us to leverage time-varying observations from the past without leaking information about the future.

7.3.6 Separability Assessment

After applied transformations, data separability was visualised using t-SNE and PCA plots, with the goal of determining which classes may be harder to separate from other classes. This is visible in figure 7.2. In the t-SNE data, it was observed that partial separability of classes in a two-dimensional setting, with MLS-0 groups being almost entirely separable from the MLS-8+ groups in the representation manifold space. Critically, it was also observed that directionality in the separation of classes, with MLS-0-8 appearing between the extremes. This indicates that the data preparation pipeline used preserves key structures in data and is likely sufficient for machine learning.

7.3.7 Training, Testing and Evaluation

It was opted to train and test models on the MGB dataset using five-fold cross-validation. Data partitioning was made at the patient level to ensure there was no leakage of observations from the same patient in the train and test set. It was then opted to use the entirety of the BMC dataset as an external validation cohort to assess the generalisation performance of the best-performing model.

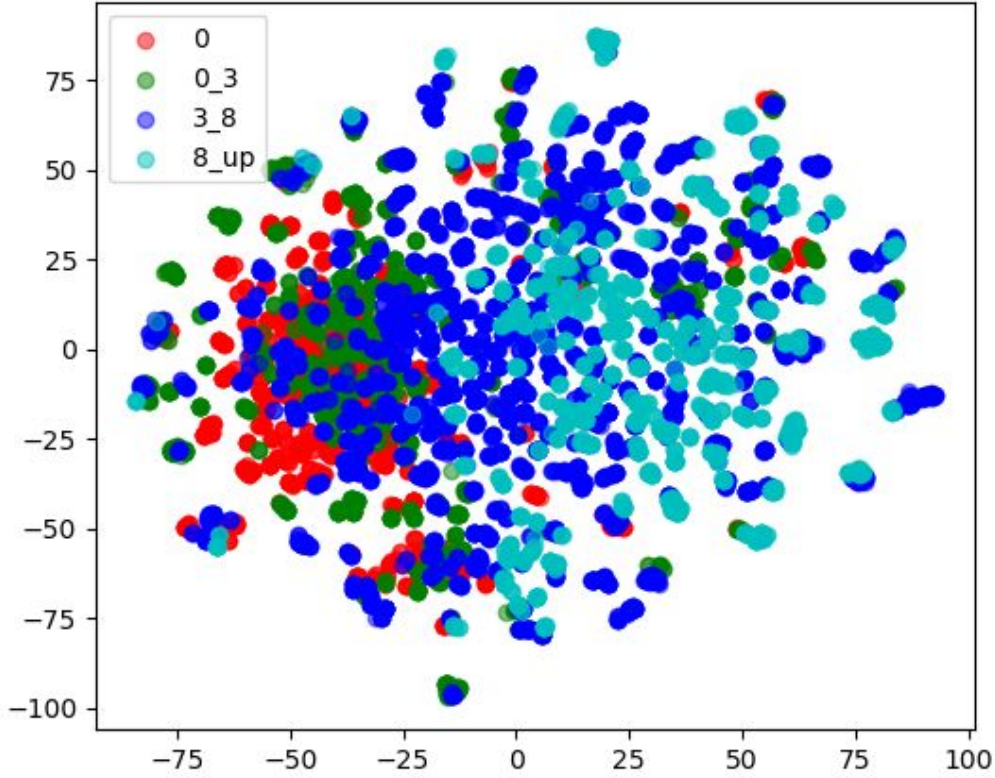


Figure 7.2: A t-SNE visualisation of the data from the 24-hours lookahead task: it was observed that in these t-SNE plots, there is directionality in the position of target clusters corresponding to the degree of severity of future mass effect events.

Given the small size of the dataset relative to the data used in chapter 5, we used relatively small models and could thus afford to perform thousands of experimental runs. As such, we opted to perform Bayesian Hyperparameter optimisation to uncover optimal model parameters instead of a Grid Search, with the objective being to maximise mean AUROC across the cross-validation folds.

7.3.8 Models Evaluated

Baseline logistic regression models, ensemble tree models, and time-series deep learning models were all evaluated. For baseline and ensemble tree models, models

using single-hour rows of patient data were trained and evaluated. For time-series deep learning models, sliding windows of 24 hours of patient data were used.

Baseline Models

It was first opted to create a simple baseline model that could act as a comparator for other methods. Accordingly, it was opted to use a regression model akin to the models commonly reported in the existing literature [272, 273, 280]. In particular, a logistic regression was created modelled after the Enhanced Detection of Edema in Malignant Anterior Circulation Stroke (EDEMA) Score system developed by Ong et al. [271], which is a regression model for predicting malignant oedema after stroke. The model leverages a binary logistic regression taking as input the following variables: the Effacement, Admission MLS, Glucose, Previous stroke, and the use of tPA or thrombectomy interventions. A logistic regression baseline was trained based on this model. This model uses the features of tPA, thrombectomy, most recent MLS, previous stroke history, and admission glucose level. This model does not make use of effacement (which is used in the original EDEMA score model), as this variable was not available in either the MGB or BMC datasets.

Ensemble Tree Models

The performance of Random Forest models and XGBoost models were assessed. The full list of hyperparameters optimised using Bayesian hyperparameter optimisation are documented in appendix B.1.

Deep Learning Models

The performance of LSTMs, Transformers, NPTs, and NPPs were assessed. Per the model, sweeps were performed using Bayesian hyperparameter optimisation over the following parameters: optimisers, batch size, model depth, embedding

dimensionality, loss functions, feature embedding dimensions, learning rate, and levels of weight decay. The full list of hyperparameters optimised over is documented in appendix B.2.

7.4 Results and Analysis

7.4.1 MGB Cross-Validation Results

Results obtained are reported in table 7.3. It was observed that deep learning models were unable to beat the baseline logistic regression models used on the MGB dataset. For ensemble tree models, the proposed models result in a mean AUROC across the four classes of 86% ($\pm 0.83\%$) for XGB for predicting the 24-hour window maximum MLS and 86% ($\pm 0.59\%$ for XGB) for predicting 8-hour window maximum MLS. In both settings, both models significantly outperformed the logistic regression baseline ($p < 10^{-3}$ across all tasks for AUROC and AUPRC using a t-test). Mean AUROC scores exceeding 80% indicate these models' strong discriminatory ability to distinguish future MLS categories.

7.4.2 BMC External Validation Results

We selected the XGB model as the best performing model, and performed in-depth analysis of generalisation performance to the unseen BMC dataset. This performance was compared to the performance of a baseline logistic-regression model trained on the BMC dataset using five-fold cross-validation. These results are reported in table 7.4. The XGB models achieve a mean AUROC of 80.28% ($\pm 0.83\%$) for predicting a 24-hour window maximum MLS and a mean AUROC of 80.63% ($\pm 0.59\%$) for predicting the 8-hour window task. These AUROC scores indicate that these models can successfully generalise to novel datasets. In the 8-hour prediction outcome, the XGB model significantly outperformed the linear

Prediction Task	Model	AUROC	AUPRC	Accuracy
24h ahead	EDEMA Score Baseline	78.43%±0.12%	49.40%±0.54%	56.12%±0.29%
	XGBoost	86.05%±0.83%	68.65%±0.95%	62.11%±1.79%
	Random Forest	86.31%±0.17%	68.39%±0.41%	61.67%±0.42%
	LSTM	69.67%±0.39%	42.32%±0.73%	53.66%±0.96%
	Transformer	61.33%±0.59%	40.22%±0.80%	53.94%±0.67%
	NPT	58.19%±0.67%	40.15%±0.34%	50.10%±0.44%
	NPP	62.94%±0.25%	39.47%±0.27%	44.89%±0.70%
8h ahead	EDEMA Score Baseline	76.66%±0.15%	48.17%±0.67%	53.51%±0.24%
	XGBoost	86.20%±0.59%	67.33%±0.52%	62.76%±1.36%
	Random Forest	85.94%±0.06%	67.25%±0.19%	62.94%±0.44%
	LSTM	68.92%±0.72%	41.09%±0.86%	53.43%±0.82%
	Transformer	60.67%±0.93%	38.99%±0.93%	52.71%±1.29%
	NPT	57.53%±0.60%	37.92%±0.55%	49.87%±0.92%
	NPP	62.28%±0.85%	38.24%±0.94%	44.66%±0.60%

Table 7.3: Prediction results of each model on the MGB dataset. $\pm$ values reflect the 95% confidence intervals.

Prediction Task	Model	AUROC	AUPRC	Accuracy
24h ahead	Regression Baseline	78.77%±0.50%	51.00%±0.47%	48.59%±3.75%
	XGBoost trained on MGB data	80.28%±0.83%	55.15%±1.41%	56.65%±0.55%
8h ahead	Regression Baseline	76.23%±0.15%	46.92%±0.33%	53.84%±2.00%
	XGBoost trained on MGB data	80.63%±0.59%	56.31%±0.71%	58.99%±0.35%

Table 7.4: Prediction results of each model on the BMC dataset. $\pm$ values reflect the 95% confidence intervals.

baseline trained on BMC data in AUROC ($p < 10^{-3}$), AUPRC ($p < 10^{-3}$), and accuracy ($p = 10^{-3}$). In the 24-hour prediction task, the XGB model significantly outperformed the baseline trained on BMC data in AUPRC ($p < 10^{-3}$) and accuracy ($p < 5 \times 10^{-3}$), while performing equivalently on AUROC. Critically, this increase in performance relative to baselines is observed despite the XGB models not being trained on BMC data. The comparative performance of XGB models on the MGB and BMC data was visualised on a per-class basis in figures 7.3 and 7.4.

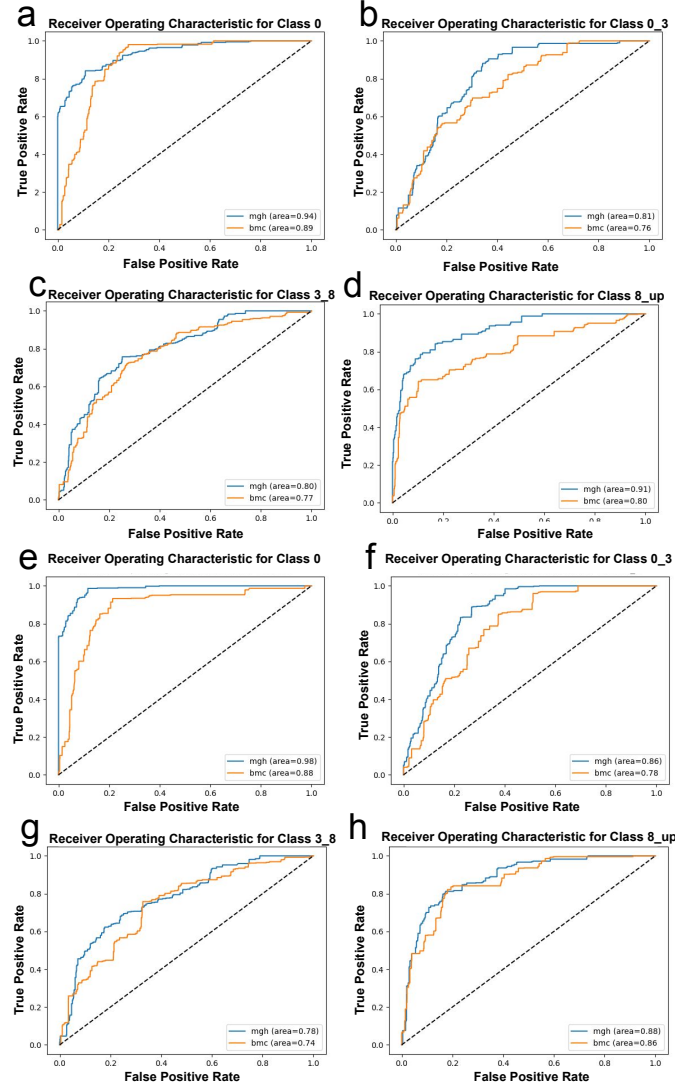


Figure 7.3: Receiver operating characteristic curves for XGBoost models grouped by the most recent MLS scan value seen by the model. Plots a - d show scores for the 8-hour prediction task; plots e - h show scores for the 24-hour prediction task. Orange lines show scores for BMC; blue lines show scores for MGH. a/e: XGBoost when most recent MLS scan can value was zero; b/f: XGBoost when most recent MLS scan can value was between in range 0-3; c/g: XGBoost when most recent MLS scan can value was between in range 3-8; d/h: XGBoost when most recent MLS scan can value was greater than 8

7.4.3 Feature Importances and Clinical Interpretation

A SHAP analysis was provided on each of the four prediction targets for the XGB models at hand. Figure 7.5 shows these SHAP bar plots for the top-performing

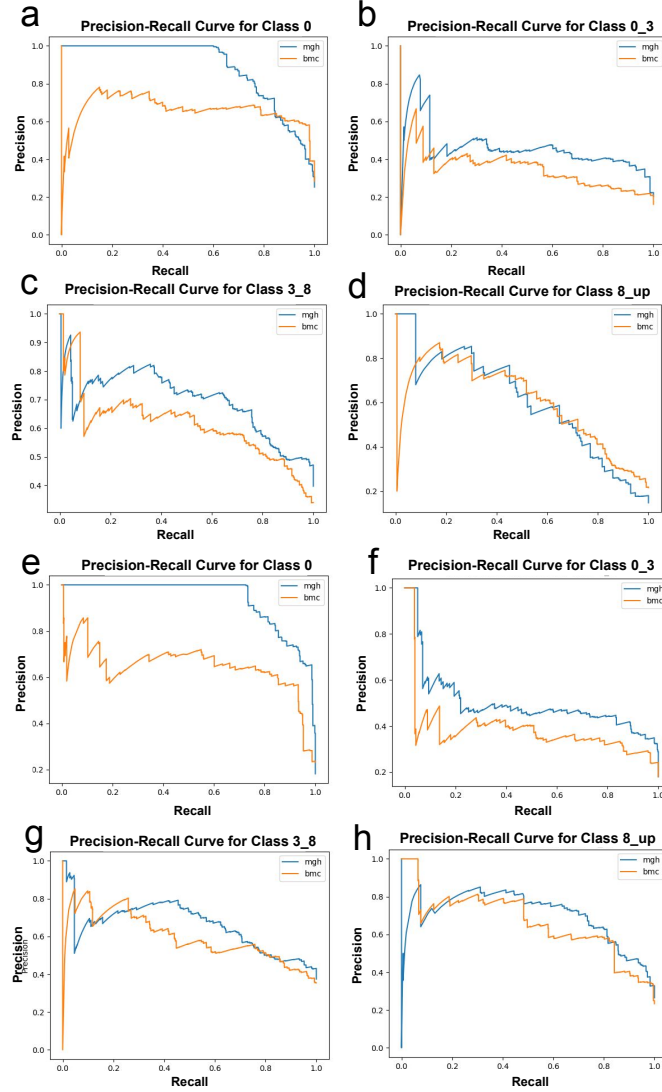


Figure 7.4: Precision recall curves for XGBoost models grouped by the most recent MLS scan value seen by the model. Plots a - d show scores for the 8-hour prediction task; plots e - h show scores for the 24-hour prediction task. Orange lines show scores for BMC; blue lines show scores for MGH. a/e: XGBoost when most recent MLS scan can value was zero; b/f: XGBoost when most recent MLS scan can value was between in range 0-3; c/g: XGBoost when most recent MLS scan can value was between in range 3-8; d/h: XGBoost when most recent MLS scan can value was greater than 8

XGB models across all tasks considered. Across each of the settings, it was observed that the most predictive features directly relate to previous imaging characteristics. These include, but are not limited to, the size of the most recent MLS event, the time and presence of the first MLS, and Pineal Gland Shift (PGS),



Figure 7.5: The 10 features with the highest SHAP values per XGBoost model grouped by the most recent MLS scan value seen by the model. Plots a - d show scores for the 8-hour prediction task; plots e - h show scores for the 24-hour prediction task. a/e: XGBoost when most recent MLS scan can value was zero; b/f: XGBoost when most recent MLS scan can value was between in range 0-3; c/g: XGBoost when most recent MLS scan can value was between in range 3-8; d/h: XGBoost when most recent MLS scan can value was greater than 8

a feature that indicates whether a PGS is present. Shift of the pineal gland may occur before MLS occurs and is potentially a more sensitive indicator of future states [281]; the extent of MLS in the prior window of time is unsurprisingly

the most significant predictor of subsequently increasing MLS. Together with the high importance conferred by the time after stroke onset, these variables reflect the natural course of oedema growth, which has been historically described to peak between 2 - 5 days [265, 267, 282–285].

Also included was ASPECTS, the Alberta Stroke Program Early CT Score (which is a topographic score of stroke extent imaged on CT of MCA stroke patients with lower score signifying greater stroke burden and less unaffected tissue [286]). SHAP analysis suggested that the ASPECTS at follow-up [286], reflecting irreversible stroke volume, is a better predictor of increasing oedema than ASPECTS at presentation [287]. For a higher degree of MLS class, PGS and the ASPECTS become increasingly important. It was also found that glucose levels and HbA1c (a marker of impaired glucose over three months) were important predictors for several of the four classifications. In a prior study, elevated glucose on presentation has been significantly associated with the development of malignant oedema and worse overall outcome [271], yet, HbA1c was not a strictly independent predictor of worsening oedema. However, higher HbA1c may suggest impaired endothelium and blood-brain-barrier integrity, resulting in worsening MLS. Given this disagreement, further study is needed to elucidate stronger correlation or causal relationships. Ultimately, these results suggest that while the most important contributors to later increased mass effect are intuitive and well-known, there are several potentially under-utilised biomarkers that, when incorporated across time in a standardised way, may improve risk stratification.

7.5 Conclusion

In summary, a pipeline for predicting future MLS in a clinical dataset was successfully developed. Critically, models could successfully generalise to novel

datasets. It was established that ensemble tree models performed better than deep learning models, most likely due to the dataset size. This finding highlights the need to select appropriate models based on the dataset characteristics. The SHAP analysis of prediction models helped identify key predictors such as previous imaging characteristics and glucose levels. These findings could guide clinicians in monitoring patients at risk of MLS more effectively, and also help them make informed care decisions.

7.5.1 Assessment of Model Performance

While tree-based models performed well in the prediction task, the deep learning models used ultimately failed to perform effectively for the task at hand. When inspected, it was found that models overfitted to training data, with some models achieving perfect accuracy scores on the training set.

Conventional wisdom suggests this failure of performance is because deep learning models, with their high number of parameters, are overparameterised for these settings. However, deep learning models can perform successfully in many settings where they are overparameterised.

Two factors that drive this unexpected success are the double descent phenomena [288] and grokking [289]. The double descent phenomenon (figure 7.6) is characterised by a reduction in test error when increasing the number of parameters past the interpolation threshold, the point where the number of parameters equals the number of data points in a model. The double descent phenomena are not guaranteed, however, and depend on features such as test data distribution and training data singular values [290].

Grokking (figure 7.7) is the name given to the phenomenon where a model that is overfitting on training data can suddenly rapidly improve its performance

on testing data in training. This phenomenon is not guaranteed and is typically observed only after a significant number of optimisation steps. Empirical results indicate weight decay is effective at driving grokking, with these smaller coefficients allowing for an over-parameterised model to still effectively fit data [289].

Ultimately, however, it was not possible to observe either double descent or grokking phenomena in the deep learning model. While the most obvious explanation for poor performance is the over-parameterisation of deep learning models relative to the dataset size, other factors are included that likely will have contributed to this difference in performance. Of particular interest is the ease of regularisation: ensemble tree methods like XGBoost are heavily regularised through multiple mechanisms like minimum splits in leaf nodes and maximum branch depths, and these simple regularisation methods help prevent overfitting.

7.6 Limitations

There are several limitations to the work in this chapter. Similar to other studies, the dataset is missing information since it is based on retrospective datasets collected from hospitals EHRs. Physical examination data was inaccessible for this study, which is considered a significant component of neurological assessment. At least 10% of patients had missing values in the following static variables: weight, pulse data, temperature, admission HbA1c, and NIH Stroke Scale score. For weight, pulse, and temperature data, at least 60% of patients were missing this information. In these instances, the majority of values were imputed. Features that are mostly missing are less likely to be informative, and this likely has a significant effect on our model coefficients. Recovering such data may reveal other features to be informative. For our dynamic variables in MGH data, we observed significantly that more events per patient reported after 2016 (see figure

7.1) due to a change in the underlying EMR system used. This functionally results in there being two distributions of data present in the training set. The effect this may have on model performance and bias is less clear. Still, it is believed that the successful external generalisation to BMC data demonstrates that data from both distributions is useful.

The temporal transformations of data may also include potential biases as they result in multiple observations of patients with higher data completion rates. The approach conducted in this thesis, which uses maximum and most recent values, while compact, may mask some step changes in data. The transformation also masks the absence of new data, which can be a feature in itself. It is also important to acknowledge that while the BMC and MGB patient cohorts significantly differ in racial and socioeconomic composition, both hospitals are still within the New England area, and generalisation beyond this region has not been assessed.

Finally, while there is a multi-year time gap between when MGB and BMC data was recorded, this study is ultimately retrospective. Further validation is required through real-time prediction in a prospective clinical study to further guarantee model performance and measure the potential impact it will have in clinical practice.

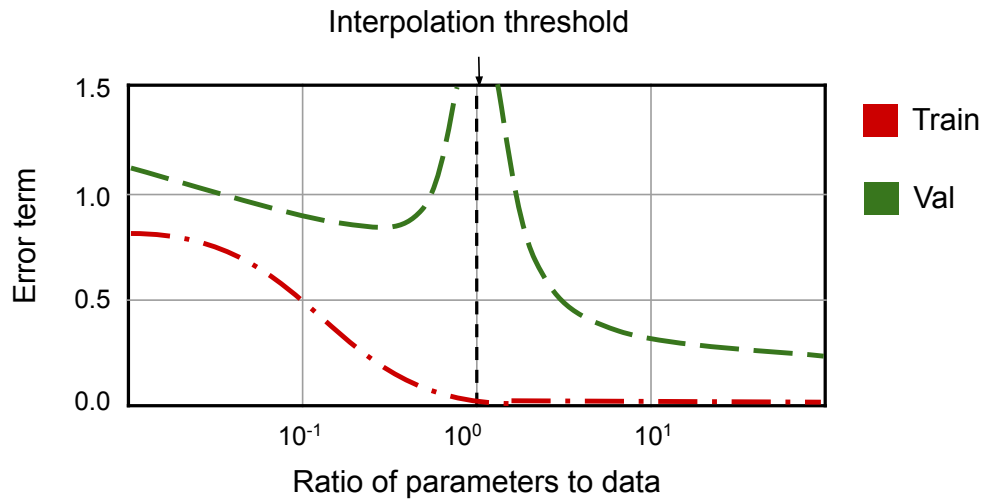


Figure 7.6: The Double-Descent phenomenon: conventional understandings in machine learning suggest that as the number of parameters in a model approaches the number of data points, there will be a rapid decrease in validation performance (indicated in green) as the model overfits training data (indicated in red). However, in deep learning, a double-descent phenomenon is frequently observed, where a model’s validation performance increases once the number of parameters significantly exceeds the amount of training data

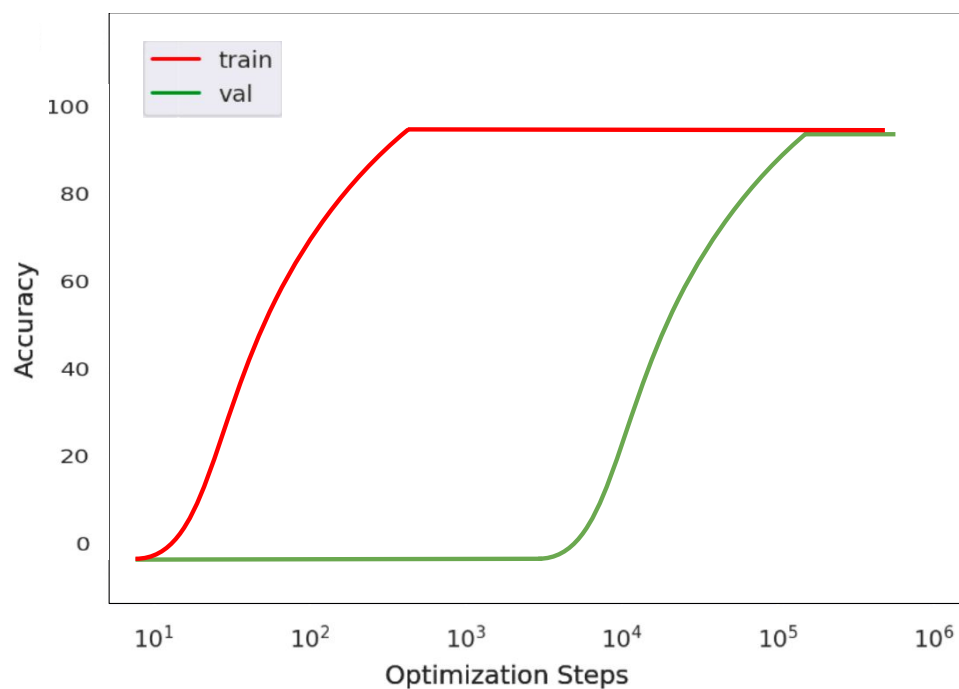


Figure 7.7: An illustration of the grokking phenomenon: note that the x-axis is logarithmically scaled. Models can initially significantly overfit on training data, as indicated by a significant difference in train dataset performance (red) and validation performance (green). However, at an order of magnitude higher number of optimisation steps, validation accuracy can significantly and suddenly increase. This phenomenon is known as grokking and appears as a new optimal minima discovered through weight decay

8

Conclusion

8.1 Novel Contributions of this thesis

This thesis makes the following novel contributions to the literature:

- In Chapter 5, I propose two new transformer-based machine learning methods: the non-parametric perceiver (NPP), and the hierarchical perceiver. I evaluate the efficacy of these novel methods on a set of benchmark medical machine learning tasks in Chapter 5, and in an applied medical machine learning task in Chapter 7.
- In Chapter 6, I develop a machine learning model for predicting the risk of adverse events from the ChAdOx1 nCoV-19 vaccine. This research was entirely novel within the context of the emerging pandemic, as is the novel finding of a potential association between blood clot risk and the absence of previous vaccinations.
- In Chapter 7, I developed a dynamic model for predicting the risk of midline shift in neurointensive care patients with ischaemic stroke. Up until this

point, all predictive models in the literature (that we have identified) have been static prediction models. This transition to dynamic prediction models allows for predictions to be updated over time, and may improve outcomes for patients over the course of their hospital stay.

8.2 Summary of Results

8.2.1 Developing Novel Deep Learning Models for Binary Classification on Electronic Health Record Data

In Chapter 5, it was demonstrated that Transformer and Perceiver models can be used as effectively as RNN models for binary classification tasks in the medical domain. Models on the MIMIC-III dataset were evaluated on the tasks of predicting shock, AF and mortality at different time horizons. Across the different machine learning models evaluated (RNNs, LSTMs, GRUs, Transformers and Perceivers) it was found that models performed well: across all tasks, a top validation accuracy of at least 97% was observed, and a top validation AUROC of at least 91% was observed. Furthermore, a new architecture was proposed, the NPP. It was found that this model also performed well with an AUROC exceeding 91% while offering linear-time efficiency, indicating the possible use of this model in more resource-restricted settings.

8.2.2 Working with High-Class Imbalance in a Low-Computing Settings

In Chapter 6, machine learning models were developed for predicting complications arising from COVID-19 vaccination. This work was conducted using the limited computational resources available through a trusted research environment, and a problem of extreme class imbalance was faced, with the positive class being only 0.04% of events in the available EHR data. Genetic algorithm approaches,

gradient boosting trees and Naive Bayes algorithms were evaluated. Naive Bayes as a method was found to be the most effective, with the highest AUROC achieved for this work being 70.1%. At this level of class imbalance, oversampling was found to be an ineffective mechanism for improving performance. Models were evaluated on age-specific subgroups. Models trained on subsections of data with an easy classification boundary were also evaluated. With the age-specific subgroups, it was identified that age-related variables impact classification decisions in older age groups, implying that there are difficulties in separating genuine vaccine-related adverse events from age-related thrombotic events. With the subsection training, it was identified that prior vaccination history can be used to create a classification boundary with an AUROC of 64.7%.

8.2.3 Working with High-Dimensional Small Sparse Data Sets

In Chapter 7, machine learning methods were applied to the task of predicting MLS events from EHR data in neurointensive care. This research using a small dataset of 675 patients was conducted and investigated with both traditional methods and the deep learning methods introduced in Chapter 5. It was found that XGBoost models achieved the highest performance in this setting, and that the novel architectures proposed were likely over-parameterised for the task. In-depth follow-up evaluation and analysis was conducted on the best-performing models: we demonstrated that the XGBoost models could generalise to an out-of-distribution dataset, with AUROC scores exceeding 80%. We also performed in-depth analysis of feature importance scores extracted from the models, which in turn can in turn inform clinical decision making in neurointensive care.

8.3 Addressing the Research Question and Objectives of the Thesis

This thesis set out with the primary goal of answering the following research question: **is it possible to build a single unified machine learning architecture appropriate for all tasks in EHR-based machine learning?** I thus started this thesis by developing models with desirable properties for different EHR-based machine learning tasks, and then moved to applied domains to assess the efficacy of these models. Ultimately, it was found that real-world data settings and computational environments may not make theoretically ideal models possible. Trusted Research Environment (TRE) settings can limit access to computational resources, and small numbers of data points can make larger models with more parameters difficult to train. The two applied settings: high volumes of static data in computationally restricted secure environments, against low volumes of dynamic data in unrestricted environments, resulted in notably different model requirements. This thesis highlights both the breadth of machine learning approaches that are required to achieve empirically useful results in medical settings and the potential limitations of using models only suited to idealised benchmark datasets.

The Secondary objective of this thesis was to generate results that could potentially improve clinical practice. Overall, I feel this thesis has generated multiple useful medical contributions. These include the identification of potential risk factors in adverse COVID-19 vaccination response and the creation of new ways to dynamically predict the risk of midline shift in ischaemic stroke. These medical contributions will need further research and development to apply to clinical practice, and I discuss how these medical contributions could be further researched in the next section.

8.4 Future work

Further Epidemiological Research

In Chapter 6, some features identified in SHAP analyses suggest a lack of previous vaccine exposure as a risk factor for vaccine complications. This observation does not appear to be reported elsewhere in the literature and should perhaps be analysed with a fully controlled retrospective epidemiological study at scale. This could be particularly important research given issues in the last 30 years around the public perception of vaccination. Given some of the historical issues around public perception of vaccine epidemiology research, however, careful thought should both go into study design and the reporting strategy of potential results. Given it is now understood that the rare thrombotic events caused by vaccination are also caused by COVID-19 infection, it may also be worth investigating if childhood vaccination has any protective effects against these virus-induced thrombotic events.

8.4.1 Causal Inference Research

In Chapter 6, one issue encountered in research was in the identification of factors that were specific to vaccine-induced thrombotic events and were not age-related thrombotic events. Further research needs to be conducted to attempt to disentangle age-related and vaccine-induced dynamics. Such research falls under the domain of causal inference [291] and in particular environment-based methods such as Invariant Risk Minimisation (IRM) [148].

IRM is designed to elucidate invariant relationships from empirical data. In particular, IRM is a learning method that seeks to identify a data representation equally optimal across different “environments”. Environments are groups of data that are identified by an oracle as being from different distributions. IRM

is, in theory, suitable for the the task of disassociating age-related and vaccine-induced dynamics: by treating different age groups as different environments, factors that are constant across environments can be extracted. Invariance is not a guarantee of causality, as not all invariant features are causal. However, given causal features are invariant, they can be used to potentially identify better candidate features to evaluate.

IRM does, however, have limitations: the original description of the method assumes that relationships between predictors and outcomes are linear [148]. Nonlinear extensions using multi-layer perceptrons have been proposed [292], but other results have suggested IRM can fail catastrophically in the non-linear domain [293]. Furthermore, IRM is affected by class imbalance in non-trivial ways [294]. Further research on using causal inference on the dataset in chapter 6 would thus also need to evaluate the IRM method at the fundamental level, and potentially propose changes to the method.

8.4.2 Further Research from Chapter 7

There are two avenues the author will explore to extend this research in Chapter 7. These are human benchmarking and integration into Markov Decision Processes (MDPs).

8.4.3 Human Benchmarking

To consider the utility of a machine learning system for predicting MLS in clinical practice, it is important to perform human benchmarking. A follow-up study to this work thus involves benchmarking the human performance of predicting MLS from available data. There are some important design considerations for a study such as this: because humans struggle with making decisions using the representations of data that machine learning models use, and as such, may use signals not

available to the machine learning model when making decisions in real-world settings, significant effort should be put into building what is considered a ‘fair’ representation of data for humans to use, to make their classification predictions. Possible examples include giving humans access to time-censored free-text clinical notes, which the machine learning model does not have access to currently.

8.4.4 Integration into Markov Decision Processes

Predictions of MLS are useful in informing when a clinician may want to scan a patient, but there were no efforts in Chapter 7 to create an optimised scanning policy. CT scanning is a limited resource in hospital settings, and scanning a patient brings with it associated costs (in particular the removal of access to other care during this scanning process). It is thus appropriate to consider building a mathematical model that proposes an optimal scanning policy that maximises patient outcomes. One possible class of approaches to consider are MDPs [295] and in particular Partially Observable Markov Decision Process (POMDP) [296]. The goal of an MDP is to find an optimal action policy (in this case whether or not to scan) that maximises the expected sum of rewards over time (in this case patient outcomes). POMDPs are an extension to the standard MDP which provides a framework for making decisions when a current state (in this case MLS) cannot be directly observed. This approach would integrate well with the current machine learning work, with the model predictions being usable as indirect observations. There is also an opportunity for novel research to be conducted in this area by using machine learning models to create custom transition probability kernels for each patient, which are a key component of POMDP predictions. The author will be working on this in further studies.

References

- [1] R Scott Evans. “Electronic health records: then, now, and in the future”. In: *Yearbook of Medical Informatics* 25.S 01 (2016), S48–S61.
- [2] Sergio Miranda Freire et al. “Comparing the performance of NoSQL approaches for managing archetype-based electronic health record data”. In: *Plos One* 11.3 (2016), e0150069.
- [3] Ashish K Jha et al. “Use of electronic health records in US hospitals”. In: *New England Journal of Medicine* 360.16 (2009), pp. 1628–1638.
- [4] Alison Callahan and Nigam H Shah. “Machine learning in healthcare”. In: *Key Advances in Clinical Informatics*. Elsevier, 2017, pp. 279–291.
- [5] Vincent X Liu et al. “The number needed to benefit: estimating the value of predictive analytics in healthcare”. In: *Journal of the American Medical Informatics Association* 26.12 (2019), pp. 1655–1659.
- [6] Edward P Ambinder. “Electronic health records”. In: *Journal of Oncology Practice* 1.2 (2005), p. 57.
- [7] Vera Ehrenstein et al. “Obtaining Data From Electronic Health Records”. In: *Tools and Technologies for Registry Interoperability, Registries for Evaluating Patient Outcomes: A User’s Guide, 3rd Edition, Addendum 2 [Internet]*. Agency for Healthcare Research and Quality (US), 2019.
- [8] Michael L Clark and Parveen Kumar. *Kumar and Clark’s Clinical Medicine*. 2017.
- [9] Kevin Donnelly et al. “SNOMED-CT: The advanced terminology and coding system for eHealth”. In: *Studies in Health Technology and Informatics* 121 (2006), p. 279.
- [10] James E Harrison et al. “ICD-11: an international classification of diseases for the twenty-first century”. In: *BMC Medical Informatics and Decision Making* 21 (2021), pp. 1–10.
- [11] James B Saxe and Jon Louis Bentley. “Transforming static data structures to dynamic structures”. In: *20th Annual Symposium on Foundations of Computer Science (SFCS 1979)*. IEEE. 1979, pp. 148–168.
- [12] Daniel R Masys et al. “Technical desiderata for the integration of genomic data into Electronic Health Records”. In: *Journal of Biomedical Informatics* 45.3 (2012), pp. 419–422.

- [13] Zhijin Wu and Zhiqiang Wu. “Exploration, Visualization, and Preprocessing of High-Dimensional Data”. In: *Statistical Methods in Molecular Biology* (2010), pp. 267–284.
- [14] David R Brillinger. *Time series: data analysis and theory*. SIAM, 2001.
- [15] Shenda Hong et al. “Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review”. In: *Computers in Biology and Medicine* 122 (2020), p. 103801.
- [16] Nishkam Ravi et al. “Activity recognition from accelerometer data”. In: vol. 5. 2005. Pittsburgh, PA. 2005, pp. 1541–1546.
- [17] Pedram Afshar et al. “A translational platform for prototyping closed-loop neuromodulation systems”. In: *Frontiers in Neural Circuits* 6 (2013), p. 117.
- [18] Karen A Wager et al. “Comparison of the quality and timeliness of vital signs data using three different data-entry devices”. In: *CIN: Computers, Informatics, Nursing* 28.4 (2010), pp. 205–212.
- [19] Cynthia C Chernecky and Barbara J Berger. *Laboratory tests and diagnostic procedures-E-Book*. Elsevier Health Sciences, 2012.
- [20] Simon de Lusignan et al. “The Oxford Royal College of General Practitioners Clinical Informatics Digital Hub: Protocol to Develop Extended COVID-19 Surveillance and Trial Platforms”. In: *Journal of Medical Internet Research (JMIR) Public Health and Surveillance* 6.3 (2020), e19773.
- [21] Hiromitsu Kobayashi. “Effect of measurement duration on accuracy of pulse-counting”. In: *Ergonomics* 56.12 (2013), pp. 1940–1944.
- [22] Noa Kallioinen et al. “Sources of inaccuracy in the measurement of adult patients’ resting blood pressure in clinical settings: a systematic review”. In: *Journal of Hypertension* 35.3 (2017), pp. 421–441.
- [23] Nguyen P Nguyen et al. “Brain midline shift detection and quantification by a cascaded deep network pipeline on non-contrast computed tomography scans”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 487–495.
- [24] Nikos K Logothetis. “What we can do and what we cannot do with fMRI”. In: *Nature* 453.7197 (2008), pp. 869–878.
- [25] Morgan P McBee et al. “Deep learning in radiology”. In: *Academic Radiology* 25.11 (2018), pp. 1472–1480.
- [26] Ewout J Hoorn and Robert Zietse. “Hyponatremia and mortality: moving beyond associations”. In: *American Journal of Kidney Diseases* 62.1 (2013), pp. 139–149.
- [27] C Thomas Wass and William L Lanier. “Glucose modulation of ischemic brain injury: review and clinical recommendations”. In: *Mayo Clinic Proceedings*. Vol. 71. 8. Elsevier. 1996, pp. 801–812.
- [28] David Edelman et al. “Utility of hemoglobin A1c in predicting diabetes risk”. In: *Journal of General Internal Medicine* 19.12 (2004), pp. 1175–1180.

- [29] John Steer, John Gibson, and Stephen C Bourke. “The DECAF Score: predicting hospital mortality in exacerbations of chronic obstructive pulmonary disease”. In: *Thorax* 67.11 (2012), pp. 970–976.
- [30] Christian P Subbe et al. “Validation of a modified Early Warning Score in medical admissions”. In: *Quarterly Journal of Medicine (QJM)* 94.10 (2001), pp. 521–526.
- [31] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), pp. 583–589.
- [32] Oliver Hennigh et al. “NVIDIA SimNet™: An AI-Accelerated Multi-Physics Simulation Framework”. In: *International Conference on Computational Science*. Springer. 2021, pp. 447–461.
- [33] Khaled El Emam, Sam Rodgers, and Bradley Malin. “Anonymising and sharing individual patient data”. In: *British Medical Journal (BMJ)* 350 (2015).
- [34] Willem G Van Panhuis et al. “A systematic review of barriers to data sharing in public health”. In: *BMC Public Health* 14.1 (2014), pp. 1–9.
- [35] Matthew BA McDermott et al. “Reproducibility in machine learning for health research: Still a ways to go”. In: *Science Translational Medicine* 13.586 (2021).
- [36] Mackenzie Graham et al. “Trust and the Goldacre Review: why trusted research environments are not about trust”. In: *Journal of Medical Ethics* 49.10 (2023), pp. 670–673.
- [37] Sanaz Kavianpour et al. “Next-generation capabilities in trusted research environments: interview study”. In: *Journal of Medical Internet Research* 24.9 (2022), e33720.
- [38] Po-Hsuan Cameron Chen, Yun Liu, and Lily Peng. “How to develop machine learning models for healthcare”. In: *Nature Materials* 18.5 (2019), pp. 410–414.
- [39] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 248–255.
- [40] Jort F Gemmeke et al. “Audio set: An ontology and human-labeled dataset for audio events”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 776–780.
- [41] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European Conference on Computer Vision*. Springer. 2014, pp. 740–755.
- [42] Fei Yin et al. “ICDAR 2013 Chinese handwriting recognition competition”. In: *2013 12th International Conference on Document Analysis and Recognition*. IEEE. 2013, pp. 1464–1470.
- [43] Julian McAuley et al. “Image-based recommendations on styles and substitutes”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2015, pp. 43–52.
- [44] Alistair EW Johnson et al. “MIMIC-III, a freely accessible critical care database”. In: *Scientific Data* 3.1 (2016), pp. 1–9.

- [45] Tom J Pollard et al. “The eICU Collaborative Research Database, a freely available multi-center database for critical care research”. In: *Scientific Data* 5.1 (2018), pp. 1–13.
- [46] Jeremy Irvin et al. “Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 590–597.
- [47] Shirly Wang et al. “Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii”. In: *Proceedings of the ACM Conference on Health, Inference, and Learning*. 2020, pp. 222–235.
- [48] Roy PC Kessels. “Patients’ memory for medical information”. In: *Journal of the Royal Society of Medicine* 96.5 (2003), pp. 219–222.
- [49] Stephanie Vanderhook and Joanna Abraham. “Unintended consequences of EHR systems: a narrative review”. In: *Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care*. Vol. 6. 1. SAGE Publications Sage CA: Los Angeles, CA. 2017, pp. 218–225.
- [50] Mathieu Boniol et al. “The global health workforce stock and distribution in 2020 and 2030: a threat to equity and ‘universal’ health coverage?” In: *BMJ Global Health* 7.6 (2022), e009316.
- [51] M Sreenivasan and Anu Mary Chacko. “Interoperability issues in EHR systems: Research directions”. In: *Data Analytics in Biomedical Engineering and Healthcare* (2021), pp. 13–28.
- [52] Timothy Stablein et al. “The catch to confidentiality: the use of electronic health records in adolescent health care”. In: *Journal of Adolescent Health* 62.5 (2018), pp. 577–582.
- [53] Subrata Chattopadhyay. “Corruption in healthcare and medicine: Why should physicians and bioethicists care and what should they do”. In: *Indian J Med Ethics* 10.3 (2013), pp. 153–9.
- [54] Andreas Eckner. “A framework for the analysis of unevenly spaced time series data”. In: *Preprint. Available at: http://www.eckner.com/papers/unevenly_spaced_time_series_analysis* (2012), p. 93.
- [55] Ari Garber et al. “Frequency, Timing, and Types of Medication Ordering Errors Made by Residents in the Electronic Medical Records Era.” In: *Southern Medical Journal* 112.1 (2019), pp. 25–31.
- [56] Stephan Rabanser, Stephan Gunnemann, and Zachary C Lipton. “Failing loudly: An empirical study of methods for detecting dataset shift”. In: *arXiv preprint arXiv:1810.11953* (2018).
- [57] JY Kang et al. “Systematic review: worldwide variation in the frequency of coeliac disease and changes over time”. In: *Alimentary Pharmacology & Therapeutics* 38.3 (2013), pp. 226–245.
- [58] Tine Jess et al. “Decreasing risk of colorectal cancer in patients with inflammatory bowel disease over 30 years”. In: *Gastroenterology* 143.2 (2012), pp. 375–381.

- [59] Juliet Bedford et al. “A new twenty-first century science for effective epidemic response”. In: *Nature* 575.7781 (2019), pp. 130–136.
- [60] Daniel T Lackland. “Racial differences in hypertension: implications for high blood pressure management”. In: *The American Journal of the Medical Sciences* 348.2 (2014), pp. 135–138.
- [61] Hao Guan and Mingxia Liu. “DomainATM: Domain adaptation toolbox for medical data analysis”. In: *Neuroimage* 268 (2023), p. 119863.
- [62] John R Zaleski. “Medical device interoperability and data integration to clinical information systems: medical device data alignment”. In: *Biomedical Instrumentation & Technology* 46.s2 (2012), pp. 65–70.
- [63] Sonali Narain et al. “Diagnostic accuracy for lupus and other systemic autoimmune diseases in the community setting”. In: *Archives of Internal Medicine* 164.22 (2004), pp. 2435–2441.
- [64] Aa Schrag, Yb Ben-Shlomo, and Niall Quinn. “How valid is the clinical diagnosis of Parkinson’s disease in the community?” In: *Journal of Neurology, Neurosurgery & Psychiatry* 73.5 (2002), pp. 529–534.
- [65] J Randall Curtis et al. “Use of the medical futility rationale in do-not-attempt-resuscitation orders”. In: *Journal of the American Medical Association (JAMA)* 273.2 (1995), pp. 124–128.
- [66] European Comission. *Rare Diseases*. 2021. URL: https://ec.europa.eu/info/research-and-innovation/research-area/health-research-and-innovation/rare-diseases_en.
- [67] Anton Pottegaard et al. “Arterial events, venous thromboembolism, thrombocytopenia, and bleeding after vaccination with Oxford-AstraZeneca ChAdOx1-S in Denmark and Norway: population based cohort study”. In: *British Medical Journal (BMJ)* 373 (2021).
- [68] Tadas Baltruvsaitis, Chaitanya Ahuja, and Louis-Philippe Morency. “Challenges and applications in multimodal machine learning”. In: *The Handbook of Multimodal-Multisensor Interfaces: Signal Processing, Architectures, and Detection of Emotion and Cognition-Volume 2*. 2018, pp. 17–48.
- [69] Mario Koppen. “The curse of dimensionality”. In: *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*. Vol. 1. 2000, pp. 4–8.
- [70] Suman Ravuri and Oriol Vinyals. “Classification accuracy score for conditional generative models”. In: *arXiv preprint arXiv:1905.10887* (2019).
- [71] Daniel Stahl and Andrew Pickles. “Fact or fiction: reducing the proportion and impact of false positives”. In: *Psychological Medicine* 48.7 (2018), pp. 1084–1091.
- [72] MP Petticrew et al. “False-negative results in screening programmes: systematic review of impact and implications.” In: *Health Technology Assessment (Winchester, England)* 4.5 (2000), pp. 1–120.
- [73] Corinna Cortes and Mehryar Mohri. “AUC optimization vs. error rate minimization”. In: *Advances in Neural Information Processing Systems* 16 (2003), pp. 313–320.

- [74] Solomon Kullback and Richard A Leibler. “On information and sufficiency”. In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86.
- [75] Tianqi Chen et al. “Xgboost: extreme gradient boosting”. In: *R package version 0.4-2* 1.4 (2015).
- [76] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 4768–4777.
- [77] Dominik Janzing, Lenon Minorics, and Patrick Blobaum. “Feature relevance quantification in explainable AI: A causal problem”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2907–2916.
- [78] Kristiina Häyrynen, Kaija Saranto, and Pirkko Nykänen. “Definition, structure, content, use and impacts of electronic health records: a review of the research literature”. In: *International Journal of Medical Informatics* 77.5 (2008), pp. 291–304.
- [79] Resham Dash et al. “Towards adult vaccination in India: a narrative literature review”. In: *Human Vaccines & Immunotherapeutics* 16.4 (2020), pp. 991–1001.
- [80] Stephen Trzeciak et al. “Significance of arterial hypotension after resuscitation from cardiac arrest”. In: *Critical Care Medicine* 37.11 (2009), pp. 2895–2903.
- [81] Joseph L Schafer and John W Graham. “Missing data: our view of the state of the art.” In: *Psychological Methods* 7.2 (2002), p. 147.
- [82] Nicholas J Horton, Stuart R Lipsitz, and Michael Parzen. “A potential for bias when rounding in multiple imputation”. In: *The American Statistician* 57.4 (2003), pp. 229–232.
- [83] Leo F Isikdogan et al. “Semifreddonets: Partially frozen neural networks for efficient computer vision systems”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 193–208.
- [84] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.
- [85] Jiquan Ngiam et al. “Multimodal deep learning”. In: *ICML*. 2011.
- [86] Wan Zhu et al. “The application of deep learning in cancer prognosis prediction”. In: *Cancers* 12.3 (2020), p. 603.
- [87] Srishti Grover et al. “Predicting severity of Parkinson’s disease using deep learning”. In: *Procedia Computer Science* 132 (2018), pp. 1788–1794.
- [88] Fabiana Tezza et al. “Predicting in-hospital mortality of patients with COVID-19 using machine learning techniques”. In: *Journal of Personalized Medicine* 11.5 (2021), p. 343.
- [89] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. “Activation functions in neural networks”. In: *Towards Data Science* 6.12 (2017), pp. 310–316.
- [90] Zhilu Zhang and Mert Sabuncu. “Generalized cross entropy loss for training deep neural networks with noisy labels”. In: *Advances in Neural Information Processing Systems* 31 (2018).

- [91] Peter L Bartlett and Marten H Wegkamp. “Classification with a Reject Option using a Hinge Loss.” In: *Journal of Machine Learning Research* 9.8 (2008).
- [92] Jonathon Byrd and Zachary Lipton. “What is the effect of importance weighting in deep learning?” In: *International Conference on Machine Learning*. PMLR, 2019, pp. 872–881.
- [93] Ali Sellami and Heasoo Hwang. “A robust deep convolutional neural network with batch-weighted loss for heartbeat classification”. In: *Expert Systems with Applications* 122 (2019), pp. 75–84.
- [94] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.
- [95] Henry J Kelley. “Gradient theory of optimal flight paths”. In: *Ars Journal* 30.10 (1960), pp. 947–954.
- [96] Arthur E Bryson. “A gradient method for optimizing multi-stage allocation processes”. In: *Proc. Harvard Univ. Symposium on Digital Computers and their Applications*. Vol. 72. 1961, p. 22.
- [97] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.
- [98] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).
- [99] Roberto Battiti. “First-and second-order methods for learning: between steepest descent and Newton’s method”. In: *Neural Computation* 4.2 (1992), pp. 141–166.
- [100] Marco Gori and Alberto Tesi. “On the problem of local minima in backpropagation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.1 (1992), pp. 76–86.
- [101] Tom Dietterich. “Overfitting and undercomputing in machine learning”. In: *ACM Computing Surveys (CSUR)* 27.3 (1995), pp. 326–327.
- [102] Tijmen Tieleman, Geoffrey Hinton, et al. “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”. In: *COURSERA: Neural Networks for Machine Learning* 4.2 (2012), pp. 26–31.
- [103] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [104] Yang You et al. “Large batch optimization for deep learning: Training bert in 76 minutes”. In: *arXiv preprint arXiv:1904.00962* (2019).
- [105] Michael Zhang et al. “Lookahead optimizer: k steps forward, 1 step back”. In: *Advances in Neural Information Processing Systems* 32 (2019).
- [106] Tomaso Poggio and Yaim Cooper. *Loss landscape: SGD can have a better view than GD*. 2020.

- [107] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive subgradient methods for online learning and stochastic optimization.” In: *Journal of Machine Learning Research* 12.7 (2011).
- [108] Jeffrey Dean et al. “Large scale distributed deep networks”. In: *Advances in Neural Information Processing Systems* 25 (2012).
- [109] Yang Zhao, Zoie Shui-Yee Wong, and Kwok Leung Tsui. “A framework of rebalancing imbalanced healthcare data for rare events’ classification: a case of look-alike sound-alike mix-up incident detection”. In: *Journal of Healthcare Engineering* 2018 (2018).
- [110] Matthew D Zeiler. “Adadelata: an adaptive learning rate method”. In: *arXiv preprint arXiv:1212.5701* (2012).
- [111] Yang You, Igor Gitman, and Boris Ginsburg. “Large batch training of convolutional networks”. In: *arXiv preprint arXiv:1708.03888* (2017).
- [112] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 448–456.
- [113] Boris Hanin. “Which neural net architectures give rise to exploding and vanishing gradients?” In: *Advances in Neural Information Processing Systems* 31 (2018).
- [114] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.
- [115] Lutz Prechelt. “Early stopping-but when?” In: *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.
- [116] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [117] Mike Schuster and Kuldip K Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.
- [118] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. “Learning long-term dependencies with gradient descent is difficult”. In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166.
- [119] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
- [120] Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. In: *arXiv preprint arXiv:1412.3555* (2014).
- [121] Paul J Werbos. “Backpropagation through time: what it does and how to do it”. In: *Proceedings of the IEEE* 78.10 (1990), pp. 1550–1560.
- [122] Ronald J Williams and Jing Peng. “An efficient gradient-based algorithm for on-line training of recurrent network trajectories”. In: *Neural Computation* 2.4 (1990), pp. 490–501.

- [123] Alexander Rehmer and Andreas Kroll. “On the vanishing and exploding gradient problem in Gated Recurrent Units”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 1243–1248.
- [124] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. “On the difficulty of training recurrent neural networks”. In: *International Conference on Machine Learning*. PMLR. 2013, pp. 1310–1318.
- [125] Jingzhao Zhang et al. “Why gradient clipping accelerates training: A theoretical justification for adaptivity”. In: *arXiv preprint arXiv:1905.11881* (2019).
- [126] Phong Le and Willem Zuidema. “Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs”. In: *arXiv preprint arXiv:1603.00423* (2016).
- [127] Yi Tay et al. “Efficient transformers: A survey”. In: *ACM Computing Surveys* 55.6 (2022), pp. 1–28.
- [128] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* 30 (2017).
- [129] Niki Parmar et al. “Image transformer”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4055–4064.
- [130] Haoyi Zhou et al. “Informer: Beyond efficient transformer for long sequence time-series forecasting”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 11106–11115.
- [131] Ronghang Hu and Amanpreet Singh. “Unit: Multimodal multitask learning with a unified transformer”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1439–1449.
- [132] Eduardo HP Pooch, Pedro L Ballester, and Rodrigo C Barros. “Can we trust deep learning models diagnosis? The impact of domain shift in chest radiograph classification”. In: *arXiv preprint arXiv:1909.01940* (2019).
- [133] Paul E Stang et al. “Health outcomes of interest in observational data: issues in identifying definitions in the literature”. In: *Health Outcomes Research in Medicine* 3.1 (2012), e37–e44.
- [134] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828.
- [135] Rob Donald et al. *Early warning of EUSIG-defined hypotensive events using a Bayesian Artificial Neural Network*. Springer, 2012.
- [136] Marzyeh Ghassemi et al. “Unfolding physiological state: Mortality modelling in intensive care units”. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2014, pp. 75–84.
- [137] Emily R Pfaff et al. “Clinical Annotation Research Kit (CLARK): computable phenotyping using machine learning”. In: *JMIR Medical Informatics* 8.1 (2020), e16042.

- [138] Mohcine Madkour, Driss Benhaddou, and Cui Tao. “Temporal data representation, normalization, extraction, and reasoning: A review from clinical domain”. In: *Computer Methods and Programs in Biomedicine* 128 (2016), pp. 52–68.
- [139] Na Hong et al. “Developing a scalable FHIR-based clinical data normalization pipeline for standardizing and integrating unstructured and structured electronic health record data”. In: *Journal of the American Medical Informatics Association Open* 2.4 (2019), pp. 570–579.
- [140] Yu Cheng et al. “Risk prediction with electronic health records: A deep learning approach”. In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM. 2016, pp. 432–440.
- [141] Lee Friedman and Oleg V Komogortsev. “Assessment of the effectiveness of seven biometric feature normalization techniques”. In: *IEEE Transactions on Information Forensics and Security* 14.10 (2019), pp. 2528–2536.
- [142] Wen Li et al. “Model generalizability investigation for GFCE-MRI synthesis in NPC radiotherapy using multi-institutional patient-based data normalization”. In: *IEEE Journal of Biomedical and Health Informatics* (2023).
- [143] Naomi J Banks. “Designing medical record abstraction forms”. In: *International Journal for Quality in Health Care* 10.2 (1998), pp. 163–167.
- [144] Hanqing Cao et al. “Predicting ICU hemodynamic instability using continuous multiparameter trends”. In: *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2008, pp. 3803–3806.
- [145] Cedric Seger. *An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing*. 2018.
- [146] George Hripcsak and David J Albers. “Correlating electronic health record concepts with healthcare process events”. In: *Journal of the American Medical Informatics Association* 20.e2 (2013), e311–e318.
- [147] Roger T Dean and William TM Dunsmuir. “Dangers and uses of cross-correlation in analyzing time series in perception, performance, movement, and neuroscience: The importance of constructing transfer function autoregressive models”. In: *Behavior Research Methods* 48 (2016), pp. 783–802.
- [148] Martin Arjovsky et al. “Invariant risk minimization”. In: *arXiv preprint arXiv:1907.02893* (2019).
- [149] Yana Dranker, He He, and Yonatan Belinkov. “IRM—when it works and when it doesn’t: A test case of natural language inference”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18212–18224.
- [150] Sardar Ansari et al. “A review of automated methods for detection of myocardial ischemia and infarction using electrocardiogram and electronic health records”. In: *IEEE Reviews in Biomedical Engineering* 10 (2017), pp. 264–298.
- [151] Varun Gupta et al. “ECG signal analysis using CWT, spectrogram and autoregressive technique”. In: *Iran Journal of Computer Science* 4.4 (2021), pp. 265–280.

- [152] M Sifuzzaman, M Rafiq Islam, and Mostafa Z Ali. “Application of wavelet transform and its advantages compared to Fourier transform”. In: (2009).
- [153] Akash Gupta, Tieming Liu, and Christopher Crick. “Utilizing time series data embedded in electronic health records to develop continuous mortality risk prediction models using hidden Markov models: a sepsis case study”. In: *Statistical Methods in Medical Research* 29.11 (2020), pp. 3409–3423.
- [154] H Lehman Li-wei et al. “Discovering shared dynamics in physiological signals: Application to patient monitoring in ICU”. In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2012, pp. 5939–5942.
- [155] Shima Ghassempour, Federico Girosi, and Anthony Maeder. “Clustering multivariate time series using hidden Markov models”. In: *International Journal of Environmental Research and Public Health* 11.3 (2014), pp. 2741–2763.
- [156] Jing Zhao et al. “Learning from heterogeneous temporal data in electronic health records”. In: *Journal of Biomedical Informatics* 65 (2017), pp. 105–119.
- [157] Spyros Tsevas and Dimitris K Iakovidis. “Dynamic time warping fusion for the retrieval of similar patient cases represented by multimodal time-series medical data”. In: *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine*. IEEE. 2010, pp. 1–4.
- [158] Lei Clifton et al. “Gaussian processes for personalized e-health monitoring with wearable sensors”. In: *IEEE Transactions on Biomedical Engineering* 60.1 (2012), pp. 193–197.
- [159] Matthias Seeger. “Gaussian processes for machine learning”. In: *International Journal of Neural Systems* 14.02 (2004), pp. 69–106.
- [160] Marco AF Pimentel, David A Clifton, and Lionel Tarassenko. “Gaussian process clustering for the functional characterisation of vital-sign trajectories”. In: *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2013, pp. 1–6.
- [161] Glen Wright Colopy, Stephen J Roberts, and David A Clifton. “Gaussian processes for personalized interpretable volatility metrics in the step-down ward”. In: *IEEE Journal of Biomedical and Health Informatics* 23.3 (2019), pp. 949–959.
- [162] Marzyeh Ghassemi et al. “A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1. 2015.
- [163] Robert Durichen et al. “Multitask Gaussian processes for multivariate physiological time-series analysis”. In: *IEEE Transactions on Biomedical Engineering* 62.1 (2014), pp. 314–322.
- [164] Joseph Futoma, Sanjay Hariharan, and Katherine Heller. “Learning to detect sepsis with a multitask Gaussian process RNN classifier”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1174–1182.

- [165] Farah Shamout, Tingting Zhu, and David A Clifton. “Machine learning for clinical outcome prediction”. In: *IEEE reviews in Biomedical Engineering* 14 (2020), pp. 116–126.
- [166] Peter Schulam, Fredrick Wigley, and Suchi Saria. “Clustering longitudinal clinical marker trajectories from electronic health data: Applications to phenotyping and endotype discovery”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1. 2015.
- [167] Lei Clifton et al. “Identification of patient deterioration in vital-sign data using one-class support vector machines”. In: *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE. 2011, pp. 125–131.
- [168] G Thippa Reddy et al. “Analysis of dimensionality reduction techniques on big data”. In: *IEEE Access* 8 (2020), pp. 54776–54788.
- [169] Carlos Oscar Sanchez Sorzano, Javier Vargas, and A Pascual Montano. “A survey of dimensionality reduction techniques”. In: *arXiv preprint arXiv:1403.2877* (2014).
- [170] Zhongheng Zhang and Adela Castelló. “Principal components analysis in clinical studies”. In: *Annals of translational medicine* 5.17 (2017).
- [171] Ahmad Taher Azar and Aboul Ella Hassanien. “Dimensionality reduction of medical big data using neural-fuzzy classifier”. In: *Soft Computing* 19 (2015), pp. 1115–1127.
- [172] Yoni Halpern et al. “A comparison of dimensionality reduction techniques for unstructured clinical text”. In: *ICML 2012 Workshop on Clinical Data Analysis*. Vol. 6. 2012.
- [173] Atiqur Rehman et al. “Performance analysis of pca, sparse pca, kernel pca and incremental pca algorithms for heart failure prediction”. In: *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. IEEE. 2020, pp. 1–5.
- [174] Xing Wei and Carsten Eickhoff. “Embedding electronic health records for clinical information retrieval”. In: *arXiv preprint arXiv:1811.05402* (2018).
- [175] Aleksei Dudchenko and Georgy Kopanitsa. “Comparison of word embeddings for extraction from medical records”. In: *International journal of environmental research and public health* 16.22 (2019), p. 4360.
- [176] Edward Choi et al. “Mime: Multilevel medical embedding of electronic health records for predictive healthcare”. In: *Advances in neural information processing systems* 31 (2018).
- [177] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444.
- [178] Margarita Cabrera-Bean et al. “Autoencoders for health improvement by compressing the set of patient features”. In: *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2020, pp. 5917–5920.

- [179] Najibesadat Sadati et al. “Representation learning with autoencoders for electronic health records: a comparative study”. In: *arXiv preprint arXiv:1801.02961* (2018).
- [180] Pascal Vincent et al. “Extracting and composing robust features with denoising autoencoders”. In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 1096–1103.
- [181] Joo-Chang Kim and Kyungyong Chung. “Multi-modal stacked denoising autoencoder for handling missing data in healthcare big data”. In: *IEEE Access* 8 (2020), pp. 104933–104943.
- [182] Hieu H Pham et al. “Interpreting chest X-rays via CNNs that exploit hierarchical disease dependencies and uncertainty labels”. In: *Neurocomputing* 437 (2021), pp. 186–194.
- [183] Byung-Hak Kim and Varun Ganapathi. “Read, Attend, and Code: Pushing the Limits of Medical Codes Prediction from Clinical Notes by Machines”. In: *arXiv preprint arXiv:2107.10650* (2021).
- [184] Julien Oster et al. “Identification of patients with atrial fibrillation: a big data exploratory analysis of the UK Biobank”. In: *Physiological Measurement* 41.2 (2020), p. 025001.
- [185] Denise Bandeira da Silva et al. “DeepSigns: A predictive model based on Deep Learning for the early detection of patient health deterioration”. In: *Expert Systems with Applications* 165 (2021), p. 113905.
- [186] Douglas P Wagner and Elizabeth A Draper. “Acute physiology and chronic health evaluation (APACHE II) and Medicare reimbursement”. In: *Health Care Financing Review* 1984.Suppl (1984), p. 91.
- [187] Inci M Baytas et al. “Patient subtyping via time-aware LSTM networks”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 65–74.
- [188] Stephanie L Hyland et al. “Early prediction of circulatory failure in the intensive care unit using machine learning”. In: *Nature Medicine* 26.3 (2020), pp. 364–373.
- [189] Roy Hirsch and Ran Gilad-Bachrach. “Trees with Attention for Set Prediction Tasks”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 4250–4261.
- [190] Andrew Cucchiara. *Applied logistic regression*. 1992.
- [191] Evangelia Christodoulou et al. “A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models”. In: *Journal of Clinical Epidemiology* 110 (2019), pp. 12–22.
- [192] Patrick Schober and Thomas R Vetter. “Logistic regression in medical research”. In: *Anesthesia & Analgesia* 132.2 (2021), pp. 365–366.
- [193] Matthew M Churpek, Trevor C Yuen, and Dana P Edelson. “Predicting clinical deterioration in the hospital: the impact of outcome selection”. In: *Resuscitation* 84.5 (2013), pp. 564–568.

- [194] Michael Sun et al. “Negative Patient Descriptors: Documenting Racial Bias In The Electronic Health Record: Study examines racial bias in the patient descriptors used in the electronic health record.” In: *Health Affairs* 41.2 (2022), pp. 203–211.
- [195] Ahmad Taher Azar and Shereen M El-Metwally. “Decision tree classifiers for automated medical diagnosis”. In: *Neural Computing and Applications* 23 (2013), pp. 2387–2403.
- [196] Mohammed Khalilia, Sounak Chakraborty, and Mihail Popescu. “Predicting disease risks from highly imbalanced data using random forest”. In: *BMC Medical Informatics and Decision Making* 11 (2011), pp. 1–13.
- [197] Tahani Daghistani and Riyad Alshammari. “Comparison of statistical logistic regression and random forest machine learning techniques in predicting diabetes”. In: *Journal of Advances in Information Technology Vol* 11.2 (2020), pp. 78–83.
- [198] Iain Robertson-Steel. “Evolution of triage systems”. In: *Emergency medicine journal* 23.2 (2006), pp. 154–155.
- [199] Stuart W Jarvis et al. “Development and validation of a decision tree early warning score based on routine laboratory test results for the discrimination of hospital mortality in emergency medical admissions”. In: *Resuscitation* 84.11 (2013), pp. 1494–1499.
- [200] Jialin Liu et al. “Predicting mortality of patients with acute kidney injury in the ICU using XGBoost model”. In: *Plos One* 16.2 (2021), e0246306.
- [201] Ivo D Dinov et al. “Predictive big data analytics: a study of Parkinson’s disease using large, complex, heterogeneous, incongruent, multi-source and incomplete observations”. In: *Plos One* 11.8 (2016), e0157077.
- [202] Fen Miao et al. “Continuous blood pressure measurement from one-channel electrocardiogram signal using deep-learning techniques”. In: *Artificial Intelligence in Medicine* 108 (2020), p. 101919.
- [203] Tian-Hao Li et al. “Electronic health records based reinforcement learning for treatment optimizing”. In: *Information Systems* (2021), p. 101878.
- [204] Rasheed El-Bouri et al. “Student-teacher curriculum learning via reinforcement learning: predicting hospital inpatient admission location”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 2848–2857.
- [205] Edward Choi et al. “Learning the graphical structure of electronic health records with graph convolutional transformer”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 01. 2020, pp. 606–613.
- [206] Zhuoning Yuan et al. “Robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification”. In: *arXiv preprint arXiv:2012.03173* (2020).
- [207] Minhao Liu et al. “Time Series is a Special Sequence: Forecasting with Sample Convolution and Interaction”. In: *arXiv preprint arXiv:2106.09305* (2021).

- [208] Aaron van den Oord et al. “Wavenet: A generative model for raw audio”. In: *arXiv preprint arXiv:1609.03499* (2016).
- [209] Haoyi Zhou et al. “Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting”. In: 35.12 (2021), pp. 11106–11115.
- [210] Omer Faruk Ertugrul. “Forecasting electricity load by a novel recurrent extreme learning machines approach”. In: *International Journal of Electrical Power & Energy Systems* 78 (2016), pp. 429–435.
- [211] Guokun Lai et al. “Modeling long-and short-term temporal patterns with deep neural networks”. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 2018, pp. 95–104.
- [212] Alec Radford et al. *Improving language understanding by generative pre-training*. 2018.
- [213] Ilija Radosavovic et al. “Humanoid Locomotion as Next Token Prediction”. In: *arXiv Preprint arXiv:2402.19469* (2024).
- [214] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).
- [215] Xiaohua Zhai et al. “Scaling vision transformers”. In: *arXiv preprint arXiv:2106.04560* (2021).
- [216] Haiping Wu et al. “Cvt: Introducing convolutions to vision transformers”. In: *arXiv preprint arXiv:2103.15808* (2021).
- [217] Zihang Dai et al. “CoAtNet: Marrying Convolution and Attention for All Data Sizes”. In: *arXiv preprint arXiv:2106.04803* (2021).
- [218] Jannik Kossen et al. “Self-attention between datapoints: Going beyond individual input-output pairs in deep learning”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 28742–28756.
- [219] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794.
- [220] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. “CatBoost: gradient boosting with categorical features support”. In: *arXiv preprint arXiv:1810.11363* (2018).
- [221] Bryan Lim et al. “Temporal fusion transformers for interpretable multi-horizon time series forecasting”. In: *International Journal of Forecasting* (2021).
- [222] Andrew Jaegle et al. “Perceiver: General perception with iterative attention”. In: *arXiv preprint arXiv:2103.03206* (2021).
- [223] Andrew Jaegle et al. “Perceiver IO: A General Architecture for Structured Inputs & Outputs”. In: *arXiv preprint arXiv:2107.14795* (2021).
- [224] Tri Dao et al. “FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness”. In: *arXiv preprint arXiv:2205.14135* (2022).

- [225] Sercan Ö Arik and Tomas Pfister. “Tabnet: Attentive interpretable tabular learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 8. 2021, pp. 6679–6687.
- [226] Juho Lee et al. “Set transformer: A framework for attention-based permutation-invariant neural networks”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3744–3753.
- [227] Shengpu Tang et al. “Democratizing EHR analyses with FIDDLE: a flexible data-driven preprocessing pipeline for structured clinical data”. In: *Journal of the American Medical Informatics Association* 27.12 (2020), pp. 1921–1934.
- [228] Jaime G. Carbonell George Philipp Dawn Song. *Gradients explode - Deep Networks are shallow - ResNet explained*. 2018. URL: <https://openreview.net/forum?id=HkpYwMZRb>.
- [229] Andrew J Pollard and Else M Bijker. “A guide to vaccinology: from basic principles to new developments”. en. In: *Nat. Rev. Immunol.* 21.2 (2021), pp. 83–100.
- [230] Merryn Voysey et al. “Safety and efficacy of the ChAdOx1 nCoV-19 vaccine (AZD1222) against SARS-CoV-2: an interim analysis of four randomised controlled trials in Brazil, South Africa, and the UK”. en. In: *Lancet* 397.10269 (2021), pp. 99–111.
- [231] Nina H Schultz et al. “Thrombosis and thrombocytopenia after ChAdOx1 nCoV-19 vaccination”. In: *New England Journal of Medicine* 384.22 (2021), pp. 2124–2130.
- [232] Andreas Greinacher et al. “Thrombotic Thrombocytopenia after ChAdOx1 nCov-19 Vaccination”. en. In: *N. Engl. J. Med.* 384.22 (2021), pp. 2092–2101.
- [233] Douglas B Cines and Andreas Greinacher. “Vaccine-induced immune thrombotic thrombocytopenia”. In: *Blood, The Journal of the American Society of Hematology* 141.14 (2023), pp. 1659–1665.
- [234] Michel Goldman and Cédric Hermans. “Thrombotic thrombocytopenia associated with COVID-19 infection or vaccination: Possible paths to platelet factor 4 autoimmunity”. en. In: *PLos Med.* 18.5 (2021), e1003648.
- [235] JB Bussel et al. “Thrombosis with thrombocytopenia syndrome (also termed vaccine-induced thrombotic thrombocytopenia)”. In: *American Society of Hematology* 25 (2021).
- [236] C R Simpson et al. “First-dose ChAdOx1 and BNT162b2 COVID-19 vaccines and thrombocytopenic, thromboembolic and hemorrhagic events in Scotland”. en. In: *Nat. Med.* 27.7 (2021), pp. 1290–1297.
- [237] *COVID-19 vaccine Market Dashboard*. 2021. URL: <https://www.unicef.org/supply/covid-19-vaccine-market-dashboard>.
- [238] Cyn-Young Park et al. “Getting Ready for the COVID-19 Vaccine Rollout”. In: (2021).

- [239] Andrew A S Soltan et al. “Rapid triage for COVID-19 using routine clinical data for patients attending hospital: development and prospective validation of an artificial intelligence screening test”. In: *The Lancet Digital Health* 3.2 (2021), pp. 78–87.
- [240] F A Klok et al. “Incidence of thrombotic complications in critically ill ICU patients with COVID-19”. en. In: *Thromb. Res.* 191 (2020), pp. 145–147.
- [241] Jean-François Llitjos et al. “High incidence of venous thromboembolic events in anticoagulated severe COVID-19 patients”. en. In: *J. Thromb. Haemost.* 18.7 (2020), pp. 1743–1746.
- [242] Emma Plested and Andrew Pollard. *A phase I/II study to determine efficacy, safety and immunogenicity of the candidate coronavirus disease (COVID-19) vaccine ChAdOx1 nCoV-19 in UK healthy adult volunteers.*
- [243] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. “A study of the behavior of several methods for balancing machine learning training data”. In: *ACM SIGKDD Explorations Newsletter* 6.1 (2004), pp. 20–29.
- [244] Simon de Lusignan et al. “Emergence of a novel coronavirus (COVID-19): protocol for extending surveillance used by the Royal College of General Practitioners Research and Surveillance Centre and Public Health England”. In: *Journal of Medical Internet Research (JMIR) Public Health and Surveillance* 6.2 (2020), e18606.
- [245] Ash K Clift et al. “Living risk prediction algorithm (QCOVID) for risk of hospital admission and mortality from coronavirus 19 in adults: national derivation and validation cohort study”. In: *British Medical Journal (BMJ)* 371 (2020).
- [246] Randal S Olson and Jason H Moore. “TPOT: A tree-based pipeline optimization tool for automating machine learning”. In: *Workshop on Automatic Machine Learning*. PMLR. 2016, pp. 66–74.
- [247] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of Statistics* (2001), pp. 1189–1232.
- [248] Hinrich Schutze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge, 2008.
- [249] Tom Fawcett. “An introduction to ROC analysis”. In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874.
- [250] Nitesh V Chawla et al. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357.
- [251] Hang Gong et al. “Insight into increased risk of portal vein thrombosis in nonalcoholic fatty liver disease”. In: *European Journal of Internal Medicine* (2023).
- [252] Ton Lisman et al. “Hemostasis and thrombosis in patients with liver disease: the ups and downs”. In: *Journal of Hepatology* 53.2 (2010), pp. 362–371.

- [253] Giuseppe Leone et al. “Blood cells diseases and thrombosis”. In: *Haematologica* 86.12 (2001), pp. 1236–1244.
- [254] Andrew J Pollard and Else M Bijker. “A guide to vaccinology: from basic principles to new developments”. In: *Nature Reviews Immunology* 21.2 (2021), pp. 83–100.
- [255] Richard S Snell. *Clinical Neuroanatomy*. Lippincott Williams & Wilkins, 2010.
- [256] Keith B Quattrocchi et al. “Quantification of midline shift as a predictor of poor outcome following head injury”. In: *Surgical Neurology* 35.3 (1991), pp. 183–188.
- [257] Ashraf Mohamed and Christos Davatzikos. “Finite element modeling of brain tumor mass-effect from 3D medical images”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005: 8th International Conference, Palm Springs, CA, USA, October 26-29, 2005, Proceedings, Part I* 8. Springer. 2005, pp. 400–408.
- [258] Yoichi Katayama and T Kawamata. “Edema fluid accumulation within necrotic brain tissue as a cause of the mass effect of cerebral contusion in head trauma patients”. In: *Brain Edema XII: Proceedings of the 12 International Symposium, Hakone, Japan, 2002*. Springer. 2003, pp. 323–327.
- [259] Charlene Jennifer Ong et al. “Enhanced detection of edema in malignant anterior circulation stroke (EDEMA) score: a risk prediction tool”. In: *Stroke* 48.7 (2017), pp. 1969–1972.
- [260] Allyson R Zazulia et al. “Progression of mass effect after intracerebral hemorrhage”. In: *Stroke* 30.6 (1999), pp. 1167–1173.
- [261] S Horstmann et al. “Sonographic monitoring of mass effect in stroke patients treated with hypothermia. Correlation with intracranial pressure and matrix metalloproteinase 2 and 9 expression”. In: *Journal of the Neurological Sciences* 276.1-2 (2009), pp. 75–78.
- [262] Connie W Tsao et al. “Heart disease and stroke statistics—2023 update: a report from the American Heart Association”. In: *Circulation* 147.8 (2023), e93–e621.
- [263] E. F. M. Wijdevicks, K. N. Sheth, B. S. Carter, et al. “Recommendations for the Management of Cerebral and Cerebellar Infarction With Swelling: A Statement for Healthcare Professionals From the American Heart Association/American Stroke Association”. In: *Stroke* 45 (2014), pp. 1222–1238.
- [264] H. H. Dasenbrock, F. C. Robertson, H. Vaitkevicius, et al. “Timing of Decompressive Hemicraniectomy for Stroke: A Nationwide Inpatient Sample Analysis”. In: *Stroke* 48 (2017), pp. 704–711.
- [265] Werner Hacke et al. “‘Malignant’ middle cerebral artery territory infarction: clinical course and prognostic signs”. In: *Archives of Neurology* 53.4 (1996), pp. 309–315.
- [266] H Bart van der Worp et al. “European Stroke Organisation (ESO) guidelines on the management of space-occupying brain infarction”. In: *European Stroke Journal* 6.2 (2021), pp. XC–CX.

- [267] Philip A Barber et al. “Computed tomographic parameters predicting fatal outcome in large middle cerebral artery infarction”. In: *Cerebrovascular Diseases* 16.3 (2003), pp. 230–235.
- [268] P. M. Pullicino, A. V. Alexandrov, J. A. Shelton, et al. “Mass effect and death from severe acute stroke”. In: *Neurology* 49 (1997), pp. 1090–1095.
- [269] Morgan E McKeown et al. “Midline shift greater than 3 mm independently predicts outcome after ischemic stroke”. In: *Neurocritical Care* (2022), pp. 1–6.
- [270] C. Waydhas. “Intrahospital transport of critically ill patients”. In: *Critical Care* 3 (1999), R83–R89.
- [271] C. J. Ong, J. Gluckstein, O. Laurido-Soto, et al. “Enhanced Detection of Edema in Malignant Anterior Circulation Stroke (EDEMA) Score: A Risk Prediction Tool”. In: *Stroke* 48 (2017), pp. 1969–1972.
- [272] Yajun Cheng et al. “External validation and modification of the EDEMA score for predicting malignant brain edema after acute ischemic stroke”. In: *Neurocritical Care* 32 (2020), pp. 104–112.
- [273] Takashi Shimoyama et al. “The DASH score: a simple score to assess risk for development of malignant middle cerebral artery infarction”. In: *Journal of the Neurological Sciences* 338.1-2 (2014), pp. 102–106.
- [274] Simiao Wu et al. “Predicting the emergence of malignant brain oedema in acute ischaemic stroke: a prospective multicentre study with development and validation of predictive modelling”. In: *Eclinicalmedicine* 59 (2023).
- [275] Martin Hanko et al. “Random forest-based prediction of outcome and mortality in patients with traumatic brain injury undergoing primary decompressive craniectomy”. In: *World Neurosurgery* 148 (2021), e450–e458.
- [276] Charlene Jennifer Ong et al. “Machine learning and natural language processing methods to identify ischemic stroke, acuity and location from radiology reports”. In: *Plos One* 15.6 (2020), e0234908.
- [277] Matthew I Miller et al. “Natural language processing of radiology reports to detect complications of ischemic stroke”. In: *Neurocritical Care* 37.Suppl 2 (2022), pp. 291–302.
- [278] Aaron M Cook et al. “Guidelines for the acute treatment of cerebral edema in neurocritical care patients”. In: *Neurocritical Care* 32 (2020), pp. 647–666.
- [279] Edward C Jauch et al. “Guidelines for the early management of patients with acute ischemic stroke: a guideline for healthcare professionals from the American Heart Association/American Stroke Association”. In: *Stroke* 44.3 (2013), pp. 870–947.
- [280] Simiao Wu and Craig S Anderson. “Precision management of brain oedema after acute ischaemic stroke”. In: *Precision Clinical Medicine* 5.3 (2022), pbac019.
- [281] Allan H Ropper. “Lateral displacement of the brain and level of consciousness in patients with an acute hemispherical mass”. In: *New England Journal of Medicine* 314.15 (1986), pp. 953–958.

- [282] Eelco FM Wijdicks and Michael N Diringer. “Middle cerebral artery territory infarction and early brain swelling: progression and effect of age on outcome”. In: *Mayo Clinic Proceedings*. Vol. 73. 9. Elsevier. 1998, pp. 829–836.
- [283] Antonio Muscari et al. “Predicting cerebral edema in ischemic stroke patients”. In: *Neurological Sciences* 40 (2019), pp. 745–752.
- [284] PM Pullicino et al. “Mass effect and death from severe acute stroke”. In: *Neurology* 49.4 (1997), pp. 1090–1095.
- [285] S Schwab et al. “The value of intracranial pressure monitoring in acute hemispheric stroke”. In: *Neurology* 47.2 (1996), pp. 393–398.
- [286] Philip A Barber et al. “Validity and reliability of a quantitative computed tomography score in predicting outcome of hyperacute stroke before thrombolytic therapy”. In: *The Lancet* 355.9216 (2000), pp. 1670–1674.
- [287] Rebecca Stafford et al. “Follow-up ASPECTS improves prediction of potentially lethal malignant edema in patients with large middle cerebral artery stroke”. In: *Journal of NeuroInterventional Surgery* (2023).
- [288] Mikhail Belkin et al. “Reconciling modern machine-learning practice and the classical bias–variance trade-off”. In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854.
- [289] Alethea Power et al. “Grokking: Generalization beyond overfitting on small algorithmic datasets”. In: *arXiv preprint arXiv:2201.02177* (2022).
- [290] Rylan Schaeffer et al. “Double Descent Demystified: Identifying, Interpreting & Ablating the Sources of a Deep Learning Puzzle”. In: *arXiv preprint arXiv:2303.14151* (2023).
- [291] Judea Pearl. *Causal inference in statistics: An overview*. 2009.
- [292] Chaochao Lu et al. “Nonlinear invariant risk minimization: A causal approach”. In: *arXiv preprint arXiv:2102.12353* (2021).
- [293] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. “The risks of invariant risk minimization”. In: *arXiv preprint arXiv:2010.05761* (2020).
- [294] Ruocheng Guo et al. “Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix”. In: *arXiv preprint arXiv:2101.07732* (2021).
- [295] Richard Bellman. “A Markovian decision process”. In: *Journal of Mathematics and Mechanics* (1957), pp. 679–684.
- [296] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. “Planning and acting in partially observable stochastic domains”. In: *Artificial Intelligence* 101.1-2 (1998), pp. 99–134.

Appendices

A

Appendices to Chapter 5

A.1 Sweep Configuration Tables

Table A.1: RNN sweep configuration

	Sweep 1	Sweep 2	Sweep 3
RNN configuration fixed parameters	- LR: 10^{-3}	- Dropout: 10%	- Dropout: 10%
	- Dropout: 10%	- Loss function: from sweep 1	- Loss function: from sweep 1
	- Depth: 8 layers	- Optimiser: from sweep 1	- Optimiser: from sweep 1
	- Feature embedding dimension: 32	- LR: 10^{-3}	- Dropout: from sweep 2
	- Batch size: 128	- Lookahead: not used	- Depth: from sweep 2
	- Lookahead: not used		- Feature embedding dimension: from sweep 2
	- Latent dimension 128		- Batch size: from sweep 2
RNN configuration varied parameters	- Loss function: cross-entropy, binary cross-entropy, batch-weighted cross-entropy, globally-weighted cross-entropy	- Depth: 4, 8, 12	- LR: 10^{-3} , 10^{-4} , 5×10^{-4}
	- Optimisers: ADAM, LAMB, RMSProp, Adagrad, Adadelat	- Feature embedding dimension: 32, 64	- Lookahead: on/off
		- Batch size: 128, 256	
		- Latent dimension: 64, 128, 256	

Table A.2: Transformer sweep configuration

	Sweep 1	Sweep 2	Sweep 3
Transformer configuration fixed parameters	- LR: 10^{-3}	- Dropout: 10%	- Dropout: 10%
	- Dropout: 10%	- Loss function: from sweep 1	- Loss function: from sweep 1
	- Depth: 8 layers	- Optimiser: from sweep 1	- Optimiser: from sweep 1
	- Feature embedding dimension: 32	- LR: 10^{-3}	- Dropout: from sweep 2
	- Batch size: 64	- Lookahead: not used	- Depth: from sweep 2
	- Lookahead: not used	- Score construction: mean score across output tokens	- Feature embedding dimension: from sweep 2
	- Latent dimension 128		- Batch size: from sweep 2
	- Attention numb heads: 8		- Attention numb heads: from sweep 2
	- Score construction: mean score across output tokens		
Transformer configuration varied parameters	- Loss function: cross-entropy, binary cross-entropy, batch-weighted cross-entropy, globally-weighted cross-entropy	- Depth: 8, 12	- LR: 10^{-3} , 10^{-4} , 5×10^{-4}
	- Optimisers: ADAM, LAMB, RMSProp, Adagrad, Adadelat	- Feature embedding dimension: 32,64	- Lookahead: on/off
		- Batch size: 64, 128	- Score construction: mean score across output tokens, token embedding
		- Latent dimension: 128, 256	
		- Attention numb heads: 8, 16	Note: NPTs cannot use token embeddings

Table A.3: Perceiver sweep configuration

	Sweep 1	Sweep 2	Sweep 3
Perceiver configuration fixed parameters	- LR: 10^{-3}	- Dropout: 10%	- Dropout: 10%
	- Dropout: 10%	- Loss function: from sweep 1	- Loss function: from sweep 1
	- Depth: 8 layers	- Optimiser: from sweep 1	- Optimiser: from sweep 1
	- Feature embedding dimension: 32	- LR: 10^{-3}	- Dropout: from sweep 2
	- Batch size: 64	- Lookahead: not used	- Depth: from sweep 2
	- Lookahead: not used	- Num cross heads 1	- Feature embedding dimension: from sweep 2
	- Latent dimension 128	- Num inducing points: 64	- Batch size: from sweep 2
	- Self Attention numb heads: 8		- Self Attention numb heads: from sweep 2
	- Num cross-heads: 1		- cross-head dimensions: from sweep 2
	- Cross-head dimensionality		- Num cross-heads: 1
	- Num inducing points: 64		- Num inducing points: 64
Perceiver configuration varied parameters	- Loss function: cross-entropy, binary cross-entropy, batch-weighted cross-entropy, globally-weighted cross-entropy	- Depth: 8, 12	- LR: 10^{-3} , 10^{-4} , 5×10^{-4}
	- Optimisers: ADAM, LAMB, RMSProp, Adagrad, Adadelat	- Feature embedding dimension: 32, 64	- Lookahead: on/off
		- Batch size: 64, 128	
		- Latent dimension: 128, 256	
		- Self Attention numb heads: 8, 16	
		- Cross-head dimensions: 64, 128	

Table A.4: RNN sweep 1 Best Results

Model name	Task	Optimiser type	Loss function	F1	AUROC
GRU	AF_4h	RMSProp	global-weighted cross-entropy	91.66%	94.29%
GRU	AF_12h	ADAM	global-weighted cross-entropy	83.18%	91.55%
GRU	Mortality_48h	Adagrad	cross-entropy	89.27%	95.68%
GRU	Shock_4h	Adagrad	global-weighted cross-entropy	84.10%	98.17%
GRU	Shock_12h	Adagrad	cross-entropy	84.10%	93.17%
LSTM	AF_4h	ADAM	cross-entropy	90.96%	94.49%
LSTM	AF_12h	Adagrad	global-weighted cross-entropy	81.84%	90.83%
LSTM	Mortality_48h	ADAM	global-weighted cross-entropy	86.47%	94.78%
LSTM	Shock_4h	ADAM	global-weighted cross-entropy	89.15%	93.69%
LSTM	Shock_12h	ADAM	global-weighted cross-entropy	86.10%	93.09%
RNN	AF_4h	Adagrad	global-weighted cross-entropy	91.00%	94.77%
RNN	AF_12h	Adagrad	global-weighted cross-entropy	84.49%	91.21%
RNN	Mortality_48h	Adagrad	global-weighted cross-entropy	91.04%	94.30%
RNN	Shock_4h	Adagrad	cross-entropy	88.24%	93.65%
RNN	Shock_12h	Adagrad	batch-weighted cross-entropy	87.35%	92.25%

Table A.5: RNN sweep 2 Best Results

Model name	Task	Feature embedding dimension	Latent dimension	Depth	Batch size	F1	AUROC
GRU	AF_4h	64	256	4	128	92.01%	94.77%
GRU	AF_12h	32	128	8	128	86.08%	93.66%
GRU	Mortality_48h	64	128	8	256	90.86%	95.25%
GRU	Shock_4h	32	128	8	128	85.85%	93.55%
GRU	Shock_12h	64	128	8	128	86.73%	94.45%
LSTM	AF_4h	32	128	8	256	90.85%	94.19%
LSTM	AF_12h	64	128	4	128	86.75%	92.38%
LSTM	Mortality_48h	32	256	4	256	88.99%	95.50%
LSTM	Shock_4h	32	128	8	256	91.24%	94.21%
LSTM	Shock_12h	32	256	4	128	80.95%	93.04%
RNN	AF_4h	64	256	4	128	90.32%	94.64%
RNN	AF_12h	32	256	4	128	87.26%	93.11%
RNN	Mortality_48h	32	128	4	128	92.79%	96.09%
RNN	Shock_4h	64	128	4	128	88.21%	93.87%
RNN	Shock_12h	64	128	8	256	86.61%	93.18%

Table A.6: RNN sweep 3 Best Results

Model name	Task	Learning rate	Lookahead parameters	F1	AUROC
GRU	AF_4h	10^{-3}	None	87.75%	93.92%
GRU	AF_12h	10^{-3}	$k5_a0.5$	81.12%	91.02%
GRU	Mortality_48h	10^{-4}	$k5_a0.5$	90.58%	93.44%
GRU	Shock_4h	10^{-3}	None	86.09%	92.26%
GRU	Shock_12h	5×10^{-3}	None	85.38%	93.42%
LSTM	AF_4h	10^{-3}	None	92.06%	94.24%
LSTM	AF_12h	10^{-3}	None	88.76%	92.79%
LSTM	Mortality_48h	10^{-4}	$k5_a0.5$	84.80%	94.99%
LSTM	Shock_4h	10^{-4}	None	86.66%	93.54%
LSTM	Shock_12h	10^{-4}	None	74.64%	93.36%
RNN	AF_4h	5×10^{-3}	$k5_a0.5$	90.37%	94.46%
RNN	AF_12h	5×10^{-3}	None	87.87%	92.44%
RNN	Mortality_48h	10^{-3}	$k5_a0.5$	90.88%	94.92%
RNN	Shock_4h	5×10^{-3}	$k5_a0.5$	88.94%	93.03%
RNN	Shock_12h	5×10^{-3}	$k5_a0.5$	76.98%	91.87%

Table A.7: Transformer sweep 1 Best Results

Model name	Task	Optimiser type	Loss function	F1	AUROC
NPT	AF_4h	LAMB	global-weighted cross-entropy	91.45%	94.23%
NPT	AF_12h	LAMB	global-weighted cross-entropy	80.93%	91.87%
NPT	Mortality_48h	LAMB	cross-entropy	83.53%	95.20%
NPT	Shock_4h	LAMB	batch-weighted cross-entropy	86.19%	94.03%
NPT	Shock_12h	Adagrad	batch-weighted cross-entropy	88.65%	93.54%
Transformer	AF_4h	Adagrad	global-weighted cross-entropy	90.05%	93.87%
Transformer	AF_12h	Adagrad	cross-entropy	84.45%	92.42%
Transformer	Mortality_48h	Adagrad	global-weighted cross-entropy	91.18%	95.56%
Transformer	Shock_4h	Adagrad	global-weighted cross-entropy	83.24%	93.56%
Transformer	Shock_12h	Adagrad	batch-weighted cross-entropy	84.27%	92.48%

Table A.8: Transformer sweep 2 Best Results

Model name	Task	# of attention heads	Dim. feature embedding	Latent dim.	Depth	Batch size	F1	AUROC
NPT	AF_4h	8	64	128	8	128	90.06%	94.70%
NPT	AF_12h	8	64	128	8	128	85.22%	93.66%
NPT	Mortality_48h	8	32	256	8	128	90.02%	95.46%
NPT	Shock_4h	16	32	128	12	64	89.45%	94.74%
NPT	Shock_12h	16	32	256	8	64	87.89%	92.93%
Transformer	AF_4h	16	64	128	12	128	90.02%	94.61%
Transformer	AF_12h	16	64	256	8	64	84.12%	93.62%
Transformer	Mortality_48h	16	32	256	8	128	91.79%	96.28%
Transformer	Shock_4h	8	64	128	12	64	86.82%	94.30%
Transformer	Shock_12h	16	64	256	8	64	87.18%	93.26%

Table A.9: Transformer sweep 3 Best Results

Model name	Task	Learning rate	Lookahead parameters	F1	AUROC
NPT	AF_4h	5×10^{-3}	None	90.37%	94.01%
NPT	AF_12h	10^{-3}	$k5_a0.5$	89.14%	94.83%
NPT	Mortality_48h	10^{-3}	$k5_a0.5$	88.83%	93.90%
NPT	Shock_4h	10^{-3}	$k5_a0.5$	86.60%	93.17%
NPT	Shock_12h	10^{-4}	$k5_a0.5$	84.54%	92.85%
Transformer	AF_4h	5×10^{-3}	$k5_a0.5$	89.44%	93.39%
Transformer	AF_12h	10^{-3}	None	82.55%	91.82%
Transformer	Mortality_48h	10^{-3}	None	91.43%	94.87%
Transformer	Shock_4h	5×10^{-3}	$k5_a0.5$	87.18%	93.07%
Transformer	Shock_12h	10^{-3}	None	85.24%	92.53%

Table A.10: Perceiver sweep 1 Best Results

Model name	Task	Optimiser type	Loss function	F1	AUROC
NPP	AF_4h	LAMB	batch-weighted cross-entropy	87.52%	93.97%
NPP	AF_12h	LAMB	cross-entropy	80.44%	92.79%
NPP	Mortality_48h	Adagrad	cross-entropy	92.30%	94.57%
NPP	Shock_4h	Adagrad	batch-weighted cross-entropy	89.29%	93.56%
NPP	Shock_12h	LAMB	batch-weighted cross-entropy	78.96%	92.37%
Time Perceiver	AF_4h	Adagrad	BCE	88.28%	93.64%
Time Perceiver	AF_12h	Adagrad	BCE	86.43%	92.84%
Time Perceiver	Mortality_48h	LAMB	BCE	88.85%	95.33%
Time Perceiver	Shock_4h	Adagrad	global-weighted cross-entropy	89.92%	94.39%
Time Perceiver	Shock_12h	Adagrad	cross-entropy	87.97%	92.79%
Hierarchical Perceiver	AF_4h	LAMB	batch-weighted cross-entropy	54.85%	75.02%
Hierarchical Perceiver	AF_12h	LAMB	batch-weighted cross-entropy	26.23%	68.90%
Hierarchical Perceiver	Mortality_48h	n/a	n/a	n/a	n/a
Hierarchical Perceiver	Shock_4h	LAMB	global-weighted cross-entropy	45.45%	75.96%
Hierarchical Perceiver	Shock_12h	LAMB	batch-weighted cross-entropy	23.32%	70.94%

Table A.11: Perceiver sweep 2 Best Results

Model name	Task	Perceiver cross-head dim.	Attention num heads	Feature embedding dim.	Latent dim.	Depth	Batch size	F1	AUROC
NPP	AF_4h	128	8	32	256	12	64	90.01%	94.16%
NPP	AF_12h	128	16	64	128	8	64	86.12%	93.26%
NPP	Mortality_48h	128	8	64	128	12	64	90.71%	94.06%
NPP	Shock_4h	128	16	64	128	8	128	90.51%	94.65%
NPP	Shock_12h	128	16	64	256	12	64	80.93%	93.36%
Time-Perceiver	AF_4h	64	16	64	256	8	128	91.52%	94.60%
Time-Perceiver	AF_12h	64	16	32	256	12	128	86.86%	93.55%
Time-Perceiver	Mortality_48h	64	16	64	128	8	64	93.25%	96.57%
Time-Perceiver	Shock_4h	64	16	64	256	12	64	87.95%	94.14%
Time-Perceiver	Shock_12h	64	8	64	128	12	128	88.10%	93.92%

Table A.12: Perceiver sweep 3 Best Results

Model name	Task	Model learning rate	Lookahead parameters	F1	AUROC
NPP	AF_4h	10^{-3}	$k5_a0.5$	88.95%	94.80%
NPP	AF_12h	10^{-3}	None	85.85%	92.06%
NPP	Mortality_48h	10^{-4}	None	89.49%	93.86%
NPP	Shock_4h	10^{-3}	$k5_a0.5$	89.72%	93.69%
NPP	Shock_12h	10^{-4}	$k5_a0.5$	78.75%	91.30%
Time-Perceiver	AF_4h	10^{-3}	None	89.69%	92.56%
Time-Perceiver	AF_12h	10^{-3}	None	85.84%	91.45%
Time-Perceiver	Mortality_48h	10^{-3}	$k5_a0.5$	88.78%	91.80%
Time-Perceiver	Shock_4h	10^{-3}	None	86.66%	92.22%
Time-Perceiver	Shock_12h	10^{-3}	None	87.34%	93.10%

B

Appendices to Chapter 7

B.1 Ensemble Tree Model Hyperparameters Optimised Over

- gamma:
 - sample distribution: uniform
 - max: 5
 - min: 0
- learning rate:
 - sample distribution: uniform
 - max: 0.5
 - min: 0.001
- max depth:
 - sample distribution: uniform
 - max: 10
 - min: 2
- minimum child weight weight:
 - sample distribution: uniform
 - max: 10

- min: 1
- alpha regularisation:
 - sample distribution: uniform
 - max: 5
 - min: 0
- lambda regularisation:
 - sample distribution: uniform
 - max: 10
 - min: 1
- subsample rate:
 - sample distribution: uniform
 - max: 1
 - min: 0.5

B.2 Deep Learning Hyperparameters Optimised Over

- **parameters:**
 - batch size:
 - * values:
 - 128
 - 256
 - depth:
 - * values:
 - 4
 - 8
 - 12
 - feature embedding dimension:
 - * values:
 - 32
 - 64
 - dropout probability:

- * values:
 - 0
 - 0.1
- latent dimension:
 - * values:
 - 128
 - 256
- loss function:
 - * values:
 - BCE
 - cross-entropy
- model learning rate:
 - * values:
 - 10^{-3}
 - 10^{-4}
 - 5×10^{-3}
- optimiser type:
 - * values:
 - ADAM
 - RMSProp
 - Adadelata
 - Adagrad
 - LAMB
 - ADAMW
- weight decay:
 - * values:
 - 0
 - 0.01
 - 0.1
 - 0.2

B.3 Model Features and Feature Construction

A breakdown of each of the features used in our machine learning models is reported. Table B.1 shows the mean values of time-dependent features following their transforms into rolling maximum or forward fill forms. Table B.2 shows the mean values of static features, including static that are time-censored (i.e. are hidden in the data until the event time and then are static in the data). All date-time features are represented as number of hours after admission (e.g. if the first MLS occurred 6 hours after admission, the value of *first_mlsdt_time_censored* would be 6).

Feature	Feature Type	Alias	BMC mean $\pm$ S.D.	MGB mean $\pm$ S.D.
Midline shift size	Continuous	size_mls	3.8112 ± 4.4172	2.9879 ± 3.8124
Pineal gland shift size	Continuous	size_pgs	1.9529 ± 2.3111	1.6468 ± 2.1884
coverage of the anterior cerebral artery (ACA) by stroke	Binary	aca	0.3664 ± 0.4818	0.2874 ± 0.4526
ASPECTS score	Continuous	aspects	3.417 ± 2.5715	3.4346 ± 2.6519
Basilar cistern effacement	Binary	bce	0.2963 ± 0.464	0.0389 ± 0.1933
Degree of Basal Cistern Effacement	Continuous	bce_severity	0.4681 ± 0.7718	0.0067 ± 0.1683
Blood Urea Nitrogen	Continuous	bun	23.233 ± 14.6538	23.7472 ± 16.8707
Creatinine	Continuous	cr	1.0611 ± 0.6018	1.0424 ± 0.7946
Diastolic blood pressure	Continuous	dbp	72.1072 ± 13.3297	71.0505 ± 12.626
If external ventricular drain is evident	Binary	evd	0.0156 ± 0.124	0.028 ± 0.165
Glucose	Continuous	gluc	165.3567 ± 52.5127	161.0816 ± 57.9846
petechial hemorrhage	Binary	hi1	0.0463 ± 0.2105	0.1275 ± 0.3334
petechial hemorrhages starting to gain mild confluence	Binary	hi2	0.0802 ± 0.2721	0.1681 ± 0.374
Hypertonic saline (23%)	Binary	hts23	did not occur	0.3297 ± 0.4701
Hypertonic saline (3%)	Binary	hts3	0.4723 ± 0.4992	0.1842 ± 0.3877
Imaging type	Categorical	imagetype	0.2963 ± 0.6333	0.4768 ± 0.7712
Mannitol	Continuous	mannitol	0.4772 ± 0.4995	0.557 ± 0.4967
Mean arterial pressure	Continuous	map	93.9988 ± 15.0368	92.4127 ± 14.7712
Sodium	Continuous	na	147.0303 ± 8.7282	144.6849 ± 8.5481
Serum osmolality	Continuous	osm	327.6512 ± 19.9785	322.2339 ± 21.7871
Administration of mannitol, hts3, or hts23	Binary	osmotics	0.6003 ± 0.4899	0.6171 ± 0.4861
Pulse	Continuous	pulse	83.6885 ± 18.5009	80.4795 ± 16.8513
Systolic blood pressure	Continuous	sbp	138.0261 ± 22.9839	135.3937 ± 23.5491
Size of the mid-cerebral artery (MCA) territory under stroke	Continuous	stroke_territory	2.4568 ± 0.9317	2.3964 ± 0.9533
Temperature	Continuous	temp	98.9129 ± 1.1815	98.9387 ± 1.2374
White Blood Cell Count	Continuous	wbc	12.228 ± 4.2489	12.0175 ± 7.237

Table B.1: Mean values ($\pm$ standard deviation) of time-varying features

Feature	Feature Type	Alias	BMC mean $\pm$ S.D.	MGB mean $\pm$ S.D.
National Institutes of Health Stroke Scale (NIHSS) on admission to hospital	Ordinal	nihss	18.7226 $\pm$ 6.114	16.7543 $\pm$ 6.0088
Presentation date & time	Continuous	pres	5.2363 $\pm$ 7.42	7.5265 $\pm$ 5.4922
History of atrial fibrillation	Binary	af	0.3003 $\pm$ 0.4585	0.4033 $\pm$ 0.4906
Body weight on admission (lbs)	Continuous	lbs	177.8138 $\pm$ 46.6747	175.914 $\pm$ 53.2195
History of hypertension	Binary	htn	0.6512 $\pm$ 0.4766	0.7125 $\pm$ 0.4526
Age (calculated on admission date)	Continuous	age_calc	63.9856 $\pm$ 16.103	64.1809 $\pm$ 13.5243
History of prior stroke	Binary	stroke	0.1021 $\pm$ 0.3028	0.1027 $\pm$ 0.3035
Thrombolysis in cerebral infarction (TICI)	Binary	tici	3.232 $\pm$ 1.3996	5.5098 $\pm$ 52.0394
Sex	Binary	sex	0.3983 $\pm$ 0.4896	0.6004 $\pm$ 0.4898
evidence of Anterior Cerebral Artery involvement in first radiography image	Binary	aca1	0.1546 $\pm$ 0.3616	2.3942 $\pm$ 48.2844
Blood urea nitrogen on admission	Continuous	bun1	16.9123 $\pm$ 7.7445	18.5472 $\pm$ 9.4841
Creatinine on admission	Continuous	cr1	1.0433 $\pm$ 0.4751	1.0628 $\pm$ 0.6324
Diastolic blood pressure at presentation	Continuous	dbp1	87.5807 $\pm$ 15.9635	80.1392 $\pm$ 15.7575
Glucose at presentation	Continuous	gluc1	157.1976 $\pm$ 77.8959	153.8541 $\pm$ 63.4576
Mean arterial pressure at presentation	Continuous	map1	111.0976 $\pm$ 19.3267	103.4581 $\pm$ 18.3841
Sodium on admission	Continuous	na1	137.9785 $\pm$ 3.0643	138.2562 $\pm$ 3.4167
Serum osmolality on admission	Continuous	osm1	306.6681 $\pm$ 14.1912	298.904 $\pm$ 10.7841
Pulse at presentation	Continuous	pulse1	84.7395 $\pm$ 18.087	79.9876 $\pm$ 21.0066
Systolic blood pressure at presentation	Continuous	sbp1	158.1299 $\pm$ 31.1483	149.6654 $\pm$ 28.859
Temperature at presentation	Continuous	temp1	97.3672 $\pm$ 1.2084	97.7409 $\pm$ 0.9896
White blood cell count on admission	Continuous	wbc1	10.5066 $\pm$ 3.5966	13.3382 $\pm$ 19.822
ASPECTs	Continuous	aspects1_time_censored	6.1265 $\pm$ 2.5842	5.1748 $\pm$ 2.7705
Time of radiographic report with first ASPECTS	Continuous	aspects1dt_time_censored	4.9773 $\pm$ 6.2683	7.2649 $\pm$ 5.5398
Time of tissue plasminogen activator (tPA) (if provided)	Continuous	tpadt_time_censored	3.9425 $\pm$ 11.9287	1.9985 $\pm$ 2.3399
Time of surgery (if occurred)	Continuous	surgt1_time_censored	36.0015 $\pm$ 26.1778	46.0255 $\pm$ 31.9329
First idline shift (MLS) value (if occurred)	Continuous	firstmls_time_censored	2.1473 $\pm$ 0.6763	3.5616 $\pm$ 2.4449
Time of first MLS (if occurred)	Continuous	firstmlsdt_time_censored	4.9773 $\pm$ 6.2683	26.1793 $\pm$ 21.3581
Time of first MLS exceeding 3mm (if occurred)	Continuous	mls3dt_time_censored	25.5607 $\pm$ 23.3029	32.2917 $\pm$ 21.0328
Time of first MLS exceeding 5mm (if occurred)	Continuous	mls5dt_time_censored	34.0614 $\pm$ 29.7701	41.5045 $\pm$ 24.2471
Time of first MLS exceeding 7mm (if occurred)	Continuous	mls7dt_time_censored	39.7188 $\pm$ 32.8855	50.1223 $\pm$ 29.3181
Time of first MLS exceeding 9mm (if occurred)	Continuous	mls9dt_time_censored	39.3872 $\pm$ 33.7902	66.4489 $\pm$ 39.1997
Time of first MLS exceeding 12mm (if occurred)	Continuous	mls12dt_time_censored	31.7068 $\pm$ 35.0105	60.7079 $\pm$ 44.5566
Time of first MLS exceeding 15mm (if occurred)	Continuous	mls15dt_time_censored	25.6747 $\pm$ 13.4274	47.9448 $\pm$ 30.6654
Time of Mechanical thrombectomy (if occurred)	Continuous	mtdt_time_censored	7.0687 $\pm$ 7.8096	5.7721 $\pm$ 7.1574
Time of first PGS exceeding 2mm (if occurred)	Continuous	pgs2dt_time_censored	28.7553 $\pm$ 24.6373	33.3508 $\pm$ 24.7709
Time of first PGS exceeding 4mm (if occurred)	Continuous	pgs4dt_time_censored	42.0975 $\pm$ 34.1785	44.686 $\pm$ 26.9842
Time of first PGS exceeding 6mm (if occurred)	Continuous	pgs6dt_time_censored	39.9537 $\pm$ 29.3839	70.041 $\pm$ 42.4269
Time of first PGS exceeding 8mm (if occurred)	Continuous	pgs8dt_time_censored	24.3457 $\pm$ 16.3489	75.2339 $\pm$ 48.6498
Time of first PGS exceeding 10mm (if occurred)	Continuous	pgs10dt_time_censored	did not occur	60.9231 $\pm$ 40.8691

Table B.2: Mean values ($\pm$ standard deviation) of static and time-censored features