UNIVERSITY OF
LIVERPOOL

Deep Learning Prognostic Models using Longitudinal Imaging Data with Applications to Age-Related Macular Degeneration

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by Joshua Thomas Bridge.

October 2022

Contents

Acknowledgements

I would first like to thank my supervisors, Professor Yalin Zheng and Professor Simon Harding, for their support and expertise. I have learned a lot from both of them throughout my PhD. My advisory panel, Prof Stephen Kaye and Dr Phil Burgess, also gave me excellent guidance and encouragement throughout my work. My group helped me throughout my PhD and made my time at UoL enjoyable. I would also like to thank my examiners for helping me to improve my thesis. I would also like to thank the patients who permitted their data to be used, allowing my work to be more than theory. Finally, I would like to thank my parents, grandma, and fiancé Jacqueline, for their ongoing support and encouragement.

Abstract

Prognostic models in the context of health are a class of clinical prediction model which aim to predict the future outcome of a disease or condition. These models are essential in planning treatment and the allocation of resources. Prognostic models can ensure that treatment is delivered only when needed. Many such models have been developed using traditional statistical methods, such as logistic regression and proportional hazards. Some are routinely used in clinic. Traditional methods rely on the relevant variables being known and easy to extract; variables or features are often difficult or even impossible to extract, especially when imaging is used. Deep learning methods can automatically extract relevant features from the image. These methods have been used extensively on classification problems, detecting diseases from imaging data; however, they are less common for prognostic modelling, especially when using longitudinal data.

In this thesis, I explore how deep learning can be used to develop prognostic models to predict the future course of disease using longitudinal data. After reviewing the previous methods and discussing their limitations, I develop novel methods which aim to be more accurate and clinically useful than previous methods. Throughout the thesis, I demonstrate the novel methods using colour fundus images of patients with age-related macular degeneration. I evaluated my models using current best practices for clinical prediction models. In real-world settings, the time interval between visits is unlikely to be the same each time; therefore, I present a method to account for uneven intervals between visits. I show results for one, two, and three time points to assess the added utility of additional time points and conclude that a single time point is sufficient in this situation. Finally, I develop deep survival models and present a method that accounts for both uneven time intervals and missing visits through a novel mixed-effects layer. I also show how clinical data can be incorporated into the model, although this does not significantly improve the performance. Unfortunately, all my developed models show poor calibration and require adjustment before being deployed in a clinical setting. This highlights the importance of assessing and revising the calibration of clinical prediction models. The methods presented in this thesis may be used in developing prognostic algorithms helping to deliver personalised healthcare.

List of abbreviations

AI	Artificial Intelligence
AMD	Age-related Macular Degeneration
APGAR	Appearance Pulse Grimace Activity Respiration (Backronym)
AREDS	Age-related Eye Disease Study
AUROC	Area Under the Receiver Operating Characteristic
BMI	Body Mass Index
BMJ	British Medical Journal
CAM	Class Activation Map
CI	Confidence Interval
CNN	Convolutional Neural Network
COVID-19	Coronavirus Disease 2019
CT	Computed Tomography
CV	Computer Vision
DHA	Docosahexaenoic Acid
DL	Deep Learning
DNN	Deep Neural Network
ELU	Exponential Linear Unit
EPA	Eicosapentaenoic acid
EPSRC	Engineering and Physical Sciences Research Council
FN	False Negatives
FP	False Positives
GA	Geographic Atrophy
GAN	Generative Adversarial Network
GiB	Gibibytes
GEV	Generalised Extreme Value
GPU	Graphics Processing Unit
GRACE	Global Registry of Acute Coronary Events
GRU	Gated Recurrent Unit
IAA	Impact Acceleration Award
IEEE	Institute of Electrical and Electronics Engineers
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IP	Intellectual property

IU	International Units
LSTM	Long Short-Term Memory
LUMPO	Liverpool Uveal Melanoma Prognosticator Online
MB	Megabytes
MIUA	Medical Imaging Understanding and Analysis
MNIST	Modified National Institute of Standards and Technology database
MSE	Mean Squared Error
NA	Not Applicable
NICE	National Institute of Care Excellence
NPV	Negative Predictive Value
OCT	Optical Coherence Tomography
PCR	Polymerase Chain Reaction
PPV	Positive Predictive Value
PROGRESS	PROGnosis RESearch Strategy
RCNN	Recurrent Convolutional Neural Network
ReLU	Rectified Linear Unit
REST API	Representational State Transfer Application Programming Interface
RGB	Red Green Blue
RMSProp	Root Mean Squared Propagation
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RT-PCR	Real-Time Polymerase Chain Reaction
SD	Standard Deviation
SD-OCT	Spectral Domain Optical Coherence Tomography
SE	Standard Error
SGD	Stochastic Gradient Descent
TN	True Negatives
TP	True Positives
US	United States
VAR	Vector Autoregression
VEGF	Vascular Endothelial Growth Factor
VGG	Visual Geometry Group

Chapter 1: Introduction

1.1 Background

Prognostic models are clinical prediction models that predict the future outcome of a disease or condition based on a patient's current or past state or condition.[1] Examples of prognostic models currently in clinical use include the APGAR score,[2] the Framingham risk score,[3] and the Liverpool uveal melanoma prognosticator online (LUMPO)[4]. These models provide a useful tool in planning treatment and the allocation of resources.

Imaging data is routinely collected in clinics to monitor the progression of various diseases. Images can contain large amounts of information which may be useful in determining a patient's current or possible future state. Prognostic models utilising these routinely collected images may aid clinicians in making informed decisions about patient treatment and care.

There is a growing interest in developing and applying prognostic models due to the increasing amount of data available to clinicians. A combination of ever-growing numbers of people with chronic disease [5] and diagnostic tests [6] provides clinicians with an overwhelming wealth of information. The number of people on waiting lists for diagnostic tests has also been steadily increasing for the past 15 years.[6] Ophthalmology, in particular, is one area with increasingly large amounts of high-resolution imaging data.

Prognostic models are attractive to clinicians as they can utilise this abundance of data to predict the outcome of disease or response to treatment. These models can potentially reduce strain on clinicians by helping with the efficient allocation of resources. Knowing when a patient is likely to progress reduces the need for frequent diagnostic tests. For example, if clinicians can predict when a patient progresses from early/intermediate to advanced age-related macular degeneration (AMD), treatment could be planned and administered appropriately to prevent further sight loss.

However, the variables or features contained in images are often highly variable and difficult or time-consuming to extract; this makes quantifying variables for use in traditional statistical models challenging. Deep learning can automatically extract useful features from images without requiring time-consuming and challenging manual intervention; this makes it particularly suited to imaging data. As a result, prognostic models developed using deep learning may improve patient care while reducing strain on clinicians.

Although many well-known models are commonly used in clinical settings,[7] many more fail to be adopted. There are several reasons that a deep learning model may fail to be adopted. Firstly, many developed models lack robust reporting and validation. Secondly, complex models are difficult to interpret and understand, leading to distrust by patients and clinicians. A study by Longoni *et al*.[8] found that patients are less likely to trust AI due to “uniqueness neglect”, believing the AI is less able to understand the unique case of each individual patient. Patients who perceive themselves as more unique are more likely to distrust AI. However, patients are more likely to trust the AI if it is presented as personalised or only used to support decision-making. Looking at previous technological revolutions, the internet drove another major shift in patient healthcare. It has been observed that patients became more trusting of the internet as a source of medical information, although clinicians remained the most trusted source.[9] As AI becomes more prominent in everyday life, we may see a shift towards acceptance of medical AI in a similar way to the acceptance of the internet.

To overcome distrust in AI, best practice guidelines for developing and reporting prediction models should be followed.[10] Throughout this thesis, I aim to follow these guidelines to assess the true usefulness of the models. Models also need to be better explained to reduce their black-box nature. Overcoming these issues will allow many more complex models to be adopted in clinical settings.

1.2 Motivation

One of the biggest sources of valuable data captured in a clinical setting is imaging data. Images contain a wealth of information about the current state of a patient; however, extracting that information can often be difficult and time-consuming. While demographic and clinical data such as age, blood glucose and blood pressure are often easy to measure and represent with a numerical value, imaging features such as area, volume, and pigment changes can be much more difficult for clinicians to quantify. Furthermore, the relevant features of a disease in an image may even be unknown. Prognostic models that can automatically extract useful information from images and give a probability of progression could benefit clinicians looking to utilise the vast amounts of imaging data available.

Traditional statistical prognostic methods rely on us being able to accurately quantify these imaging features, which may not always be possible. Deep learning allows us to extract relevant features from models automatically. For example, several risk factors for progression to advanced AMD are commonly reported; however, the strength of association is quite varied.[11] For example, AMD was previously associated erroneously with optic disc pallor[12]. Image characteristics such as subtle colour changes are often difficult to detect and quantify.

1.3 Aims and objectives

In this thesis, I aim to present the development of novel methods to predict the future progression of disease using longitudinal images. Any useful model must overcome a few challenges to make it clinically practical. Imaging data poses a significant challenge in traditional statistics as the variables must be extracted from the image before being fed into a model. This can be time-consuming or even impossible when the important variables of the image are difficult to quantify or even unknown. Deep learning can automatically extract these variables or features.

Therefore, the central aim is to develop a deep learning methodology to create models which predict progression of a disease or condition at any time point, using longitudinal images while accounting for uneven time intervals, missing images, and right-censored data.

1.4 Contributions

The main contributions of my work presented in this thesis are:

- I have developed a novel interval scaling method, allowing for uneven time intervals between patient visits. This allows for more clinical utility and means that one model can be used no matter the visit or screening interval.
- I have implemented mixed-effects in deep learning to account for the relationship between images. This provides an alternative to simply

concatenating features when multiple images are used and allows for better modelling of the random effects. In addition, missing images can be accounted for using this model.

- I have developed a joint mixed-effects and survival model to create prognostic models for deep learning. This combines deep learning feature extraction by a CNN with a traditional statistical model, allowing us to make inferences about the underlying distribution, such as how the risk of progression increases with time.

1.5 Publications

The following published papers have resulted from the work presented here:

- **Bridge, J.,** Harding, S., & Zheng, Y. (2020). Development and validation of a novel prognostic model for predicting AMD progression using longitudinal fundus images. *BMJ Open Ophthalmology*, *5*(1), e000569.
- **Bridge, J.,** Harding, S., & Zheng, Y. (2021, July). End-to-end deep learning vector autoregressive prognostic models to predict disease progression with uneven time intervals. In *Annual Conference on Medical Image Understanding and Analysis* (pp. 517-531). Springer, Cham.
- **Bridge, J.,** Meng, Y., Zhao, Y., Du, Y., Zhao, M., Sun, R., & Zheng, Y. (2020). Introducing the GEV activation function for highly unbalanced data to develop COVID-19 diagnostic models. *IEEE Journal of Biomedical and Health Informatics*, *24*(10), 2776-2786.
- **Bridge, J.,** Meng, Y., Zhu, W., Fitzmaurice, T., McCann, C., Addison, C., Wang, M., Merritt, C,. Franks, S,. Mackey, M,. Sun, R,. Zhao, Y,. & Zheng, Y. (2022). Development and external validation of a mixed-effects deep learning model to diagnose COVID-19 from CT imaging. *medRxiv*.

1.6 Thesis structure

The thesis is set out as follows: In Chapter 2, I present a literature review of prognostic modelling and deep learning, focusing on AMD. In Chapter 3, I describe the datasets I used in the thesis to demonstrate the developed methodologies. My first models are presented in Chapter 4, with a novel interval scaling mechanism used to achieve one of the main objectives. In Chapter 5, I introduce a novel method to account for missing data. I present my final model accounting for missing and right-censored data in Chapter 6 and apply the model to the AREDS data. Finally, in Chapter 7, I review and discuss the thesis and present my conclusions.

Chapter 2: Literature review

In this chapter, I present a brief overview of the field when writing this thesis. I begin with an introduction to clinical prediction models highlighting the differences between diagnostic and prognostic model research. Next, I outline the measures used to assess the performance of clinical prediction models. I then outline computer vision and how artificial intelligence interprets and analyses imaging data. Then, I describe and discuss deep learning and its applications to image classification tasks, focusing on convolutional neural networks (CNNs). Section 2.5 gives a brief overview of Age-related Macular Degeneration (AMD), which is used to demonstrate the methods that I have developed in Chapters 4 and 6. Finally, I briefly review and critically appraise the key literature for prognostic models, mainly focusing on deep learning and AMD prognosis.

2.1 Clinical prediction models

Clinical prediction models fall into two main categories, diagnostic and prognostic.[10] While there is considerable overlap in the theory and methods underpinning these two classes of model, they differ significantly in their aims. Diagnostic models aim to predict the current status or condition of a patient. Prognostic models aim to predict what the future status or condition is likely to be in the future. This thesis focuses on prognostic modelling[13]; however, a similar methodology can sometimes be applied to both problems. I have also developed a diagnostic model to demonstrate how one of my novel methods works before using it in a more complex prognostic model.

Examples of diagnostic models include predicting whether a patient has a particular disease or not or predicting the current stage of the disease. These models are quite common, both in traditional statistics and machine learning. These models are often binary, where only two classes are possible, but they may also be multiclass, where we aim to diagnose the patient from a list of possible diseases.

Examples of prognostic models include predicting whether a patient will progress to another stage of disease in the future or even whether a patient will die from a disease. However, these models are more challenging to develop and validate and are less common than diagnostic models, especially when using deep learning.

There are two broad types of models that I have considered in this thesis: classification types and survival types. Classification types give a binary outcome of future disease, progression, or no progression. Survival models are more complicated and give a probability of progression by a specified time point.

2.2 Performance measures

When developing a clinical prediction model, three criteria would make a model clinically useful:

1. The model should be able to discriminate between diseases. In prognostic models, the model should determine which patients will progress to future stages of the disease.
2. The model should not systematically over- or under-estimate risk, which would lead to harmful predictions. For example, a model which underestimates risk will lead to patients being denied necessary treatment or not being told that it will progress. Conversely, a model which overestimates risk will lead to unnecessary treatment, increasing stress on the patient and overwhelming clinical services. Both of these should be minimised.
3. The model should be clinically useful. A clinical prediction model that provides no clinical benefit is simply an academic exercise and may be detrimental to patients' mental health if they are informed of future outcomes without benefit. While the benefit to a patient is not always clear and must usually be assessed from a clinical perspective, the clinical benefit compared to other models or no model can be assessed numerically in several ways.

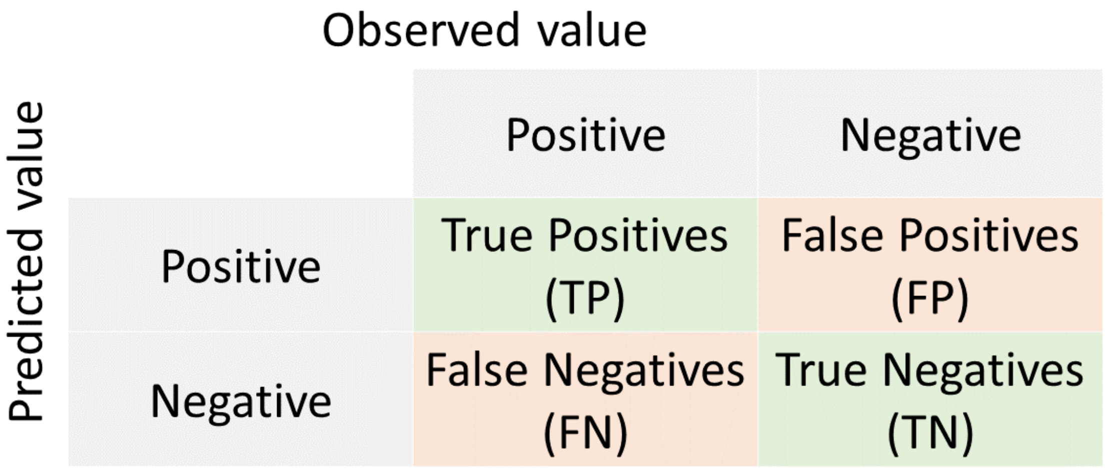
These criteria can be tested by assessing the model's 1. discrimination, 2. calibration, and 3. clinical usefulness.

2.2.1 Binary prediction

Clinical prediction models are often binary, meaning they only predict two outcomes. For example, a model may be developed to diagnose a disease. The outcomes are disease or no disease, and the model returns a probability of having the disease.

2.2.1.1 Discrimination

Discrimination assesses how well the model discriminates between disease and no disease and is the most commonly reported type of performance measure, and these measures are often easy to interpret. When comparing the predicted and observed outcomes, a confusion matrix can be constructed showing the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Whether the patient is predicted as positive or negative depends upon the predicted probability, with a cut-off of 0.5 commonly used. True positives are when the algorithm correctly predicts a positive; true negatives occur when the algorithm correctly predicts negative; false positives are when the algorithm predicts positive, but the outcome is negative, and false negatives are when the algorithm predicts negative, but the outcome is positive. I show a confusion matrix for the binary case in *Figure 2.1*, but this can easily be extended to the multiclass case.



		Observed value	
Predicted value		Positive	Negative
	Positive	True Positives (TP)	False Positives (FP)
	Negative	False Negatives (FN)	True Negatives (TN)

Figure 2.1: Confusion matrix for the binary classification case.

Popular discrimination measures include accuracy, sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), and area under the receiver operating characteristic curve (AUROC).

False positives can also be called a type I error and occur when overestimating risk. On the other hand, false negatives are type II errors when underestimating the risk.

The simplest and most intuitive performance measure is accuracy, defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.1}$$

In simple terms, it is the percentage of predictions that the model got correct. While the accuracy is very simple both in terms of calculation and explainability, it is not suitable when classes are imbalanced. For example, if a disease has a prevalence of only 1%, then a model can easily attain an accuracy of 99% by always predicting no disease. In the real world, data is highly likely to be unbalanced.

Sensitivity is the proportion of positives predicted as positive, while specificity is the proportion of negatives predicted as negative. The sensitivity is given as

$$Sens = \frac{TP}{TP + FN}, \tag{2.2}$$

and the specificity is

$$Spec = \frac{TN}{TN + FP}. \tag{2.3}$$

Models with higher specificities have lower type I error rates, and models with higher sensitivities have lower type II error rates.

Sensitivity and specificity are often opposing, and a balance is needed between them. A higher sensitivity may be needed in certain settings, while a higher specificity may be preferred in others. The sensitivity and specificity can be altered by changing the cut-off point; as the cut-off point increases, the sensitivity decreases with an increase in specificity. The receiver operating characteristic (ROC) curve plots the sensitivity and 1-specificity at different cut-off points. The area under the ROC (AUROC) can be used as an overall measure of sensitivity and specificity at different cut-off points. An example ROC curve is presented in Figure 2.2.

The AUROC can be difficult to interpret and may not be the most useful measure for assessing the real-world performance. The AUROC gives the performance over all thresholds; however, a final threshold must be chosen and used in practice. There are situations where sensitivity or specificity may be preferred, and the model with the best AUROC may not necessarily be the best overall model. Instead, giving the sensitivity and specificity at a range of thresholds is useful to show how the model performs in real-world values. Nonetheless, the AUROC can provide useful information about the model.

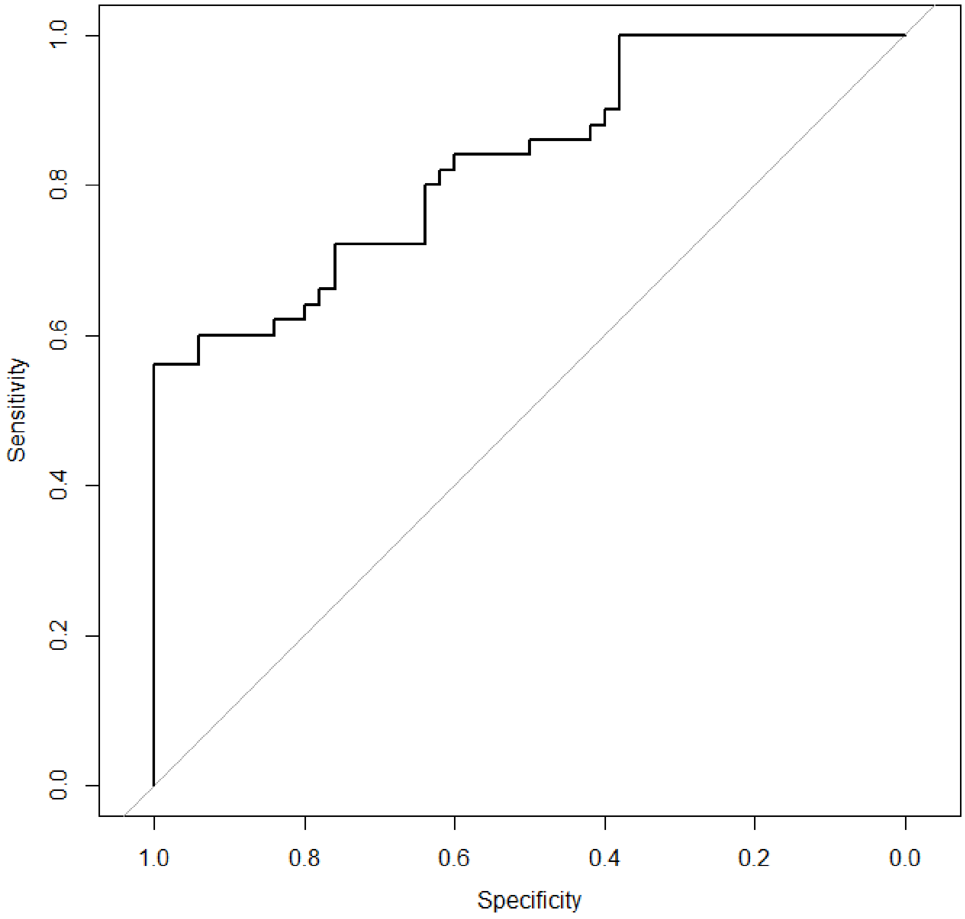


Figure 2.2: Example of an ROC curve. Confidence bands can be added.

It is also important to consider the disease prevalence when considering the model performance. The PPV and NPV are linked to the sensitivity and specificity but take the disease prevalence into account. The PPV is the number of true positive cases divided by the total number of predicted positives

$$PPV = \frac{TP}{TP + FP} = \frac{Sens \times Prevalence}{Sens \times Prevalence + (1 - Spec) \times (1 - Prevalence)}, \qquad (2.4)$$

Similarly, the NPV is the number of true negatives divided by the total number of predicted negatives

$$NPV = \frac{TN}{TN + FN} = \frac{Spec \times (1 - Prevalence)}{Spec \times (1 - Prevalence) + (1 - Sens) \times Prevalence}. \qquad (2.5)$$

Assessing discrimination is essential in predictive modelling; however, good discrimination does not guarantee that models provide reliable or useful predictions. Even models with excellent discrimination can provide unreliable and unsafe predictions with no clinical benefit.

2.2.1.2 Calibration

Calibration assesses how well the expected and observed outcomes agree. Calibration, described as "the Achilles heel of predictive analytics", is often overlooked in machine learning but is vital to assessing a clinical prediction model’s safety.[14] If the risks are systematically over-or under-estimated, clinicians and patients may make incorrect decisions. Models can have excellent discriminative performance but poor calibration performance, leading to harmful predictions with high confidence.

Model calibration can be split into four levels: mean calibration, weak calibration, moderate calibration, and strong calibration. Each level becomes more stringent than the last, with strong calibration described as utopic and impossible to achieve in practice.[15]

Mean calibration is satisfied if the average predicted risk equals the observed event rate, also known as the calibration-in-the-large. Logistic regression can be used to assess calibration-in-the-large. However, as the lowest level of calibration, it is not enough on its own to determine if the model predictions are over- or under-estimated.

Weak calibration is attained when there is no over- or under-estimation of the risks. As with mean calibration, logistic calibration can be used to assess weak calibration. If the calibration intercept is zero and the calibration slope is 1, the predicted risks are neither under- nor over-estimated. Confidence intervals may be constructed to reject the claim of weak calibration.

The model is moderately calibrated if the predicted risks correspond to the observed event rates. Moderate calibration can be assessed using flexible calibration curves; this often detects miscalibration that may be missed by the logistic framework used for mean and weak calibration. The calibration curve plots the predicted probability against the observed proportion and can show if the model over or underestimates the risk. A well-calibrated model will have a calibration curve which follows the diagonal. Deviation from calibration can be observed across all probabilities; this means we can assess whether a model is calibrated around a particular threshold of interest. An example of a calibration curve is shown in Figure 2.3. This level of calibration is what any clinical prediction model should aim for.

Strong calibration is similar to moderate calibration in that it requires the predicted risks to correspond to the observed risks; however, it adds the additional requirement that this must be true for every covariate pattern (pattern observed in the matrix of the covariances between variables). Strong calibration requires the true model to be known, making this calibration level impossible.

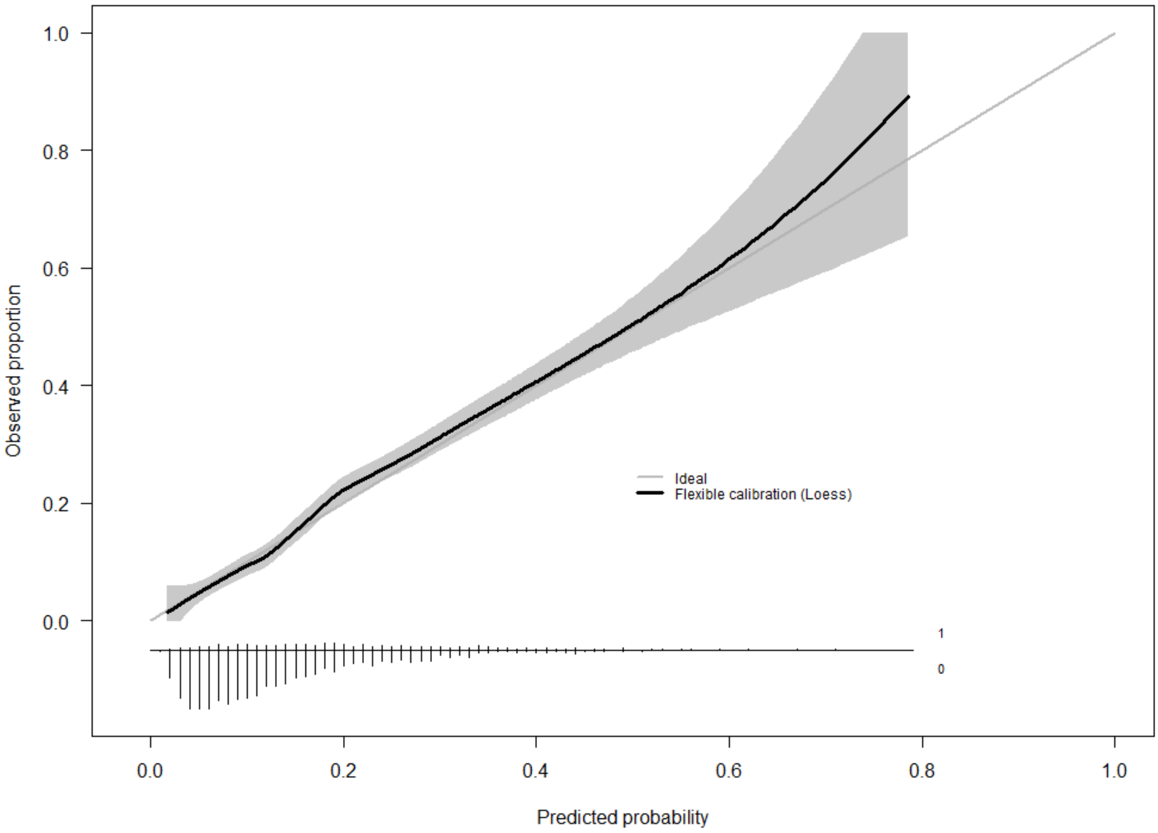


Figure 2.3: An example of a calibration curve.

2.2.1.3 Clinical usefulness

Models that are no better than either current models or treat all/none approaches provide no benefit to clinicians or patients. Models intended for deployment in clinical settings must be able to justify their use. Often the consequences of a false negative may be more severe than a false positive and vice versa; this can depend upon the setting, the disease being studied, or even the amount of funding currently available and demand for the service. The costs and harms must be quantified to assess the optimal threshold at which treatment should be given; however, this can often be difficult to quantify and differ between services and settings. Decision curve analysis assesses net clinical benefit at a range of thresholds.[16] Models can be plotted against each other and treat all/none approaches to assess the threshold at which each method reaches zero net clinical benefit. A cost-benefit ratio can also be added to the graph. An example of a decision curve is shown in Figure 2.4.

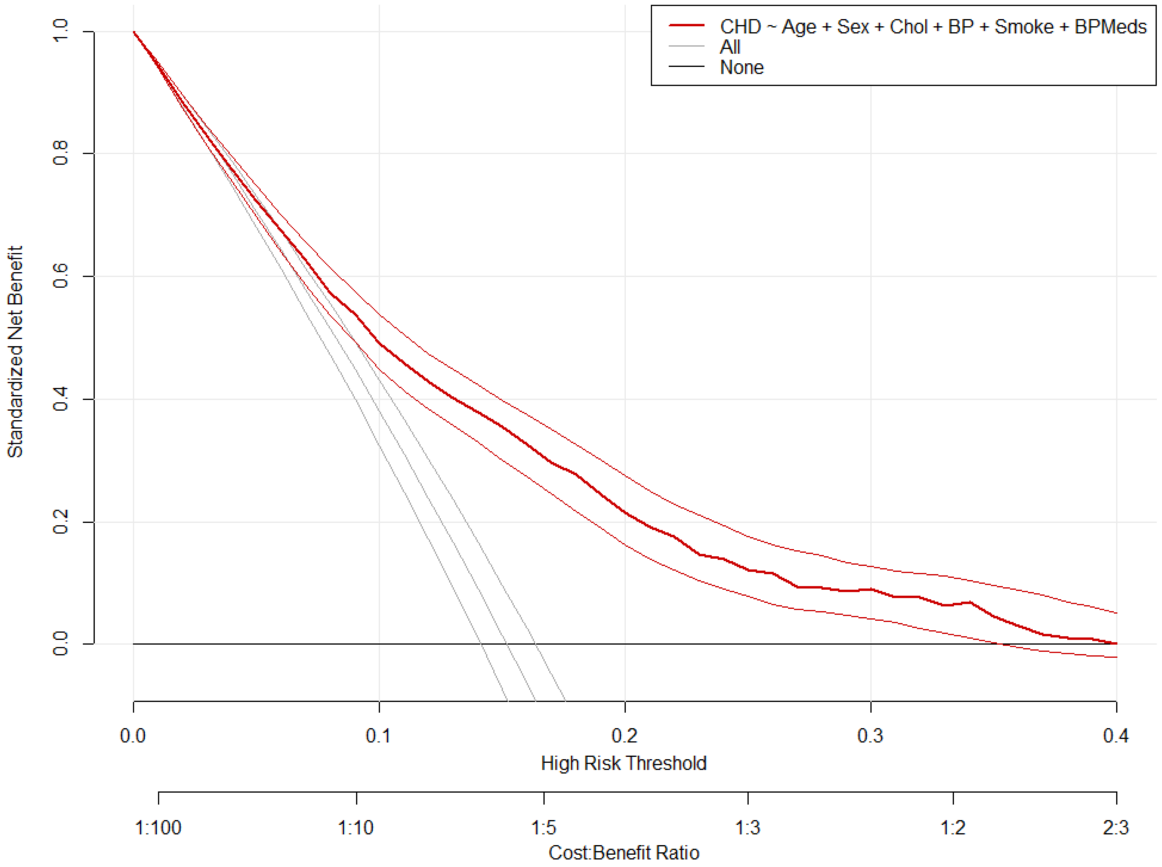


Figure 2.4: An example of a decision curve. This model shows improved net benefit over the treat-all approach.

2.2.2 Survival model performance

Classification performance measures and survival performance measures differ slightly. Performance measures for survival models need to account for censoring. Censoring occurs when the actual event is not observed. There are three main types of censoring: right, left, and interval. The most common type of censoring is right censoring, where the patient leaves the study before the event is observed. There are many reasons for right censoring, including the patient leaving the study, dying, or the study ending before the event is observed. Another type of censoring is left censoring when the event has already occurred before the patient is enrolled in the study. Finally, interval censoring occurs when the event occurs between two time points, but it is unclear exactly when. In this thesis, I only consider right censoring as it may be of the most use in my work.

Performance can be measured across all time points or time-dependent and measured at chosen time points. Measuring at specific time points shows how the performance degrades over time. Although performance can theoretically be measured for any future time point, it should ideally be assessed at clinically relevant time points.

2.2.2.1 Discrimination

For the assessment of discrimination in survival models, overall performance can be measured using Harrell's concordance index (C-index). The right-censored C-index generalises the AUROC to censored data. It is easier to define the C-index in terms of Somers' D_{XY} rank correlation.[17]

Given bivariate random variables (X_1, Y_1) and (X_2, Y_2) sampled independently from the same distribution, then Kendall's tau is given by

$$\tau(X, Y) = E[sign(X_1 - X_2)sign(Y_1 - Y_2)]. \tag{2.6}$$

This can be extended to account for censoring by introducing censoring indicators R_i and S_i, for X_i and Y_i, respectively.[17] These indicators are positive for right-censored data, negeative for left-censored data, and zero for no censorship. Then, the censored sign difference is given by

$$csign(a, b, c, d) = \begin{cases} 1, & \text{if } a > c \text{ and } b \geq 0 \geq d, \\ -1, & \text{if } a < c \text{ and } b \leq 0 \leq d, \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

The concordance-discordance difference between the two observations (X_i, R_i, Y_i, S_i) and (X_j, R_j, Y_j, S_j) is the product of $csign(X_i, R_i, X_j, R_j)$ and $csign(Y_i, S_i, Y_j, S_j)$. We can then redefine Kendall's tau from above for censored data

$$\tau(X,Y) = E\left[csign(X_i, R_i, X_j, R_j)csign(Y_i, S_i, Y_j, S_j)\right]. \tag{2.8}$$

Then Somers' D_{XY} is defined as

$$D_{XY} = \frac{\tau(X,Y)}{\tau(X,X)}. \tag{2.9}$$

The C-index is then given as

$$\text{C-index} = (D_{XY} + 1)/2. \tag{2.10}$$

interpretation of the C-index is similar to the AUROC, with 1 indicating perfect concordance, 0 indicating perfect discordance, and 0.5 indicating random concordance. The Hmisc package in R contains the rcorr.cens function to calculate both the D_{XY} and C-index with standard error so that the confidence intervals can be constructed. The rcorrp.cens function can compute the U-statistics for testing whether one predictor is more concordant than another.

Assessing the model's discrimination performance across all time points gives a valuable overview of how well the model performs; however, performance is likely to change depending on how far into the future the model is used to predict. Therefore, it is useful to measure performance at specific times. Time-dependent measures measure the model performance at specific time points. For example, the performance at one, two, and three years could be assessed. A time-dependent ROC curve for each time point of interest can be constructed, accounting for right-censored data. A method proposed by Beyene and Ghouch uses imputation to estimate the actual unobserved survival time for censored data and a kernel function to smooth the ROC.[18] Compared to the previous methods for estimating time-dependent ROC curves for censored data, their method improved results on a simulated dataset. The simulated data also showed that bootstrapping with a sample size of 2000 gives a good approximation to the sampling distribution of the estimated AUROC. Bootstrapping involves sampling with replacement from the results to obtain a sample the same size as the original sample. The performance is then calculated. This is repeated many times (often 2000 times), and a distribution of 2000 model performance measures is obtained. This distribution can then be used to obtain confidence intervals of the performance measures. The advantage of bootstrapping over other confidence interval construction methods is that symmetry of the distribution is not assumed. I have used their implementation in the cenROC package in parts of this thesis.

2.2.2.2 Calibration

Calibration curves can be extended to survival models to assess the model calibration. Similar to the ROC curves, calibration curves for survival models are time-dependent.[19] For binary outcomes, the calibration measures the agreement between the observed and estimated probabilities of an outcome. For time-to-event outcomes, calibration measures the agreement between the observed and estimated probabilities of an event occurring within a particular time. Calibration could be assessed by stratifying samples into risk categories and fitting a Kaplan-Meier curve for each stratum. This stratified approach is not ideal as the risk categories are often arbitrary.[19]

Smoothed calibration curves can be constructed using Cox-Snell residuals by comparing the residuals on the cumulative probability scale against the right-censored survival time data.[20]

2.2.2.3 Clinical usefulness

Decision curves have been extended to censored data.[21] Calculating the net benefit becomes problematic when the actual survival time is unknown. Kaplan-Meier survival probabilities can estimate the number of true and false positives to estimate the net benefit. This may lead to non-monotonic relationships between the predicted probabilities and the model sensitivity and specificity;[22] however, this does not pose a problem for the decision curves.

2.3 Images

The research I present in this thesis focuses on using imaging data to predict the outcome of disease. Here, I will briefly outline how computers store and analyse imaging data.

2.3.1 Computer vision

Computer vision (CV) explores how computers can be used to process and analyse imaging data and aims to mimic human vision. Examples of computer vision tasks include object classification, object detection, and motion analysis. CV is a wide and varied field, often involving multidisciplinary groups to tackle problems as diverse as disease detection, self-driving vehicles, and facial recognition. Deep learning is often used in CV as it provides methods for the automated analysis of images.

2.3.2 Digital images

Digital images are often stored as matrices of pixel values; these matrices are known as channels. For example, greyscale image pixels take values between 0 and 255, with 0 being black, 255 being white, and values in between being different shades of grey. Colour images are formed by combining different matrices. One of the most common methods to store colour images is using red, green, and blue channels, known as the RGB colour model. This trichromatic system is based upon the three types of cones found in the retina (short, medium, and long)[23 24]. Combining these three channels, each with 256 possible pixel values, we can store over 16 million different colours. An example of an image broken down into its red, green, and blue channels can be seen in Figure 2.5.

2.3.3 Image augmentation

When using imaging data for deep learning, we augment the data using various transformations during training. These augmentations make the algorithm more robust to unseen images and reduce overfitting, although data augmentation is not as good as collecting additional data. Typical image augmentations include adjusting image brightness, rotating the image, zooms, flips, and mirrors. Examples of these augmentations are shown in Figure 2.6. Images are also often resized, and pixel values are normalised between 0 and 1 before being used in a deep learning model.

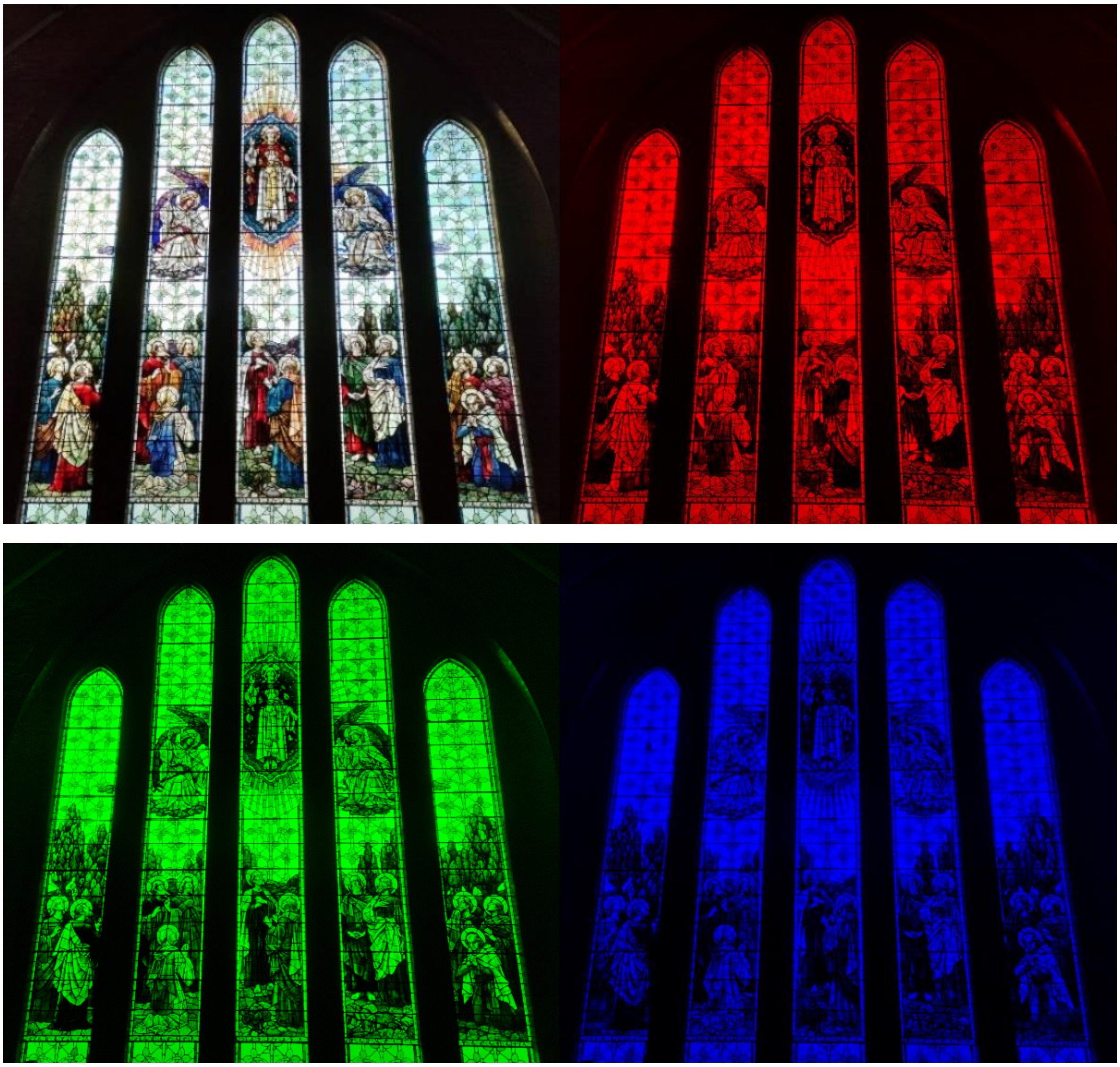


Figure 2.5: Example image showing the original image (top left) and the image broken down into red, green, and blue channels.

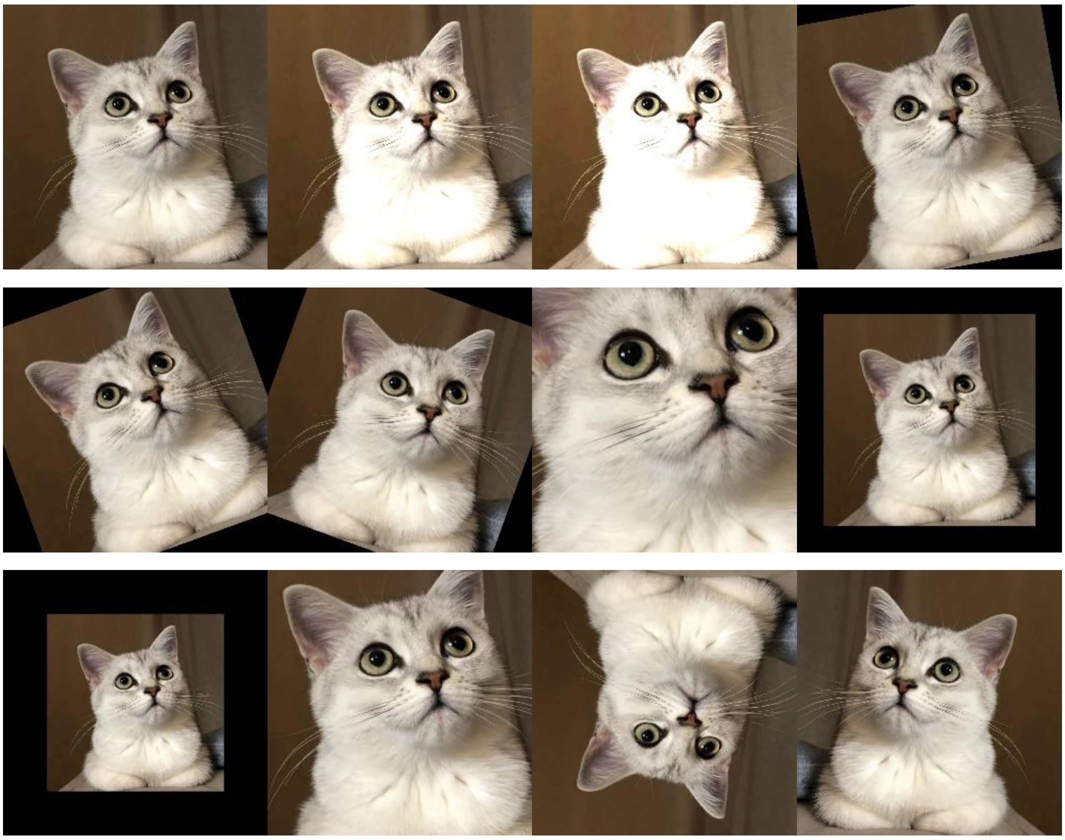


Figure 2.6: Various image augmentations performed upon the same image.

2.4 Deep learning

Deep learning is a subset of machine learning characterised by large networks of many layers[25]. There are several types of deep learning, including deep neural networks (DNNs) and deep reinforcement learning. Here I concentrate on DNNs. Although neural networks were first proposed in the 1960s, they have only gained widespread adoption in image applications from around 2010 due to increased computational resources[26]. Deep learning has been applied to various tasks, including image analysis, language processing, drug discovery, and self-driving cars.

Several layers are used concurrently in deep learning to form a network; after each layer, an activation is placed to alter the layer's output. The first layer in a neural network is known as the input layer, the final layer is known as the output layer, and the intermediate layers are known as hidden layers as we do not observe them. Each layer contains units, also called neurons, which are connected to the units in other layers. The values of these units are determined by the value of the units they are connected to in the previous layers and the model parameters, also called weights.

When selecting the parameter values of a deep learning model, we aim to reduce the distance between the observed and predicted outcome; we measure this distance using a loss function. Finding the optimal parameters to minimise this loss function is known as training.

The most common approach to training a deep learning model is gradient descent, first proposed by Cauchy in 1847. A deep learning model is trained iteratively by setting the parameters and calculating the loss function, the parameters are then changed slightly, and the loss function is recalculated. The gradient of the loss function from the previous step is used in an optimisation algorithm to update the parameters for the next step. Unfortunately, when training a neural network, the error signal and loss function gradient used to update the parameters can become increasingly small as the gradient moves through the layers, preventing convergence to the optimal parameters. This is known as the vanishing gradient problem.[27] Conversely, the error signal could "blow up" and tend to infinity, causing the exploding gradient problem. A few methods have been proposed to reduce the risk of vanishing or exploding gradients, which are discussed later in the thesis.

In image analysis, the data and model are often too computationally expensive to fit in the available memory; therefore, we often split the dataset into batches of data and iterate over these batches. We call one iteration over a single batch a step and one iteration over the entire dataset an epoch. Usually, the data is shuffled after every epoch to avoid the algorithm encountering the same order of batches in each epoch.

Here I describe the basic building blocks of deep learning: the layers, activation functions, loss functions, and optimisation algorithms. When discussing these, the focus will be on computer vision and image analysis; however, the layers can also be used in other applications, such as natural language processing.

2.4.1 Deep learning layers

DNNs consist of many consecutive layers.[25] There are several types of layers that I describe here. The layers used in deep learning can consist of simple operations, such as additions, multiplications, averages, and dot products; however, more complex layers combine operations. Here, I briefly outline some of the most commonly used layers in deep learning. Although these layers can be applied to various tasks, I will use image analysis for illustration as that is the main focus of the thesis.

2.4.1.1 Fully connected layers

The most straightforward layer of a neural network is the fully-connected layer, also called a dense layer. Each unit is connected to each unit in the next layer; this makes the layer computationally expensive.[28] Given a vector of features X and some parameters or weights W with some bias b, the fully connected layer is calculated by

$$Y = WX + b \tag{2.11}$$

Networks consisting solely of fully-connected layers are known as fully-connected networks. Fully-connected neural networks have been used in computer vision when the size of the image is small. For example, the MNIST dataset consists of images of hand-written digits (0-9) of size 28×28 pixels.[29] Examples of these images are shown in Figure 2.7. These images can be flattened to a vector of length 784, and a fully-connected neural network can be used to classify which digit is written. On the MNIST dataset, a simple fully-connected network can achieve impressive performance; however, fully-connected networks become too computationally complex when high-resolution images are used. An example of a fully-connected network, illustrated with a binary classification task, is shown in Figure 2.8.

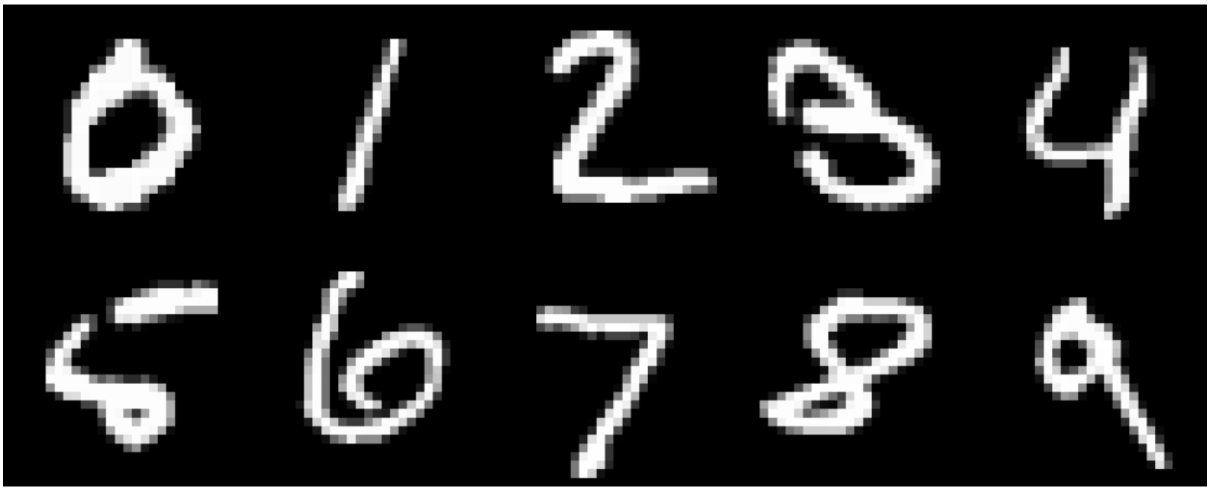


Figure 2.7: Example images from the MNIST handwritten digit recognition dataset.

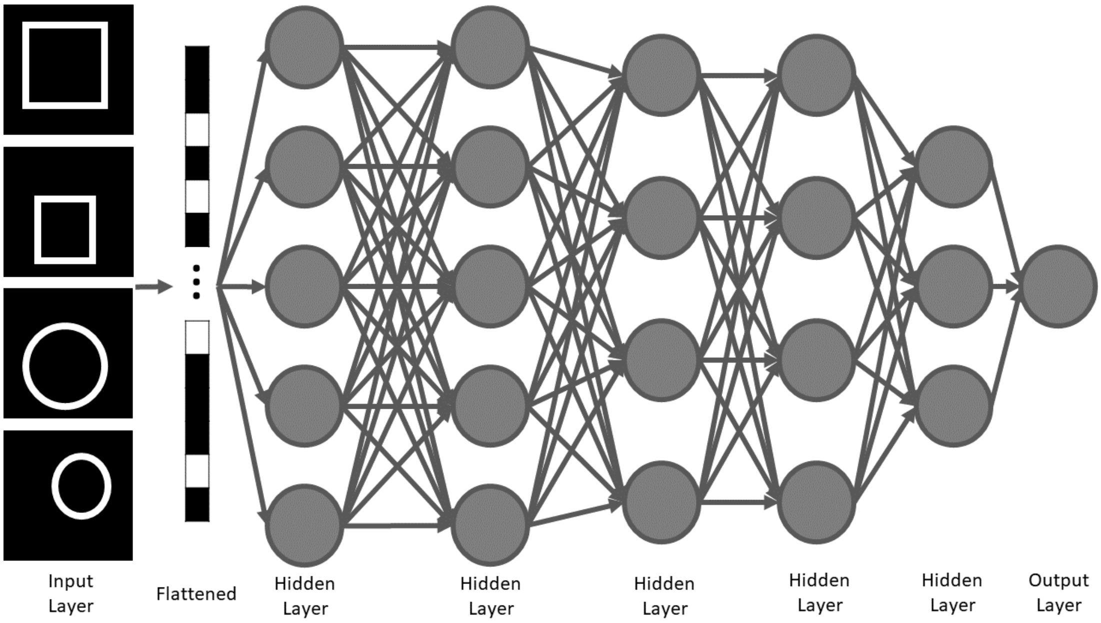


Figure 2.8: An example of a fully-connected neural network. The task used to illustrate the network is a simple binary classification task of square vs circle. The image is flattened to a vector representation. There are five hidden layers; the first two hidden layers have five units each, the subsequent two hidden layers have four units each, and the final hidden layer has three units. The output layer consists of a single unit. An activation function is usually applied to the final layer to obtain a probability of the shape being a circle.

2.4.1.2 Convolutional layers

Convolutional neural networks (CNNs) replace one or more matrix multiplications with a convolution. While fully-connected layers have each unit connected to each unit in the previous layer, units in a convolutional layer are only connected to previous units in their receptive field; this dramatically reduces computational complexity.[28] This receptive field can be thought of as a field of vision for that particular unit. Additionally, convolutional layers preserve some of the spatial information lost when using a fully-connected network, as we no longer need to flatten the image to a vector form. For example, in computer vision, two-dimensional (2D) convolutions are often used to reduce an image into a representation in smaller dimensions. These smaller representations can then be passed to a fully-connected layer for classification. The convolutional layer reduces the image dimensions by reducing the height and width of the image while expanding the image depth. The depth dimension is made up of channels; for example, an RGB image has three channels, red, green, and blue.

The parameters of a convolutional layer are filters or convolutional kernels; these filters are the size of the receptive field, which is chosen as a hyperparameter. Each filter is convolved over the image and then multiplied and accumulated. In simpler terms, convolving is like sliding the filter across the image, moving a fixed number of pixels at a time. The number of pixels that we move along is called the stride. This results in a feature map. Each filter creates a different feature map representing different image features; these multiple feature maps can be stacked to produce a three-dimensional (3D) representation. At the edges of the image, padding may be used to produce a feature map of the same width and height as the original input; padding with all zeros is most commonly used. The convolutional layer is shown graphically in Figure 2.9. The representation's height and width can be reduced by increasing the stride; this centres the filter at larger intervals. The effects of using a stride of size two are shown in Figure 2.10. In these diagrams, integer values are used for illustration; however, the values may not necessarily be integers.

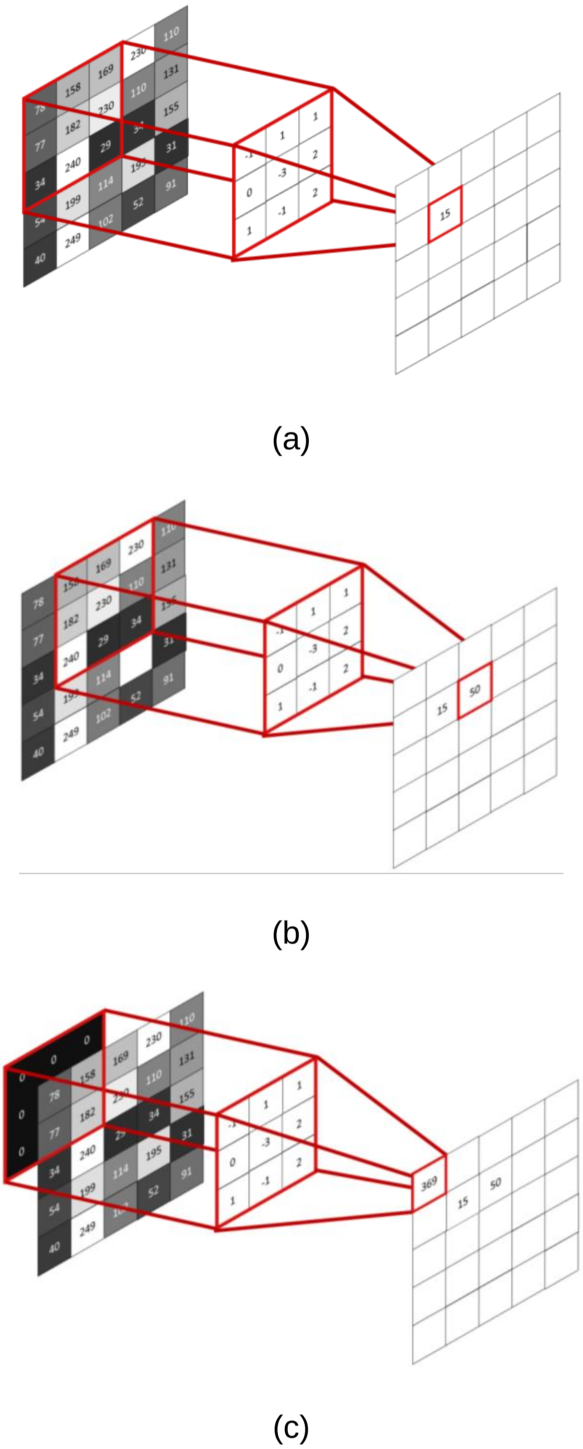


Figure 2.9: The convolutional layer takes an image and a filter the same size as the receptive field, consisting of parameters to be learned. (a) The dot product of the filter and receptive field is calculated, and the value is output. (b) We then slide the filter along the image and repeat. (c) At the edges of the image, zero padding is used.

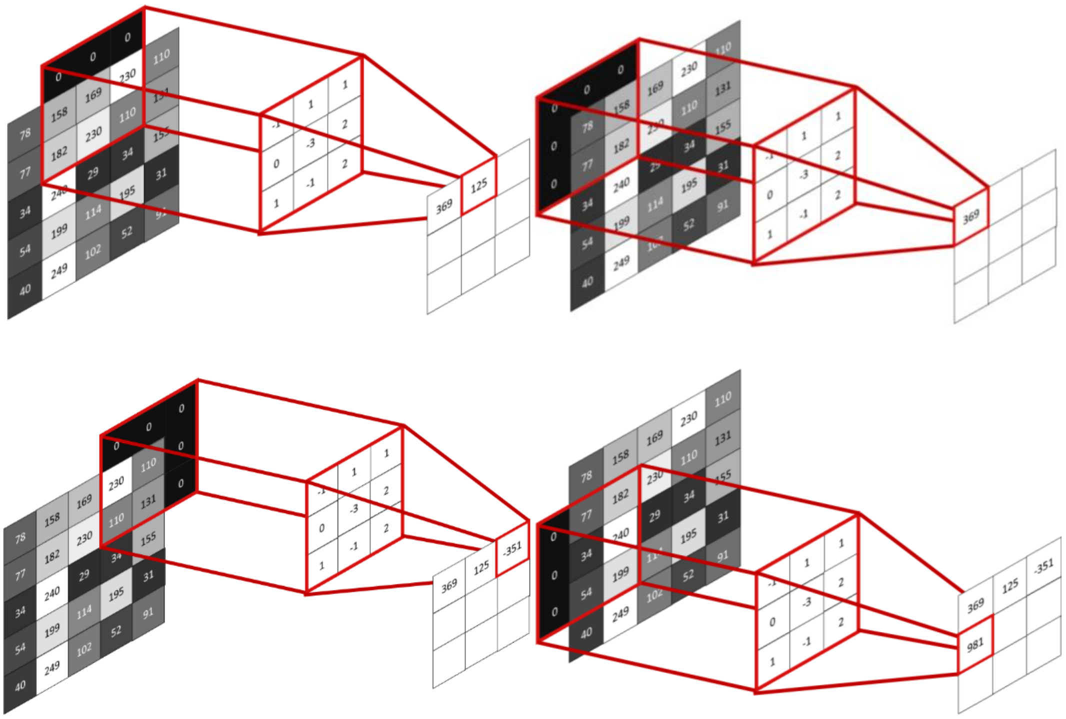


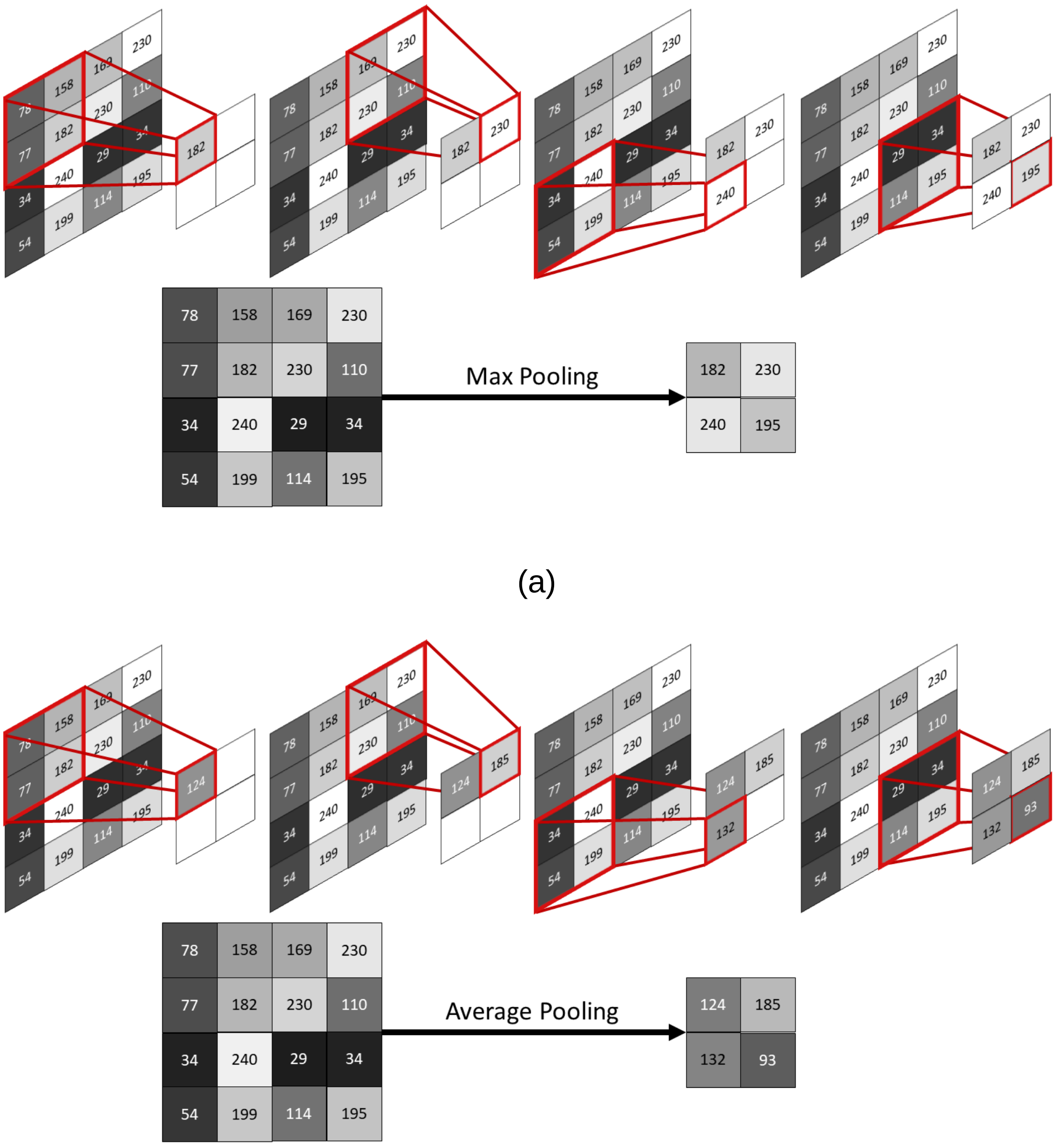
Figure 2.10: Convolution with a stride size of 2. The filter moves along the image by two pixels instead of pixel by pixel.

2.4.1.3 Pooling layers

The feature maps produced by convolutional layers are often sensitive to the location of the identified features. However, it is unlikely that those features will always be in the same location of the image each time; therefore, pooling can make the model more robust to the location of features in the image. In addition, pooling effectively downsamples or downsizes the image, reducing the model's computational requirements.

Like a convolutional layer, each unit in the pooling layer is connected only to the units of the previous layer within a receptive field[28]. Pooling layers do not have a filter, and therefore there are no parameters to train. The two main types of pooling are max pooling and average pooling. Max pooling takes the maximum value of the receptive field, while average pooling uses the mean value. Similar to a convolutional layer, the size of the receptive field, stride size, and padding can be chosen. Pooling layers are applied to each channel (or feature map) separately.

An illustration of how max and average pooling works is shown in Figure 2.11. Integer values are shown, and the output is converted to pixel values for illustration; however, when pooling is applied to hidden layers, values may not necessarily be integers. An example of both max and average pooling applied to an image is shown in Figure 2.12. The image resized using max-pooling appears slightly brighter than average pooling, especially around the eyes and whiskers; this is due to max-pooling selecting the maximum value in each receptive field.



78	158	169	230
77	182	230	110
34	240	29	34
54	199	114	195

182	230
240	195

(a)

78	158	169	230
77	182	230	110
34	240	29	34
54	199	114	195

124	185
132	93

(b)

Figure 2.11: Diagram illustrating how (a) max pooling and (b) average pooling work. A stride of size 2 is used, resulting in an image that is two times smaller in both height and width. Integer values and the output is converted to pixel values to better display the difference between max and average pooling.

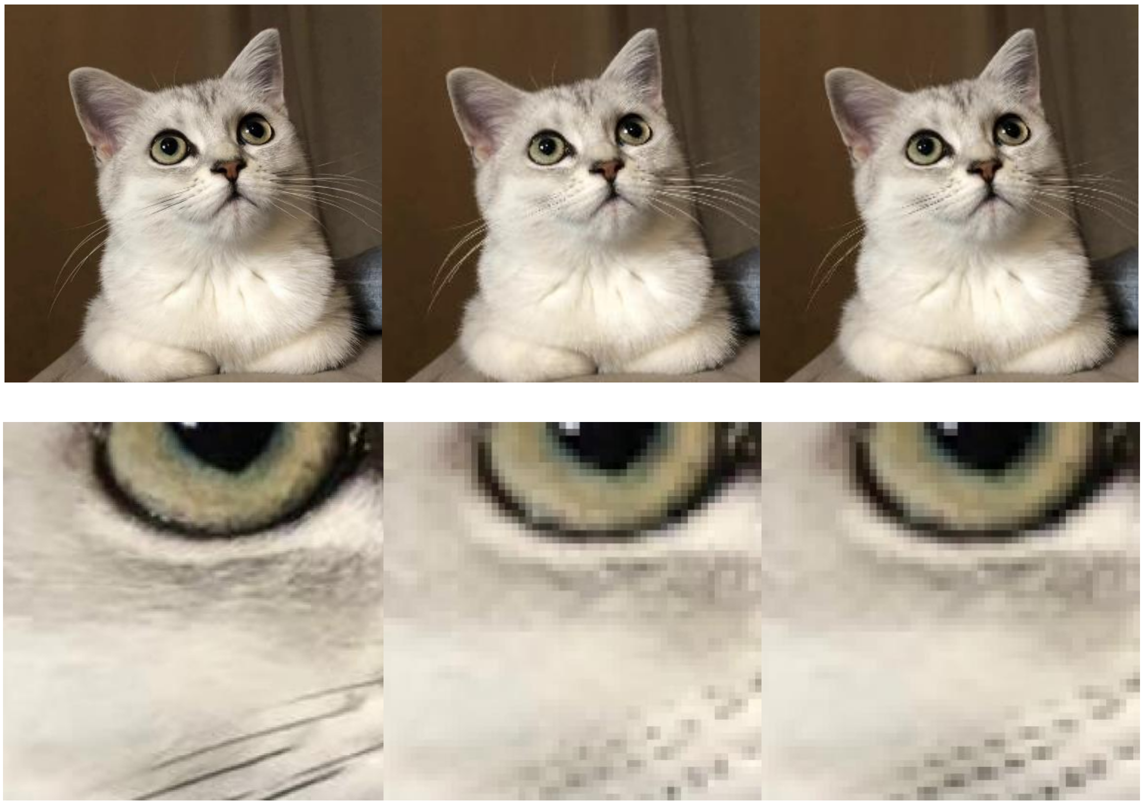


Figure 2.12: (a) Original image with dimensions 1024×1024*, (b) image resized to* 256×256 *using max pooling with a stride of* 4 *and kernel size* $(2, 2)$*, and (c) image resized to* 256×256 *using average pooling with a stride size of* 4 *and kernel size* $(2, 2)$*.*

Max and average pooling subsample within a receptive field, which is a small patch of the whole channel (or feature map). Instead, we could apply pooling to the whole channel by setting the size of the receptive field to be the size of the channel; this is known as global pooling. Global pooling reduces the channel to a single value and can be used to obtain a feature vector from the feature maps. These feature vectors can then be fed into a fully-connected network for classification. Global pooling can be used as an alternative to flattening the feature maps, resulting in a much smaller feature vector.

2.4.1.4 Recurrent layers

In convolutional neural networks, the activation only flows forward from the input layer to the output layer; these networks are called feedforward neural networks[28]. For some applications, we may want to retain past information. Recurrent neurons can reinput their output into themselves, retaining previous information through feedback connections; this makes them particularly useful when dealing with sequences. A comparison between a feed-forward neuron and a recurrent neuron is shown in Figure 2.13, and a diagram of recurrent neurons unrolled over time is shown in Figure 2.14.

Recurrent neural networks (RNNs), which use these recurrent cells, often suffer from the vanishing gradient problem. Long short-term memory (LSTM) is one type of recurrent unit and was created to overcome the vanishing gradient problem.[30] LSTM outputs two vectors at each time point t: one with the short-term state h_t and one with the long-term state c_t. Inside the unit there are three gates: an input gate i_t, an output gate o_t, and a forget gate f_t. The cell remembers the information over several time points while the gates help control the flow of information through the cell. The input gate controls which parts of the input are stored in the long-term state, the forget gate controls which parts of the long-term state are forgotten and the output gate controls which parts of the long-term state are output. An LSTM unit is displayed in Figure 2.15.

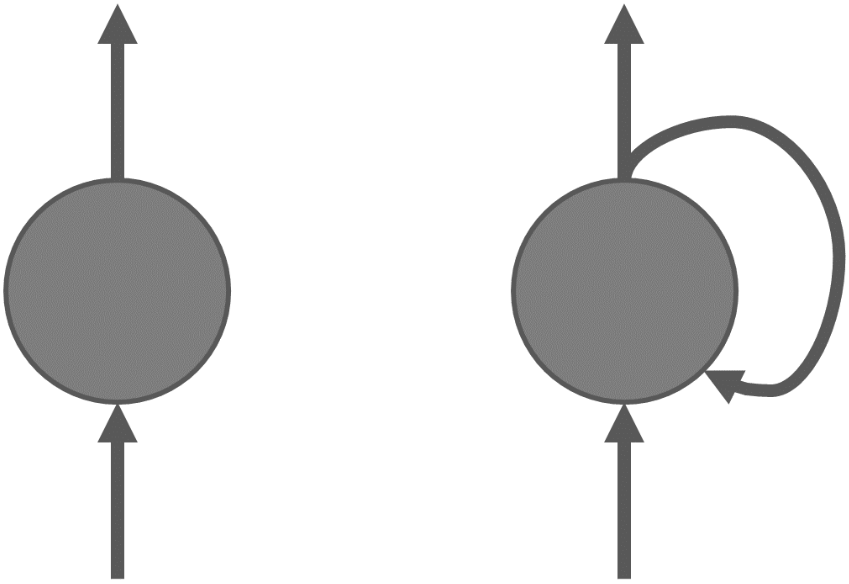


Figure 2.13: A basic representation of a feedforward neuron (left) and a recurrent neuron (right).

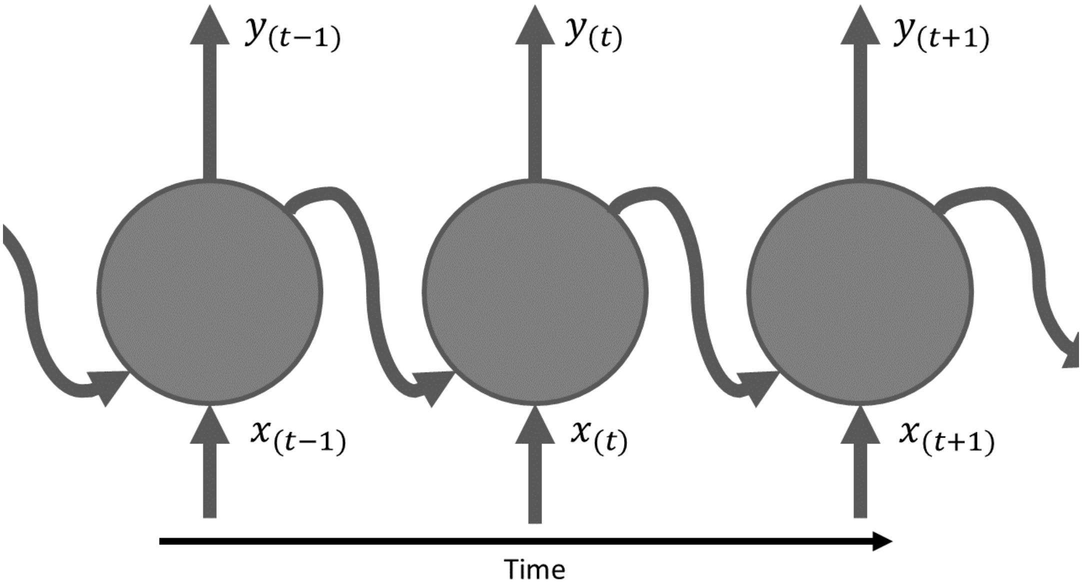


Figure 2.14: Recurrent neurons unrolled over three time points.

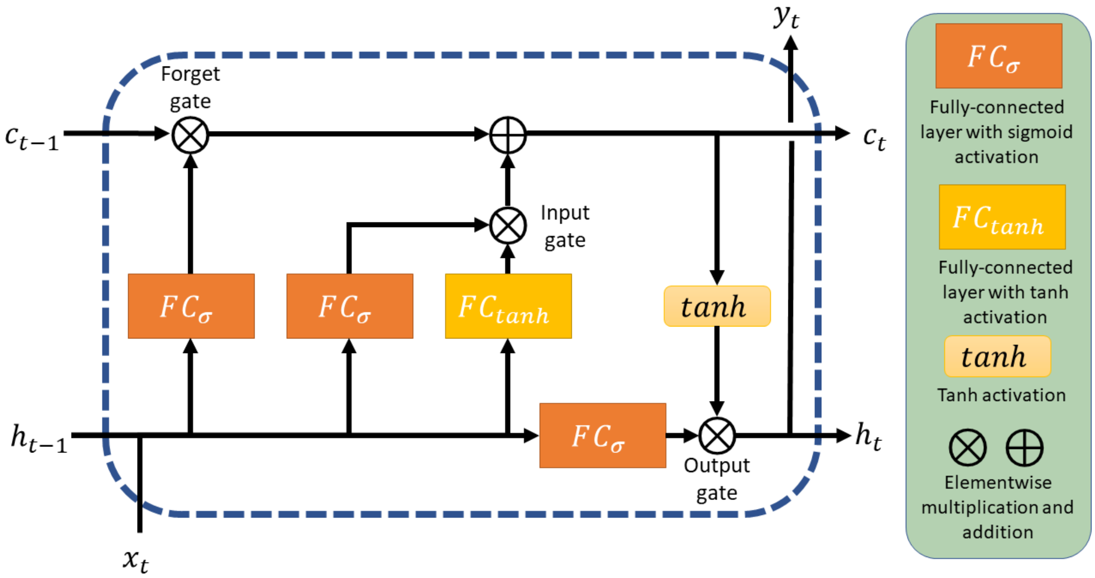


Figure 2.15: Diagram of an LSTM cell.

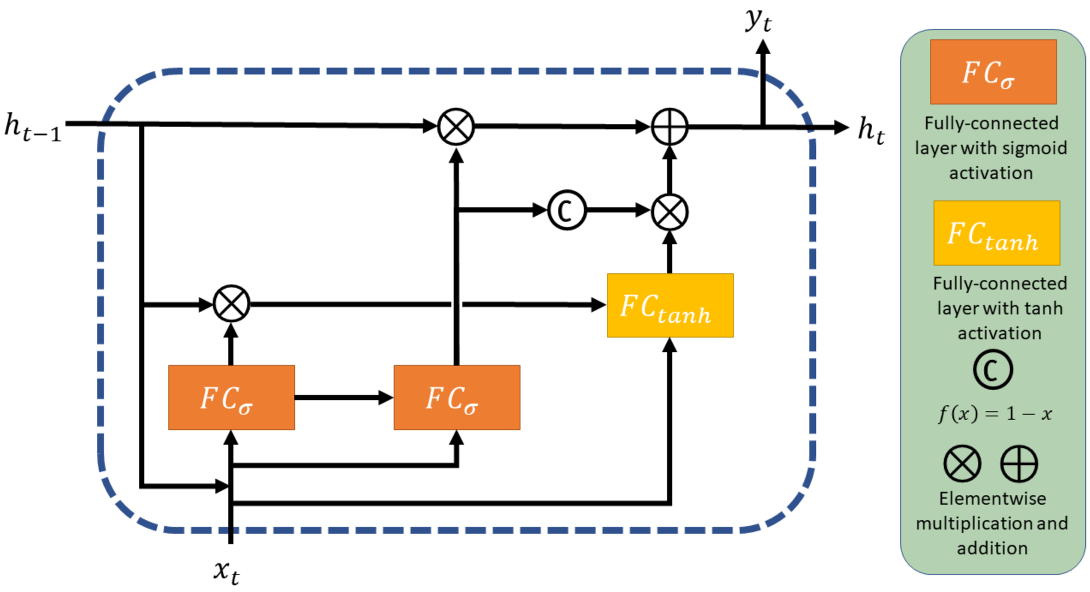

y_t
h_{t-1}
h_t
x_t
FC_σ
FC_σ
FC_{tanh}
FC_σ
Fully-connected layer with sigmoid activation
FC_{tanh}
Fully-connected layer with tanh activation
$f(x) = 1 - x$
Elementwise multiplication and addition

Figure 2.16: Diagram of a GRU cell.

The main aim of this LSTM unit was to overcome the problems with the error back-flow, which could solve the vanishing gradient problem. Before the introduction of the LSTM, the practical usefulness of RNNs, especially for many time steps, was in question. RNNs often encountered either vanishing or exploding gradients with as few as five time steps.[31] LSTM was able to train stably with as many as 1000 time points. With the introduction of LSTM, interest in RNNs was renewed. However, LSTM units are complex and computationally expensive; there was room for improvement in simplifying the module.

The gated recurrent unit (GRU) cell greatly simplifies the gates used in LSTM by merging the input and forget gates and removing the output gate.[32] In addition, the two vectors containing the short- and long-term states are also combined into a single vector. This modification greatly reduces computational complexity; however, studies comparing LSTM and GRU have shown there to be little difference in performance.[33-35] A diagram of a GRU cell is shown in Figure 2.16. Although the GRU did not significantly improve the LSTM in terms of performance, the reduced complexity has led to significantly reduced training times and computational requirements. This improvement brought the stable training of large RNNs to a larger group of researchers and further improved the practical usefulness of RNNs.

Some networks utilise recurrent and convolutional layers, known as recurrent convolutional neural networks (RCNNs).[36]

2.4.1.5 Dropout

Deep learning models often have millions of parameters; some models can even have billions[37] or trillions of parameters.[38] This makes models prone to overfitting, where the model performs exceptionally well on the training data but fails to generalise to unseen data.

There are many ways to alleviate overfitting. First, we could use L1 and L2 regularisation. We could use a validation set evaluated during training and stop training when the validation performance stops improving. Finally, we could augment the data discussed to simulate extra samples.

Another option to reduce overfitting in deep learning models is dropout[39]. During training, at each step, we can randomly drop units out of the model with probability p. A common value for p is 0.5, meaning that each unit has a 50% probability of being dropped from the model at each step. This prevents the model from learning parameters that make just a few units useful, as those units could be removed from the model during training.

2.4.1.6 Batch normalisation

When training a deep learning model, the parameters of each layer are iteratively updated, meaning that the layer's inputs (and their distribution) are constantly changing during training. This makes the training sensitive to parameter initialisation and requires lower learning rates, increasing training time. To overcome these problems, Ioffe and Szegedy proposed normalising the inputs to the layer during training using batch normalisation.[40] They found that batch normalisation greatly reduces training time and acts as a regularisation.

During training, batch normalisation begins by calculating the mini-batch mean and variance

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \text{, and } \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2. \tag{2.12}$$

Then values are normalised

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{2.13}$$

where ϵ is a small smoothing term added to avoid division by zero. Finally, a scale and shift are applied to the normalised values giving

$$y_i = \gamma \hat{x}_i + \beta, \tag{2.14}$$

where γ and β are parameters to be learned.

During inference, the mean and standard deviation of the whole training set is used rather than the mini-batch mean and standard deviation.

As a regularisation technique, batch normalisation greatly reduces the need for dropout. Combining batch normalisation and dropout concurrently in the same network may seem logical to increase the amount of regularisation; however, this can result in significantly worse performance.[41]

2.4.2 Activation functions

An activation function determines the output of each layer or node in a neural network. Activation functions can be non-linear and are applied to the hidden and output layers. A common activation function for the hidden layer is the hyperbolic tangent function $f(x) = \tanh(x)$. Activation functions are analogous to link functions in generalised linear mixed models for the output layer. In binary classification tasks, the sigmoid activation is most commonly used. Activation functions are vital for introducing non-linearity into deep learning models. Without activation functions, using multiple layers would be equivalent to using a single layer.

2.4.2.1 Identity and linear activations

The simplest activation function is the identity activation function, which is the same as applying no activation to the node $f(x) = x$.

2.4.2.2 Tanh

The hyperbolic tangent (tanh) function is often used for the hidden layer activation. The tanh function is S-shaped and rescales values to between -1 and 1. The tanh function is given by

$$f(x) = \tanh(x)\,. \tag{2.15}$$

The tanh function is continuous and everywhere differentiable.

2.4.2.3 Sigmoid

The sigmoid function, also called the logistic function, is often used on the output layer to rescale the output between 0 and 1, forming a probability distribution. It is often used for binary or multiclass classification when the classes are not independent. The sigmoid function is

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{2.16}$$

Like the tanh function, the sigmoid function is S-shaped, continuous, and differentiable everywhere. The sigmoid activation is also commonly used for binary segmentation, where pixels are either objects or backgrounds.

2.4.2.4 Softmax

The softmax function can be used in a similar to the sigmoid to produce a probability distribution from the output layer values; however, the softmax function is used for multiclass classification when the classes are independent[42]. The softmax is given by

$$f(x) = \frac{e^{x}}{\sum_{i=1}^{n} e^{x}}, \tag{2.17}$$

where n is the number of classes, this produces a vector of probabilities for each class. The softmax function is continuous and differentiable. Similar to the sigmoid activation, the softmax can be used for segmentation; however, the softmax is used where multiple different objects are segmented.

2.4.2.5 ReLU

The choice of hidden layer activation is vital for vanishing gradients, as the S-shaped function saturates in both positive and negative extremes. The rectified linear units (ReLU) activation function was proposed to mitigate the vanishing gradient problem[27]. The ReLU activation is given by

$$f(x) = \max(0, x). \tag{2.18}$$

This function sets values less than 0 to zero and is the identity function for values greater than 0, which means the ReLU activation is computationally efficient. In addition, unlike the tanh, sigmoid, and softmax activations, the ReLU activation is non-differentiable at zero.

One of the advantages of the ReLU activation is that all values lower than zero are set to zero; only around half of the units are activated, resulting in a sparse activation. However, this can result in the units remaining at zero, known as the dying ReLU problem. Several variants of the ReLU activation have been proposed to overcome this problem. The leaky ReLU[43] adds a slight gradient to the negative units

$$f(x) = \begin{cases} x, & if\ x \geq 0, \\ \alpha x, & if\ x < 0. \end{cases} \tag{2.19}$$

The hyperparameter α is often chosen to be 0.01, but other values may be chosen. The parametric ReLU activation is the same as the leaky ReLU with the value of α learned by the algorithm during training[44]. Another variant of leaky ReLU is randomised leaky ReLU[45], where the value of α is chosen from a uniform random distribution

$$\alpha \sim U(l, u) \tag{2.20}$$

during training and fixed to the average

$$\alpha_{test} = \frac{l + u}{2} \tag{2.21}$$

during inference.

The exponential linear unit (ELU)[46] activation function is similar to leaky ReLU. The ELU is given as

$$f(x) = \begin{cases} x, & if\ x \geq 0, \\ \alpha(\exp(x) - 1), & if\ x < 0, \end{cases} \tag{2.22}$$

where $\alpha > 0$. The ELU activation function is more computationally expensive than the other ReLU variants; however, training time is reduced because the mean unit activations are pushed closer to zero, similar to batch normalisation.

2.4.2.6 Swish

The swish activation function was found via an automatic search[47]. The swish function is defined as

$$f(x) = x\sigma(x), \tag{2.23}$$

where $\sigma(x)$ is the sigmoid activation. In experiments, the swish activation function was found to outperform the ReLU activation.[47]

2.4.3 Commonly used CNN architectures

The majority of work in image classification has focused on CNN-based models. A few famous and well-developed CNN architectures are commonly used for various tasks. The work presented in this thesis uses these CNN architectures as a backbone, and here I describe some of the most common CNN architectures.

2.4.3.1 ImageNet

Before discussing architectures, it is important to introduce one of the main driving forces behind image recognition over the last decade. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[48] is an annual competition that evaluates computer vision algorithms in image classification and object detection. The original ImageNet dataset contained over 1 million images from 1000 classes. It is used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a competition between network architectures and deep learning techniques. These classes consist of real-world photographs such as dogs, cats, flowers, and cars; often, the images are complex scenes. Models are usually assessed in their accuracy of correctly predicting the class (top-1 accuracy) or having the correct class in their top five predictions (top-5 accuracy).

ImageNet has undoubtedly pushed researchers to continually improve image recognition algorithms; however, there are some issues with using ImageNet. Firstly, models have exceeded human-level performance. For example, the current best classification algorithm at the time of writing[49] for top-5 has an error rate of 0.98% compared to 5.1% for humans, and the best algorithm for top-1 accuracy has an error rate of 9.8%.[50] We are reaching a saturation point where algorithms can no longer improve, and a more challenging dataset is needed. It is also possible that some of the 0.98% error rate is due to mislabelling in the dataset[51]. Assessing the models based on the error rate may not necessarily be the most appropriate evaluation method if we are concerned with how wrong the errors are. Models with higher error rates but predictions closer to the actual answers may be preferred. For example, an algorithm may misclassify a bird as the wrong species of bird, while an algorithm with a lower error rate may classify the bird as a flower. Nonetheless, ImageNet is still a hugely influential dataset and has led to many great innovations in architecture design.

Often these models are available with parameters already trained on ImageNet. Using pretrained parameters means that the parameters are already initialised to reasonable values for feature extraction; therefore, we only need to fine-tune these parameters, and the training time is greatly reduced.

2.4.3.2 LeNet

One of the first CNNs to be developed was LeNet. In this groundbreaking series of work, LeCun et al. showed that single-layer networks do not generalise well; however, multilayer constrained networks using convolutional layers (called shift-invariant feature detectors in 1989) have much better performance.[52] The authors identified that handwriting recognition models often consist of multiple modules, and a multi-layer neural network may be able to replicate these modules. LeNet used multiple stacked layers to develop a model which could recognise hand-written digits. Although this model was simple and limited in applicability, the initial work was improved upon, and LeNet-5 was created, showing improved performance compared to previous methods on handwritten digits obtained from the US Postal Service,[53] [54] with minimal pre-processing of the images. LeNet showed that hand-crafted features were no longer necessary, allowing more complex problems to be solved quicker and cheaper than using previous methods.

LeNet-5 consists of 7 layers. The original implementation of LeNet-5 takes a 32×32 image as its input. First, the image is passed to a convolutional layer with 6 filters and a kernel size of 5×5, followed by a tanh activation function. The dimensions of the feature maps are then reduced using an average pooling layer with a stride size of 2 and a kernel size of 2×2. Next, a second convolutional layer is used with 16 filters and a tanh activation, followed by another average pooling layer. This results in a feature map of size $5 \times 5 \times 16$. A final convolutional layer, with 120 filters and kernel size 5×5, reduces the feature map into a feature vector of length 120. A fully-connected layer follows this with 84 units and tanh activation. Finally, a full-connected layer with 10 units (one for each class) and softmax activation produces a probability of the image showing each class. This architecture is outlined in Figure 2.17.

The MNIST dataset presented LeNet-5 with 58,527 images of a single digit from 500 different writers. The model achieved a final error rate of just 0.8% in the test set.[29] This performance matched the performance of the previous best method using support vector machines; however, the number of multiply-add operations was greatly reduced when using LeNet-5 from 28,000 to 401, providing a much more practically applicable method.

LeNet-5 was initially run on a Sun-4/260 workstation with 128MB of memory. Due to these hardware restrictions, the network size could not be increased much more, and LeNet-5 is restricted to small, simple image recognition tasks, such as handwriting and digit recognition. Despite this, LeNet-5 was the breakthrough work that brought attention to CNNs and advances in both computation and deep learning meant that CNNs could be applied to a broader variety of applications.

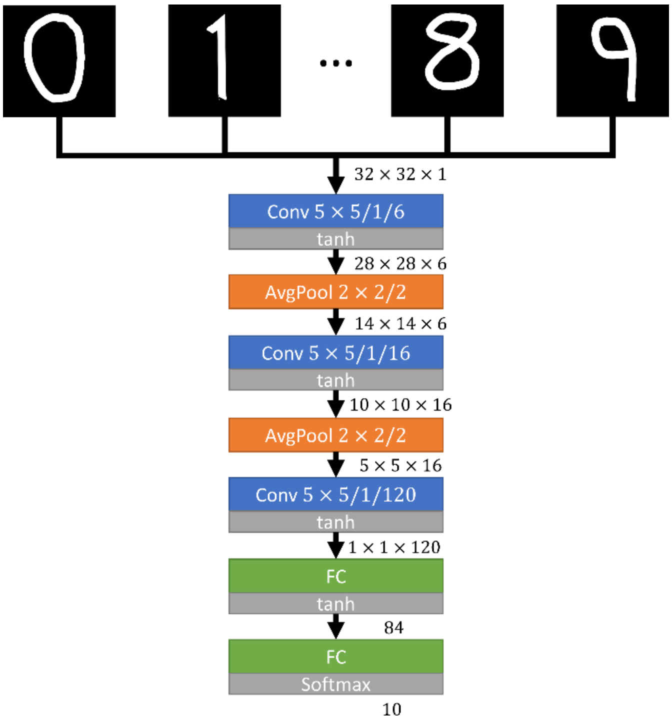


Figure 2.17: Diagram of the LeNet-5 architecture. Handwritten digits, such as those in MNIST, are used as examples. Convolutional layer parameters are displayed as kernel size/strides/filters, pooling layer parameters are displayed as kernel size/strides, and output sizes are shown below each layer.

2.4.3.3 AlexNet

In 2012, a new CNN called AlexNet[55] won the ILSVRC[48]. With advances in computing hardware, AlexNet was able to add more layers creating a larger and deeper network which can classify higher-dimensional colour images. As a result, AlexNet uses colour images with size 227×227 as its input by default.

AlexNet was the first CNN to begin stacking convolutional layers without a pooling layer between them; the authors noted that removing these stacks resulted in a drop in model performance. Dropout and data augmentation were also used to reduce overfitting. AlexNet was applied to the ImageNet dataset. Only cropping was used to reduce the image size, with no additional image pre-processing required. AlexNet achieved a top-5 error rate of 17%, with the second-place competitor only achieving 26%.[55] These impressive results show huge improvements over the other methods in the competition. Despite the problems associated with ImageNet, which I have discussed, AlexNet clearly shows more accurate classifications on a large-scale dataset. The improvement is clear, although the results lack the fundamental robust analysis we expect today, such as confidence intervals. The reasons behind this improvement are also well justified in the paper. The authors describe how they offset the risk of overfitting with a larger model by introducing dropout and data augmentations.

The authors also gave some qualitative evidence of what the network had learned by showing some examples of images with predictions and some groups of images that the network identified as similar. This qualitative analysis is not as sophisticated as the model visualisation methods used today; however, it was an attempt to reduce the black-box nature of DL.

A diagram of AlexNet is shown in Figure 2.18. The following year the ILSVRC was won by ZFNet, which is very similar to AlexNet with slightly improved hyperparameters.

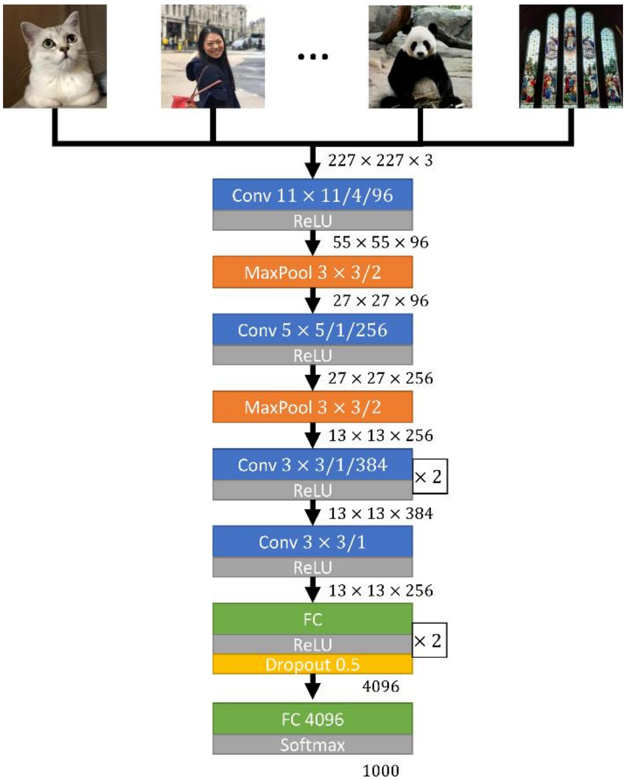


Figure 2.18: Diagram of the AlexNet architecture. Natural images, such as those in ImageNet, are used as an example. Convolutional layer parameters are displayed as kernel size/strides/filters, pooling layer parameters are displayed as kernel size/strides, and output sizes are shown below each layer.

2.4.3.6 VGG

VGGNet[56] was runner-up in the ILSVRC 2014 for classification with a top-5 error rate of 7.32%. There are two variants: VGG16 and VGG19, with VGG19 being a deeper version of VGG16. VGGNet stacks convolutional layers similar to AlexNet; however, VGGNet has many more units in the convolutions. The creators of VGGNet concluded that an increased model depth results in improved model accuracy.[56] Although this is somewhat true, deeper neural networks require more data and regularisation to prevent overfitting. I would argue that increased model depth leads to improved accuracy on the training data, but it may result in reduced accuracy on external data. Both VGG16 and VGG19 are displayed in Figure 2.19.

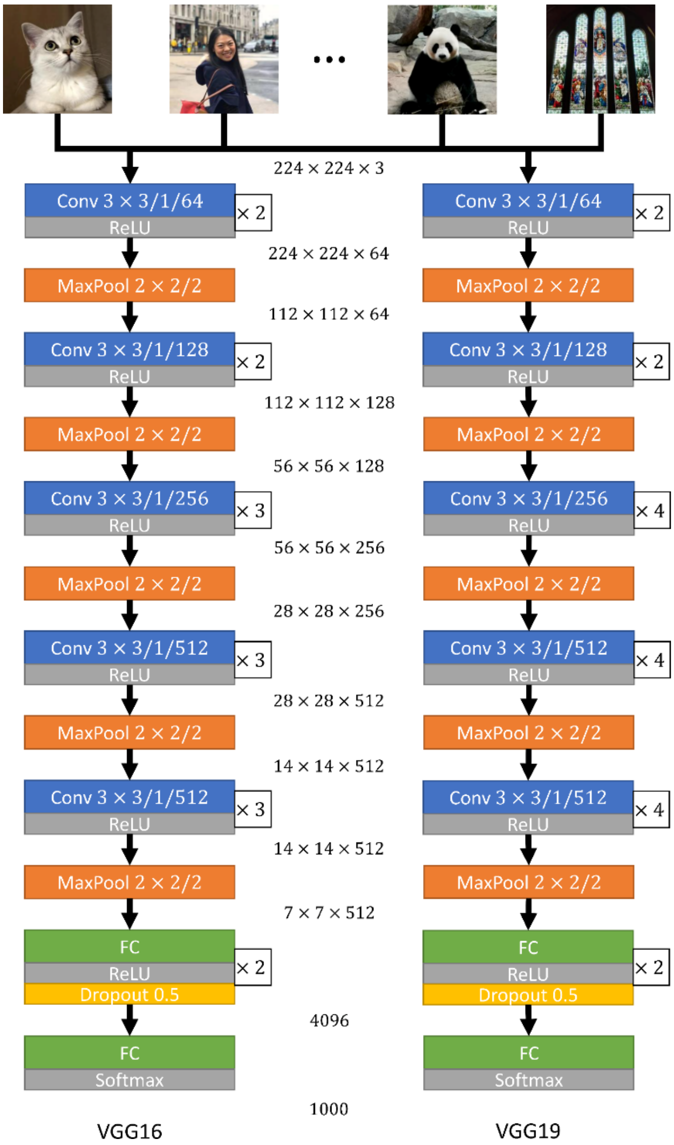


Figure 2.19: Diagram of the VGGNet architectures (Left: VGG16; Right: VGG19). Natural images, such as those in ImageNet, are used as an example. Convolutional layer parameters are displayed as kernel size/strides/filters, pooling layer parameters are displayed as kernel size/strides, and output sizes are shown below each layer.

2.4.3.7 Inception

The architecture which beat VGGNet for classification is called Inception or GoogLeNet.[57] The name Inception is a reference to the 2010 film.[58] Inception achieved a top-5 error rate of 6.67%; this is a 56.5% relative reduction in error rate compared to AlexNet two years prior. The authors of Inception reached a similar conclusion to the authors of VGGNet a deeper network leads to improved performance; however, Inception also increases the width of the network. The main feature of Inception is the Inception module. The input feature maps are copied to separate branches with different kernel sizes, which can capture features at several scales. Padding ensures that the output size is equal despite the differing kernel sizes. The module uses 1×1 kernels, which act as bottleneck layers, reducing dimensionality and computation. The motivation behind this approach is that salient objects in images vary greatly in size; sometimes, the object may be close to the camera and occupy a large part of the image; other times, the object may be further away, occupying a smaller section. Therefore, using various kernel sizes to capture both global and local features can be beneficial. A diagram of an Inception module is shown in Figure 2.20, and the full Inception-v1 is shown in Figure 2.21.

Although VGGNet improved significantly over AlexNet, the primary motivation was to simply make the network deeper. The authors of Inception-v1 confirmed that the assumption of a deeper network leads to improved performance; however, the addition of a wider network is why Inception-v1 won the ILSVRC challenge that year.

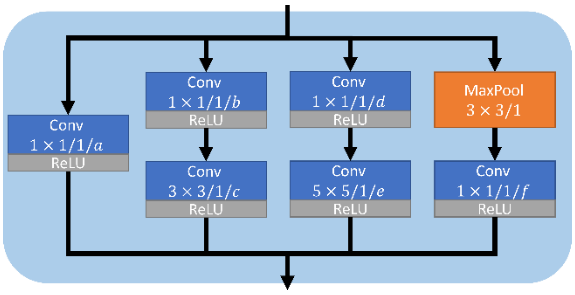


Figure 2.20: Diagram of an Inception module.

Inception-v2 tweaked the Inception-v1 architecture replacing the 5×5 convolution with two 3×3 convolutions. Although this increases the number of convolutions, it reduces computational complexity. One 5×5 convolution is 2.78 times more computationally expensive than a 3×3 convolution ($5^2/3^2 = 2.78$). For similar reasons, asymmetric convolutions were introduced where instead of a 3×3 convolution, a 3×1 convolution is followed by a 1×3 convolution. They also altered the inception network to reduce the grid size while expanding the filter bank, reducing the computational complexity. Diagrams of the modules for Inception-v2 are shown in Figure 2.22.

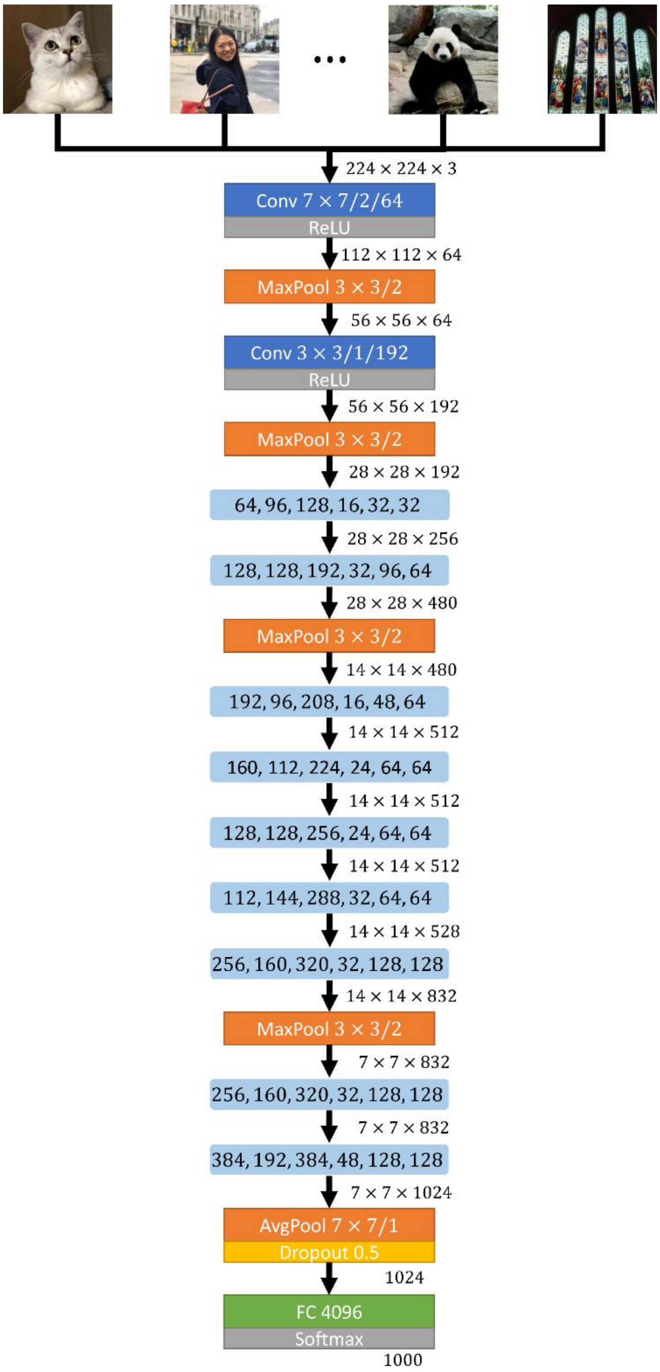


Figure 2.21: Diagram of the Inception-v1 architecture. Natural images, such as those in ImageNet, are used as an example. Convolutional layer parameters are displayed as kernel size/strides/filters, pooling layer parameters are displayed as kernel size/strides, and output sizes are shown below each layer. Numbers inside the light blue boxes correspond to the a, b, c, d, e, f, and g values in Figure 2.20.

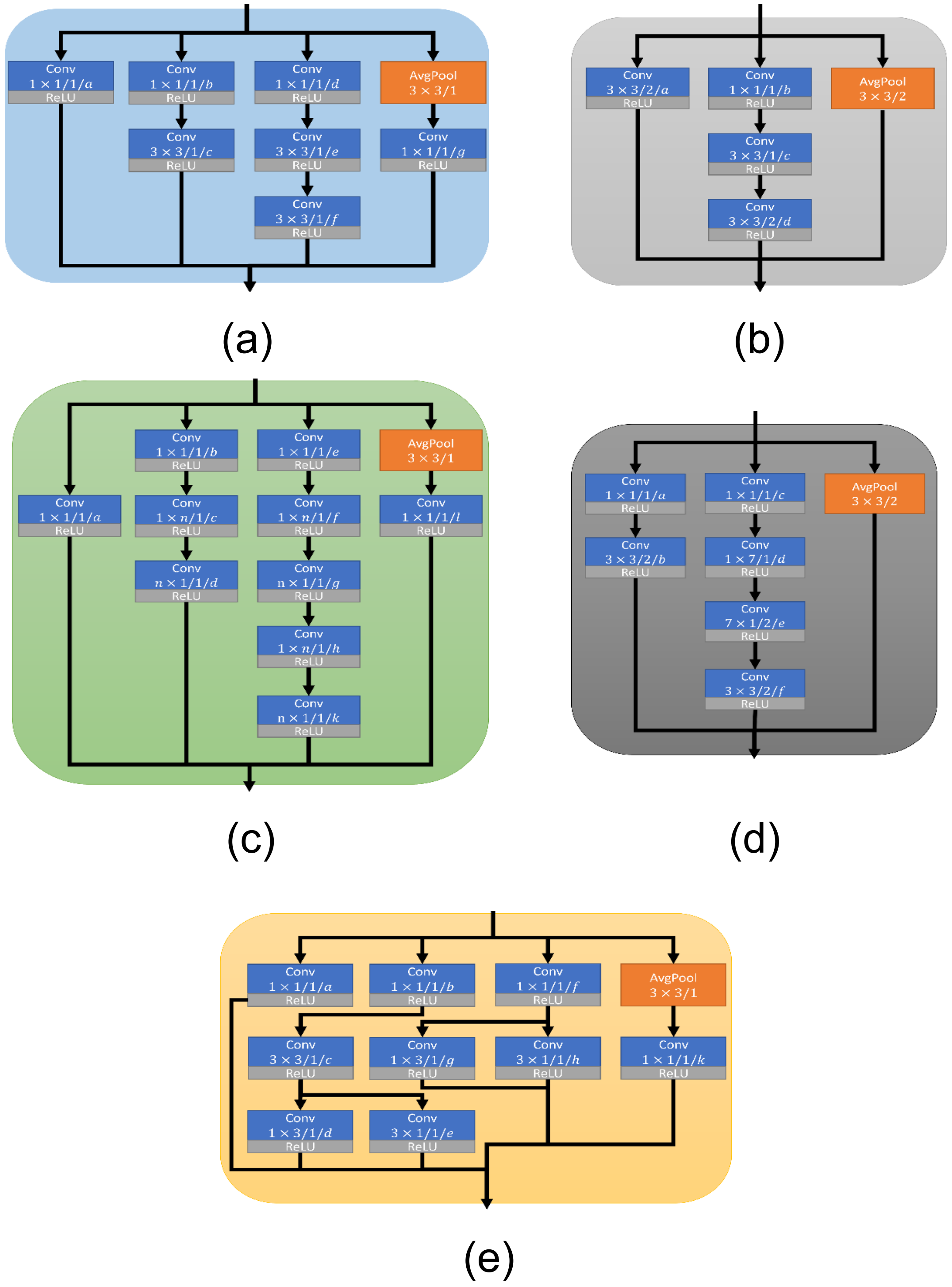


Figure 2.22: Inception-v2 and v3 modules.

In the same paper, four additional changes were made to this Inception-v2 architecture, with the resulting altered architecture and training scheme being referred to as Inception-v3. First, the RMSProp optimiser was used. Secondly, label smoothing was used to add uncertainty to the labels and reduce overfitting. Third, the 7×7 convolution was replaced with three 3×3 convolutions, similar to the replacement of the 5×5 convolution. Finally, batch normalisation was applied to the auxiliary output layer. Inception-v3 is available as a pretrained model in TensorFlow; however, the implementation is slightly different to the paper. In Figure 2.23, the Tensorflow implementation is shown.

A more streamlined and simplified architecture was later proposed called Inception-v4[59]. Inception-v4 aims to further increase the efficiency of the Inception module by making it deeper and wider. The first change to the architecture added branches to the stem before the inception modules. Secondly, the Inception modules were made more uniform; this change was due to a different library being used to train, which meant the model no longer needed to be trained in partitions. Finally, a new reduction block was introduced.

In this thesis, I choose to use Inception-v3 in all experiments. The combination of depth and width in the network combined with batch-normalisation results in an architecture with good performance and high generalisability on many tasks. The replacement of larger kernel size convolutions with a series of smaller ones also makes the overall computational complexity reasonable.

2.4.3.8 ResNet

A residual network (ResNet)[60] won the ILSVRC in 2015, lowering the top-5 error rate to just 3.57%. The primary motivation behind ResNet is to overcome the problem of parameters being too close to zero, causing the outputs to be zero. ResNet adds a skip connection that controls the flow of gradients to reduce the chances of the vanishing or exploding gradient problem. Figure 2.24 displays the skip connection for a two and three-layer module.

There are several variants of ResNet architecture proposed in the original paper.[60] Resnet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet152. ResNet50 is the most widely used as it is a reasonable trade-off between performance and computational requirements. ResNet-18 and ResNet-34 are made up of modules with two convolutional layers, while ResNet-50, ResNet-101, and ResNet-152 modules have three convolutional layers. I show architectures for ResNet-18, ResNet-34, ResNet-50, and ResNet-101 in Figure 2.25.

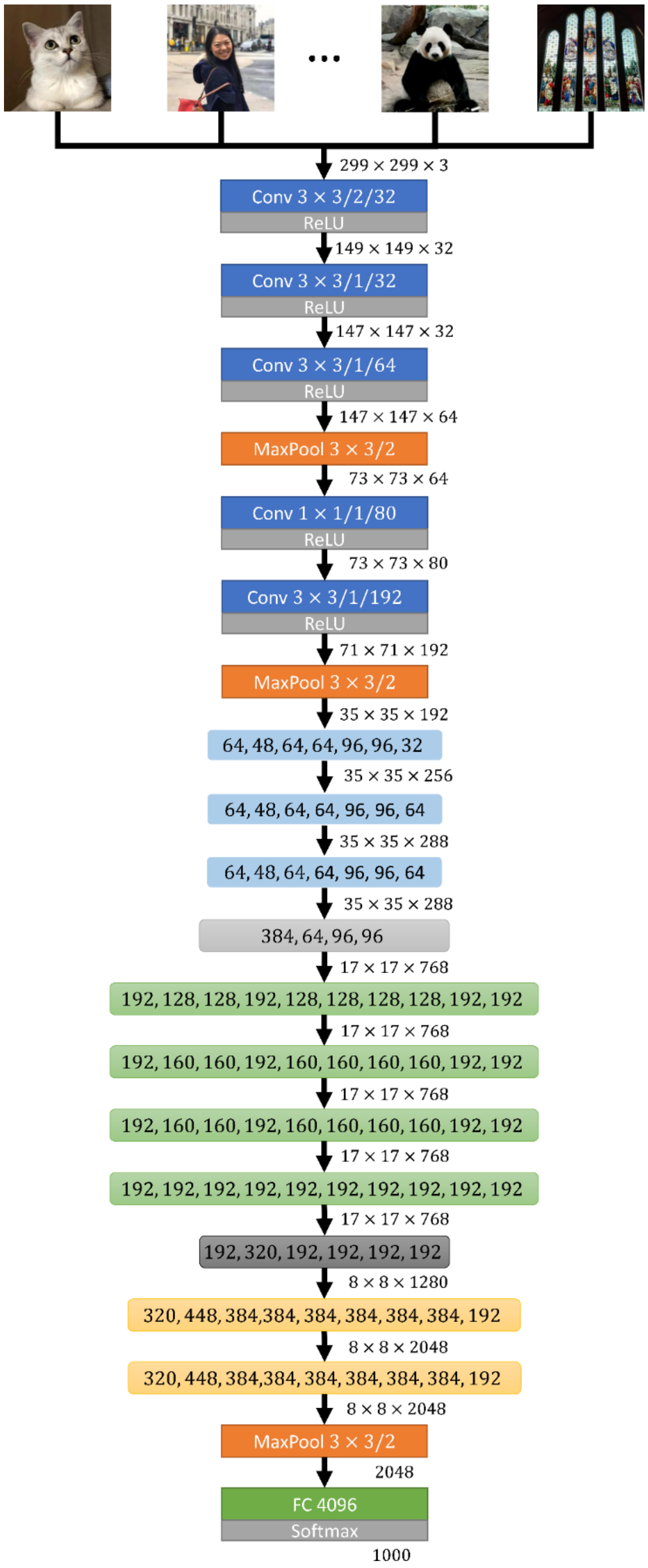


Figure 2.23: Inception-v2 and v3 architecture. Natural images, such as those in ImageNet, are used as an example. Convolutional layer parameters are displayed as kernel size/strides/filters, pooling layer parameters are displayed as kernel size/strides, and output sizes are shown below each layer.

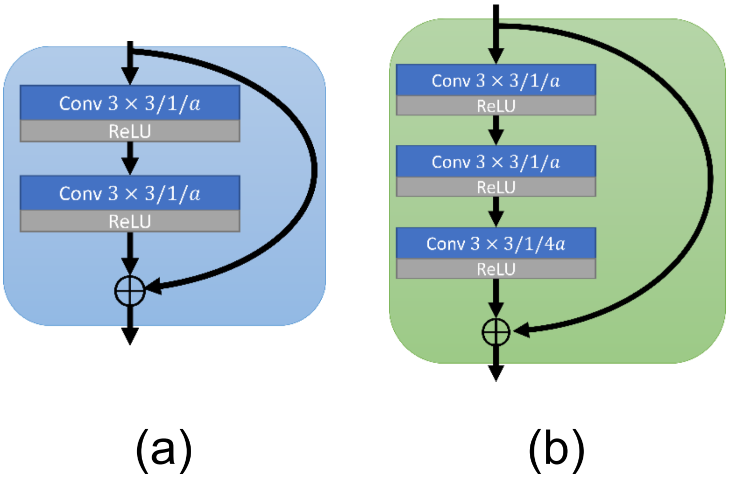


Figure 2.24: ResNet modules showing (a) a skip connection over two layers and (b) a skip connection over three layers.

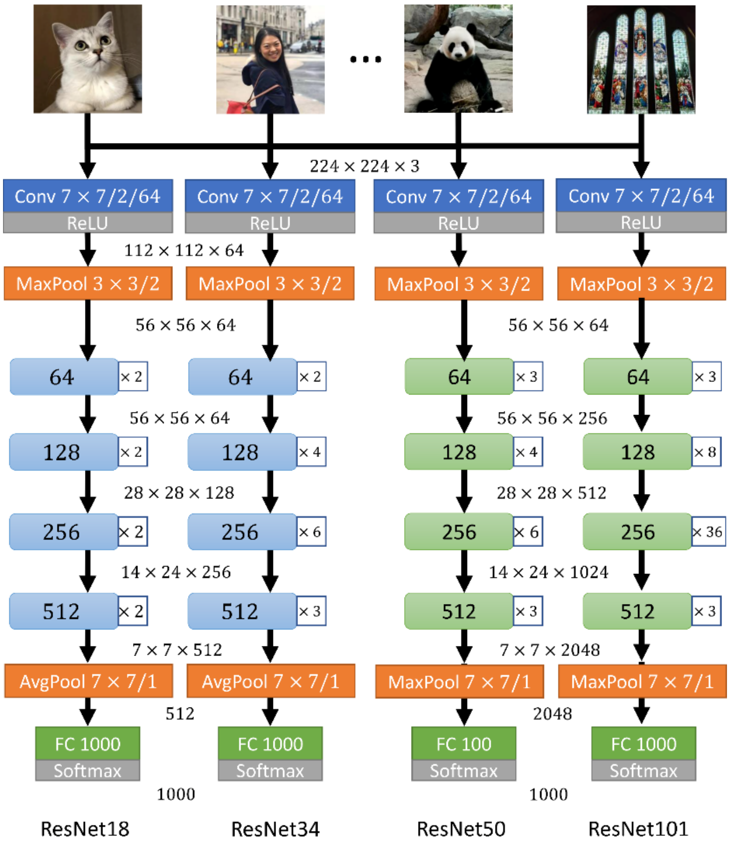


Figure 2.25: Architectures for ResNet-18, ResNet-34, ResNet-50, and ResNet-101. Natural images, such as those in ImageNet, are used as an example. Convolutional layer parameters are displayed as kernel size/strides/filters, pooling layer parameters are displayed as kernel size/strides, and output sizes are shown below each layer.

2.4.3.9 Attention and Transformers

The most recent of the CNNs I have presented here were created in 2015. Since then, there have been advances in CNN architecture, such as NASNet[61] and EfficientNet;[62] however, they had many more parameters than previous architectures. NasNet large has 88.9 million parameters, and EfficientNetB7 has 66.7 million, compared to Inception V3 with 23.9 million; this greatly increases the computational requirements. I chose to use Inception V3 as it is widely used and has reasonable requirements.

Beyond CNN architecture, attention is another area that I could have utilised in my work. A Google paper title “Attention is all you need” sparked interest in attention by showing that a novel architecture called a Transformer using only attention mechanisms could improve language translation tasks.[63] A paper by Woo et al.[64] introduced a convolutional block attention module, which infers attention in both the channel and spatial dimensions. This module could be easily integrated into existing networking architectures and has little additional computational cost. They found improvements in classification across a range of datasets. More recently, architectures solely based on transformers have been developed.[65]

Although much work has been focused on attention, another paper also by Google claimed that “Attention is not all you need” and that attention mechanisms are poorly understood, leading to inductive bias. For this reason, I chose to use Inception V3 only throughout my work as the method I developed could be implemented in other future networks, including those with attention mechanisms.

2.4.4 Loss functions

When training a model and estimating parameters, we aim to minimise the distance between the observed and predicted values. The function used to measure this distance is known as a loss function (or a cost function). It may seem obvious to simply use the accuracy or one minus the accuracy; however, accuracy is not a distance metric and is not suitable for minimising the distance between observed and predicted outcomes. Loss functions are often based on maximum likelihood estimators with the sign reversed.

There are many loss functions to choose from depending on the model used and the type of data being modelled. I will briefly discuss common loss functions and their attributes.

2.4.4.1 Binary cross-entropy

For binary classification tasks, binary cross-entropy is often chosen for the loss function. Binary cross-entropy is based on the maximum likelihood estimator of the Bernoulli distribution. Consider n random variables, $X_1, X_2, \dots, X_n$ independent and identically with the Bernoulli distribution, $X_1, X_2, \dots, X_n \sim Bern(p)$ where

$$X_i(Event) = 1 \text{ and } X_i(No\ Event) = 0. \tag{2.24}$$

The probability mass function (pmf) of X_i is given by

$$p(x) = \begin{cases} p & if\ x = 1, \\ 1 - p & if\ x = 0, \end{cases} \tag{2.25}$$

which can be written as

$$p(x) = p^x(1 - p)^{1-x}. \tag{2.26}$$

The likelihood is given by the joint distribution of the samples $X_1, X_2, \dots, X_n$

$$\mathcal{L}(p) = \prod_{i=1}^{n} p_i^{x_i}(1 - p_i)^{1-x_i}. \tag{2.27}$$

This likelihood function needs to be maximised; however, this is the same as maximising the log-likelihood, which has a slightly easier form to compute

$$l(p) = \log(\mathcal{L}(p)) = \sum_{i=1}^{n} [\log(p_i)\, x_i + \mathrm{lo\,g}(1 - p_i)\,(1 - x_i)]. \tag{2.28}$$

For the binary cross-entropy loss function, this is divided by the total number of samples. Therefore, for an algorithm with n predicted probabilities $p_i \in [0,1]$ for n observed events $x_1, x_2, \dots, x_n$. The binary cross-entropy loss function is given by

$$BCE = -\frac{1}{n}\sum_{i=1}^{n} [\mathrm{x_i}\log(p_i) + (1 - x_i)\log(1 - p_i)]. \tag{2.29}$$

This loss function is used for binary classification or when classes are not mutually independent. For example, a classifier may have "fluffy" and "animal" as two classes; a cat may be classified as both "fluffy" and an "animal".

2.4.4.2 Categorical cross-entropy

When dealing with more than two classes, the generalisation of the Bernoulli distribution to the multiclass case can be used. Let $X_1, X_2, \dots, X_n$ be n random variables independent and identically with the categorical distribution with K possible events, where

$$X(Event\ 0) = 0, X(Event\ 1) = 1, \dots, X(Event\ K) = K. \tag{2.30}$$

The pmf for the categorical distribution for the i^{th} event is given by

$$f(x = i) = p_i. \tag{2.31}$$

Then likelihood function can be written as

$$\mathcal{L}(p) = \prod_{i=1}^{K} p_i^{x_i}. \tag{2.32}$$

Taking the log and reversing the sign, the categorical cross-entropy loss function is given as

$$CCE = -\sum_{i=1}^{K} x_i \log(p_i). \tag{2.33}$$

Categorical cross-entropy is most useful when there are more than two mutually independent classes.

2.4.4.3 Mean squared error

Given observed values x and predicted probabilities p, the mean squared error (MSE) of n samples is

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(p_i - x_i)^2. \tag{2.34}$$

In some of the work presented in this thesis, I choose to use the MSE as the loss as there are several properties of the MSE which make it suitable as a loss function:

(1) Firstly, the MSE is the second central moment of the error and can be decomposed into the bias and variance of the estimator

$$MSE(x) = Bias^2(x) + Var(x). \tag{2.35}$$

This is useful due to the bias-variance trade-off where variance can be reduced by increasing the bias. The bias-variance trade-off can lead to models which overfit and fail to generalise to new unseen data. The MSE aims to reduce both bias and variance simultaneously.

(2) In the case of unidimensional predictions, the MSE becomes the Brier score,[66] which additionally has useful properties. The Brier score can be decomposed into uncertainty, reliability, and resolution[67]. Similarly, it can be decomposed into calibration and refinement. Refinement is closely related to the ROC curve and can therefore be used to measure the model's discrimination. The calibration component can be used to measure statistical calibration. Hence, the Brier score considers both model discrimination and calibration. The importance of both discrimination and calibration is discussed earlier in the chapter.

(3) For linear regression problems, the MSE is convex with only one minimum; this guarantees convergence in some scenarios[28].

However, there is a disadvantage to using the MSE; as the MSE squares the error, larger errors are more heavily weighted than smaller errors which can be undesired in some scenarios[68]. Therefore, the mean absolute error may be more suitable for some applications.

2.4.4.4 Other loss functions

Many more loss functions are used in specific situations, such as segmentation. The focal cross-entropy loss function was proposed by Lin et al.[69] for object detection tasks where there is an extreme imbalance between the foreground and background

$$FL = -(1 - p_t)^{\gamma} \log(p_t)\,, \tag{2.36}$$

where $p_t = xp + (1 - x)(1 - p)$ and γ is the focusing parameter which needs to be tuned using cross-validation. The focal loss becomes equivalent to cross-entropy when $\gamma = 0$.

The hinge loss[70] is most commonly used for classification with support vector machines

$$H(p) = \max(0, 1 - xp)\,. \tag{2.37}$$

The Kullback-Leibler divergence[71] assesses the distance between two probability distributions. The distance is given by

$$KL = x \log\left(\frac{x}{p}\right). \tag{2.38}$$

The K-L divergence is a distance, not a metric and is also known as the relative entropy.

Loss functions designed specifically for image segmentation include the Sørensen-Dice loss[72 73]

$$DL = 1 - \frac{2xp + 1}{x + p + 1}, \tag{2.39}$$

Jaccard loss

$$J_{loss} = 1 - \frac{|A \cap B|}{|A \cup B|}, \tag{2.40}$$

and active contour loss[74]

$$AC = \int_C |\nabla u| ds + \int_\Omega ((c_1 - v)^2 - (c_2 - v\,)^2) u dx, \tag{2.41}$$

where C is the curve, Ω is the domain, u and v are the predicted and ground truth segmentations, and c_1 and c_2 are the energies of the foreground and background.

Both the Sørensen-Dice loss and Jaccard loss can be expressed as scores

$$DS = \frac{2|A \cap B|}{|A| + |B|}, \tag{2.42}$$

where and

$$J_{score} = \frac{|A \cap B|}{|A \cup B|}, \tag{2.43}$$

where A is the prediction and B is the ground truth.

2.4.5 Supervision

There are three basic types of machine learning: supervised, unsupervised, and reinforcement learning. When data is labelled, supervised learning may be used.[25] The deep learning algorithm takes an input and returns a label (or a probability of that label) for the input. For example, X-ray images may be classed as healthy or diseased. When data is unlabelled, unsupervised learning can be used.[28] Unsupervised learning aims to detect patterns and cluster data without those patterns being explicitly labelled by a human. An example of unsupervised learning is in recommendation engines, where items are clustered to recommend similar items without the need for specific labels for those items. Reinforcement learning involves the algorithm interacting with an environment and being rewarded or punished based on its action; the algorithm aims to maximise the reward.

Supervised learning allows us to classify data into known classes; this can be useful when developing an algorithm to predict a certain disease; however, labelling the data for training can be expensive. Two approaches to reducing the cost of labelled data for supervised learning are semi-supervised learning and weakly-supervised learning. Semi-supervised learning combines supervised and unsupervised learning with some labelled and some unlabelled data. The algorithm may be able to better learn the underlying relationship of the labelled data by incorporating the unlabelled data. Semi-supervised learning reduces the cost of annotating data by concentrating on just a few good-quality labels. Weakly-supervised learning instead reduces the cost of annotating the dataset by utilising noisy labels. The annotations can be made by non-experts or made quickly with less care than in supervised learning. There are many reasons for noisy labels, including non-expert annotations, incomplete annotations, and imprecise annotations.

2.4.6 Parameter initialisation

Under certain conditions, such as using the MSE loss in linear regression, the loss function is continuous and convex with a single minimum, as shown in Figures 2.26-2.28; however, the loss function is often much more complex. The loss function often has local minima and plateaus where the algorithm could get stuck. Local minima and plateaus can be avoided by carefully selecting the initial parameter estimates

($\hat{\theta}_0$). An initial parameter estimate closer to the minimum will also reduce the number of iterations needed for convergence.

There are several possible choices for parameter initialisation. Parameters could be initialised to zero, one, or another constant value. Initialising all parameters to the same constant can lead to the model simply learning the same weights. Another option is to initialise the parameters with random values, for example, using the standard normal distribution. Initialising weights to the standard normal distribution can lead to the vanishing gradient problem.

There have been a few initialisers proposed to overcome the vanishing gradient problem. These initialisers choose random distribution parameters based on the number of input units, n_{in} and the number of output units, n_{out}.

The Glorot (also called Xavier)[27], He[44], and LeCun[75] are three popular initialisers. For the Glorot[27] uniform initialiser, the limits are given by

$$limits = \pm\sqrt{\frac{6}{n_{in} + n_{out}}}. \tag{2.44}$$

The limits for the He[44] uniform initialiser are

$$limits = \pm\sqrt{\frac{6}{n_{in}}}. \tag{2.45}$$

Limits for the LeCun[75] uniform initialiser are given by

$$limits = \pm\sqrt{\frac{3}{n_{in}}}. \tag{2.46}$$

These initialisers also have variants based on the normal distribution, all with mean zero and variances $2/(n_{in} + n_{out})$, $2/n_{in}$, and $1/n_{in}$ for the Glorot, He, and LeCun normal initialisers, respectively.

These initialisers appear relatively similar, with only subtle changes in each. The He initialiser has the largest limits for both normal and uniform variants. Unlike the He and LeCun initialisers, the Glorot initialiser depends on the number of input units and output units. When $n_{in} = n_{out}$ the Glorot initialiser becomes the Lecun initialiser, and when $n_{in} < n_{out}$ the Glorot initialiser gives smaller limits than the LeCun initialiser and vice versa.

In this work, pretrained neural networks are utilised for the backbone networks. Pretrained networks have already been applied to classification problems. The parameters are already set for some feature extraction and just need to be fine-tuned for my particular task.

2.4.7 Optimisation algorithms

Stochastic gradient descent (SGD) is one of the simplest optimisation algorithms used in deep learning[28]. Given initial parameters w_i the updated parameters are calculated as

$$\theta_{i+1} = \theta_i - \eta G, \tag{2.47}$$

where η is the learning rate, and G is the gradient of the loss function $G = \nabla L(\theta)$. The learning rate is a hugely important parameter and should be chosen carefully so that the algorithm can converge to optimal parameters.

This subsection discusses the importance of selecting a suitable learning rate. I then introduce momentum, which can help to avoid local minima. Finally, I briefly outline several other optimisation algorithms, which all build upon each other.

2.4.7.1 Learning rate

As previously discussed, the gradient of the loss function is used to update the model parameters at each step. The most common method used to calculate the loss function gradient is backpropagation[76]. An optimiser uses the gradient to iteratively select parameters which minimise the loss.

Although other methods have been proposed, gradient descent and its various extensions account for most optimisation algorithms currently used. For simplicity, the loss function can be imagined as a U-shaped curve on a graph of parameter values vs loss function. The size of the step taken is called the learning rate.

The choice of learning rate is crucial to ensure the algorithm converges in a reasonable amount of time. A learning rate that is too big could miss the minima and may even diverge, as demonstrated in Figure 2.26. A learning rate that is too small will take too long to converge; this can be expensive and time-consuming, especially when large computationally complex models are used; this is demonstrated in Figure 2.27. A correct learning rate converges to the minima quickly, as displayed in Figure 2.28.

The optimal learning rate will likely be different at different stages of training. Several learning rate scheduling algorithms have been proposed to alter the learning rate during training. One option is to reduce the learning rate when the performance plateaus. For example, the learning rate could be halved if the loss does not reduce for three epochs.

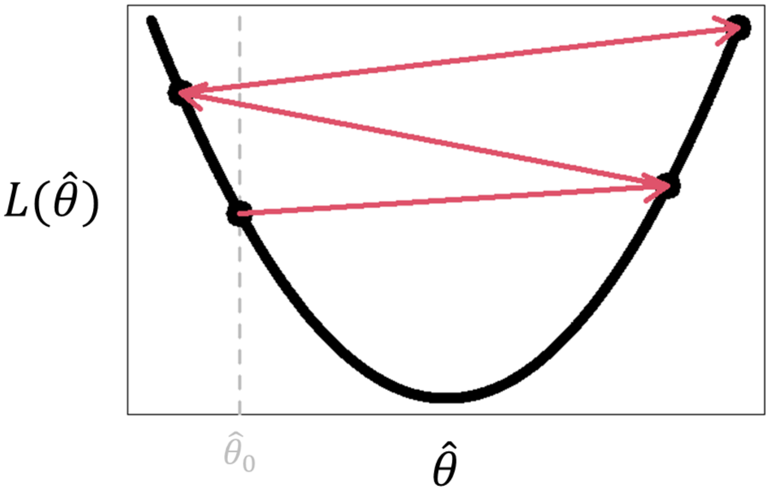


Figure 2.26: The learning rate is too high; the algorithm diverges from the minimum.

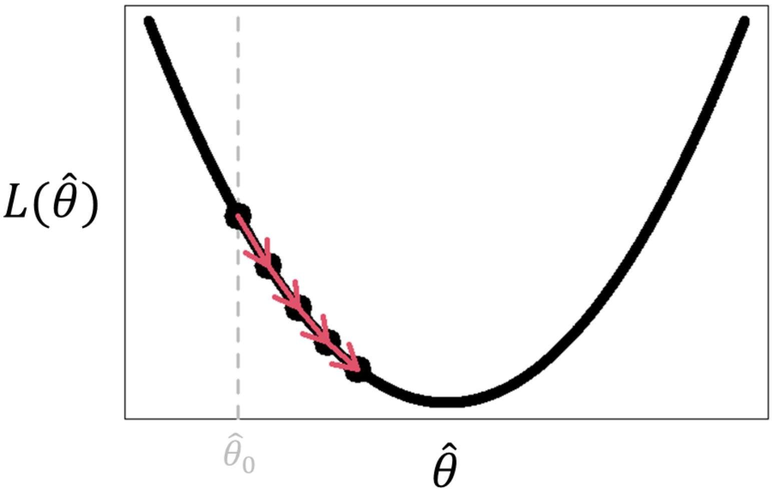


Figure 2.27: The learning rate is too low; the algorithm takes too long to converge.

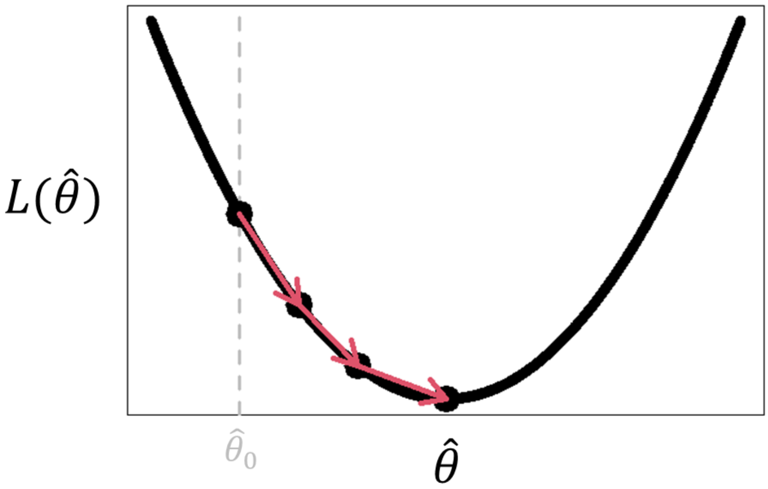


Figure 2.28: The learning rate is well-chosen and converges to the minimum within a few iterations.

2.4.7.2 Momentum

The algorithm does not consider the previous gradients when gradient descent takes a step. Momentum can be used to consider these previous gradients and helps avoid local minima.[77] Simple momentum is given by

$$\theta_{i+1} = \mu\theta_i - \eta G, \tag{2.48}$$

where μ is the momentum parameter often chosen to be 0.9. An alternative to classical momentum is Nesterov momentum[78], where the gradient is calculated as

$$G = \nabla L(\theta + \mu\theta). \tag{2.49}$$

Momentum avoids local minima and can speed up training by considering the previous gradient when choosing the next step. The downside to momentum is that an extra hyperparameter (η) is introduced; however, a value of 0.9 often gives good results.

2.4.7.3 AdaGrad

While SGD follows the steepest gradient at each step, adaptive gradient (AdaGrad) considers the earlier gradients to learn the general geometry[79]. The previous gradients are incorporated into the updated parameter estimates by averaging the previous squared gradients

$$S_{i+1} = S_i + G_i \otimes G_i, \tag{2.50}$$

where $\otimes$ is the element-wise multiplication operator.

The parameter estimates are then updated with

$$\theta_{i+1} = \theta_i - \frac{\eta G}{\sqrt{S_{i+1} + \epsilon}}, \tag{2.51}$$

where ϵ is added to avoid division by zero. It may be noticed that AdaGrad is equivalent to SGD with a learning rate of $\frac{\eta}{\sqrt{S_{i+1}+\epsilon}}$. This can be considered an adaptive learning rate, with the current steps learning rate based upon the gradient of the previous steps. The following optimizers discussed in this section all build upon AdaGrad and, therefore, can also be thought of as having adaptive learning rates.

The choice of learning rate is essential; however, with AdaGrad, the impact of a slightly suboptimal learning rate is reduced. This was a significant advancement for optimisation algorithms, reducing the time spent tuning the initial learning rate. Unfortunately, the learning rate is often reduced too quickly in deep learning, and the algorithm fails to converge to the local minimum.

2.4.7.4 RMSProp

Root mean squared propagation (RMSProp) improves upon AdaGrad by decaying the older gradients while giving more weight to the more recent gradients.[80] The weighted squared gradients are calculated as

$$S_{i+1} = \beta S_i + (1 - \beta) G_i \otimes G_i, \tag{2.52}$$

where β is the momentum.

Then the parameter updates for RMSProp are

$$\theta_{i+1} = \theta_i - \frac{\eta G}{\sqrt{S_{i+1} + \epsilon}}. \tag{2.53}$$

This was a significant improvement upon AdaGrad and enabled adaptive gradients to be used on deep learning problems. The introduction of adaptive gradients to deep learning was so significant that it became one of the major changes in Inception-v3.

2.4.7.5 Adam

RMSProp enabled deep learning to benefit from adaptive gradients, while momentum helps the algorithm to converge faster. Adaptive momentum (Adam) combines RMSProp with momentum[81]. Given chosen decay rates β_1 and β_2, the biased momentum is

$$m_{i+1} = \beta_1 m_i + (1 - \beta_1)G, \tag{2.54}$$

and the unbiased momentum is

$$\widehat{m}_{i+1} = \frac{m_{i+1}}{1 - \beta_1^i}. \tag{2.55}$$

The biased squared gradient is calculated as

$$S_{i+1} = \beta_2 S_i + (1 - \beta_2)G, \tag{2.56}$$

and the unbiased squared gradient is

$$\hat{S}_{i+1} = \frac{S_{i+1}}{1 - \beta_2}. \tag{2.57}$$

Then the parameter updates are given by

$$\theta_{i+1} = \theta_i - \frac{\eta \widehat{m}_{i+1}}{\sqrt{\hat{S}_{i+1} + \epsilon}}. \tag{2.58}$$

2.4.7.6 Adamax

An extension to Adam based on the infinity norm has been proposed[81]. Similar to Adam, Adamax requires two decay parameters, β_1 and β_2, the momentum is then calculated as

$$m_{i+1} = \beta_1 m_i + (1 - \beta_1)G. \tag{2.59}$$

The weighted infinity norm is then calculated using

$$S_{i+1} = \max(\beta_2 S_i, |G|)\,. \tag{2.60}$$

The learning rate is updated to

$$\eta_{i+1} = \frac{\eta_i}{1 - \beta_1^i}. \tag{2.61}$$

Finally, the parameter estimates are updated using

$$\theta_{i+1} = \theta_i - \frac{\eta_{i+1}\, m_{i+1}}{S_{i+1} + \epsilon}. \tag{2.62}$$

2.4.8 Explainable AI

Explainability is a hot topic in AI. Deep learning algorithms are often black-box in nature; an input is given, and an output is returned without explanation for that output. For example, when using an image, the user of an algorithm may want to know precisely what areas of the image the algorithm is using to justify the output.

Models that automatically extract features could potentially extract the wrong features, finding a pattern that "cheats" by looking at something that does not tell us anything about the outcome. One example of this is chest x-ray imaging, where patients are diagnosed as having or not having a lung condition. Patients with lung conditions may already need some intervention, such as ventilation or a chest drain; this equipment may be visible on the x-ray. A deep learning algorithm could potentially see this equipment and use that to classify patients instead of using the radiographical features. Model visualisation and explainability are vital in deep learning. Several methods exist that allow us to see what the algorithm looks at, such as class activation maps (CAMs), saliency maps, and Shapley values.

In this thesis, I use saliency maps, in particular SmoothGrad.[82] When using images, the output could be explained by finding which pixels in the image most strongly influence the algorithm. One common method to identify these important pixels is to generate a saliency map using gradients.

I use binary classification as an example in this section, but the method extends to multiclass classification. Given an input image I, an algorithm computes an activation function F. If the activation is piecewise differentiable, then a saliency map can be constructed using the gradient of the activation function of the image

$$S(I) = \frac{\partial}{\partial x} F(I). \tag{2.63}$$

In simple terms, this saliency map displays how much a small change in each image pixel would influence a change in the output.

Using gradients in this way often results in noisy saliency maps. SmoothGrad reduces this noise by adding noise to the image and calculating the average of many saliency maps. The SmoothGrad saliency map, S_{SG}, is therefore given as

$$S_{SG}(I) = \frac{1}{n}\sum_{i=1}^{n} S\big(I + N(0, \sigma^2)\big), \tag{2.64}$$

Where n is the number of saliency maps to be averaged and $N(0, \sigma^2)$ is the Gaussian distribution with mean 0 and variance σ^2. Due to the averaging of the multiple saliency maps, SmoothGrad results in less noisy saliency maps, which improves the interpretability and explainability of the deep learning algorithm.[82] The idea behind SmoothGrad is simple but justifies why the method works in theory. The lack of quantitative performance measures may appear to be a major issue with the paper; however, as the authors explain, quantifying such a subjective problem is impossible. Nonetheless, qualitative results in the paper and subsequent studies back up the method, showing the apparent improvement that SmoothGrad provides with much clearer, more focused saliency maps. A large-scale study asking users to rank which qualitative method is best may provide some quantification of whether the method is an improvement; however, this inevitably would have its own biases.

The final maps produced by SmoothGrad often produce clearer saliency maps as the random noise typically seen in saliency maps is averaged out. This simple trick provides a good solution to the problem of random noise in saliency maps; however, computing multiple saliency maps can greatly increase the computation time.

The interpretation of these visualisations can be challenging. Oramas *et al.*[83] combined the explanation and interpretation of deep learning models by identifying relevant features for each class and averaging the visualisations of that set of features. During inference, the prediction is then presented along with the visual features used to make the prediction.

Most methods rely on gradients or intermediate features, another approach proposed by Li *et al.,*[84] uses a slot-attention based method. The fully-connected layer is replaced with a SCOUTER layer which contains an attention module for each class. A confidence is then obtained for each class. Using a novel SCOUTER loss, the model can explain why it chose one class and also why it did not choose another class.

McCoy *et al.*[85] argue that explainability is not confined to AI algorithms and is actually part of a larger problem of treatment choice explainability in general. They conclude that explainability should not be sought uncompromisingly and that robust evaluation should be the main aim.

Although explainability is undoubtedly essential in AI, it is not the focus of this thesis. Throughout the work I present here, I will use SmoothGrad as the maps are easy to produce and interpret.

2.5 Age-related macular degeneration (AMD)

I use age-related macular degeneration (AMD) as an example to demonstrate and evaluate the methods presented in this thesis.[86] Images of AMD that I used were taken from the age-related eye disease study (AREDS). In this subsection, I first give a general background of AMD and the risk factors associated with AMD. I then discuss colour fundus photography, the imaging modality that I use in this thesis, before discussing the features of AMD visible on colour fundus images.

2.5.1 Clinical features and risk factors

AMD is a degenerative retinal condition and a leading cause of sight-loss worldwide. Damage is caused to the centre of the retina, which results in blurring and distortion of the central vision.[87] AMD is classified into early (sometimes referred to as age-related maculopathy because vision is not yet affected), intermediate, and late. Patients with early or intermediate AMD are at risk of developing late-stage or advanced AMD, resulting in severe loss of central vision.[86] Around 196 million people are estimated to be living with AMD globally, with around 10.4 million of them suffering with vision impairment from end-stage AMD.[87 88] There are two forms of late-stage AMD, atrophic (dry) and neovascular (wet). Repeated injections of anti-vascular endothelial growth factor agents are an effective treatment for neovascular AMD; however, no treatment currently exists for atrophic AMD.[87 89 90] Prognostic models predicting progression to end-stage AMD may help plan treatment for neovascular AMD and rehabilitation for atrophic AMD.

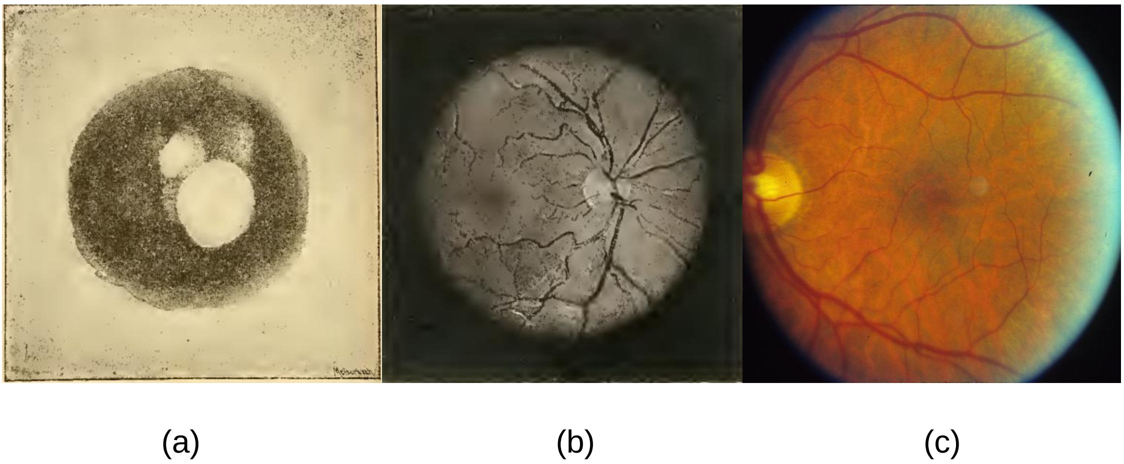
There are several clinical risk factors associated with AMD. A systematic review by Chakravarty et al.[11] identified commonly reported risk factors with varying strengths of evidence. Age, smoking status, cataract surgery, and a family history of AMD were strongly associated with developing AMD. Body mass index (BMI), cardiovascular disease, hypertension, and higher plasma fibrinogen showed a moderate association. Other risk factors with weaker and more inconsistent associations included gender, ethnicity, diabetes, iris colour, cerebrovascular disease, and cholesterol levels. They concluded that only smoking, cataract surgery, and a family history of AMD were consistent risk factors. This systematic review highlighted the inconsistency in the reporting of AMD. Overall, the systematic review was well-conducted. The authors of the review conceded that they did not consider genetic or dietary factors; these factors are considered important in AMD. In their meta-analysis, the authors did consider whether the studies accounted for confounding; however, the method used to account for confounding was not assessed. Some studies may have reported that they adjusted for confounding variables but did not use appropriate methods. This highlights an issue with aggregate data meta-analysis, which may be solved using individual participant data meta-analysis.[91]

The AREDS identified several risk factors for progression to late-stage AMD while controlling for age, gender, and treatment group. Factors associated with neovascular AMD included ethnicity and smoking. Factors associated with atrophic AMD included education, BMI, smoking, and the use of antacids. Cataract surgery has been associated with the risk of developing AMD; however, cataract surgery is not associated with the risk of progressing to advanced forms of AMD.[92]

2.5.2 Colour fundus photography

The imaging modality used for investigating AMD in this thesis is colour fundus photography. In 1851, Hermann von Helmholtz created his Augenspiegel or ophthalmoscope, as it is known in English. This invention allowed clinicians to view the interior surface of the posterior eye, known as the fundus. Coincidentally, Frederick Scott Archer invented the collodion process for photography in the same year. Ten years later, James Clerk-Maxwell produced the first colour photograph. These advances in optics and photography allowed the development of colour fundus photography.

Jackman and Webster produced the first successful image of a living human retina in 1886.[93] Limited by exposure time and visible light, the image is basic, with only the optic disc and the largest vessels visible. Although these images appear unimpressive by today's standards, they were incredibly exciting advances for the time. Due to further advances in photography, Friedrich Dimmer obtained much more detailed images in 1907, with an exposure time of as little as 10 seconds.[94] Modern fundus photographs are coloured and show much more detail. Figure 2.29 shows how fundus photography has progressed. An example of a colour fundus image showing the main geography of a colour fundus photograph is shown in Figure 2.30. A cross-sectional diagram of the eye showing its main anatomical features is shown in Figure 2.31.



(a) (b) (c)

Figure 2.29: (a) Fundus photograph obtained by Jackson and Webster in 1886. (b) Fundus photograph obtained by Dimmer in 1907. (c) Colour fundus photograph taken from the AREDS dataset.

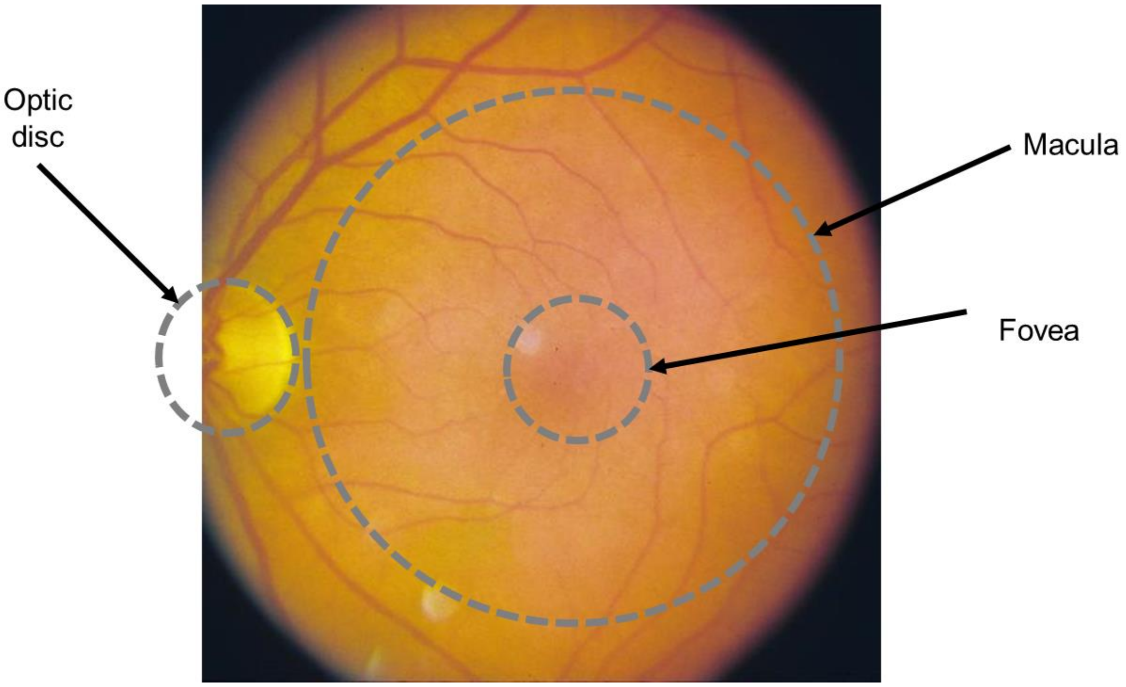


Figure 2.30: Example of a colour fundus photograph highlighting the main features. This image is taken from the AREDS dataset.

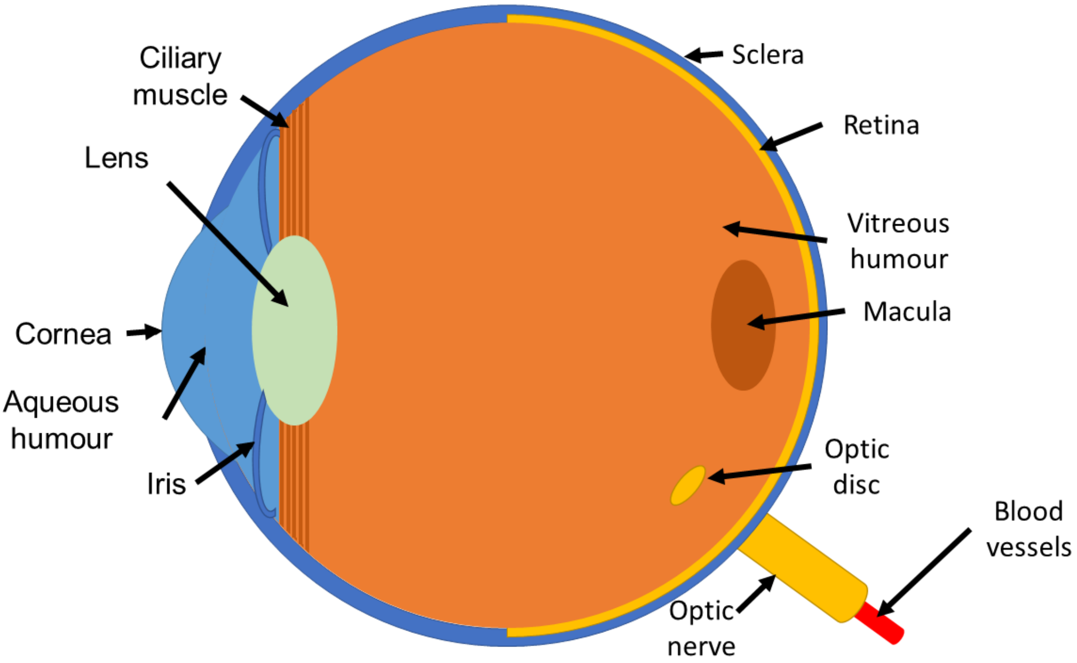


Figure 2.31: Cross-sectional diagram of the eye showing the main features.

2.5.3 AREDS

The AREDS and AREDS2 were clinical trials investigating the risk factors of AMD and cataracts. These large-scale longitudinal studies evaluated the effect of vitamins on the progression of age-related eye diseases. The study was initially prompted by the widespread use of vitamins in the United States to treat AMD and cataracts without clear evidence of their efficacy and safety.[95] A total of 4,757 participants aged 55 to 80 years old were recruited for the study. Large amounts of demographic and clinical data were collected as well as colour fundus photographs. The photographs used in AREDS are stereoscopic to give a 3D effect; however, only the right stereoscopic image is used in this thesis.

The main aim of AREDS was to assess the benefit of vitamins and supplements in patients with age-related eye conditions. The first AREDS found that the odds of developing advanced AMD in high-risk groups could be significantly reduced with antioxidants and zinc, with an odds ratio of 0.72 (95% CI: 0.52, 0.98) compared to a placebo.[96] This mix of vitamins became the original AREDS formulation. AREDS2 expanded on this work and found that adding lutein, zeaxanthin, DHA, and EPA to the original AREDS formulation gave no statistically significant reduction in the risk of progressing to advanced AMD over the AREDS formula alone, with a hazard ratio of 0.89 (98.7% CI: 0.75,1.06). The authors concluded that the AREDS2 formula did not further reduce the odds of developing advanced AMD; however, lutein and zeaxanthin could replace carotenoids to reduce the risk of lung cancer in former smokers.[97]

AREDS also examined the features of AMD identified on colour fundus imaging; I outline these in the next section. Another primary outcome of AREDS was a set of severity scores, which I discuss in Section 2.5.5.

AREDS and AREDS2 have led to important conclusions either through reports published by the AREDS group or through external studies using the data collected by AREDS; however, there are some significant issues with the studies. The AREDS2 study population was enrolled from US clinical centres and consisted of 97% white participants.[98] As the efficacy of vitamin supplements depends greatly upon genetic factors;[99] the findings are limited to a white US population. Additionally, participants were relatively well-nourished compared to the general population.[96 100] Report number 8 of AREDS acknowledges that the study population may differ from the general population and the effect on the generalisability of the results is unknown. Any conclusions made from the AREDS data may only apply to a well-nourished, white, elderly US population. The AREDS2 formula is built upon the original AREDS formula. As it had already been shown that the original AREDS formula significantly reduced the odds of progression to advanced AMD, the AREDS2 group determined it was unethical to have a group of participants taking no vitamin supplements. As a result, AREDS2 had no true control group.[101] The original AREDS study reported no statistically significant increase in the risk of adverse events; however, AREDS2 found that the original formulation may increase the risk of developing lung cancer in former smokers. It is unclear why the original study did not identify this increased risk for former smokers. Additionally, AREDS did not meet its original endpoints, and the conclusions were based on unplanned post-hoc sub-group analyses. This adds considerable bias to the results obtained in the original study.

Despite the shortcomings of AREDS and AREDS2, the data and images collected by these studies form one of the most extensive publicly available longitudinal imaging datasets. For this reason, I chose to use the AREDS data to demonstrate the methods I have developed while acknowledging the limitations.

2.5.5 Imaging features of AMD

Several features of AMD are visible on colour fundus imaging as identified by the AREDS group.[102] These features can be classed into three broad types:

(1) Firstly, there are retinal elevations caused by retinal detachment. There are three categories of retinal elevation, serous sensory retinal detachment, retinal pigment epithelial detachment, and drusenoid pigment epithelial detachment. If a pigment epithelial detachment is dome-shaped, this suggests a serous pigment epithelial detachment. Shallow pigment epithelial detachments can be either serous or fibrovascular. Finally, irregular pigment epithelial detachments are probably fibrovascular.

(2) Secondly, retinal pigment epithelial abnormalities may be observed. These include geographic atrophy, depigmentation (hypopigmentation), and increased pigmentation (hyperpigmentation). Geographic atrophy is characterised by a sharply demarcated area of the retinal pigment epithelium depigmentation. This area is circular or scalloped and, to meet the definition, should be greater than one-eighth the diameter of the optic disc. Areas of

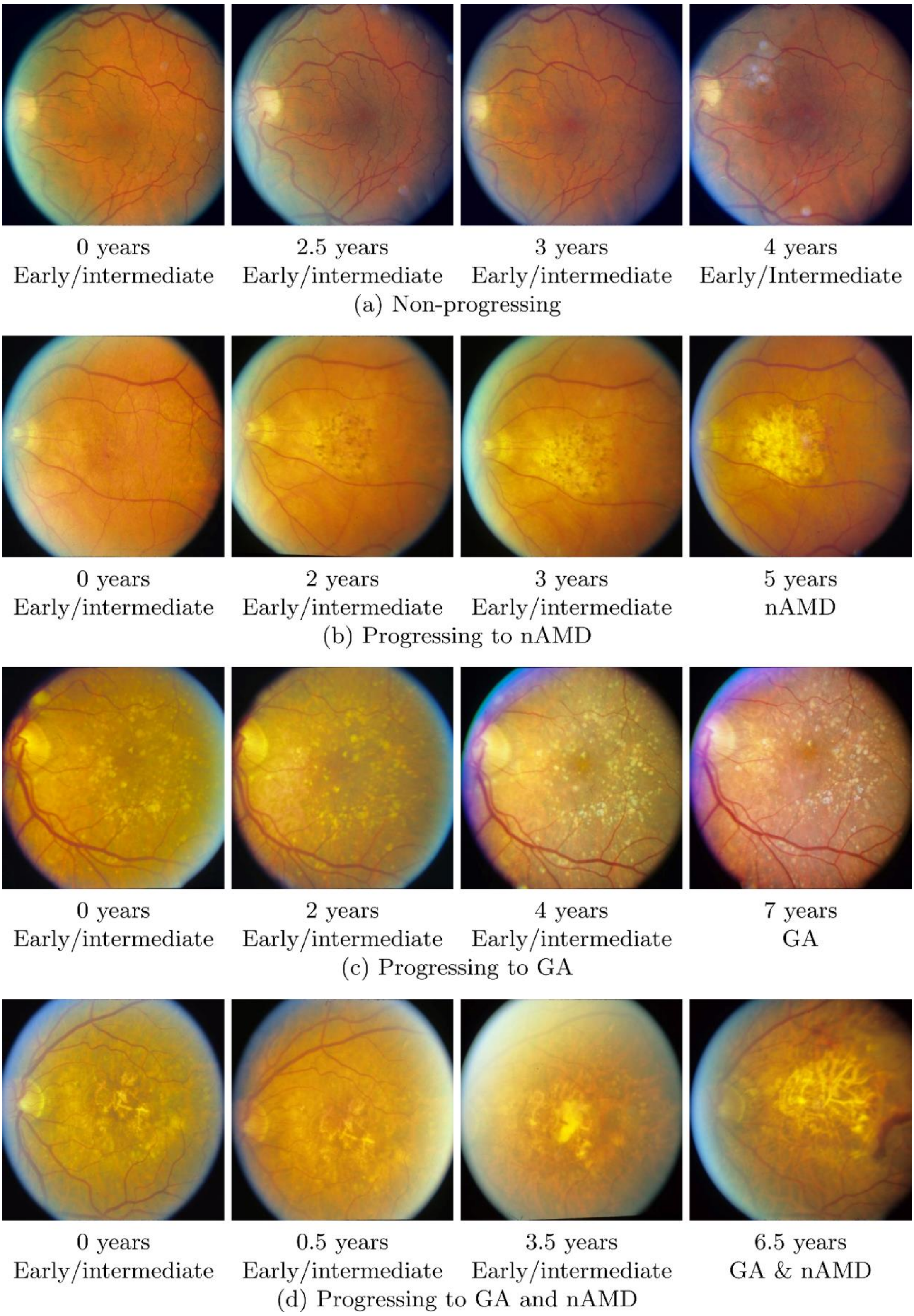
pigmentation that are smaller than this, non-circular in shape, or appear around a subretinal fibrous scar are classified as depigmentation rather than geographic atrophy. Increased pigmentation appears as clumps of grey or black pigment beneath the retina.

(3) Thirdly, yellow-whitish lipid deposits under the retinal pigment epithelium are called drusen. These drusen can vary greatly in shape and size. Some drusen may be small, round, and flat, while others are large, irregular, and thick. Drusen are classified by size, hardness, and area covered. Drusen can be small ($< 63\mu m$), intermediate ($\geq 63\mu m$ and $< 125\mu m$), or large ($\geq 125\mu m$).

AREDS classified AMD into three main categories early, intermediate, and advanced.[103] Early AMD is defined as having multiple small or intermediate drusen with no advanced AMD. Intermediate AMD is defined as extensive intermediate or large drusen with no advanced AMD. Advanced AMD is defined as having developed either geographic atrophy, neo-vascular disease, or both.

Based on these categories, AREDS created a four-step AMD severity scale.[102] The first three stages are early/intermediate AMD. The fourth stage requires geographic atrophy (GA) in the central subfield (atrophic AMD), neovascular AMD, or both. Evidence of neovascular AMD includes fibrovascular pigment epithelial detachment, subretinal pigment epithelial haemorrhage, subretinal fibrous tissue, or photocoagulation.

Other studies have also identified further changes caused by AMD in the optic disc. A study by Law et al. found that eyes with large areas of AMD were more likely to be classified as glaucomatous than eyes with smaller areas of AMD.[104] This study identified that optic disc changes caused by AMD can resemble glaucoma, making them difficult to identify, especially in patients with glaucoma. Optic disc pallor has also been associated with AMD, even in early-stage patients.[12] Examples of how a patient may progress are shown in Figure 2.32.



(a) Non-progressing

(b) Progressing to nAMD

(c) Progressing to GA

(d) Progressing to GA and nAMD

Figure 2.32: Example longitudinal images of age-related macular degeneration. The first three images of each patient profile display early or intermediate AMD. The fourth image of (a) shows the patient does not progress, while (b), (c), and (d) show the patient progressing to advanced forms of AMD. These images are from AREDS.

2.6 COVID-19 diagnosis

In Chapter 5, I demonstrate a novel part of my final prognostic model by using CT imaging taken during the COVID-19 pandemic. Although this is not directly related to my main aims, it is vital for me to assess the added benefit of the new method. There are three main methods for the diagnosis of COVID-19, these are lateral flow testing, RT-PCR, and radiographic assessment (CT, for example). A systematic review of lateral flow testing concluded that it has a high specificity, but the sensitivity can be highly variable, with values as low as 38.32% and as high as 99.19%.[105] This sensitivity is found to be larger for symptomatic patients than for asymptomatic.[106 107] A study examining the real-life clinical sensitivity of RT-PCR described the clinical sensitivity of RT-PCR as 'moderate at best',[108] with a sensitivity of 47.3% (95%CI: 44.4, 50.3). From early in the pandemic, CT imaging has been considered the gold-standard for the diagnosis of COVID-19,[109] with a higher sensitivity than RT-PCR.[110]

2.6 Prognostic models

In 2013, a partnership of researchers met to unify prognosis research into a coherent framework. The prognosis research strategy (PROGRESS) framework was formed to improve the standard of prognostic research. Before the PROGRESS framework was developed, the terminology was often inconsistent, with poor reporting standards. PROGRESS identified four types of prognosis research:[1]

(1) Overall prognosis[111]
(2) Prognostic factors[112]
(3) Prognostic models[7]
(4) Predictors of treatment effect[113]

Overall prognosis research aims to estimate the likely outcome of a disease for a group of people. For example, uveal melanoma is a form of cancer that develops in the choroid of the eye and can metastasise to other areas, such as the liver. Only around 70% of patients survive for five years following a clinical diagnosis.[114] This informs us about the overall likely outcome for a group of patients and can be used to make healthcare decisions at a population level. For survival rates, this may mean decision-makers can allocate treatment resources based on the current number of patients diagnosed with the cancer.

Prognostic factor research aims to find what factors are associated with increased risk. For example, tumour thickness and proximity to the optic disc are associated with an increased risk of eye loss in uveal melanoma.[115] These factors can be used to identify most at-risk individuals and develop effective treatments that target that factor.[1]

Prognostic model research uses the identified prognostic factors to develop models that estimate the risk of a particular outcome in individual patients. In contrast to overall prognostic research, prognostic model research aims to predict the future outcome for each patient rather than for a group of patients. The importance of using observed factors to predict the likely outcome of a disease in an individual patient dates back as far back as Hippocrates.[116] For example, the Liverpool uveal melanoma prognosticator online (LUMPO) was developed to predict metastatic death in patients diagnosed with uveal melanoma. The LUMPO model is a semi-parametric Markov multi-state model and includes demographic, clinical, and genetic data as covariates. This model achieved a C-index of 0.862 (bootstrapped 95% confidence interval [0.84, 0.88]), and calibration curves show reasonable performance. A multicentre external validation study of LUMPO reported a pooled C-index of 0.72 (0.68, 0.75). Calibration curves suggested that recalibration may be needed for some clinical settings.

The final type of prognostic research aims to assess how treatments affect the prognosis in individual patients. This type of research investigates why some patients benefit from treatment while others do not by identifying the factors associated with improved treatment outcomes. For example, younger age is strongly associated with improved outcomes using anti-vascular endothelial growth factor (anti-VEGF) treatments for AMD.[117] These studies help prevent unnecessary treatment for patients who will not benefit from it. The findings of these studies may be helpful but may not necessarily change clinical practice; for example, although younger patients have a greater benefit from anti-VEGF treatment, older patients still benefit.

In this thesis, I focus on the third type of prognostic research, prognostic model development. Most prognostic models are developed using traditional statistical techniques; however, prognostic models are starting to be developed using more novel machine learning methods, especially deep learning. The main advantage of machine learning is its ability to automatically incorporate varied data sources, such as images and unstructured text, into the model with many predictors. However, these also provide disadvantages as the algorithm's output needs to be interpreted to ensure that the algorithm is making sensible predictions.

In this section, I review and critically appraise the published literature relevant to the methods presented in this thesis.

2.6.1 Traditional statistical prognostic models

The majority of prognostic models are developed using traditional statistical methods. These methods are well established and are suited to using easily extractable data such as demographics and clinical characteristics. In addition, these models are easily interpretable and can often be given as a written equation.

One of the oldest prognostic models, still in use today, is the Apgar score[2]. The Apgar score was developed to assess the well-being of newborn babies. While the Apgar score has been found to be inadequate for assessing the risk of mortality on the individual level, it can predict an increase in relative risk for cerebral palsy.

Another well-known example is the Framingham risk score. This model is used to predict the risk of developing cardiovascular disease in community settings.[118] The Framingham risk score has been extensively validated; however, it has been found to overestimate the risk of developing cardiovascular disease.[119] Therefore, recalibration of the model may be needed.

Some prognostic models have been adopted into national guidelines. For example, the global registry of acute coronary events (GRACE) risk score[120] [121] is now recommended in NICE guideline 185 to assess the risk of future cardiovascular events.

In the context of AMD specifically, a few models have been developed using traditional statistical methods. AREDS Report 17 developed a severity scale for the risk of developing advanced AMD.[122] A 6-step drusen scale and a 5-step pigmentary abnormality scale were combined into a 9-step scale to estimate the 5-year risk of progressing to advanced AMD. Probabilities were obtained by calculating the percentage of patients who had progressed in each step group. Patients in step group 1 are assessed as having a 0.3% probability of progression, and patients in step group 9 are estimated to have a 53.2% probability of progression in 5 years.

The original gradings were compared with replicate gradings from other clinicians to display the reproducibility of the scale. The original and replicate gradings agreed in 63.4% of eyes, with 63.4% agreement within one step and 93.6% agreement within two steps. The original and replicated scales also had an unweighted κ score of 0.58 with a standard error (SE) of 0.015. This κ score can be interpreted as a moderate strength of agreement.[123] This 9-step score is difficult to use and unsuitable for routine clinical examinations.[124] Therefore, a second simplified score was developed and presented in AREDS Report 18.[125]

The simplified score is given on the patient level rather than the eye level. Each eye is given one risk factor if there is one or more large drusen and another point if there are any pigment abnormalities. Summed across both eyes, the patient has between zero and four risk factors. Each patient’s estimated probabilities of progression are given as 0.5%, 3%, 12%, 25%, and 50% for zero to four risk factors, respectively. This simplified scale is much easier to calculate and understand, making it more usable in clinic. As part of AREDS2, the probabilities for the full and simplified severity scores were recalculated on the new data collected.[126] The AREDS and AREDS2 5-year rates did not differ significantly, suggesting that the severity scale does not differ significantly between studies.

There are several significant problems with the AREDS severity score, which reduce its usefulness. Firstly, the highest risk score only gives a 50% probability of progression. This may make it challenging to prioritise patients because even the most at-risk patients still only have a 50% probability of progression. Secondly, the probability of progression is a continuous outcome; categorising these continuous outcomes like this is inefficient and unnecessary.[127] Categorising continuous predictors leads to a loss in information and power and results in models with poor predictive performance and clinical usefulness.[128] Finally, the probabilities are presented as probabilities of progression for individual patients; however, the probabilities are the percentages of patients in each risk score group who have progressed. Under the PROGRESS framework, this is the first type of prognostic research which informs us about the proportion of patients progressing at the population level. For the simplified score, 50% of patients on step 4 progress to advanced AMD; this does not inform us about the probability of progression of individual patients. This type of model may still be useful but should not be used on the individual patient level.

Other models have been developed that are more suited to predicting individual patient-level progression. Many of these models are developed using data from AREDS as it is the largest available dataset with vast amounts of demographic, clinical, imaging, and genetic data available.

Seddon et al.[129] developed a model using Cox proportional hazards.[130] Final predictors included genetic factors, environmental factors, and drusen size. Data was taken from AREDS, with 819 of 2937 included patients identified as progressing to advanced AMD. For five-year predictions, the best model attained an AUROC of 0.876 (SE=0.012); for ten-year survival, the AUROC was 0.908 (SE=0.009) in the test sample.

Another similar model using a survival analysis approach used ten genetic loci, age, sex, education, BMI, smoking status, and AMD status to predict AMD progression.[131] Data in this study were also taken from AREDS, with 834 of 2951 patients identified as progressing. The final model combining environmental and genetic factors achieved an AUROC of 0.911 for the 10-year prediction. The authors noted that the inclusion of genetic factors led to significantly improved predictions.

A logistic regression model has also been proposed to predict AMD progression[132]. This study used data from a Korean population consisting of 10,890 patients aged over 50 years. Early AMD was found in 318 patients, and 157 of these were followed up for 4.4 years. Final model predictors included: drusen characteristics, hyper- and hypo-pigmentation, sex, age, smoking status, protein levels, and globulin levels. This model achieved an AUROC of 0.84 (95% CI: 0.75, 0.92).

2.6.2 Two-stage models

When features of progression can be observed, it is possible to use deep learning to extract features and then use a traditional statistical model. Four of the five models for AMD progression mentioned in the previous section include drusen characteristics as predictors.[122 125 129 132] However, these characteristics can be difficult and time-consuming to quantify and extract manually. One way to overcome this problem is to automatically extract features using a deep learning or similar method and then pass the extracted predictors to a deep learning model. Although deep learning may be used to extract the predictors, the prognostic model is still a traditional prognostic model. These two-stage methods have been used in diverse fields such as neurology[133] and cardiology.[134]

One model used this approach[135] with spectral-domain optical coherence tomography (SD-OCT) for AMD progression. OCT is an imaging modality that uses light waves to create a cross-sectional image of the retina. Images can be taken in slices to form a 3D view of the retina.[136] On OCT images, individual layers of the retina can be observed. De Sisternes et al. segmented the layers of the retina using thresholding and morphological operations.[137] From these segmentations, 11 features were quantified. A Poisson model was then used to predict the time to progression. Their proposed model attained a mean overall AUROC of 0.74 (95% CI: 0.58, 0.85), with the best predictive performance occurring at 11 months with an AUROC of 0.92 (95% CI: 0.83, 0.98). This high performance appears impressive; however, it is unclear whether the prediction is particularly useful as 11 months may be an unusual time point for clinicians to choose to predict at. The large confidence intervals for the mean overall AUROC indicates that the model may not generalise well and may provide poor estimates at some time points.

A similar method of feature extraction from SD-OCT images was used by Niu et al.;[138] however, this model aimed to predict the GA growth using a random forest model. For predicting GA growth in patients without signs of GA, the model attained a Sørensen-Dice score of around 0.74 (standard deviation (SD)=0.17). This model is interesting in that it does not just attempt to predict GA but also the growth of GA, which may provide some added utility to clinicians who may want to assess the possible extent of disease.

Banerjee et al.[135] also used a similar feature extraction method; however, they instead used an RNN consisting of LSTM, batch normalisation, and full-connected layers. Yim et al. noted that these two-stage models, with the image being segmented before classification, can help identify the anatomical changes that lead to a higher risk of progression[139].

The main disadvantage of this two-stage approach is that the predictors must still be known and extractable, as the deep learning model extracting the features still requires annotated training data to train the algorithm. For volumetric data such as OCT images, the individual layers can be segmented and the volume easily estimated; however, automatically extracting these predictors is more difficult on colour fundus images. For AMD, the risk factors are not fully understood, and some important features of the image may be missed. However, deep learning can be used to automatically extract features from images;[53] so the two stages can be combined into a single stage by using a single deep learning prognostic model that takes the raw images as input, extracts the important features, and estimates the risk of progression.

2.6.3 Deep learning prognostic models

Before the work presented in this thesis, some prognostic models were developed in deep learning. My work builds upon those previous methods.

One of the first models using deep learning for prognostic modelling was DeepSurv,[140], which merged a deep neural network with a Cox proportional hazards model.[130] DeepSurv is not intended for imaging data and consists of a fully-connected neural network followed by a survival layer. DeepSurv showed comparable or improved performance on simulated and real data experiments. The authors attributed the improved performance on some tasks to the improved flexibility of the model. The proposed method performed particularly well on more complex data with nonlinear features. This method was soon extended to imaging data with DeepConvSurv[141]. Replacing the fully-connected network with a CNN enables DeepConvSurv to better handle unstructured data such as images. For example, experiments were conducted on a dataset of pathological images of lung cancer. Due to the size of the images, they were split into patches. Compared with DeepSurv, this method showed an improved C-index of 0.629 versus a 0.602 obtained using DeepSurv.

Several models have been developed to predict progression to advanced AMD in particular. Most of these models use a single time point to predict the future progression, although more than one image may be used if there are multiple fields[142] or the images are stereoscopic.[143]

Grassman et al.[144] proposed an ensemble model composed of six CNNs (AlexNet, GoogLeNet, VGGNet, Inception-v3, ResNet101, and Inception-ResNet-v2). This model aimed to predict the patient's probability of being in one of 13 classes at some future time point. The 13 classes consisted of the 9-step severity scale[122], neovascular AMD, geographic atrophy, neovascular AMD and geographic atrophy, and ungradable images. Using data from AREDS, each CNN was trained to predict one of the 13 classes. A random forest classifier was then used to ensemble the CNNs and obtain a final prediction. The algorithm attained an overall accuracy of 63.3%. No measures of uncertainty, such as confidence intervals, were reported. The lack of uncertainty measures is a problem throughout deep learning and hinders the ability to truly assess how well a model performs.

Babenko et al.[143] aimed to predict the progression from early/intermediate AMD to neovascular AMD within one year. Stereo images were fed into the algorithm as pairs into the model based on Inception-v3. A late fusion approach was used to deal with the pairs, with each image fed into a separate Inception-v3 network with shared weights and the softmax activation applied; the output was then averaged. The model was trained and tested on AREDS with 10-fold cross-validation. The model attained an AUROC of 0.88 (SD=0.02), compared to 0.83 (SD=0.03) for the AREDS 9-step scale and 0.78 (SD=0.2) for the AREDS 4-step scale.

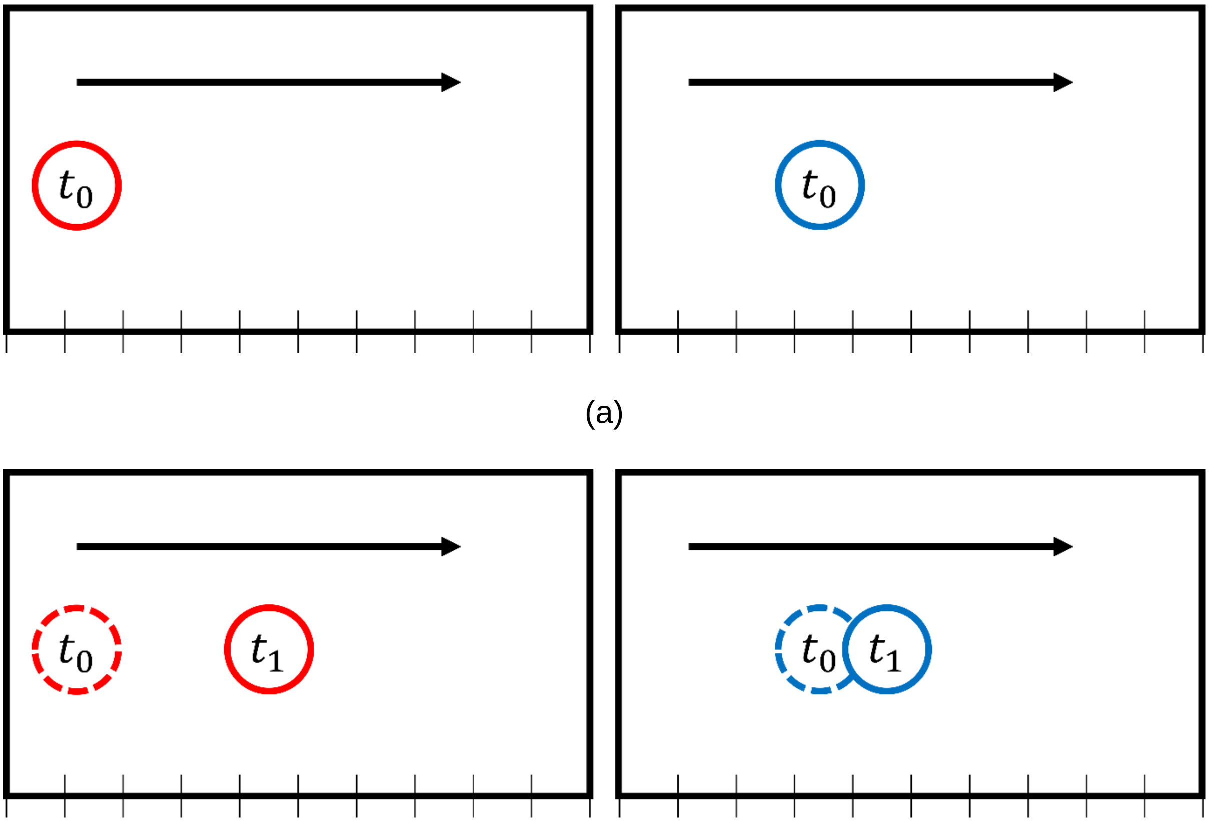
Traditional statistical models for AMD progression often use genetic data. A model proposed by Yan et al.[145] combined both imaging and genetic data. The model aimed to predict if the patient would progress from early/intermediate to advanced AMD. Inception-v3 was used to predict the current AMD severity of the image. The severity score was then concatenated with the genetic data, and fully-connected layers were used to obtain an estimated probability of progression.

Peng[146] et al. combined a CNN with a Cox proportional hazards model[130]. The model consisted of four Inception-v3 architectures that extracted features of drusen and pigment abnormalities for the left and right eyes, resulting in 512 extracted features. Feature selection was then used to reduce the dimensionality. Cox proportional hazards models were then used to predict the progression to advanced AMD. The model achieved a five-year C-index of 0.86, exceeding the performance of two retinal specialists.

It is impossible to determine which of the above models is best, as they were all developed on slightly different datasets and often attempted to solve different problems. Different types of utilised data, such as stereoscopic images or genomics, may be used even when using the same dataset.

So far, all the models I have described only use a single time point to predict the future progression of disease. A single time point shows the current state of the disease, but it does not show the rate of progression.

To illustrate this, I use the example of a ball rolling along the floor. Looking at a single image of the ball in Figure 2.33 (a), it is impossible to know how fast it moves. If two images of the ball are taken at different times, such as in Figure *2.33 (b)*, the progression of the ball between the two points can now be observed. Images taken at different time points are longitudinal and are the main focus of my thesis.



(a)

(b)

Figure 2.3: An example to illustrate the importance of longitudinal imaging data. In the first image (a), the blue ball may be predicted to reach the end of the image

before the red ball, as at t_0 the blue ball has progressed further. When another time point is given in image (b), the red ball can be seen moving more quickly and will therefore reach the end before the blue ball.

More recently, longitudinal images have been used to predict the progression of AMD[147]. Several OCT images were taken at 30-day intervals. A DenseNet architecture[148] with shared weights was used to extract features from each OCT scan, and an RNN was used to capture the temporal relationship. Finally, a fully-connected layer predicted the probability of progression.

Incorporating longitudinal images will likely improve performance as it allows the algorithm to capture the rate of progression; however, previously, regular visit intervals are required. While even intervals may be possible in clinical studies, in real-world clinical settings, it is unrealistic as patients often miss visits or may have their intervals changed. Regular visits at exactly 30-day intervals for imaging are impossible in most applications. It is important to account for uneven intervals between visits to determine the rate of progression between longitudinal images. Figure 2.34 extends the example shown in Figure 2.33 to show the importance of accounting for the time intervals between images. As mentioned in Section 2.2.2, censoring is also an important consideration. It is likely that some patients will not progress to advanced AMD during the follow-up time and will be right-censored. Therefore, methods that deal with imaging data taken at uneven and irregular intervals and right-censored data are needed.

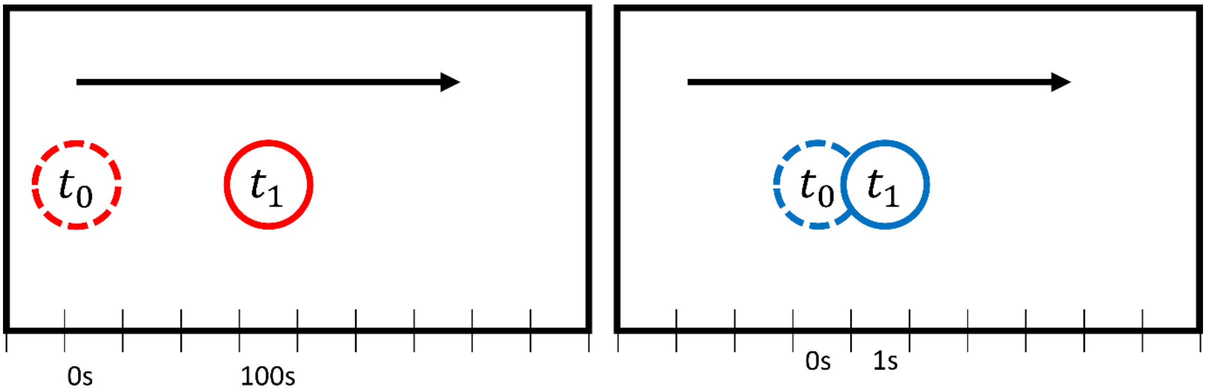


Figure 2.34: The same red and blue balls from Figure 2.33 now have times showing that the intervals between them are not the same. It is important to consider the times to capture the rate of progression.

A previous method was developed by Wang *et al.*[149] to deal with uneven intervals in an Alzheimer's data set, although this method was not applied to imaging data. Their method simply concatenates the time times onto the end of the variables. This method is unsuitable for imaging data as there is no way to concatenate the image tensor with a time scalar. The method was not fully explained, making it difficult to determine how this could be adapted to imaging data. Other methods aiming to deal with the uneven interval problem have been developed using LSTM;[150] [151] however, these were both applied to patient health records. A new method is needed which can deal with imaging data.

2.7 Summary

In this chapter, I have briefly outlined how deep learning has been applied in medicine. Deep learning is a wide and varied field that has grown massively with many methodological developments in recent years.

I have described how prediction models are a valuable tool for clinicians and can relieve pressure on resources. Prognostic models aim to predict the future outcome of disease and can be used to plan future treatment. For imaging data, deep learning shows excellent promise in prognostic modelling.

Deep learning is a relatively new field with several barriers to implementation. Firstly, novel methods are being developed without any real-world validation. The developed models often suffer from overfitting and do not generalise well to other images outside the development dataset. External validation using prospective data in the intended setting is needed before models can be trusted. Secondly, models are often black-box and difficult to interpret; this leads to mistrust in the model by both clinicians and patients. Visualisation methods such as saliency maps may be able to reduce this black-box nature. Thirdly, these models are often complex and computationally expensive, meaning expensive machines may be needed to utilise the developed models. Methods to greatly reduce computational complexity are needed.

Using a traditional statistical model may overcome these barriers; however, the main attraction of deep learning lies in its ability to automatically extract useful features from complex data sources such as images. The solution may be combining statistics and deep learning to create more trusted and robust models that benefit from aspects of both disciplines. It is vital to test each algorithm thoroughly according to best practice guidelines and to visualise what the model is using to reach the given decision.

At the start of my PhD, there were four main unsolved challenges that I aimed to overcome when developing a deep learning prognostic model for longitudinal imaging data:

- First, the model must account for uneven intervals between visits. For example, while a patient may be asked to come into the clinic annually, patients will likely miss visits, or the clinician may choose to alter the screening interval. Additionally, if the model is used to set screening intervals, the interval will change based on the model prediction, and the model will be invalid after its first use.
- Second, the model must account for missing data. For example, patients may miss visits, they may only have a baseline visit or images could be corrupted and lost.
- Thirdly, it would be useful to make predictions at several future time points. This will allow clinicians to choose when they want the prediction to be made.
- Finally, not all patients will likely be observed progressing, but they may progress after observation; this is known as right censoring. Right-censored data is common in clinical contexts.

It may be possible to create multiple models to solve these issues; however, this would be inefficient as it would require splitting the training data into multiple sets corresponding to each model. Furthermore, many models would also be required to account for each combination of visit intervals and missingness, making this option impractical.

Chapter 3: Data

Two types of dataset are used to demonstrate the methodology presented in this thesis. The first type is an ophthalmological dataset taken from AREDS, described in Chapter 2. The AREDS dataset consists of demographic, clinical, imaging, and genetic data from patients with AMD. I begin by using the imaging data, which are colour fundus images. I then use some of the demographic and clinical data in Chapter 6, to show how data other than imaging data can be incorporated into my models.

The second type of dataset I used is a COVID-19 dataset. There are two datasets used here, one from hospitals in China used for model development and the second from hospitals in Russia used for external geographical validation. This dataset is not longitudinal or prognostic but was used to demonstrate one of the novel components of another prognostic model. The images are slices of computed tomography (CT) scans.

3.1 AREDS

The AREDS dataset is one of the most extensive publicly available longitudinal datasets. I have already described the study in Chapter 2 and some ground-breaking research resulting from it. The study recruited 4,757 participants aged 55 to 80 years old. I chose this dataset as it is one of the most comprehensive longitudinal imaging datasets with a follow-up of up to 10 years and is freely available from the National Eye Institute upon request.

As mentioned in Chapter 2, AREDS and the follow-up study AREDS2 were long-term studies by the US National Eye Institute, primarily assessing risk factors associated with AMD and cataracts.[95] As well as studying the risk factors and prognosis of patients with AMD and cataracts, the studies also assessed the effects of nutrients on the progression of AMD and cataracts. Participants were randomly assigned to one of four arms:

1. placebo
2. zinc (80mg) and copper (2mg)
3. vitamin C (500mg), vitamin E (400IU), and beta-carotene (15mg)

4. zinc (80mg), copper (2mg), vitamin C (500mg), vitamin E (400IU), and beta-carotene (15mg)

The main conclusion of the studies was that patients with intermediate AMD or advanced AMD in one eye only, who were assigned to the formulation in arm 4, had a 25% risk reduction in progression from intermediate to advanced AMD. There was no reduction in cataract risk observed in any of the arms. AREDS2 added omega-3 fatty acid or lutein and zeaxanthin to the formulation, and no significant reduction in the risk of developing advanced AMD.[152]

Although 4,757 patients were recruited in AREDS, not all of these were included in my work. One of the main reasons a patient may need to be excluded is that they have already progressed at baseline or they have no follow-up visits. It is impossible to predict progression for patients who have already progressed, and without at least one follow-up, we cannot know if the prediction was correct. The exact numbers of patients used for each of my developed models are given in the relevant chapters, as one model was able to account for missing visits and could therefore use more of the available data.

3.1.1 Limitations of the dataset

Although the AREDS and AREDS2 datasets are large-scale multi-centre studies and two of the best publicly available longitudinal datasets, they have significant problems.

Firstly, the images were initially collected on film and later digitised. Unfortunately, the digitisation project resulted in minor artefacts visible on some images. Examples of two images showing artefacts caused by the digitisation process are shown in Figure 3.1. The algorithm may confuse these artefacts with features of AMD, leading to a wrong classification. Modern colour fundus photography is digital without the need for film, and any algorithms developed on the AREDS dataset may not necessarily generalise well to more modern photography.

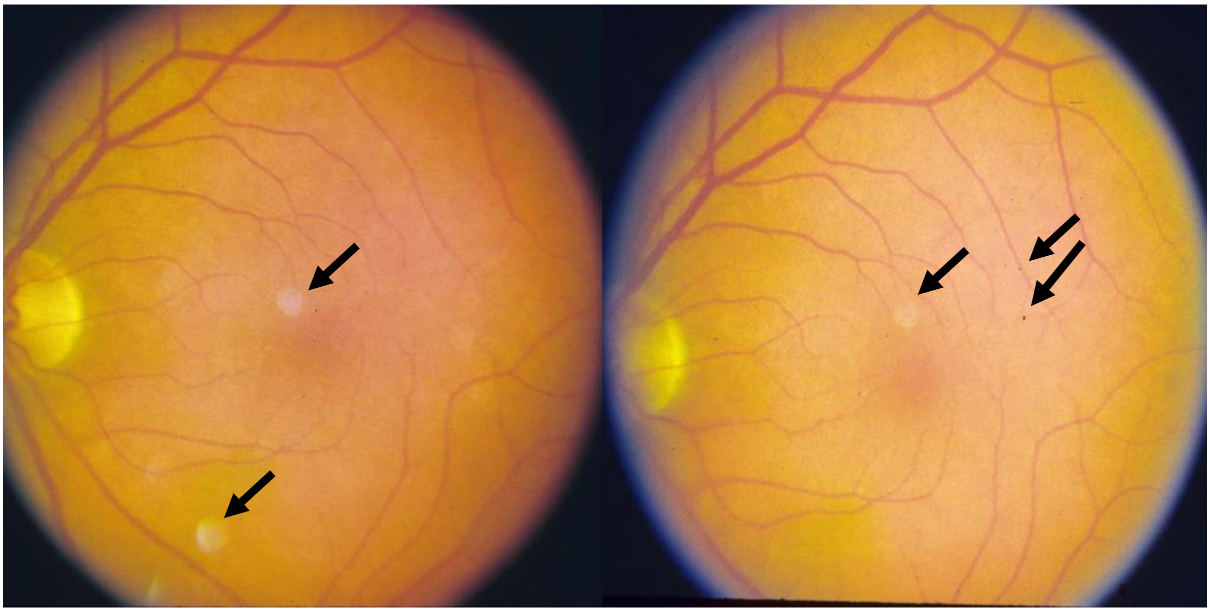


Figure 3.1: Examples of images showing artefacts caused by digitisation.

The AREDS dataset also contains large amounts of missing data. Images may be missing for many reasons, such as the patient progressing to advanced AMD, dropping out from the study, non-attendance at visits, or the image being lost. Explanations for missing data are not given, and some demographic and clinical data are left blank. Blank could mean the value is unknown, not collected, zero, or the same as the last visit for longitudinal data. For example, all patients have their smoking status at baseline recorded. The smoking status at each visit should then also be recorded; however, this is often left blank. This could mean that the clinician forgot to collect or record the smoking status, it could mean that the patient indicated they do not smoke, or it could mean that the smoking status has not changed from baseline. As the proportion of missingness is large and we do not know if the data is missing completely at random or not, there is no statistical method that can account for this missing data.[153] This makes the variable unusable as it is unclear if the data are truly missing. However, for the missing images, we can assume that the images are missing at random, and I will test how many images may be removed before there is a significant reduction in model performance. The image quality may also cause issues. I was unable to ask clinicians for their opinion on each image in this dataset quality to obtain quantitative values; however, a study into image quality in the AREDS2 dataset found that only 0.7% of images were deemed ungradable by clinicians.[154]

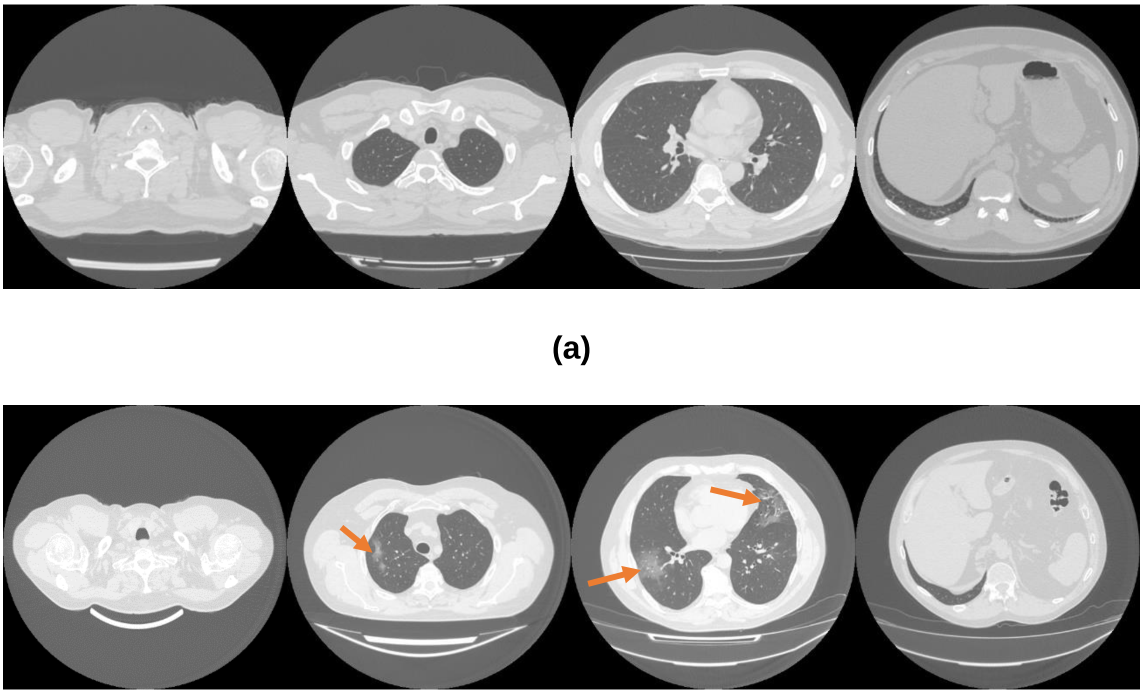
3.2 COVID-19

The COVID-19 data are used to demonstrate one of the methods I have developed. For this, I was able to obtain a training set and an external validation set. All the images used in this dataset are of computed tomography (CT) imaging. CT scans are made of many 2D images (slices) that give a 3D-like structure when combined. Some scans can be made up of as few as ten slices, while others may have as many as 500. This can cause a challenge in deep learning as many algorithms require the same number of images used in each patient.

The first dataset is taken from a group of hospitals in Moscow, Russia[155] and the second set is taken from a group of hospitals in China.[156] This allowed me to perform external geographical validation and assess how the model may generalise to data from other countries. All data were retrospectively collected with the diagnosis made by expert consensus examining radiographical features of the scan. Examples of slices from a healthy and COVID-19 patient from the MosMed dataset are shown in Figure 3.2. The example images show how COVID-19 features may not necessarily be present in each slice of a diseased patient, and a slice-based approach to diagnosis may not be suitable. These two datasets were chosen as they are large-scale, from two different countries, publicly available, and have been widely used.

3.2.1 MosMed

The training and internal validation sets were collected from a consortium of hospitals in Moscow, Russia.[155] The scans were all performed between March 1, 2020, and April 25, 2020. The dataset consisted of 254 healthy patients and 1,141 patients with COVID-19. All images were included in the analysis.



(a)

(b)

Figure 3.2: Example slices of CT scans in the MosMed dataset from (a) a healthy patient and (b) a patient with COVID-19. In the two middle scans, COVID-19 features can be observed in patient (b), as highlighted by the orange arrows.

3.2.2 Zhang et al.

The external validation data are from a consortium of hospitals in China.[156] All scans were performed between January 25, 2020, and March 25, 2020. The dataset contained many duplicate patients, and I removed scans so that each patient only had one scan. I found 243 healthy and 553 COVID-19 scans suitable for use.

While the training dataset was ready to use, the external validation dataset had several issues I needed to address before evaluating any models on this set. Firstly, many scans were repeated in the dataset, with the same patient appearing more than once in the same set. This introduced significant bias as it artificially inflated the size of the dataset. I removed scans so that only one scan per patient remained. Secondly, many of the scans were not centred on the lungs and looked at other body parts which were higher or lower. Some scans did not contain any slices of the lung. I removed slices which did not show any lung. Thirdly, there was a mix of masked and unmasked scans in the dataset; I removed any masked scans. A flowchart showing the number of included patients is displayed in Figure 3.3.

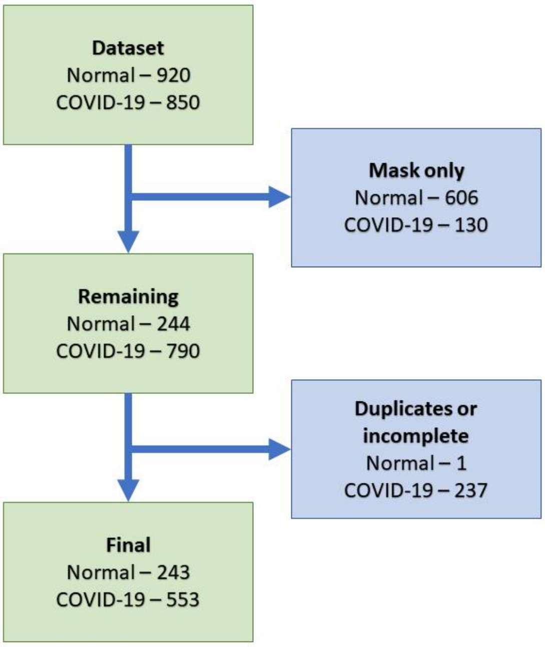


Figure 3.3: Patients included in the external validation dataset.

3.2.3 Unbalanced data

In Chapter 4, I develop a novel activation function that aims to overcome unbalanced data. To test this activation function, I needed to obtain highly unbalanced data. At the start of the pandemic, we had many negative and a few positive patients; this gave me the perfect opportunity to test my idea.

The first dataset I used was a balanced dataset, which would allow me to vary the level of imbalance to assess the effect. This widely used toy dataset consists of community-acquired pneumonia and healthy x-ray images.[157] I used 1,583 images from each class. I then used 800 images for training, 200 for validation, and 583 for testing from each class. I then removed pneumonia-positive images to obtain pneumonia:healthy images at ratios of 1:10, 1:25, and 1:50.

The second dataset is from the COVID-19 image data collection and contains COVID-19 positive x-rays.[158] At the time of doing this piece of work, this was the biggest and most trusted source of COVID-19 images. For training and validation, I used images from the Italian Society of Radiology[159] and images from other sources for testing. For healthy images, I used the ChestX-Ray dataset for training and validation,[160] and images from Kermany *et al.*[157] and the Shenzhen Hospital X-ray dataset[161] for testing. Images which had distinguishing features, such as notes or which were less than 256 pixels in either height or width, were removed. This resulted in 30 COVID-19 and 40,240 healthy images for training, 15 COVID-19 and 20,120 healthy images for validation, and 84 COVID-19 and 1,907 healthy images for testing. In the training and validation set, this equated to a COVID-19:healthy ratio of 1:1341.

Chapter 4: Interval scaling

In this chapter, I describe the development of a method to overcome the problem of uneven time intervals between visits. I proposed a novel interval screening mechanism that weights observations at time points closer to the prediction time as more important and informative than those further away. Here, I demonstrate how the novel mechanism can be incorporated into a deep learning model using the AREDS dataset and assess whether longitudinal imaging can improve the prediction of progression to advanced AMD. In all the examples I present to demonstrate my methods, I use three images taken at three time points to predict the outcome at a fourth time point for the same individual; however, any of the methods can be easily extended to any number of time points.

The novel interval screening mechanism in this chapter was applied to a network using a GRU and posted to arXiv[162] while under review and was then accepted to be published in BMJ Open Ophthalmology after implementing reviewer comments.[163] I then developed a similar method replacing the GRU with a more straightforward method inspired by time series leads to a less computationally complex model with no significant loss in model performance. I presented this alternative method at MIUA 2021.[164]

4.1 Introduction

In Chapter 2, I presented a general overview of the currently available methods for prognostic modelling in deep learning. I discussed that most work focuses on using a single time point, and the model may benefit from using longitudinal data to capture the rate of disease progression. In particular, I highlighted some of the previous models for AMD prognosis, most notably the work of Babenko et al.[143] and Yan et al.[145], which used a single time point. Patients may progress at different rates for some diseases, and longitudinal imaging may be needed to capture this progression. In this chapter, I explore how longitudinal images may be used in deep learning and whether this can improve the prediction of AMD progression.

In Chapter 2, I gave one example of longitudinal imaging data used in a deep learning prognostic model; however, the images were collected at even time intervals.[147] In reality, patients often visit the clinic at uneven intervals due to missed appointments, busy clinics, or their screening interval adjusted. Potential applications of prognostic models include aiding in setting screening intervals, which allows low-risk patients to be seen less frequently; however, this would immediately invalidate models requiring even time intervals. Furthermore, ignoring the uneven intervals between visits is likely to reduce the accuracy of the predictions, as the model is given no frame of reference. Therefore, a method which can account for uneven intervals between screening was needed. It would also be useful to choose the future time point to predict. Finally, even when using a single time point, it would be helpful to be able to alter the time interval between the observation and prediction to choose the future prediction point.

In the work presented in this chapter, my main aim was first to develop a method to predict disease progression using longitudinal images taken with different time intervals between observations. The method needed to allow a prediction to be made at any chosen future time and not a set future time point. I then applied this method to predict progression to advanced AMD and assessed whether adding additional time points improves performance or if a single observation is sufficient. Finally, I aimed to simplify the method to reduce the computational complexity of the model.

4.2 Methods

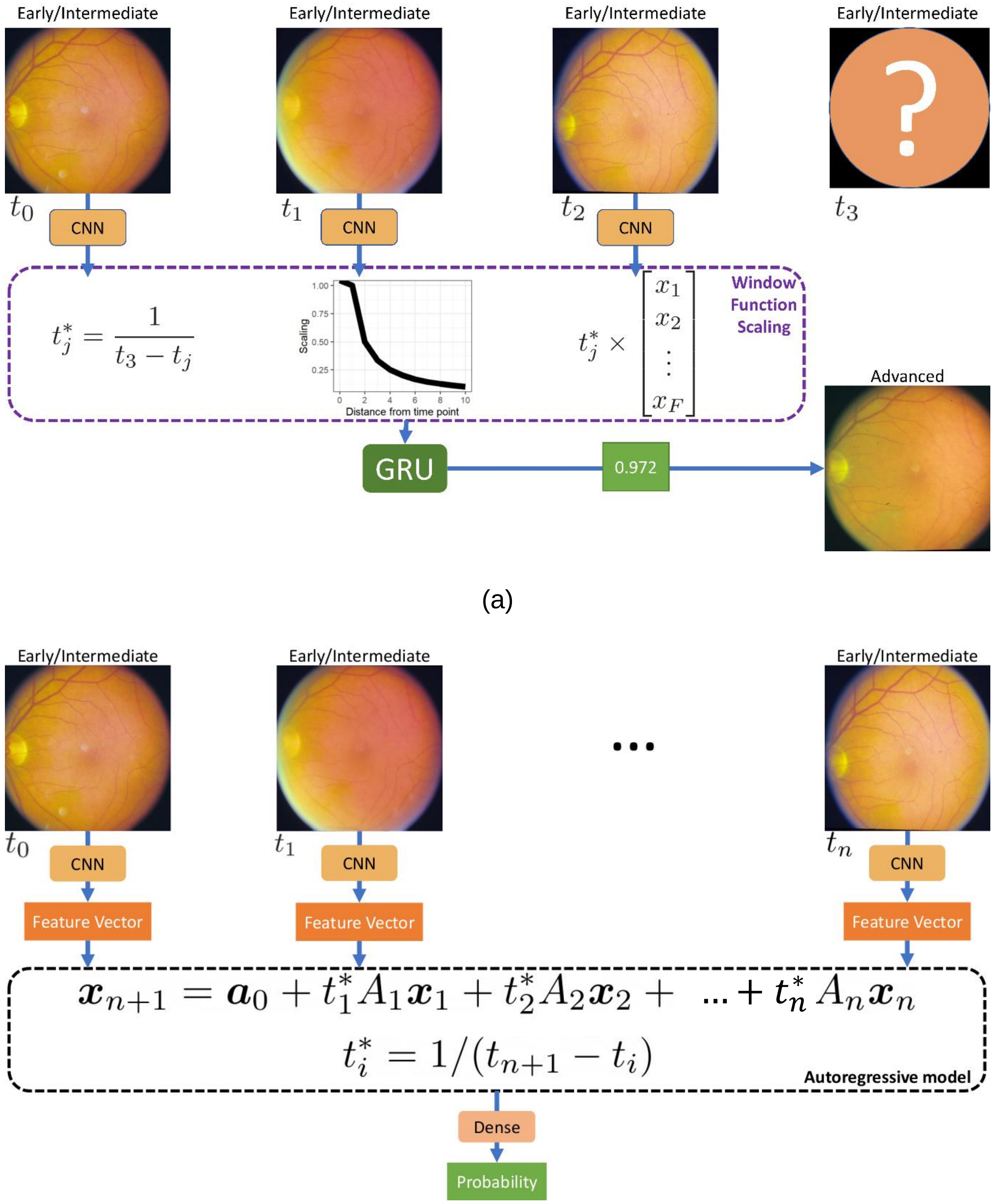
In this chapter, I describe two methods I developed to predict progression to advanced AMD using longitudinal imaging while accounting for uneven intervals between visits.

To formalise the problem, given a set of N images from a single patient $(X_0, \dots, X_i, \dots, X_N)$ observed at times $(t_0, \dots, t_i, \dots, t_N)$ I aimed to predict whether the patient will have progressed by their next visit at t_{N+1}, where $t_{i+1} - t_i = t_i - t_{i-1}$ does not necessarily hold. This is a binary outcome, where

$$y = \begin{cases} 1, & \text{if the patient progresses to advanced AMD,} \\ 0 & \text{if the patient does not progress.} \end{cases} \tag{4.1}$$

Both of the methods I present here consist of three main stages. First, I used a CNN with shared weights to extract features from each image; the feature vectors were then scaled to account for the uneven time intervals using a novel interval scaling mechanism. Finally, based on those scaled feature vectors, I predicted the probability of progression by a chosen time. The first method used a GRU, while the second aimed to reduce computational complexity by replacing the GRU with a more straightforward linear combination. Finally, I used a novel activation function for both methods I created to better deal with unbalanced data.

In this work, I used three previous images ($N = 3$) to predict the diagnosis at a fourth time point. I compared both methods to assess whether the complexity reduction reduces performance. Overviews of both methods are shown in Figure 4.1.



(a)

(b)

Figure 4.1: Overviews of (a) the GRU network and (b) the VAR network.

4.2.1 CNN

As I described in Chapter 2, a CNN uses convolutional layers to reduce data dimensionality. Several layers, such as convolutional, pooling, and batch normalisation layers, can be used to reduce the image to a single vector of features. In this way, a CNN can be considered an automatic feature extractor that reduces the image to a single vector of features the algorithm finds useful or important. These features can then be passed to another layer, such as a fully-connected or recurrent layer, to obtain a probability of progression.

I began by extracting features using a CNN for each image for both methods presented here. To reduce computational complexity, weights were shared between the CNNs; this means only one set of parameters needs to be trained. I chose to use Inception-v3[165] pretrained on ImageNet.[48] as it is relatively efficient and generalisable to many applications. The architecture and reasoning behind Inception-v3 are described in Chapter 2. After the convolutional layers, I reduced the representation of each image to a single vector using average pooling. Then I applied a dropout layer with a 50% probability. This results in three feature vectors, $F_1, F_2, \text{and } F_N,$ one for each of the three images $X_1, X_2, \text{and } X_N,$ with each vector of length 2048.

4.2.2 Interval scaling

One of the main objectives of the thesis was to develop methods capable of dealing with uneven time points. To achieve this, I developed interval scaling. As previously discussed, visits closer to the prediction time are more likely to be useful to the algorithm than visits further away. Interval scaling weights the extracted features so that more recent features are given a higher priority; this enables the algorithm to better capture the rate of progression.

For each sequence of images observed at times $(t_0, \ldots, t_i, \ldots, t_N)$ and outcome prediction time t_{N+1} I rescaled each time such that $t_i^* = 1/(t_{N+1} - t_i)$. I then multiplied each feature vector by its corresponding t_i^* to create a scaled feature vector. This prioritises the observations closer to the prediction time, allowing the network to account for these uneven time intervals. This scaling also accounts for uncertainty in the model features caused by older, outdated observations, as features from distant time points will be given lower importance.

To demonstrate my interval scaling method, I will give a brief example. Patients may be observed at 0 years, 1 year, 4 years, and 6 years, with the patient having progressed to advanced AMD on the fourth observation. Defining the times as $t_0 = 0, t_1 = 1, t_2 = 4,$ and $t_3 = 6$, the interval scaling values are given as

$$\begin{aligned} t_0^* &= \frac{1}{t_3 - t_0} = \frac{1}{6-0} = \frac{1}{6}, \\ t_1^* &= \frac{1}{t_3 - t_1} = \frac{1}{6-1} = \frac{1}{5}, \\ t_2^* &= \frac{1}{t_3 - t_2} = \frac{1}{6-4} = \frac{1}{2}. \end{aligned} \tag{4.2}$$

These values were then multiplied by the corresponding feature vector, $F_0,\ F_1,$ and F_3 resulting in three rescaled feature vectors.

Previous models developed using set intervals between visits could only predict specific future time points. For example, if patients are observed at yearly visits, the prediction can only be made one year in advance. To predict at other time points, another model needs to be developed. It would be useful to be able to pick any future time point to predict. With interval scaling, the future time point t_3 can be set to any desired time using one model. Multiple values of the future prediction time point can be chosen to obtain probabilities of progression for a range of future time points.

4.2.3 GRU

The first method I developed uses a GRU to predict the probability of progression. As I discussed in Chapter 2, RNNs are suited to sequence data such as audio or video data. In this work, the multiple time points form a sequence; therefore, an RNN, such as GRU, may be suitable.

The three feature vectors of length 2048 were concatenated into a 3 × 2048 matrix before being passed to the GRU with a single output unit. I used the tanh function for the activation function and the sigmoid function for the recurrent activation function.

The GRU layer could be replaced by another RNN layer, such as LSTM; however, as discussed in Chapter 2, the GRU layer is much less computationally complex and often attains comparable performance. GRU may also be more stable if longer time sequences are used in future studies.

4.2.4 VAR

Although GRU layers are less computationally complex than LSTM layers, they are still relatively complex. Therefore, a complex recurrent layer may not be needed and could be replaced with a simpler solution. I developed an alternative solution inspired by vector autoregression (VAR) in time series analysis.

From the CNN, there are three feature vectors from previous time points. Instead of predicting the future outcome based on those vectors, I aimed to use the previous feature vectors to produce a future feature vector. This new feature vector is then used to predict the outcome.

Autoregressive models take previous values and perform regression upon them to predict the next value in the sequence. These models are commonly used in meteorology to predict daily temperatures and in finance to predict stock prices. The pth order autoregressive model can be written as

$$x_{n+1} = a_{n+1} + a_n x_n + \cdots + a_{n-p} x_{n-p} + \epsilon_t, \tag{4.3}$$

where x_i is the value at time i, a_i is the parameter associated with x_i to be estimated and e_t is some stationary noise.

Autoregressive models can be extended to the multivariate case using vector autoregression

$$\boldsymbol{x_{n+1}} = \boldsymbol{a_{n+1}} + A_n \boldsymbol{x_n} + \cdots + A_{n-p} \boldsymbol{x_{n-p}} + \boldsymbol{e_t}, \tag{4.4}$$

with the values at each time point now being vectors $\boldsymbol{x}_i$ and their corresponding parameters A_i being matrices. VAR considers the relationship between features rather than just considering each feature separately.

As with the GRU method, I utilised interval scaling to account for uneven time intervals. Each feature vector in the VAR equation was weighted to give features closer to the prediction time greater importance than those further away. The interval scaling was then given by $t_i^* = 1/(t_{n+1} - t_i)$. Therefore, the VAR equation with the interval scaling applied is given by

$$\boldsymbol{x_{n+1}} = \boldsymbol{a_{n+1}} + t_1^* A_1 \boldsymbol{x_1} + \cdots + t_{n-p}^* A_{n-p} \boldsymbol{x_{n-p}} + \boldsymbol{e_t}. \tag{4.5}$$

The future prediction time point can be chosen by altering t_{n+1}. This resulted in a single feature vector which was then used to predict the probability of progression. For this, I used a fully-connected layer with one output unit.

In traditional statistics, the parameters are estimated using methods such as ordinary least squares or Newton's method. In this work, I incorporated the parameter estimation into the deep learning framework and used backpropagation to learn the parameters.

4.2.5 Single time point

One of my main objectives was to assess whether including multiple time points improves performance over using a single time point. I compared my novel multiple time point methods with a single time point method. A GRU layer cannot be used for the single time point model, so I replaced the GRU with a fully connected layer with a single output unit. Interval scaling was applied to the single time point to account for the difference in times between the single observation and the outcome time; this allows for any future time point to be chosen, similar to my longitudinal methods. To ensure fair comparisons, all other hyperparameter settings remained the same.

4.2.6 GEV activation

During my studies, I encountered the issue of highly unbalanced data. When datasets have one class which significantly outweighs the others, the model tends to overfit on the dominant class. Methods to overcome this problem include oversampling the underrepresented class or undersampling the overrepresented class. Unfortunately, these resampling methods often lead to even more overfitting and often do not adequately alleviate the problem. Other possible solutions include reweighting the loss function or using the focal loss.[69]

I proposed an alternative solution based on the Generalised Extreme Value (GEV) theory.[166] Instead of focusing on the loss, my solution focuses on the activation function. The activation function is based on the GEV distribution and is given by

$$f(x) = \begin{cases} exp\left(-exp\left(-\frac{x-\mu}{\sigma}\right)\right), & if\ \xi = 0, \\ exp\left(-\left(1+\xi\frac{x-\mu}{\sigma}\right)^{-\frac{1}{\xi}}\right), & if\ \xi \neq 0, \end{cases} \tag{4.6}$$

where $\mu, \sigma,$ and ξ are parameters to be estimated by the deep learning algorithm. The GEV activation can be used as a direct replacement for the sigmoid activation function and showed significantly improved performance in experiments when the classes are extremely imbalanced. When the data is balanced, the GEV activation performs similarly to the sigmoid activation.[166]

The parameters $\mu, \sigma,$ and ξ can be initialised to be approximately the sigmoid function. To find suitable initial parameters, I used non-linear least squares with the Gauss-Newton algorithm to find parameters of the GEV function, which approximates the sigmoid curve. The parameters that most closely approximate the sigmoid activation are $\mu = -0.590813, \sigma = 1.660529,$ and $\xi = -0.273071$. A graph showing the sigmoid and the GEV activation with these parameters is shown in Figure 4.2. Initialising the GEV activation parameters to these values means that the model output will be similar to using the sigmoid activation during the early stages of training. As training progresses, the GEV curve changes from the sigmoid curve to adapt to the unbalanced data.

I have also extended the GEV activation function to the multiclass problem with the multiclass generalised extreme value (mGEV) activation function.[167] The mGEV begins with the GEV activation before normalising the probabilities

$$mGEV(GEV) = \frac{GEV}{\sum_i GEV}. \tag{4.7}$$

The mGEV activation function showed improved performance over the softmax activation on a multiclass classification task with highly unbalanced classes.

As the GEV activation is a replacement for the sigmoid activation, it could also replace the sigmoid function in the swish activation function. I used the GEV activation as the final activation for both the GRU and the VAR models.

In the following subsection, I give the results of experiments showing that the GEV activation function can outperform the sigmoid function with and without oversampling.

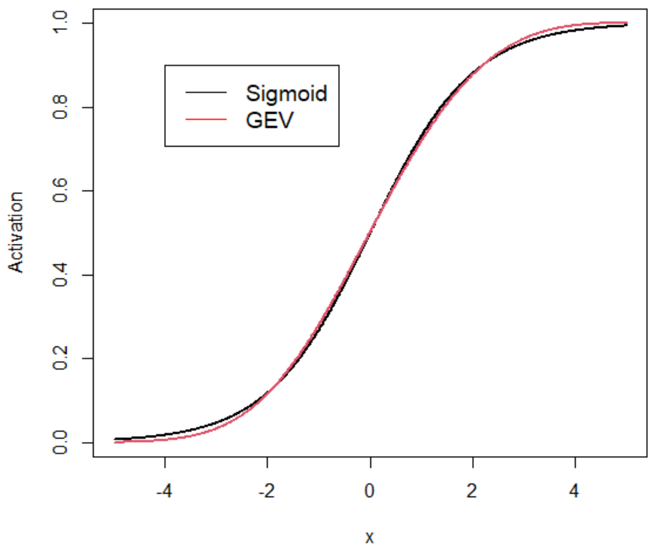


Figure 4.2: Sigmoid and GEV activation with parameters set to approximate the sigmoid.

4.2.6.1 GEV Experiments

Using the datasets described in Section 3.2.3, I aimed to test whether the GEV activation is better than the sigmoid function with and without oversampling.

Firstly, results on the balanced dataset are shown in Table 4.1. The results show that the sigmoid without oversampling works reasonably well up to a ratio of 1:10; however, with an imbalance ratio of 1:25, the sensitivity is significantly lower, as can be seen from the 95% confidence intervals. The sigmoid activation with sampling performs better with reasonable performance up to a ratio of 1:25; however, with a balance of 1:50, oversampling fails and performs worse than using sigmoid alone. My novel GEV activation function maintains good performance across all levels of imbalance, although there is still some loss in performance.

These results highlight the danger of overfitting when using oversampling. At the 1:50 ratio, using the sigmoid activation with oversampling achieves perfect specificity, but the sensitivity is massively lowered. It seems that oversampling at higher levels of imbalance may actually make performance worse.

Table 4.1: Model performance on the balanced set at different levels of imbalance using the sigmoid activation, the sigmoid activation with oversampling, and my GEV activation function.

Ratio P:N	Method	AUROC	Sensitivity	Specificity
1:1	Sigmoid	0.993 (0.990, 0.996)	0.962 (0.947, 0.978)	0.962 (0.947, 0.975)
	Sigmoid + OS	-	-	-
	GEV	0.994 (0.991, 0.997)	0.966 (0.943, 0.980)	0.967 (0.953, 0.975)
1:10	Sigmoid	0.975 (0.966, 0.983)	0.919 (0.897, 0.941)	0.947 (0.929, 0.965)
	Sigmoid + OS	0.981 (0.975, 0.970)	0.930 (0.909, 0.950)	0.933 (0.913, 0.953)
	GEV	0.985 (0.978, 0.991)	0.959 (0.943, 0.975)	0.959 (0.943, 0.975)
1:25	Sigmoid	0.916 (0.961, 0.981)	0.792 (0.760, 0.825)	0.954 (0.937, 0.971)
	Sigmoid + OS	0.959 (0.948, 0.970)	0.875 (0.848, 0.902)	0.923 (0.901, 0.944)
	GEV	0.971 (0.961, 0.981)	0.919 (0.897, 0.941)	0.966 (0.951, 0.980)
1:50	Sigmoid	0.609 (0.575, 0.644)	0.419 (0.378, 0.459)	0.931 (0.911, 0.952)
	Sigmoid + OS	0.529 (0.520, 0.539)	0.058 (0.039, 0.077)	1.0 (1.0, 1.0)
	GEV	0.941 (0.928, 0.954)	0.828 (0.798, 0.859)	0.937 (0.917, 0.956)

Next, I performed experiments comparing the sigmoid activation and my GEV activation on the real imbalanced dataset. Results are shown in Table 4.2. In this case, there is a non-statistically significant increase in the AUC, with an increase in sensitivity and a decrease I specificity. These results may be surprising as it seems that the GEV does not provide much advantage based on AUROC; however, the GEV provides a better balance between sensitivity and specificity, indicating that overfitting is reduced.

Table 4.2: Model performance using the sigmoid and GEV activation on the unbalanced x-ray dataset.

Activation	AUROC	Sensitivity	Specificity
Sigmoid	0.750 (0.690, 0.809)	0.488 (0.381, 0.595)	0.932 (0.921, 0.944)
GEV	0.820 (0.770, 0.870)	0.798 (0.712, 0.884)	0.778 (0.759, 0.796)

Finally, using the CT imaging dataset, results for the sigmoid activation and GEV activation are shown in Table 4.3. The model using the sigmoid activation on this dataset classified all images as healthy; this model is clinically useless. The model using the GEV activation gave much more balanced results, although the results may still not be good enough for clinical use.

Table 4.3: Model performance using the sigmoid and GEV activation on the unbalanced CT dataset.

Activation	AUC	Sensitivity	Specificity
Sigmoid	0.561 (0.502, 0.620)	0.0 (0.0, 0.0)	1.0 (1.0, 1.0)
GEV	0.675 (0.621, 0.730)	0.628 (0.561, 0.695)	0.651 (0.579, 0.723)

These experiments suggest that the GEV activation can perform at least as well as the sigmoid activation function and, when the data is unbalanced, can give the model a much better balance between sensitivity and specificity.

4.2.7 Loss function

Several possible choices for loss functions were described in Section 2.4.4. For binary classification, the binary cross-entropy loss function is most commonly used; however, for the reasons I have discussed, I chose to use the MSE. To reduce overfitting, I made two important changes to the usual method of calculating the MSE. Firstly, I weighted the loss to help account for the unbalanced data. For the negative cases, I used the weighting

$$w_0 = \frac{1}{n_0} \times \frac{n}{2}, \tag{4.8}$$

for the positive cases, I used

$$w_1 = \frac{1}{n_1} \times \frac{n}{2}, \tag{4.9}$$

where n_0 is the total number of negative cases, n_1 is the total number of positive cases, and $n = n_0 + n_1$ is the total number of cases. This class weighting can be combined with the GEV activation to deal with unbalanced data. I also transformed the target labels such that the negative and positive labels are

$$p_0 = \frac{1}{n_0 + 2} \tag{4.10}$$

and

$$p_1 = \frac{n_1 + 1}{n_1 + 2}. \tag{4.11}$$

This rescaling is based on Bayes' rule applied to the out-of-sample data and was proposed by Platt to reduce overfitting.[168]

4.2.8 Data and preprocessing

I used data from AREDS, which is described in Chapter 3. I included eyes with three previous visits with early/intermediate AMD and a fourth visit showing either early/intermediate or advanced AMD. This resulted in 5,144 eyes in the dataset, with 641 (12.5%) eyes progressing to advanced AMD. Eye profiles were split into 60% for training, 20% for validation, and 20% for testing on the patient level. This data split was stratified, so each set had 12.5% progressing eyes. This data split is shown in Table 4.4.

All images were cropped to remove the excess black background around the colour fundus image. During training, random augmentations were applied to the images to improve the generalisability of the model to unseen images. These random augmentations were a brightness adjustment of between 80% and 120%, a rotation of ± 10 degrees, and a random flip in the horizontal and vertical directions. All images were then resized to 299×299 pixels using bilinear interpolation, the default image size for Inception-v3. Finally, pixel values were divided by 255 to rescale between 0 and 1.

Table 4.4: Number of eyes in each data split.

	Training	Validation	Testing	Total
Progressing	385	128	128	641
Non-progressing	2701	901	901	4503
Total	3086	1029	1029	5144

4.2.9 Performance metrics

I have described the performance measure for survival models in 0 2. I assessed the models using the C-index for discriminative performance across all time points. I used the time-dependent censored AUROC at one, two, and three years for the performance at individual time points. The calibration at one, two, and three years was assessed using calibration curves. Finally, I used decision curves to show the clinical utility over the treat none and treat all approaches.

4.2.10 Training and inference

All training and inference were performed on a Linux machine running Ubuntu 18.04, with 32 GiB of available memory and a 12GiB Titan X GPU. Python 3.6 Tensorflow 2.4.2 was used for development, and R 4.1.2 was used for analysis.

The censored C-index was calculated using the rcorr.cens function in the Hmisc package. Dynamic AUROCs were calculated using the censROC package with 95% confidence intervals calculated using 2000 sample bootstrapping. Finally, survival calibration curves were created using the rms package; the rmda package was used to create the decision curves.

4.3 Results

In this section, I present results for the GRU model and then for the VAR model using both two and three time points. There are no similar previously proposed models that can be used with the differing time intervals between observations for me to use for comparison. Instead, for comparison, I present results using a single time point with a fully-connected layer to assess whether the additional time points are needed. In all models, I used the proposed interval scaling mechanism, which enables observations and predictions at any time point to be used. Results are presented for predictions at one, two, and three years to display the ability of the interval scaling to choose the prediction time. Finally, I assessed the time taken to compute the GRU and VAR components of the models to assess if one method is faster to compute than the other.

4.3.1 GRU

The validation dataset was used to select the best model based on the lowest loss; the results are likely to be biased. Results on the testing dataset are likely to be less biased; however, they are taken from the same population, so some bias will remain.

Results on the validation dataset show that the model using a single time point achieved excellent results with a C-index of 0.914 (95%CI: 0.889, 0.940). When using two time points, the method achieved a C-index of 0.898 (95% CI: 0.869, 0.928). Using three time points, the C-index was 0.897 (95% CI: 0.868, 0.925). The single time point method attained a C-index of 0.862 (0.826, 0.897) on the testing data. The GRU method with two time points had a C-index of 0.889 (0.856, 0.922) and a C-index of 0.884 (0.849, 0.919) when using three time points.

Full results, including time-dependent AUROCs at one, two, and three years are shown in Table 4.5 and Table 4.6. ROC curves are shown in Figure 4.3. The true positive rate and false-positive rates plotted against risk thresholds are shown in Appendix A.

Table 4.5: Results using the GRU model with two and three time points and the fully-connected model for one time point on the validation dataset.

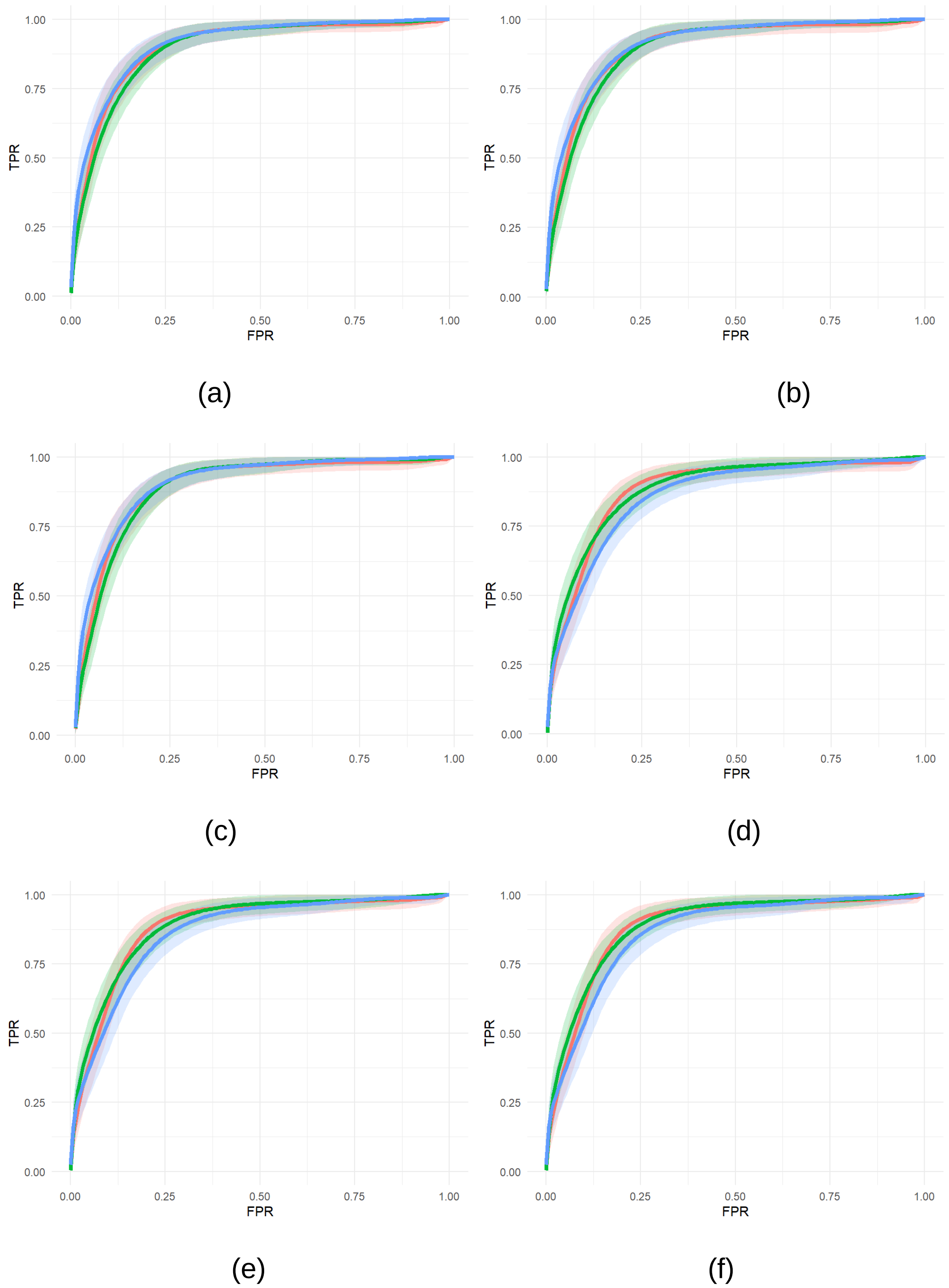
Number of time points	C-index	Years	AUROC
1	0.914 (0.889, 0.940)	1	0.916 (0.887, 0.940)
		2	0.942 (0.920, 0.963)
		3	0.940 (0.918, 0.961)
2	0.898 (0.869, 0.928)	1	0.900 (0.868, 0.929)
		2	0.931 (0.899, 0.961)
		3	0.933 (0.898, 0.96)
3	0.897 (0.868, 0.925)	1	0.899 (0.869, 0.925)
		2	0.931 (0.909, 0.963)
		3	0.930 (0.906, 0.961)

Table 4.6: Results using the GRU model with two and three time points and the fully-connected model for one time point on the testing dataset.

Number of time points	C-index	Years	AUROC
1	0.862 (0.826, 0.897)	1	0.869 (0.813, 0.905)
		2	0.929 (0.895, 0.954)
		3	0.954 (0.931, 0.974)
2	0.889 (0.856, 0.922)	1	0.888 (0.850, 0.919)
		2	0.95 (0.927, 0.972)
		3	0.964 (0.944, 0.983)
3	0.884 (0.849, 0.919)	1	0.893 (0.859, 0.925)
		2	0.947 (0.924, 0.967)
		3	0.965 (0.945, 0.983)

The calibration curves, shown in *Figure 4.4*, *Figure 4.5*, and *Figure 4.6*, indicate that all models are poorly calibrated. The risk is overestimated in all of the developed models, leading to overtreatment and wasted resources. Therefore, the risks need to be recalibrated using methods such as isotonic regression before the model can be used in a clinical setting.

Finally, decision curves are shown in Figure 4.7. These decision curves show that all models provide improved net clinical benefit over the treat-all or treat-none approaches in both the validation and testing sets. However, the models using two and three time points do not provide an improved net benefit over using the single time point method with the interval scaling.



(a)

(b)

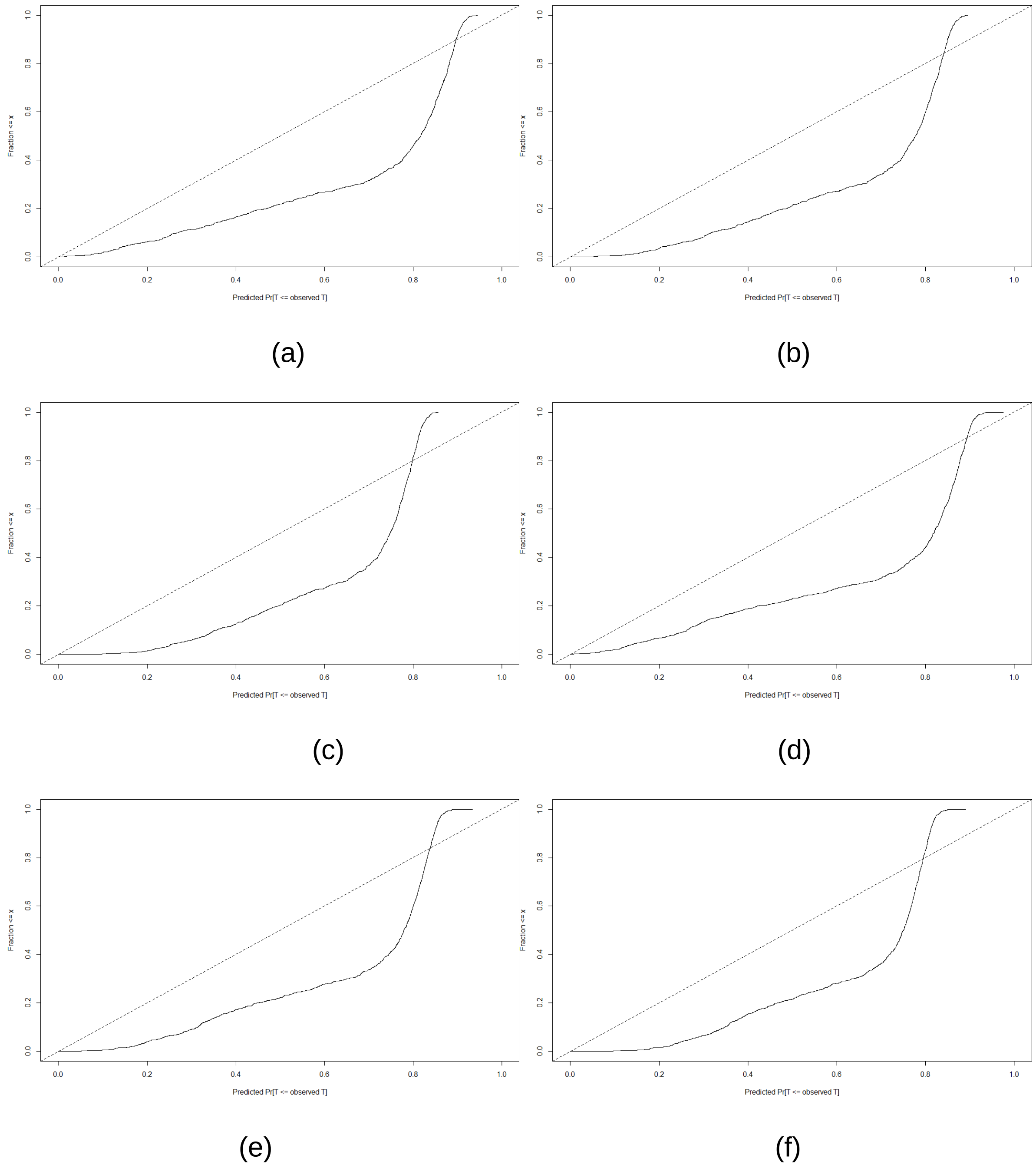
(c)

(d)

(e)

(f)

Figure 4.3: ROC curves for the single time point (blue) and GRU models with two (red) and three (green) time points on the validation dataset at (a) one year, (b) two years, and (c) three years and on the testing dataset at (d) one year, two years, and (f) three years. Confidence bands were calculated using 2000 sample bootstrapping.

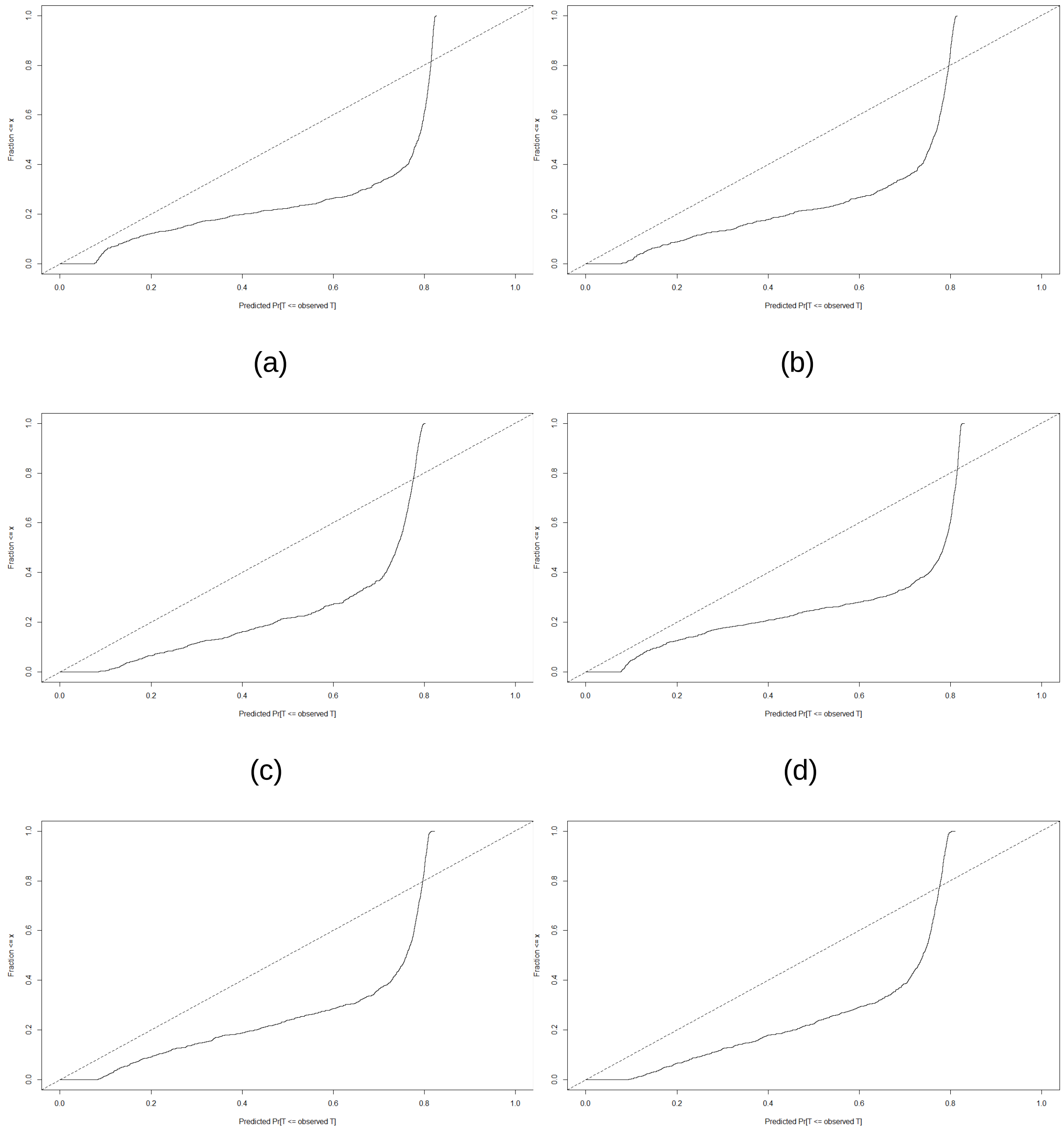


(a) (b)

(c) (d)

(e) (f)

Figure 4.4: Calibration curves for the single time point model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, two years, (f) and three years.

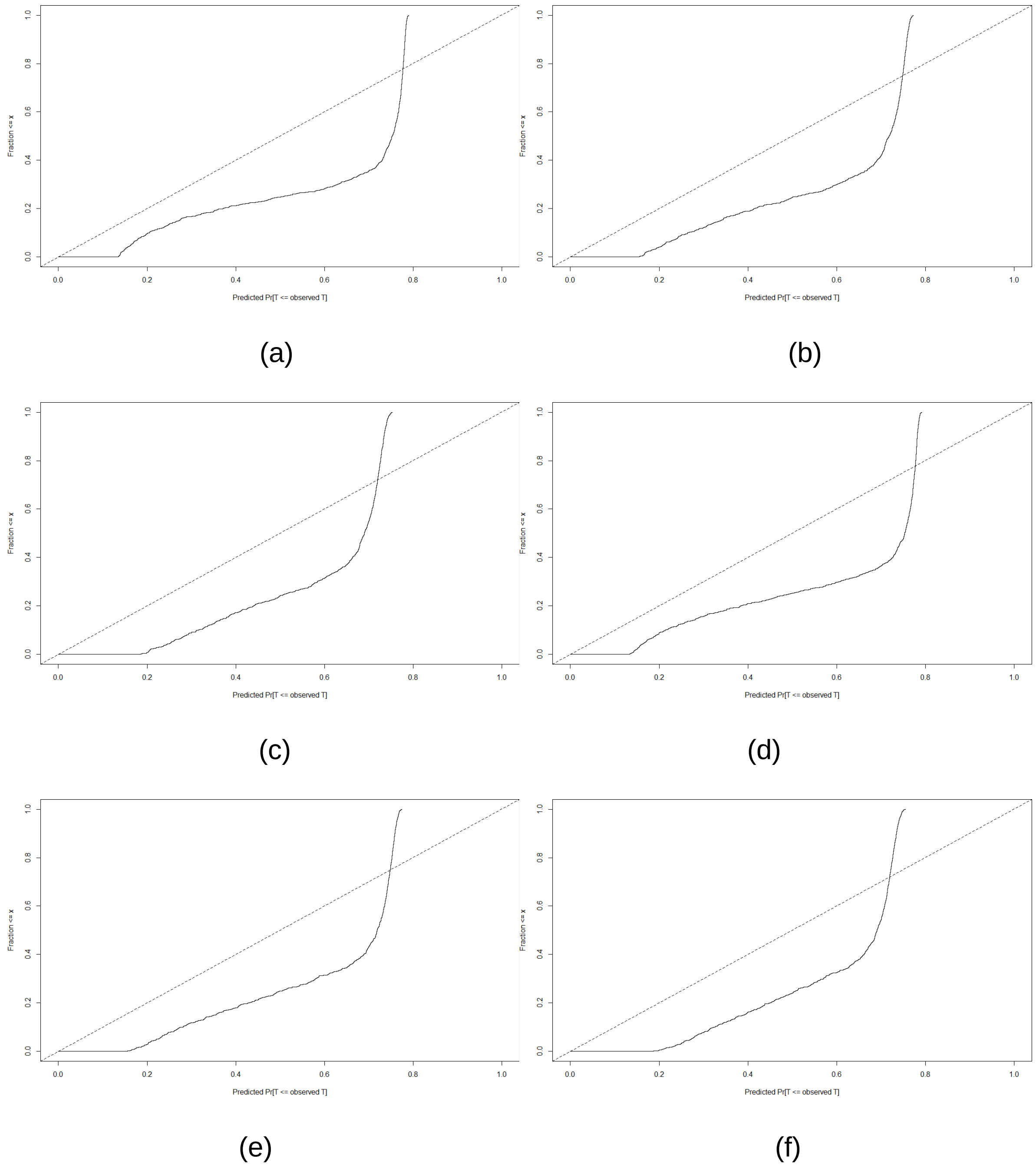


(a) (b)

(c) (d)

(e) (f)

Figure 4.5: Calibration curves for the GRU model with two time points model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, two years, (f) and three years.

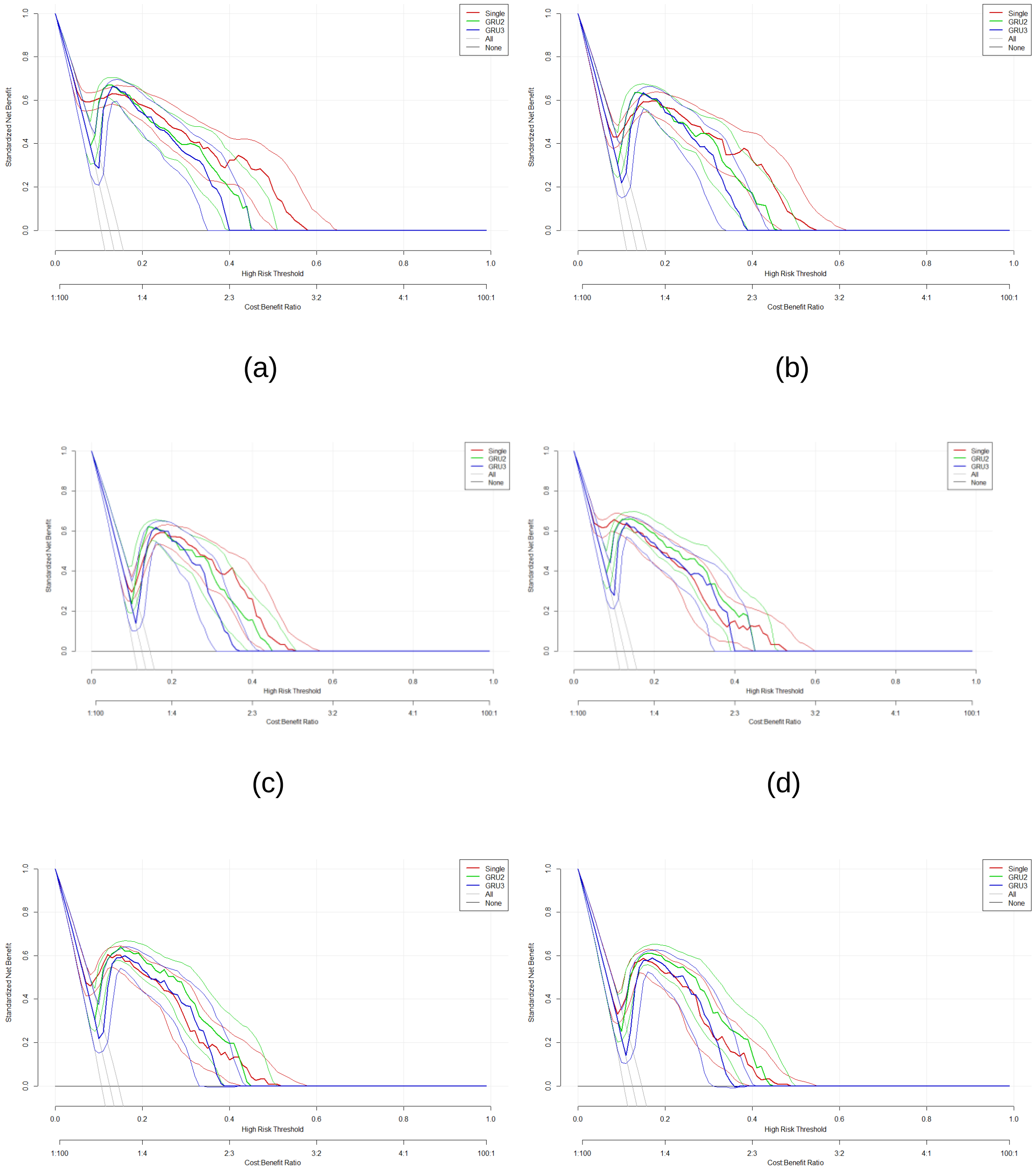


(a) (b)

(c) (d)

(e) (f)

Figure 4.6: Calibration curves for the GRU model with three time points model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.7: Decision curves for the single and GRU models on the validation dataset at (a) one year, (b) two years, and (c) three years and on the testing dataset at (d) one year, (e) two years, and (f) three years. Confidence bands were calculated using 2000 sample bootstrapping.

4.3.2 VAR

Using the VAR method, the model attained similar results as the GRU model. The single time point method results are the same as obtained in the GRU model experiment and are repeated in the tables for comparison.

For the validation dataset, using two time points, the VAR method achieved a C-index of 0.898 (95% CI: 0.868, 0.928). Using three time points, the C-index was 0.897 (95% CI: 0.868, 0.925). For the testing data, the VAR method with two time points had a C-index of 0.889 (0.856, 0.922) and a C-index of 0.884 (0.849, 0.919) when using three time points.

Full results, including time-dependent AUROCs at one, two, and three years are shown in Table 4.7 and Table 4.8. ROC curves are shown in Figure 4.8. The true positive rate and false-positive rates plotted against risk thresholds are shown in Appendix A.

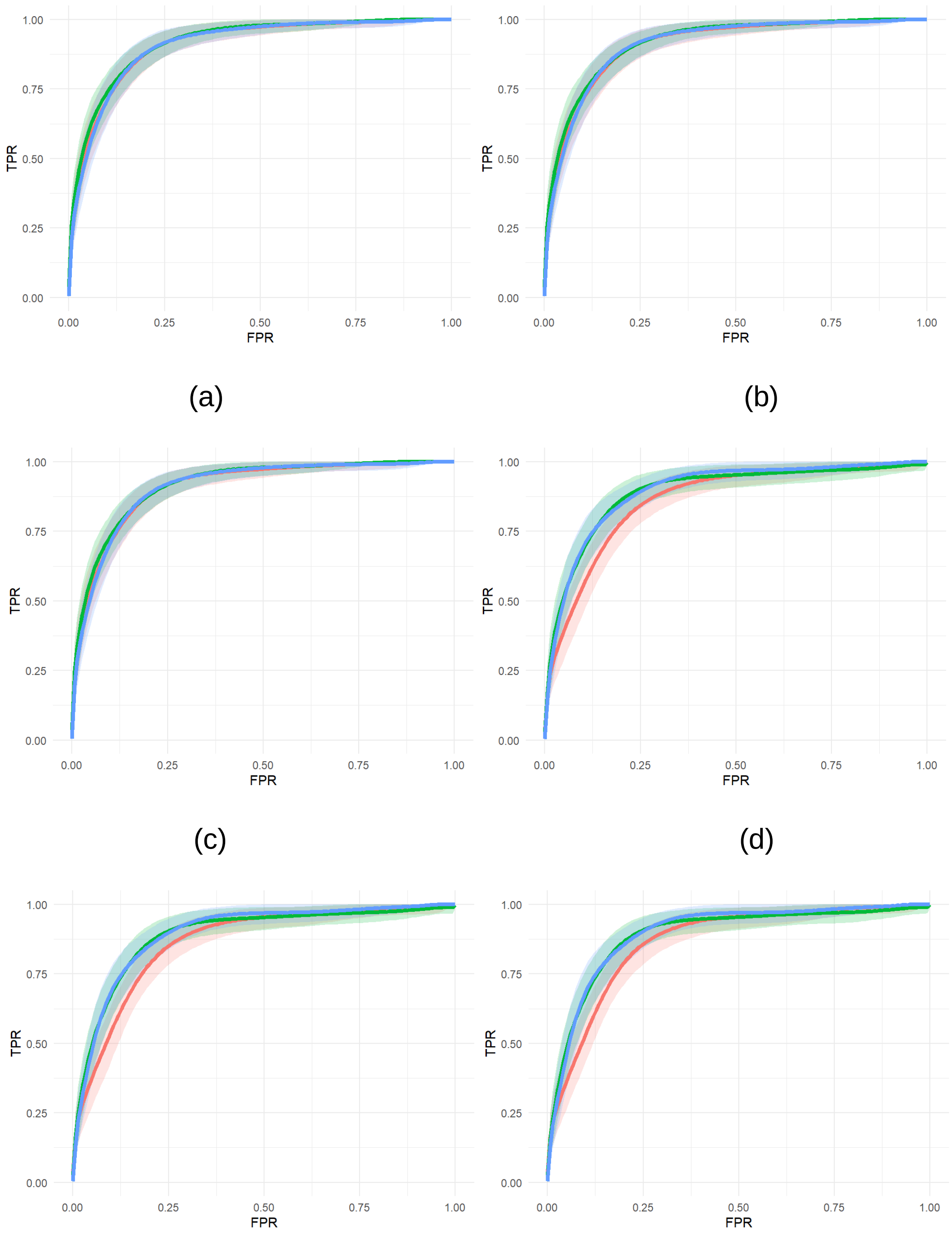
As with the GRU method, the calibration curves show systematic overestimation of risk by all models, as displayed in Figure 4.9 and Figure 4.10. These models would need to be updated prior to being deployed. Figure 4.11 shows that the models provide improved clinical utility over the treat-all and treat-none methods, similar to the GRU models.

Table 4.7: Results using the VAR model with two and three time points and the fully-connected model for one time point on the validation dataset.

Number of time points	C-index	Years	AUROC
1	0.914 (0.889, 0.940)	1	0.916 (0.887, 0.940)
		2	0.942 (0.920, 0.963)
		3	0.940 (0.918, 0.961)
2	0.917 (0.893, 0.942)	1	0.923 (0.895, 0.946)
		2	0.950 (0.929, 0.970)
		3	0.949 (0.927, 0.968)
3	0.908 (0.881, 0.935)	1	0.914 (0.885, 0.939)
		2	0.944 (0.921, 0.965)
		3	0.942 (0.917, 0.963)

Table 4.8: Results using the VAR model with two and three time points and the fully-connected model for one time point on the testing dataset.

Number of time points	C-index	Years	AUROC
1	0.862 (0.826, 0.897)	1	0.869 (0.813, 0.905)
		2	0.929 (0.895, 0.954)
		3	0.954 (0.931, 0.974)
2	0.892 (0.857, 0.927)	1	0.895 (0.854, 0.928)
		2	0.947 (0.923, 0.970)
		3	0.958 (0.936, 0.977)
3	0.892 (0.859, 0.924)	1	0.900 (0.867, 0.929)
		2	0.942 (0.920, 0.962)
		3	0.963 (0.945, 0.980)

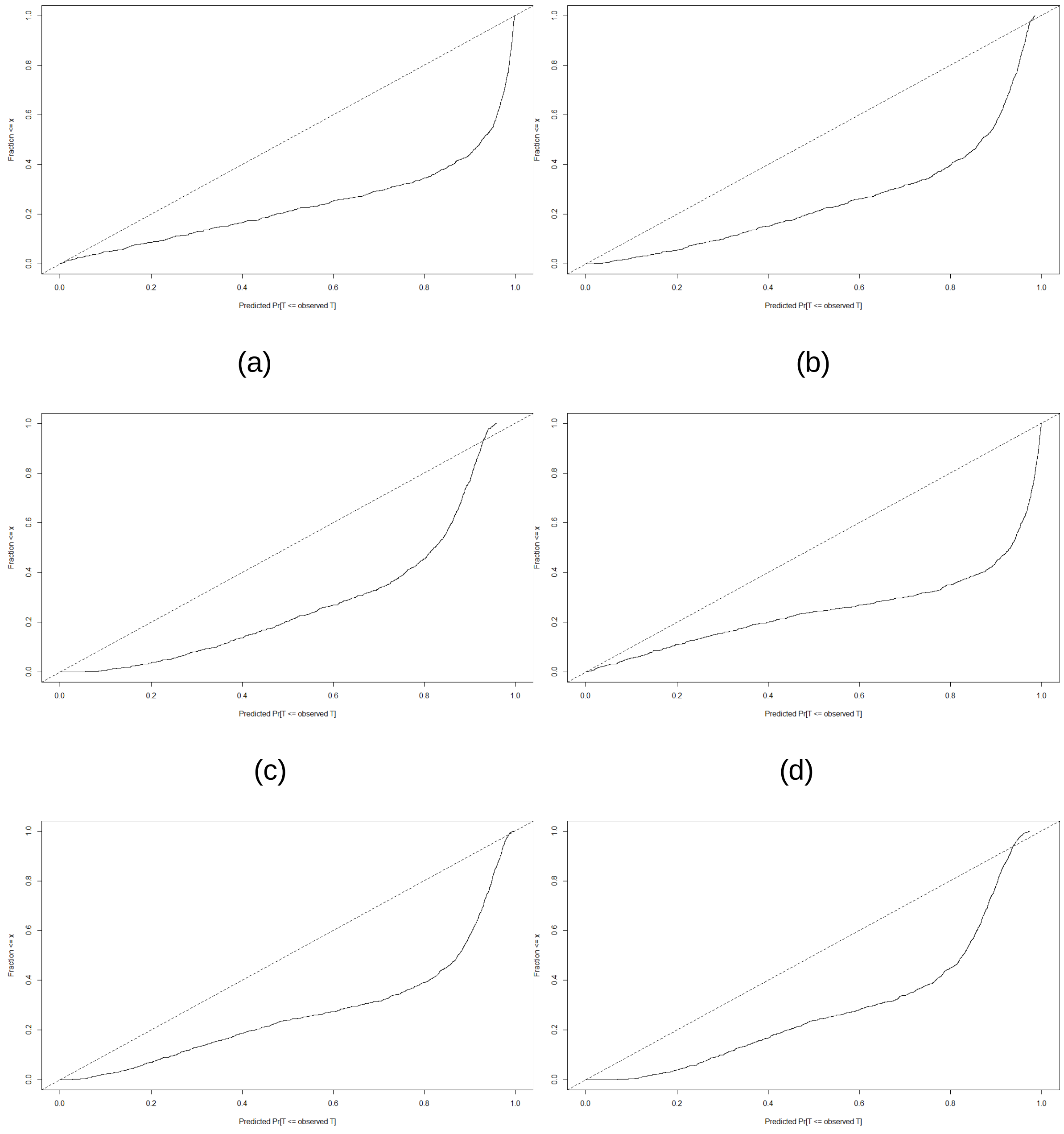


(a) (b)

(c) (d)

(e) (f)

Figure 4.8: ROC curves for the single time point (red) and VAR models with two (green) and three (blue) time points on the testing dataset at (a) one year, (b) two years, and (c) three years and on the testing dataset at (a) one year, (b) two years, and (c) three years. Confidence bands were calculated using 2000 sample bootstrapping.



(a)

(b)

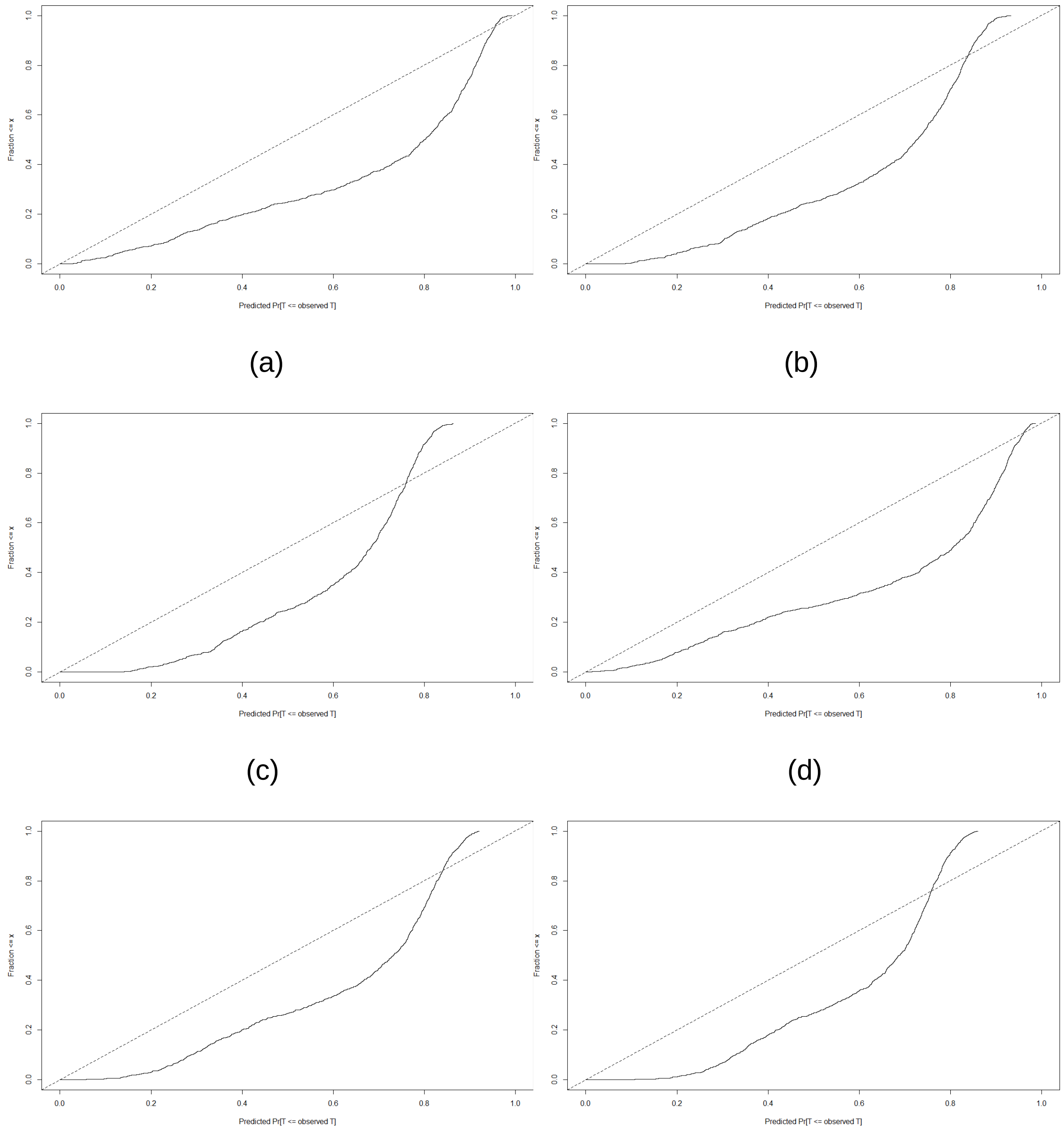
(c)

(d)

(e)

(f)

Figure 4.9: Calibration curves for the VAR model with two time points model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

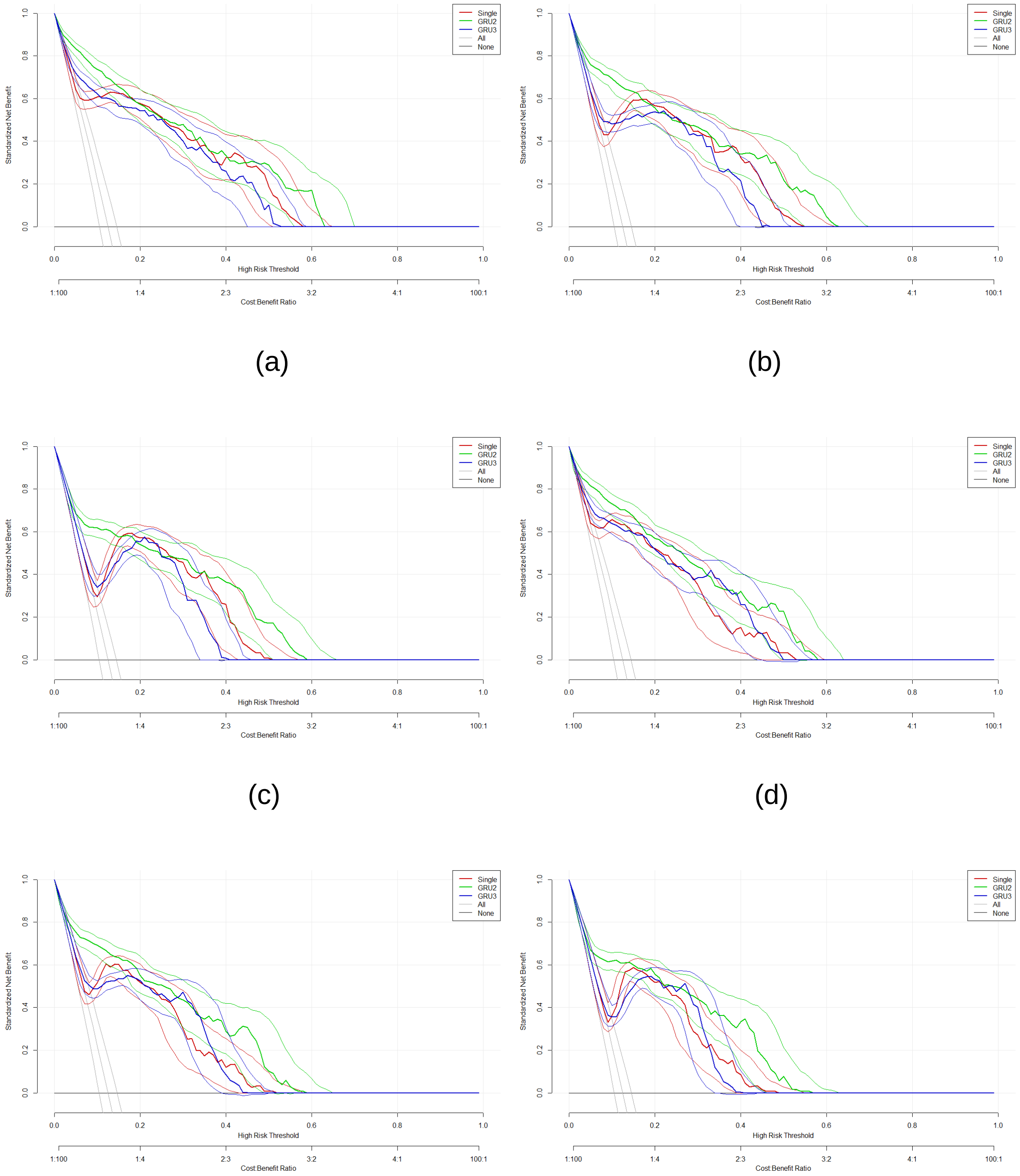


(a) (b)

(c) (d)

(e) (f)

Figure 4.10: Calibration curves for the VAR model with three time points model on the testing dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, I two years, (f) and three years.



(a)

(b)

(c)

(d)

(e)

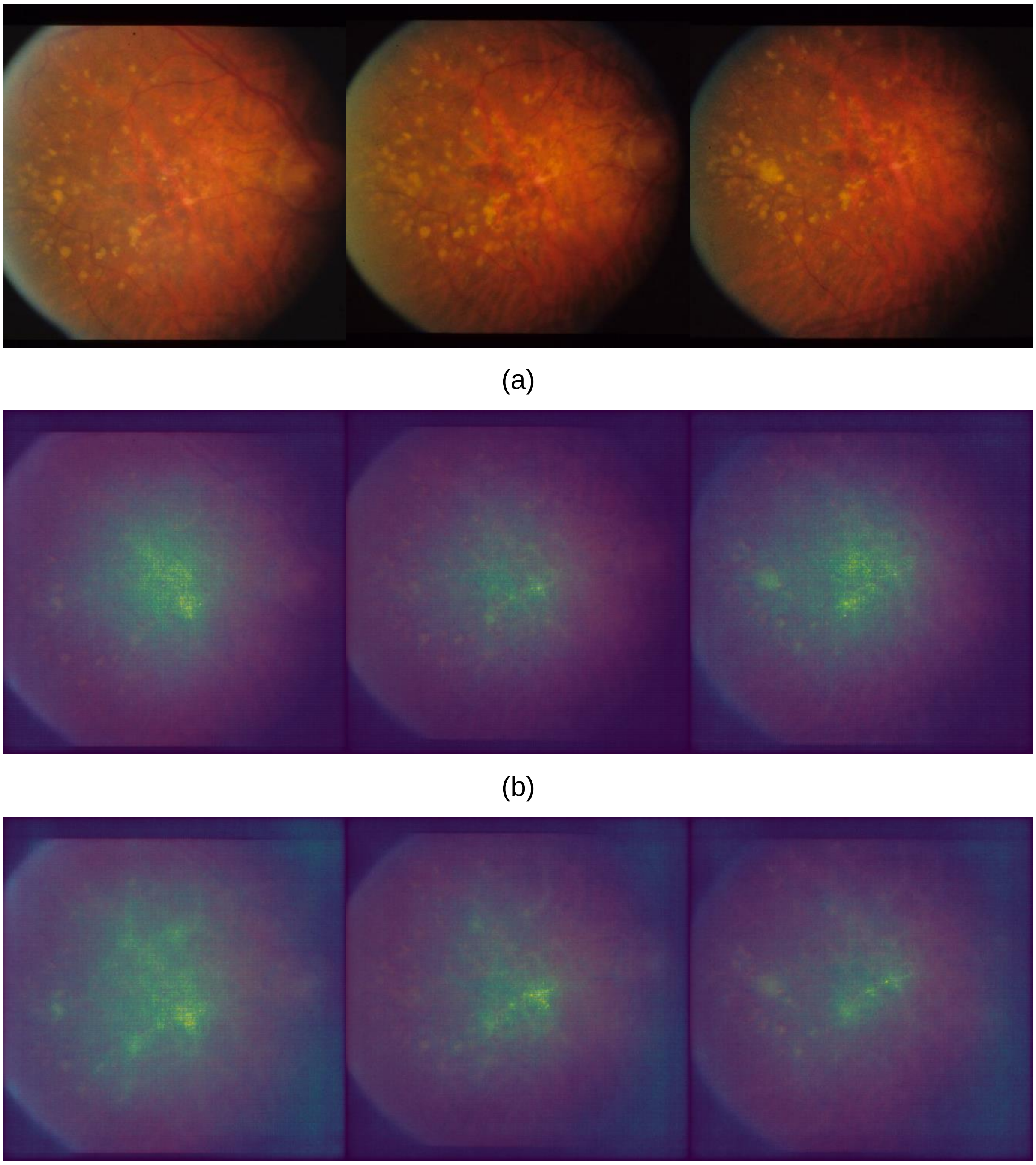
(f)

Figure 4.11: Decision curves for the single and VAR models on the validation dataset at (a) one year, (b) two years, and (c) three years and on the testing dataset at (d) one year, (e) two years, and (f) three years. Confidence bands were calculated using 2000 sample bootstrapping.

4.3.3 Saliency maps

In this section, I briefly show saliency maps for the GRU and VAR models to check if the model is correctly identifying the expected features. I created these maps using SmoothGrad,[82] with 20 samples and a noise of 0.2. As discussed in Chapter 2, the most prominent features for AMD are drusen; optic disc characteristics may also be useful in the classification.

The first set of saliency maps, shown in Figure 4.12, is for a patient progressing after one year. The GRU model correctly identified that the patient would progress with 86.2% probability; the VAR model also correctly predicted that this patient would progress with 82.4% probability. Figure 4.13 shows a patient progressing after two years. The GRU predicted the patient would progress with a probability of 72.7%, while the VAR model predicted a probability of progression at two years of 72.1%.

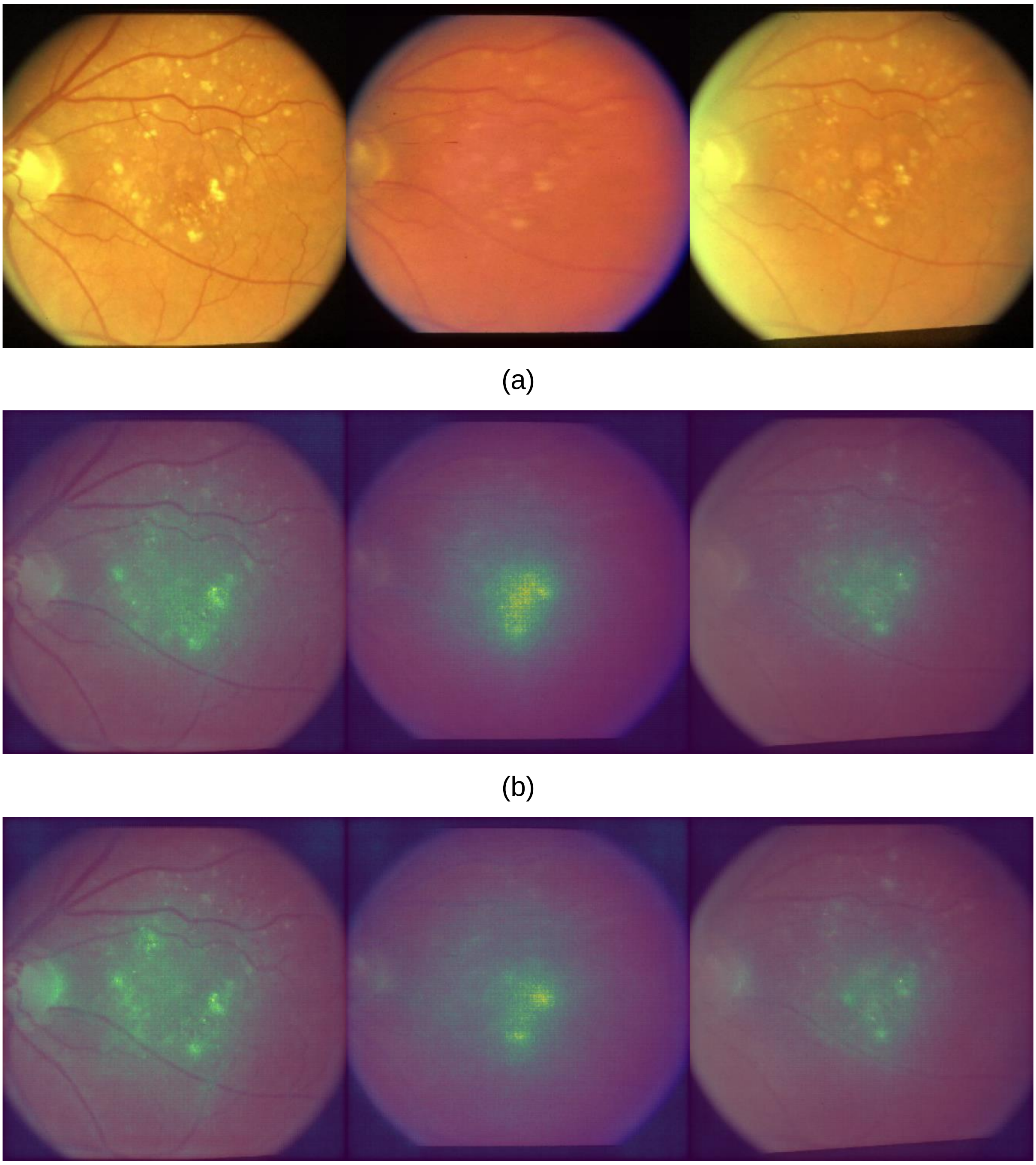


(a)

(b)

(c)

Figure 4.12: Saliency maps for a progressing patient for the three time point models showing (a) the original image, (b) saliency maps for the GRU model, and (c) saliency maps for the VAR model.



(a)

(b)

(c)

Figure 4.13: Saliency maps for a progressing patient for the three time point models showing (a) the original image, (b) saliency maps for the GRU model, and (c) saliency maps for the VAR model.

These saliency maps show that the algorithm identified the drusen visible in the image. The optic disc is also highlighted as useful in the prediction in the second patient. This indicates that the algorithms successfully learned the correct features within the images. Although the VAR model is slightly less confident in its predictions, the saliency maps appear more precise, with the drusen more prominently highlighted.

4.3.4 Computational complexity

The GRU layer has 6,153 parameters, while the VAR layer has 12,587,009 parameters. Therefore, the non-statistically significantly improved performance may not be worth the additional parameters; however, the VAR layer is much simpler than the complicated GRU layer. Therefore, the simpler VAR layer may be easier and faster to compute than the complicated GRU layer.

To test if one layer is faster than the other to compute, I ran the top layers of each network 1000 times each and compared the time taken for inference. As the bottom of both networks is the same, consisting of a CNN and pooling layers, I removed the bottom layers and only compared the VAR and GRU classification layers.

The GRU layers took an average of 0.0579s per inference, and the VAR layers took an average of 0.0535s per inference. This is a difference of 0.0044s. I used Welch's two-sample t-test to test whether this difference is statistically significant. The t-test gave a p-value of 1.68e-5 and 95% confidence intervals of (0.0024, 0.0064), indicating that the new VAR layer takes significantly less time for inference despite having many more parameters.

4.4 Discussion

The results show that using a single time point to predict progression to advanced AMD provides acceptable discriminative performance. Including multiple time points in this particular example did not significantly alter performance. However, this may not necessarily be true in all cases, and extra time points may benefit some applications. This highlights the need to assess whether more complex longitudinal models are justified. In addition, external validation is required to confirm whether this conclusion generalises to other data outside of the population.

There are many reasons that longitudinal data may not provide significantly improved predictions. Firstly, most patients with AMD may progress at similar rates, or the difference may not be observable over a three-year interval. A single observation may be enough to estimate the rate of progression. Secondly, there may be other clinical and demographic factors which affect progression. The inclusion of these other factors in the model may improve the prediction.

Despite the conclusion that a single time point appears to be sufficient, my proposed interval scaling mechanism can still be used for single time points, allowing any future time point to be selected for prediction. In my work, I have only shown results for one, two, and three years into the future; however, the predictions could be made at more regular intervals or further into the future, such as at the five-year mark. Sufficient data must exist for making these predictions; for example, a model trained on data with only a two-year follow-up could not be used to make reliable predictions for three years. My interval scaling method is computationally efficient and easily implemented and can be used with various future deep learning architectures beyond CNNs.

There is some debate about the usefulness of confidence intervals in decision curve analysis. It makes sense to choose whichever model provides the most clinical benefit, even if the benefit is not statistically significant. Additionally, the confidence intervals make the decision curve slightly harder to read. However, in this case, as increasing the number of time points increases the complexity of the model, it may be useful to consider the statistical significance of any improved benefits. Decision curves without confidence intervals, which may be slightly easier to read, are shown in Appendix A.

The VAR model adds considerably more parameters to the model than the GRU model; however, I have shown that the computation time may be significantly less as the operations are simpler than those in the GRU. Although the VAR model is faster, there is not a significant difference in predictive performance; this indicates that complex recurrent units such as LSTM and GRU may not be necessary. Future work using such units should consider whether simpler units can be utilised to reduce computational cost without sacrificing model performance.

The GEV activation function may provide an alternative to the sigmoid activation. This activation function provides better performcne when data is highly unbalanced and performs similar to the sigmoid activation when data is balanced. More recently, Alexandridis *et al.*[169] used the Gumbel distribution, which is equivalent to the GEV with $\mu = 0,\ \sigma = 1$, and $\xi = 0$, for segmentation and also observed improved performance over the sigmoid activation.

Since the publication of this model, another similar method has been proposed for glaucoma,[170] highlighting the current research interest in this area. The solution used in the glaucoma paper is more complex with multiple LSTM units. In their work, only accuracy and an ROC curve are reported, without individual time points or the right-censored nature of the data being considered.

4.4.1 Limitations

There are a few limitations of my work which could be improved in future studies. Firstly, I only had access to a single dataset. External validation is needed to assess how well the developed methods generalise to other populations. Future studies could examine whether additional time points are useful in other applications and diseases.

Although the discrimination and clinical usefulness of the developed models are good, they are poorly calibrated with systematic overestimation of the risk of progression across all models. These models would need to be recalibrated using methods such as isotonic regression to be suitable for clinical use. I have not recalibrated the models in this work as my aim has been chiefly methodological development, and I do not recommend deployment without further validation and sensitivity analysis. Recalibration and deployment are beyond the scope of my aims.

In the previously published work, I assessed progression or no progression and found that multiple time points significantly improve prediction over single time point models. However, a single time point may be sufficient when assessing prediction performance at individual time points in this thesis. This highlights that multiple time points may be more suitable in other applications.

In these models, I have only used imaging data. However, performance may be improved by adding clinical and demographic data. Possible clinical variables that could affect AMD progression have been briefly described in Chapter 2.

4.5 Conclusions

In this chapter, I have presented the deep learning methods I have developed, which can utilise longitudinal data for prognostic modelling. In particular, interval scaling allows for uneven time points between visits and for any future time for prediction to be chosen. In the example I gave to demonstrate the methods, a single time point was sufficient to predict progression to advanced AMD.

Chapter 5: Mixed-Effects Model

One of the challenges discussed in the introduction is the presence of missing data. Until now, I have considered models with an image available at each time point. In reality, patients are likely to miss visits or have just begun being monitored, so they may have fewer images available. It is possible to develop multiple models for each number of available images; however, this is inefficient and not optimal. A model for each possible number of images would be needed, and patients without the minimum number of required images would need to be excluded from that model. Imputation could be used to infer missing data; however, this is computationally expensive and may not produce good results for imaging data. Methods for imputing some modalities such as CT and MRI exist,[171] but when using photographs taken at different time points these methods are not suitable. Any imputation method for colour fundus imaging would need to deal with differences in lighting, differences in optic disc location (including the optic disc being missing), and differences in orientation. A method which could account for these missing images is needed. I investigated whether incorporating a mixed-effects model into the deep learning architecture could solve this problem. Mixed-effects models are a type of statistical model that consider both fixed- and random-effects and can account for missing data, provided that data is missing at random; this is the method used in my work.

Rather than immediately implementing mixed-effects into a prognostic model, I first tried my mixed-effects model on a diagnostic problem. This enabled me to assess the suitability of the mixed-effects layer and how well it accounts for missing data. The development of this methodology happened while the COVID-19 pandemic was occurring, and a wealth of CT volumetric data became available. Therefore, the proposed mixed-effects method is illustrated in a binary diagnostic algorithm for COVID-19. While the diagnosis of COVID-19 is not directly related to my PhD objectives, the methodology is the same. Although I could have used existing methods for CT slice imputation,[171] my aim was to develop a method which could be used on any image type including colour fundus imaging.

Many algorithms to diagnose COVID-19 were developed during the pandemic; however, these algorithms were often of low quality and unsuitable for real-world use.[172] While developing and externally validating my model, I aimed to show how a deep learning algorithm could be developed following statistical best practices.

The work presented in this chapter was made possible by a collaboration between the University of Liverpool, Amazon Web Services, and Alces Flight Software. Amazon Web Services kindly gave $50,000 of credits to be used on their platform. Funding was also received from the EPSRC Impact Acceleration Award (IAA). I submitted this work to medRxiv, and it is currently under peer review.[173]

5.1 Introduction

COVID-19 is an infectious respiratory disease which began in 2019 and was later declared a pandemic. Symptoms of COVID-19 are hugely varied, with some patients being asymptomatic while others develop severe viral pneumonia, which can be fatal. Through vaccination programmes, some countries have achieved relative control over the spread of the virus; however, future outbreaks and new emerging strains are expected to remain for many years or even decades to come. Even as vaccines have become more available, the continued threat of vaccine-resistant strains means that robust and reliable methods of detecting the disease must be developed.

The standard test used worldwide for confirming suspected COVID-19 is reverse transcription-polymerase chain reaction (RT-PCR); however, this detection method is not perfect. A review of RT-PCR for COVID-19 diagnosis found that while specificity is high at around 95%, sensitivity can be much lower at only 70%.[174] Early testing is vital to reducing false negatives, as tests taken further away from symptom onset will have much lower sensitivity.[175] Caution should be taken when using RT-PCR to rule out COVID-19. Therefore, computerised tomography (CT) can often be used to confirm a negative diagnosis; however, this can increase pressure on radiology departments. With many cases worldwide and the need for CT diagnosis, automated algorithms may prove helpful in screening patients for COVID-19.

Many models have been developed to classify COVID-19; however, these models are often of poor quality with a high risk of bias.[172] As a result, few developed models are suitable for clinical deployment. Roberts et al. identified three main pitfalls that lead to unsuitable models.[176] Firstly, models often lack adequate documentation; this prevents reproducibility. Secondly, best practice guidelines for developing and reporting prediction models are not followed. Finally, many studies do not include external validation to display the method's generalisability. External validation is vital to assess if the model is applicable outside the study sample.

In Chapter 3, I discussed CT imaging, the dataset used in this chapter, and the 3D-like nature of CT imaging. Deep learning can aid in automatically classifying CT images with high accuracy. When analysing CT scans using deep learning, there are two main approaches:

(1) Analysing the slices separately and then concatenating the features
(2) Merging the slices and treating the scan as a 3D structure

The first approach often uses a feature extraction network (such as a CNN) and pooling layers to obtain a feature vector for each image. It then uses a pooling layer to concatenate the slices into one feature vector for all slices. A fully connected layer is then used to classify the feature vector. This approach may not sufficiently model the spatial relationship between slices. Studies of the distribution of COVID-19 through the lungs have observed that some lobes may be more affected than others, and it is important to develop a model which accounts for this.

Bai et al.[177] developed a model which used a pretrained CNN called EfficientNetB4 to extract features from each slice. It is not mentioned how many slices were used for each patient; however, the dataset contained an average of 111.8 slices per patient. A series of fully-connected layers with batch normalisation and dropout were then used to reduce the size of each feature vector to 32. Average pooling then concatenated the features into a single vector. Finally, a fully-connected layer with sigmoid activation was used to obtain a probability of COVID-19.

On internal testing, this method achieved an accuracy of 0.96 (95% CI: 0.90, 0.98) with a sensitivity of 0.95 (95% CI: 0.83, 1.0) and specificity of 0.96 (95% CI: 0.89, 0.99).[177] The authors also reported an external validation accuracy of 0.87 (95% CI: 0.82, 0.90) with a sensitivity of 0.89 (95% CI: 0.81, 0.94) and specificity of 0.86 (95% CI: 0.80, 0.90). This impressive performance appears reasonably well-maintained in external data; however, there are a few issues with aspects of this study. Firstly, the proposed architecture uses a series of fully-connected layers followed by batch normalisation and dropout layers concurrently. It has previously been shown empirically and theoretically that combining batch normalisation and dropout produces worse performance.[41] Secondly, although data were collected from two countries, most COVID negative scans came from one country (81.4% from the US) while most COVID positive scans came from the other (97.7% from China). Additionally, all COVID negative scans were collected between 2017 and 2019, meaning that some or all negative scans were collected before the pandemic. These temporal and geographical differences between the COVID positive and COVID negative scans will likely add bias to the data.

COVNet also followed the slice-based approach using ResNet50 pretrained on ImageNet to extract features from each slice.[178] A max-pooling layer was then used to concatenate the features into a single feature vector. Finally, they used a fully-connected layer with sigmoid activation to estimate the probability of the scan showing COVID-19. The model was used to classify scans as COVID-19, community-acquired pneumonia, or non-pneumonia. They used 3,918 CT scans from 2,969 patients for training and 434 scans from 353 patients for testing. For predicting COVID-19, the authors reported an AUROC of 0.95 (95% CI: 0.93, 0.97), with a specificity of 0.90 (95% CI: 0.83, 0.94) and sensitivity of 0.96 (95% CI: 0.93, 0.98). Although images were collected from six separate centres, the images were pooled and then split into training, validation, and testing. As noted by the authors, the training and testing data came from the same hospitals. It may have been better to only use one centre for external validation to assess how well the model generalises.

The second method does enable the model to assess the spatial relationship between slices by concatenating the slices into a 3D volume. However, the slices do not form an actual 3D structure, especially when few slices are used. An example of a model developed using this approach is CoviNet,[179] which uses a 16-layer 3D CNN, followed by pooling and a fully-connected layer to obtain a probability. This approach was applied to a dataset obtained from the US consisting of 397 scans from 171 COVID-19 negative patients and 349 scans of 213 patients without COVID-19. The paper did not specify how the diagnosis was reached or if the COVID-19 patients had any other lung diseases visible on the scans. On this data, the model attained an accuracy of 0.75, sensitivity of 0.917, and specificity of 0.583. The model was also applied to the MosMed dataset and achieved an accuracy of 0.941, with a sensitivity of 0.922 and specificity of 0.971. Confidence intervals were not reported. The second dataset was not used for external validation, as the model was retrained on data from the new dataset.

These methods require the same number of slices, but the number of slices can differ in reality. If the scan has more slices than used in the model, then slices can be removed to reach the required size. However, some scans may have fewer than the required number of slices. Therefore, a new method was needed to account for the spatial relationship between slices without assuming a 3D structure, which could account for missing images if smaller scan sizes are used.

The main aims of the work presented in this chapter were:

(1) To develop a deep learning mixed-effects method capable of handling missing data, which accounts for the relationship between slices without assuming a 3D structure
(2) To demonstrate the method on a dataset of CT images
(3) To externally geographically validate the model
(4) To follow best practice guidelines for the development and validation of clinical prediction models to overcome some of the shortcomings of previously developed models

5.2 Methods

The method I developed begins by taking a CT volume with 20 slices, although any number of slices could be chosen, provided the number is kept constant within the dataset. Features are then extracted from each slice to obtain a feature vector. Like the models in Chapter 4, I used a CNN as a feature extractor. These feature vectors are then passed to a novel mixed-effects layer which models the spatial relationship between slices while also dealing with missing slices. The mixed-effects layer results in a single combined feature vector, which can be classified using a fully-connected layer. An overview of the method is shown in Figure 5.1.

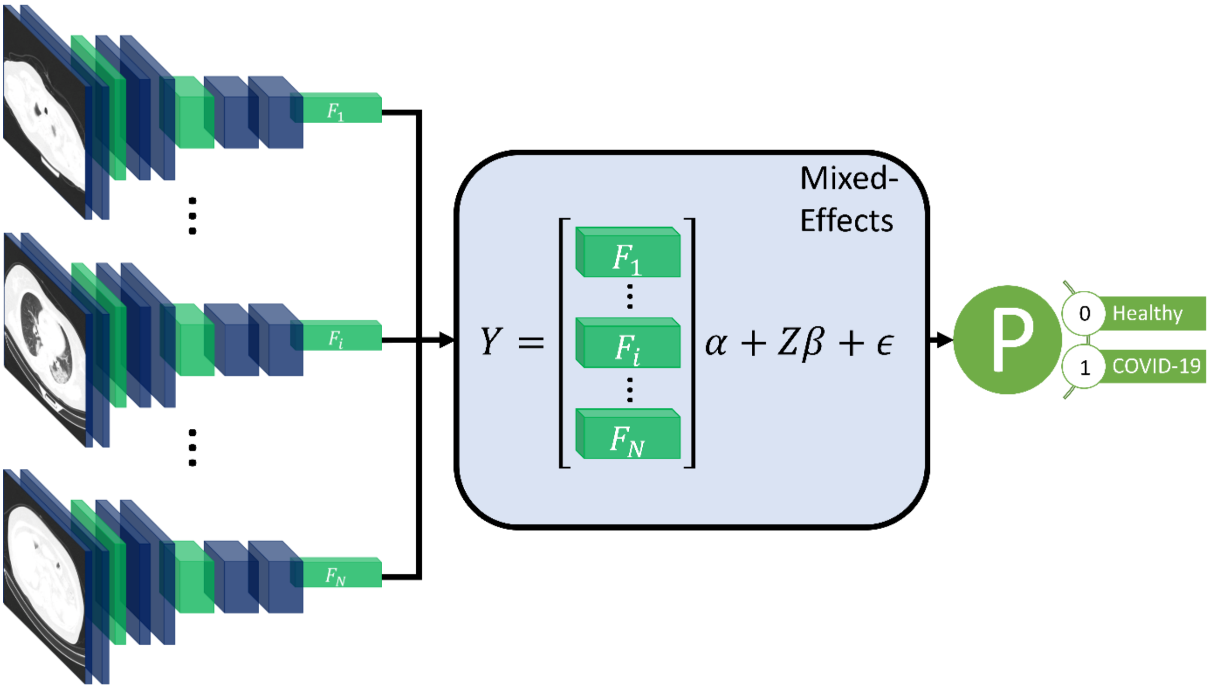

F_1
F_i
F_N
Mixed-Effects
$Y = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix} \alpha + Z\beta + \epsilon$
P
0 Healthy
1 COVID-19

Figure 5.1: The overall framework of the model.

5.2.1 Feature extractor

Several methods could be used to extract relevant features from the images. Here I chose to use the same pretrained CNN as in 0 4, Inception-v3. As previously described, Inception-v3 is relatively computationally efficient and highly generalisable to many applications. The parameters were pretrained on ImageNet to reduce the time taken to convergence.

One CNN was used for each slice, with the parameters shared across each CNN to reduce the computations in training the parameters. I then used average pooling to reduce the representations to a single feature vector for each slice. This resulted in a feature matrix of shape 20×2048. To prevent overfitting, I then applied dropout with a probability of 60%.

5.2.2 Mixed-effects

Previous methods of extracting features from the 2D slices would usually use a pooling layer to concatenate the features into a single vector. My approach aimed to consider the spatial relationship between slices using a mixed-effects model consisting of both fixed and random effects.

Mixed-effects models are commonly used in traditional statistics to model spatial relationships.[180] [181] For example, the fixed-effects part can model the relationship within the slices, while the random-effects part can model the spatial relationship between slices. This allows the spatial correlations to be modelled. Mixed-effects models can also handle missing data, provided the data are missing at random.

Mixed-effects models are of the form

$$Y_i = X_i\alpha + Z_i\beta + e_i. \tag{5.1}$$

In the above equation, Y_i is a vector of outcomes for the i^{th} scan. The fixed effects are modelled by $X_i\alpha$ where X_i is the design matrix of the i^{th} scan obtained from the feature extractor, and α is a vector of fixed effects parameters to be learned. The random effects are modelled by $Z_i\beta$ where Z_i is the random effects design matrix and β is a vector of random effects parameters to be learned. The vector e_i gives the unknown random errors of the i^{th} scan.

The fixed-effects design matrix can be constructed using the feature vectors extracted using the CNN. I then added a vector of ones for the intercept term. The fixed-effects design matrix is then given by

$$X = [\mathbf{1}, \boldsymbol{F1}, \boldsymbol{F2}, \boldsymbol{F3}], \tag{5.2}$$

where $\boldsymbol{F1}$, $\boldsymbol{F2}$, and $\boldsymbol{F3}$ are feature vectors extracted by the CNN.

There are many choices for the random-effects design matrix; in this work, I simply used an identity matrix of size 20×20, as I used 20 slices; however, this easily generalises to any chosen number of slices. I also added a vector of ones for the intercept. The random-effects matrix is

$$Z = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \tag{5.3}$$

The parameter vectors, α and β, need to be estimated as parameters in the deep learning network. Therefore, I randomly initialised the mixed-effects layer parameters using the Gaussian distribution with mean 0 and standard deviation 0.05.

There are two assumptions that are made by the mixed-effects model. Firstly, it is assumed that the random effects are normally distributed with mean 0 and some variance G, such that

$$\beta \sim N(0, G). \tag{5.4}$$

The second assumption is independence between the random effects and the error term.

One previous piece of work by Xiong et al. used a kind of mixed-effects model in deep learning for Gaze estimation. However, their work used the same design matrix for both the random- and fixed-effects. Additionally, the parameters were estimated in a two-step process. First, an expectation-maximisation algorithm was used to estimate the random-effects parameters, while the fixed-effects parameters were estimated using backpropagation. My model allows a different random-effects matrix to be used, and all parameters are estimated within the same deep learning framework.

5.2.3 Classification layer

The mixed-effects model results in a single vector of length 20, the same as the number of slices. This vector can be considered a feature vector with modelled spatial relationships. I used a fully-connected layer with sigmoid activation to obtain a probability of the scan displaying COVID-19. I also added L1 and L2 regularisation with values of 0.1 and 0.01 to the kernel to reduce overfitting. This gives a single value between 0 and 1, the estimated probability of COVID-19 being present in the scan.

5.2.4 Loss function

As with the models I presented in 0 4, I used the mean squared error with the balanced classes and rescaled targets, as previously described. I denote this loss L_{MSE}.

As previously mentioned, one of the assumptions made in mixed-effects models is that the random effects parameters, $\beta,$ are approximately normally distributed with mean zero. Initially, I assumed that the final learned parameters would satisfy this assumption; however, this could not be guaranteed. I then considered how I could enforce the normality of the parameters to ensure this assumption is met. My solution was to introduce a second loss function which encourages these parameters to satisfy that assumption. I output the random-effects parameters along with the predicted probability; I could then apply a loss function to this output. To enforce a mean of zero, I added a component to the loss function, which calculates the absolute value of the mean, $E(\beta)$. This penalises the algorithm if the mean of the parameters is further from zero. The normal distribution has a skewness of zero and a kurtosis of three (or excess kurtosis of zero). These values can be calculated by

$$Skew(\beta) = \frac{\sqrt{n(n-1)}}{n-2} \frac{E\left[(\beta - \bar{\beta})^3\right]}{\left(E\left[(\beta - \bar{\beta})^2\right]\right)^{3/2}} \tag{5.5}$$

and

$$Kurt(\beta) = \frac{1}{n^2} \sum_{i=1}^{n} \left(\frac{E\left[(\beta - \bar{\beta})^4\right]}{\left(E\left[(\beta - \bar{\beta})^2\right]\right)^2} \right), \tag{5.6}$$

where n is the number of parameters and $\bar{\beta}$ is the mean of the parameters. I added skewness and kurtosis components to this new loss function to push the learned random-effects parameter to follow an approximately normal distribution. The loss function for the random effects parameters was then given by

$$L_{fixed} = |E(\beta)| + |Skew(\beta)| + |Kurt(\beta) - 3|. \tag{5.7}$$

The final loss function is then

$$L = L_{MSE} + L_{fixed}. \tag{5.8}$$

A factor could be added to weight one part of the loss function more than the other; however, I weighted both losses equally in the total loss in this work.

5.2.5 Missing data

Some scans may have more or fewer slices. When there are more than the required number of slices, the slices can be uniformly sampled throughout the scan to obtain the required number of samples; however, when there are fewer than the required number of slices, there needs to be a method capable of handling missing images.

One option is to develop multiple models for different scan sizes. This option would require many models to be trained and is not practical. As previously discussed, one advantage of mixed-effects models is their ability to deal with missing data. Blank images can pad the scan at uniform intervals when faced with missing images. This is one reason why I chose to incorporate mixed-effects into my algorithm. I assessed the ability of my method to deal with different amounts of missing slices in scans.

5.2.6 Comparisons models

Many models have been developed for COVID-19 diagnosis, but there are too many to feasibly compare against them all. I compared my method against the three models described in the introduction to this chapter. As noted by Roberts et al.[176], the reporting and documentation of many models is insufficient, even when code is made available. Therefore, I attempted to recreate the models according to their descriptions and code, where it was available. All training and hyperparameter settings were kept the same across all experiments to make fair comparisons.

5.2.7 Performance measures

Many previous studies focus on assessing the discriminative performance of models with measures such as AUROC, sensitivity, and specificity. As I have discussed in 0 2, while discrimination is important, assessing the calibration of clinical models is vital.

I assessed the overall discriminative performance with the AUROC, using the pROC package in R. To calculate 95% confidence intervals, I used DeLong's method, which is a nonparametric asymptotically exact method for calculating AUROC confidence intervals. The AUROC assesses performance across all probability thresholds; the performance can also be assessed at specific thresholds. I assessed the sensitivity, specificity, PPV, and NPV at cut-off points of 0.3, 0.4, 0.5, 0.6, and 0.7. I used the epiR package in R to calculate these, with Jeffrey's prior used to calculate the 95% confidence intervals. I aimed to achieve moderate calibration to ensure that the model does not over- or underestimate the probabilities. Moderate calibration can be assessed using calibration curves. I used the CalibrationCurves package in R, which is based on the RMS package but adds 95% confidence intervals to the curves. Finally, I assessed the clinical usefulness of the model using decision curves. Decision curves show the net benefit of the proposed method over treating all or no patients as having COVID-19. I also used saliency maps to show which areas of the image were considered important by the algorithm in making the decision.

5.2.8 Training and inference

Training and hyperparameter tuning were performed on an Amazon Web Services p3.8xlarge node with four Tesla V100 16GiB GPUs and 244GiB of available memory. Inference and analysis were then performed on a Linux machine running Ubuntu 18.04, with a Titan X 12 GiB GPU and 32GiB of available memory. All models were developed using Tensorflow 2.4, with analysis performed using R 4.0.5. I reduced the computational cost of the models by using 16-bit floating-point precision, with only the mixed-effects and final fully-connected layer using 32-bit floating-point precision.

For training, I chose the Adam optimiser with an initial learning rate of 1e-4. If the validation loss did not improve within three epochs, I reduced the learning by 80%. I stopped training if the validation loss did not improve for ten epochs, so time and energy were not wasted.

5.2.9 Data

Sample size calculations are rare in deep learning. In the last chapter, I used a rule-of-thumb to estimate the number of samples needed in the testing dataset to assess the model calibration; however, it would be useful to know how many samples are needed to train the model. A systematic review of sample-size determination in machine learning highlighted that methods to calculate sample sizes in deep learning often rely on learning curve approaches.[182] These methods train multiple models on different amounts of data to assess how the model performance increases with increasing data. The performance can be plotted against dataset size to identify the required amount of data to achieve a required performance.[183] [184] This relies on having similar data available for a similar task. In many situations, suitable data may not be available.

The main barrier to applying traditional sample size calculations in deep learning is the number of parameters. For example, using Inception-v3 for a binary classification task with a prevalence of 50% and an expected c-statistic of 0.8 would require over half a billion samples according to the pmsampsize in R. This number is clearly too high and deep learning models are successfully trained using many fewer samples. In this work, I used a different approach that enables the sample size calculations from traditional statistical modelling. The final fully-connected layer could be treated as the classifier, with the other layers treated as the feature extractor. In this way, only the parameters in the final layer are included in the sample size calculation.

The model I proposed has 21 parameters in the final layer. Based on previous studies, I expected a disease prevalence of 80% and a c-statistic of around 0.8. This gave an estimated required sample size of 923. For validation, I aimed for 200 samples from each class.

In 0 3, I described the CT datasets used to demonstrate my developed method. First, I used data from the MosMed dataset for model training and internal validation, split into two-thirds for training and one-third for internal validation. The internal validation data is used to select the model during training, and the results on this set are likely to be biased. Second, I used the data from the Zhang et al. dataset for external geographic validation. A summary of the datasets and data splits is shown in Table 5.1.

Table 5.1: Summary of the CT datasets used to demonstrate the developed method.

Dataset	Location	Use	Healthy/COVID19
MosMed Training	Moscow, Russia	Training	169/856
MosMed Validation	Moscow, Russia	Internal Validation	85/285
Zhang et al.[156]	China	External Validation	243/553

To improve the generalisability of the model to unseen data, I applied data augmentation during training. I randomly adjusted the brightness and contrast to between 80% and 120% of the original values, randomly rotated the images by ± 5 degrees, and cropped the image by up to 20%. The values of these augmentations were chosen using a uniform distribution. Finally, I flipped the images horizontally and vertically, with a probability of 50% each, which was chosen using a random bit. All augmentations were performed on the scan level. All images were resized to 256×256 pixels, and pixel values were divided by 255 to normalise the image between 0 and 1.

In this work, I used 20 slices from each scan. If scans had greater than 20 slices, I uniformly sampled slices from the scan to obtain 20 slices total. For example, one scan in the external validation dataset had 19 slices, and a blank slice was added to make 20 slices.

5.3 Results

In this section, I present the results of my developed method on both the internal validation dataset and the external geographical validation dataset. I also present saliency maps for some of the images to display how the algorithm identifies the correct areas of the image displaying COVID-19. I then briefly perform two sensitivity analyses to assess the effects of missing and noisy data on the algorithm.

5.3.1 Internal validation

First, I present results on the internal validation set. At the end of each training epoch, the loss on this dataset was calculated. The model that achieves the best loss was chosen as the final model; therefore, the results on this dataset are biased and may not represent how the model performs on unseen data. The full results are shown in Table 5.2.

On the internal validation set, the mixed-effects method achieved an AUROC of 0.936 (95% CI: 0.910, 0.961). This AUROC is a statistically significant increase over the method by Bai et al. with an AUROC of 0.731 (95% CI: 0.674, 0.80) and CoviNet with an AUROC of 0.801 (95%CI: 0.748, 0.853). However, there was no significant difference in AUROC compared to CovNet, which attained an AUROC of 0.935 (95% CI: 0.912, 0.959). ROC curves are shown in Figure 5.2. The calibration curves in Figure 5.3 show that the mixed-effects model is reasonably well-calibrated, although some recalibration may be beneficial. The comparison models all show inadequate calibration. In Figure 5.4, I show a decision curve for the mixed-effects model. The decision curve shows that the model improved net benefit over the treat-all or treat-none approaches.

5.3.2 External validation

The same model trained on the MosMed data was then used to classify the Zhang et al. dataset images. The external dataset is taken from a separate country, making it an external geographical dataset. Results on this dataset will be less biased and show how well the model generalises to other settings. The full results are shown in Table 5.3.

On the external validation dataset, my model had an AUROC of 0.930 (95% CI: 0.914, 0.947). As with the internal validation data, my model achieved a statistically significant improvement over the Bai et al. model, with an AUROC of 0.805 (95% CI: 0.774, 0.836) and CoviNet with an AUROC of 0.651 (95% CI: 0.615, 0.691). However, unlike the internal validation data, there was also a statistically significant improvement over the CovNet model, with an AUROC of 0.808 (95% CI: 0.775, 0.841). ROC curves are shown in Figure 5.5. The calibration curves in Figure 5.6 show similar results to the internal validation results. The mixed-effects model shows calibration close to the perfect calibration, with the Bai et al., CoviNet, and CovNet models showing poor calibration. Similar to the internal validation results, the decision curve in Figure 5.7 shows that the mixed-effects model provided an improved net clinical benefit over the treat-all or treat-none approaches.

Table 5.2: Performance on the internal validation dataset.

Model	AUROC	Threshold	Sensitivity	Specificity	PPV	NPV
Bai et al.	0.731 (0.674, 0.80)	0.3	0.0 (0.0, 0.042)	1.0 (0.987, 1.0)	NA	0.77 (0.724, 0.812)
		0.4	0.012 (0, 0.064)	0.996 (0.981, 1.0)	0.50 (0.013, 0.987)	0.772 (0.725, 0.814)
		0.5	1.0 (0.958, 1.0)	0.0 (0.0, 0.013)	0.230 (0.188, 0.276)	NA
		0.6	1.0 (0.958, 1.0)	0.0 (0.0, 0.013)	0.230 (0.188, 0.276)	NA
		0.7	1.0 (0.958, 1.0)	0.0 (0.0, 0.013)	0.230 (0.188, 0.276)	NA
CoviNet	0.801 (0.748, 0.853)	0.3	0.459 (0.350, 0.570)	0.898 (0.857, 0.931)	0.574 (0.448, 0.693)	0.848 (0.802, 0.886)
		0.4	0.706 (0.597, 0.80)	0.761 (0.708, 0.810)	0.469 (0.380, 0.559)	0.897 (0.851, 0.932)
		0.5	0.824 (0.726, 0.898)	0.596 (0.537, 0.654)	0.378 (0.308, 0.452)	0.919 (0.870 0.954)
		0.6	0.918 (0.838, 0.966)	0.446 (0.387, 0.505)	0.331 (0.271, 0.394)	0.948 (0.895, 0.979)
		0.7	0.965 (0.90, 0.993)	0.246 (0.197, 0.30)	0.276 (0.226, 0.331)	0.959 (0.885, 0.991)
CovNet	0.935 (0.912, 0.959)	0.3	0.941 (0.868, 0.981)	0.839 (0.791, 0.879)	0.635 (0.544, 0.719)	0.98 (0.953, 0.993)
		0.4	0.965 (0.90, 0.993)	0.825 (0.775, 0.867)	0.621 (0.533, 0.704)	0.987 (0.964, 0.997)
		0.5	1.0 (0.958, 1.0)	0.796 (0.745, 0.842)	0.594 (0.509, 0.676)	1.0 (0.984, 1.0)
		0.6	1.0 (0.958, 1.0)	0.779 (0.726, 0.826)	0.574 (0.490, 0.655)	1.0 (0.984, 1.0)
		0.7	1.0 (0.958, 1.0)	0.761 (0.708, 0.810)	0.556 (0.473, 0.636)	1.0 (0.984, 1.0)
Mixed-Effects (Ours)	0.936 (0.910, 0.961)	0.3	0.588 (0.476 0.694)	0.961 (0.932, 0.981)	0.820 (0.70, 0.906)	0.887 (0.846, 0.920)
		0.4	0.659 (0.548, 0.758)	0.933 (0.898, 0.959)	0.747 (0.633, 0.840)	0.902 (0.862, 0.933)
		0.5	0.753 (0.647, 0.840)	0.909 (0.869, 0.940)	0.711 (0.606, 0.802)	0.925 (0.888, 0.953)
		0.6	0.812 (0.712, 0.888)	0.884 (0.841, 0.919)	0.676 (0.577, 0.766)	0.940 (0.905 0.960)
		0.7	0.906 (0.823 0.958)	0.832 (0.783, 0.873)	0.616 (0.525, 0.702)	0.967 (0.937, 0.986)

Table 5.3: Performance on the external validation dataset.

Model	AUROC	Threshold	Sensitivity	Specificity	PPV	NPV
Bai et al	0.805 (0.774, 0.836)	0.3	0.0 (0.0, 0.015)	1.0 (0.993, 1.0)	NA	0.695 (0.661, 0.727)
		0.4	0.0 (0.0, 0.015)	1.0 (0.993, 1.0)	NA	0.695 (0.661, 0.727)
		0.5	1.0 (0.985, 1.0)	0.0 (0.0, 0.007)	0.305 (0.273, 0.339)	NA
		0.6	1.0 (0.985, 1.0)	0.0 (0.0, 0.007)	0.305 (0.273, 0.339)	NA
		0.7	1.0 (0.985, 1.0)	0.0 (0.0, 0.007)	0.305 (0.273, 0.339)	NA
CoviNet	0.651 (0.610, 0.691)	0.3	0.0 (0.0, 0.015)	1.0 (0.993, 1.0)	NA	0.695 (0.661, 0.727)
		0.4	0.0 (0.0, 0.015)	1.0 (0.993, 1.0)	NA	0.695 (0.661, 0.727)
		0.5	0.008 (0.001, 0.029)	0.991 (0.979, 0.997)	0.286 (0.037, 0.710)	0.695 (0.661, 0.727)
		0.6	0.160 (0.117, 0.213)	0.929 (0.905, 0.949)	0.50 (0.385, 0.615)	0.716 (0.681, 0.749)
		0.7	0.551 (0.487, 0.615)	0.694 (0.654, 0.733)	0.442 (0.385, 0.50)	0.779 (0.740, 0.815)
CovNet	0.808 (0.775, 0.841)	0.3	0.305 (0.247, 0.367)	0.969 (0.951, 0.982)	0.813 (0.718, 0.887)	0.760 (0.727, 0.791)
		0.4	0.354 (0.294, 0.418)	0.955 (0.934, 0.971)	0.775 (0.686, 0.849)	0.771 (0.737, 0.802)
		0.5	0.387 (0.325, 0.451)	0.940 (0.917, 0.959)	0.740 (0.655, 0.814)	0.777 (0.744, 0.808)
		0.6	0.432 (0.369, 0.497)	0.937 (0.913, 0.956)	0.750 (0.670, 0.819)	0.790 (0.756, 0.820)
		0.7	0.473 (0.409, 0.538)	0.931 (0.907, 0.951)	0.752 (0.675, 0.818)	0.801 (0.768, 0.831)
Mixed-Effects (Ours)	0.930 (0.914, 0.947)	0.3	0.675 (0.612, 0.733)	0.935 (0.911, 0.954)	0.820 (0.760, 0.871)	0.867 (0.838, 0.894)
		0.4	0.741 (0.681, 0.795)	0.904 (0.877, 0.927)	0.773 (0.713, 0.825)	0.888 (0.859, 0.913)
		0.5	0.778 (0.720, 0.828)	0.882 (0.853, 0.908)	0.744 (0.686, 0.797)	0.90 (0.872, 0.924)
		0.6	0.827 (0.774, 0.873)	0.859 (0.827, 0.887)	0.720 (0.664, 0.772)	0.919 (0.892, 0.941)
		0.7	0.885 (0.838, 0.922)	0.828 (0.794, 0.859)	0.694 (0.639, 0.744)	0.942 (0.918, 0.961)

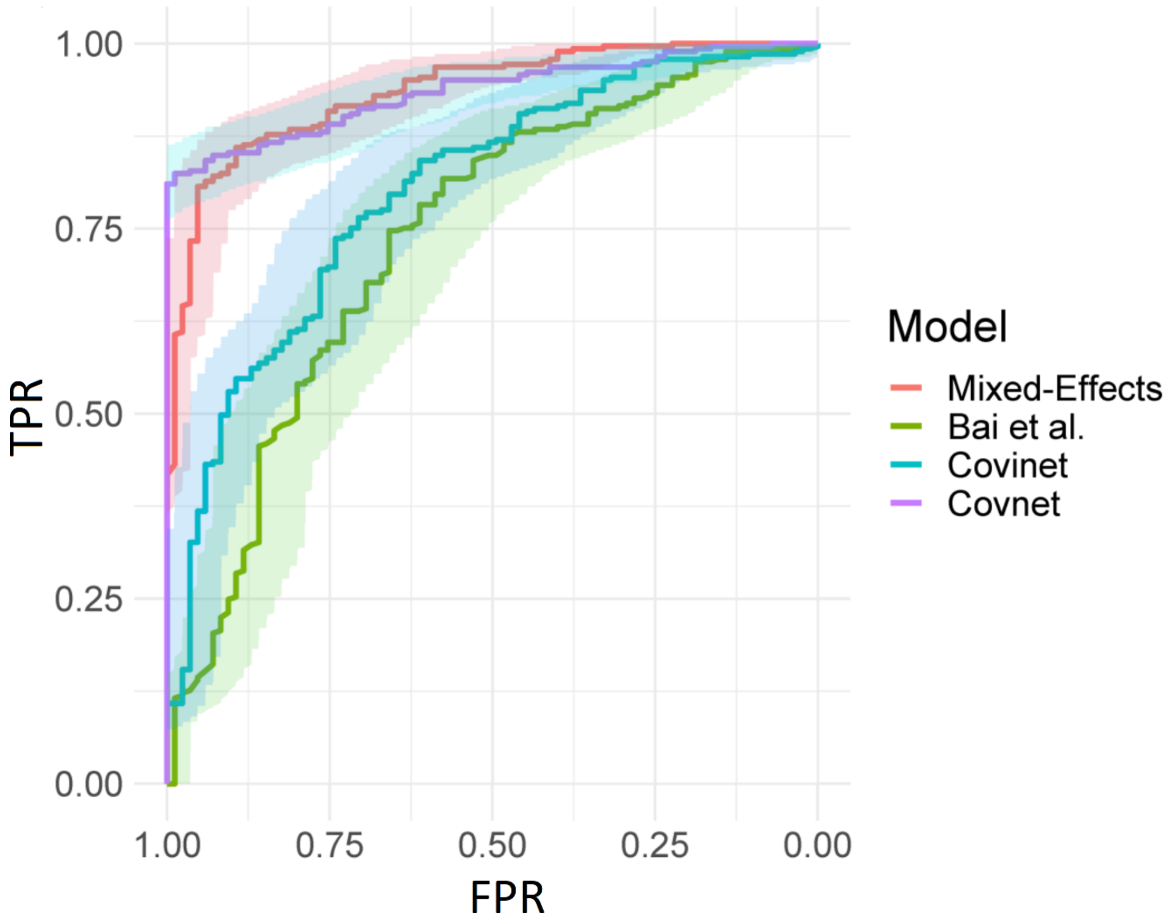


Figure 5.2: ROC curves for the internal validation dataset. The confidence bands showed that the mixed-effects and Covnet models performed significantly better than the other two.

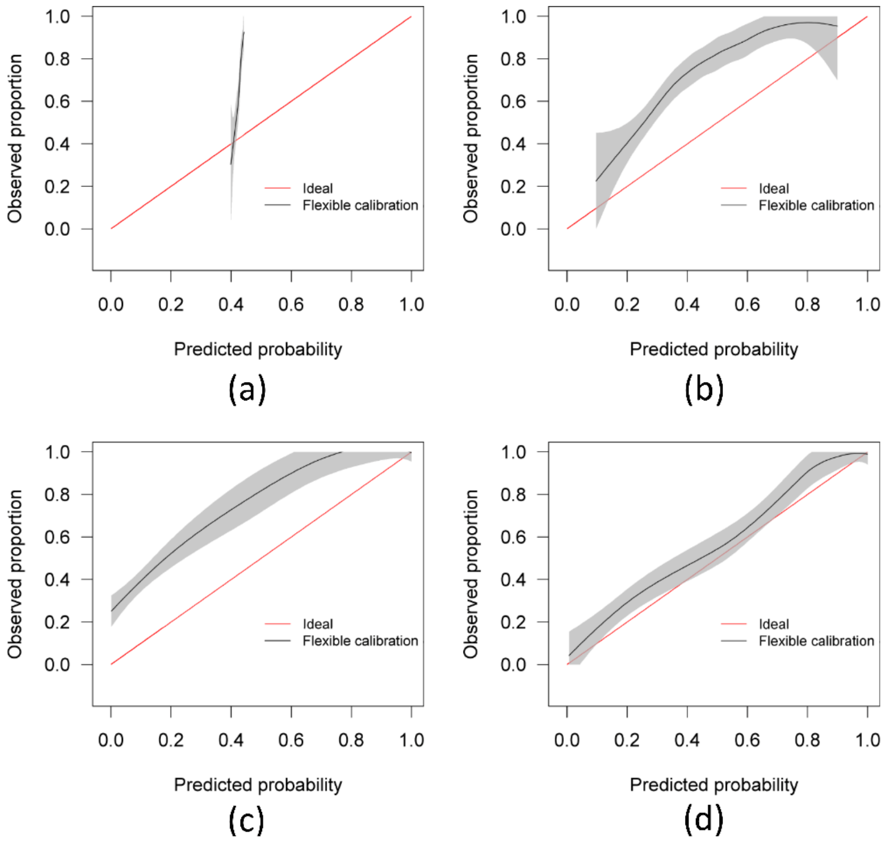


Figure 5.3: Calibration curves on the internal validation dataset for (a) the Bai et al. model, (b) the CoviNet model, (c) the Covnet model, and (d) the proposed mixed effects model.

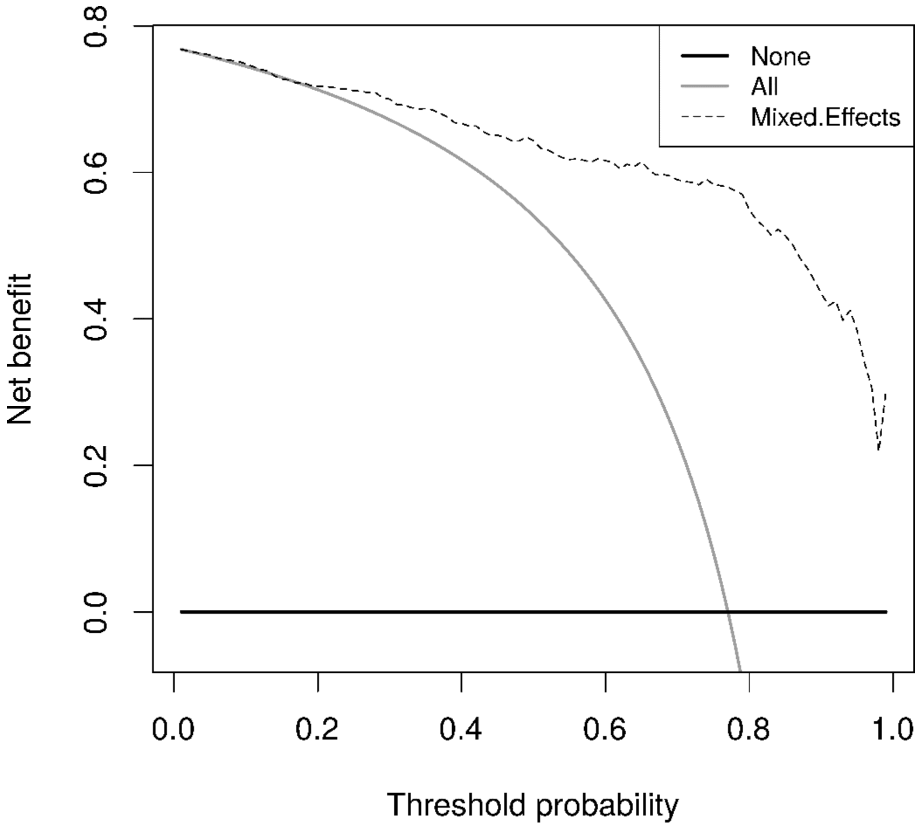


Figure 5.4: Decision curve for the mixed-effects model on the internal validation dataset.

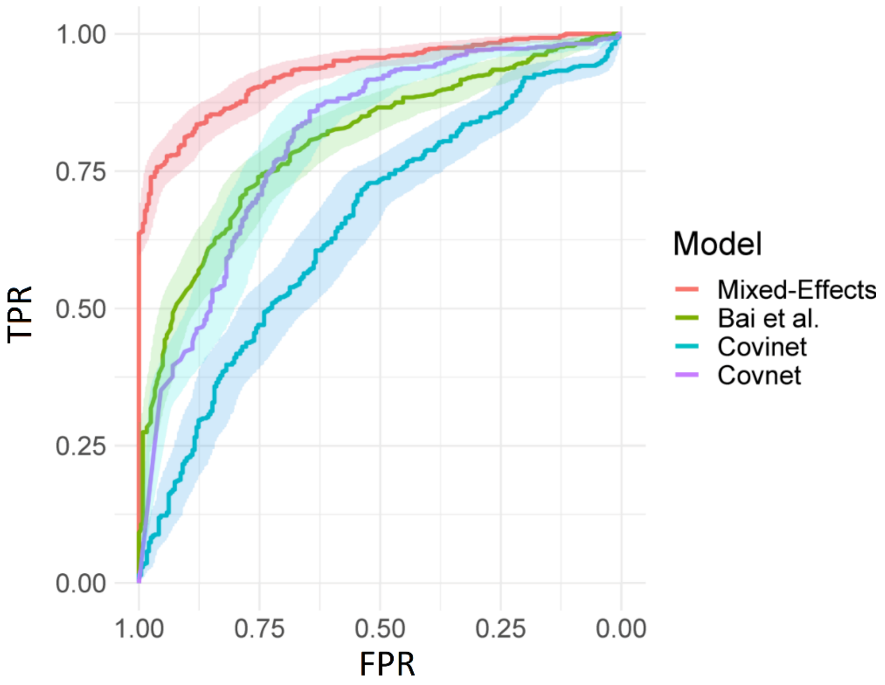


Figure 5.5: ROC curves for the external validation dataset. The proposed mixed-effects model appears to perform significantly better than the other models.

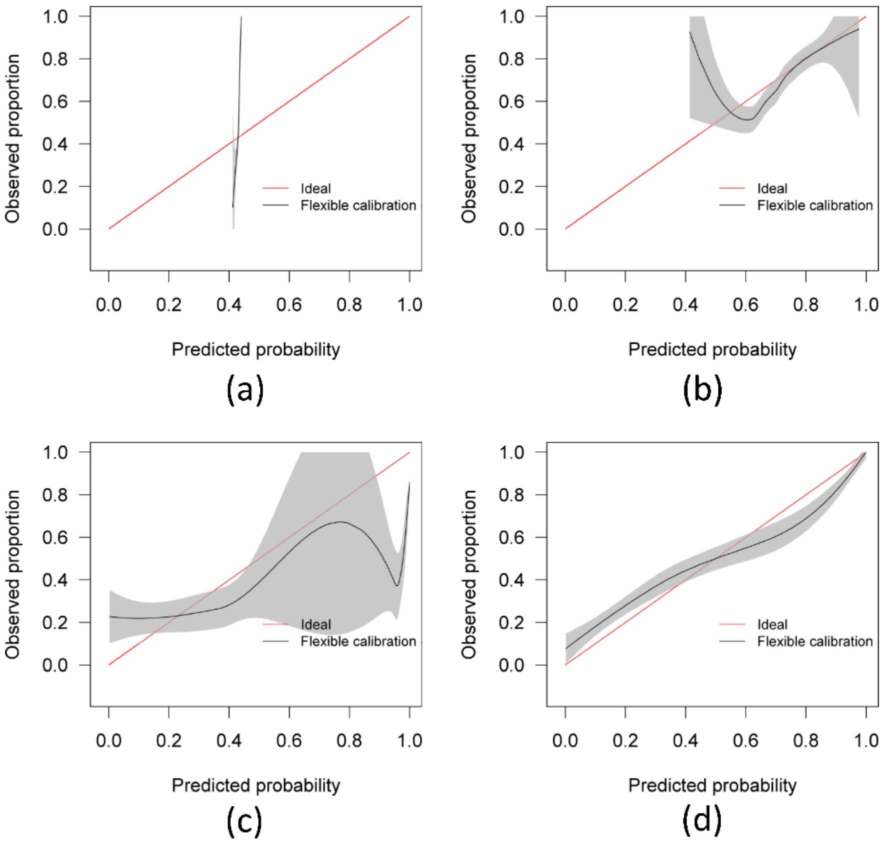


Figure 5.6: Calibration curves on the external validation dataset for (a) the Bai et al. model, (b) the CoviNet model, (c) the Covnet model, and (d) the proposed mixed effects model.

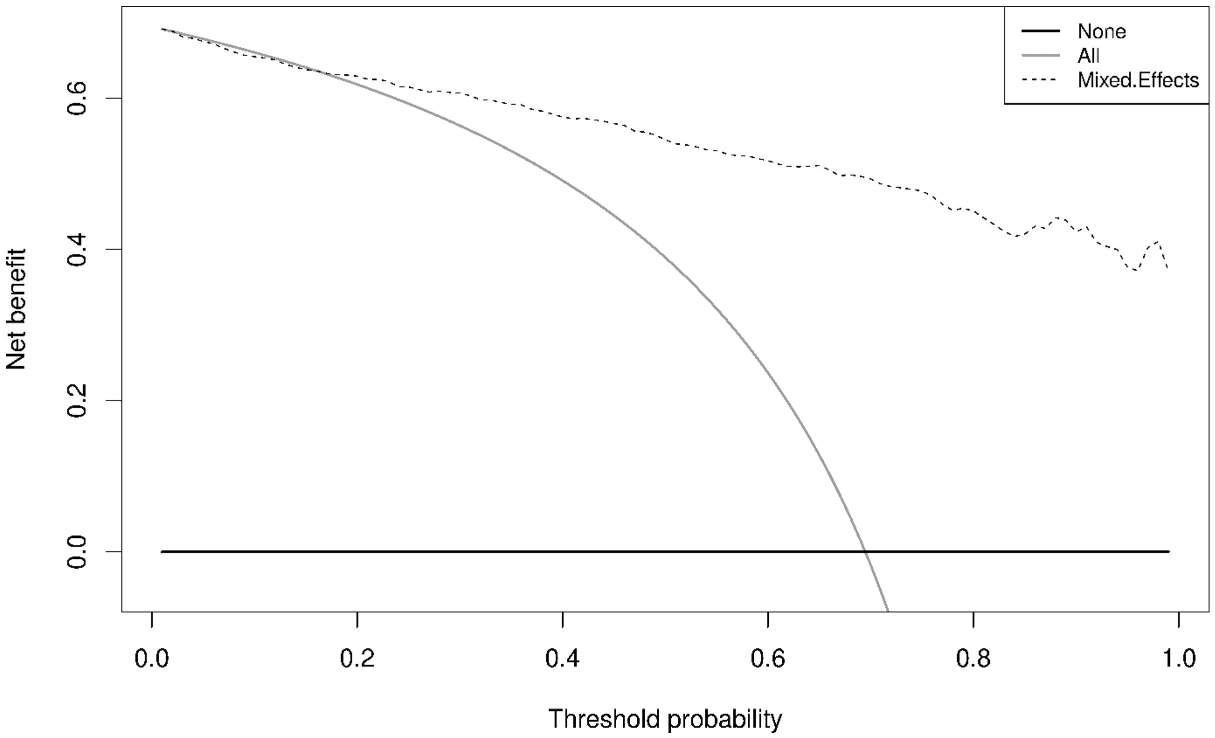


Figure 5.7: Decision curve for the mixed-effects model on the external validation dataset

5.3.3 Saliency maps

Saliency maps of four scans using the mixed-effects model from the external validation dataset are shown in Figure 5.8. I used SmoothGrad to create these maps,[82] with 20 samples and a noise of 0.2. Four consecutive slices demonstrate how areas of disease move through the scans. The model correctly identified diseased areas of the scans. This suggests that the model learned the correct features.

5.3.4 Missing data

To assess the model's capability to account for missing data, I removed slices from the external validation scans at regular intervals and calculated the AUROC. The model was not retrained to deal with missing data explicitly. I assessed performance at levels of missingness from 0 to 50% in 5% increments. The results in Figure 5.9 show that my mixed-effects method can reasonably handle missing data. At 20% missingness, there was a statistically significant decrease in the AUROC; however, even at 50% missingness, the AUROC was 0.890 (95% CI: 0.868, 0.912).

5.3.5 Random noise

Deep learning models using imaging data may be vulnerable to adversarial attacks, with small amounts of noise in the image causing the model to make wrong decisions with falsely high confidence. I performed a sensitivity analysis on my model by adding a small amount of random noise to the external validation dataset and calculating the AUROC that my model achieves. I added small Gaussian noises to the normalised images with mean 0 and standard deviations from 0 to 0.005 in increments of 0.001. Example images with noise are shown in Figure 5.10, with the AUROC values shown in Figure 5.11. The AUROC drop steadily with increasing levels of noise. Algorithms using CT scans may be particularly susceptible to adversarial attacks of random noise as small changes in lung appearance may be difficult to see. Although label noise may have been a more realistic and more interesting noise to investigate, I did not have chance to test that type of noise in my work.

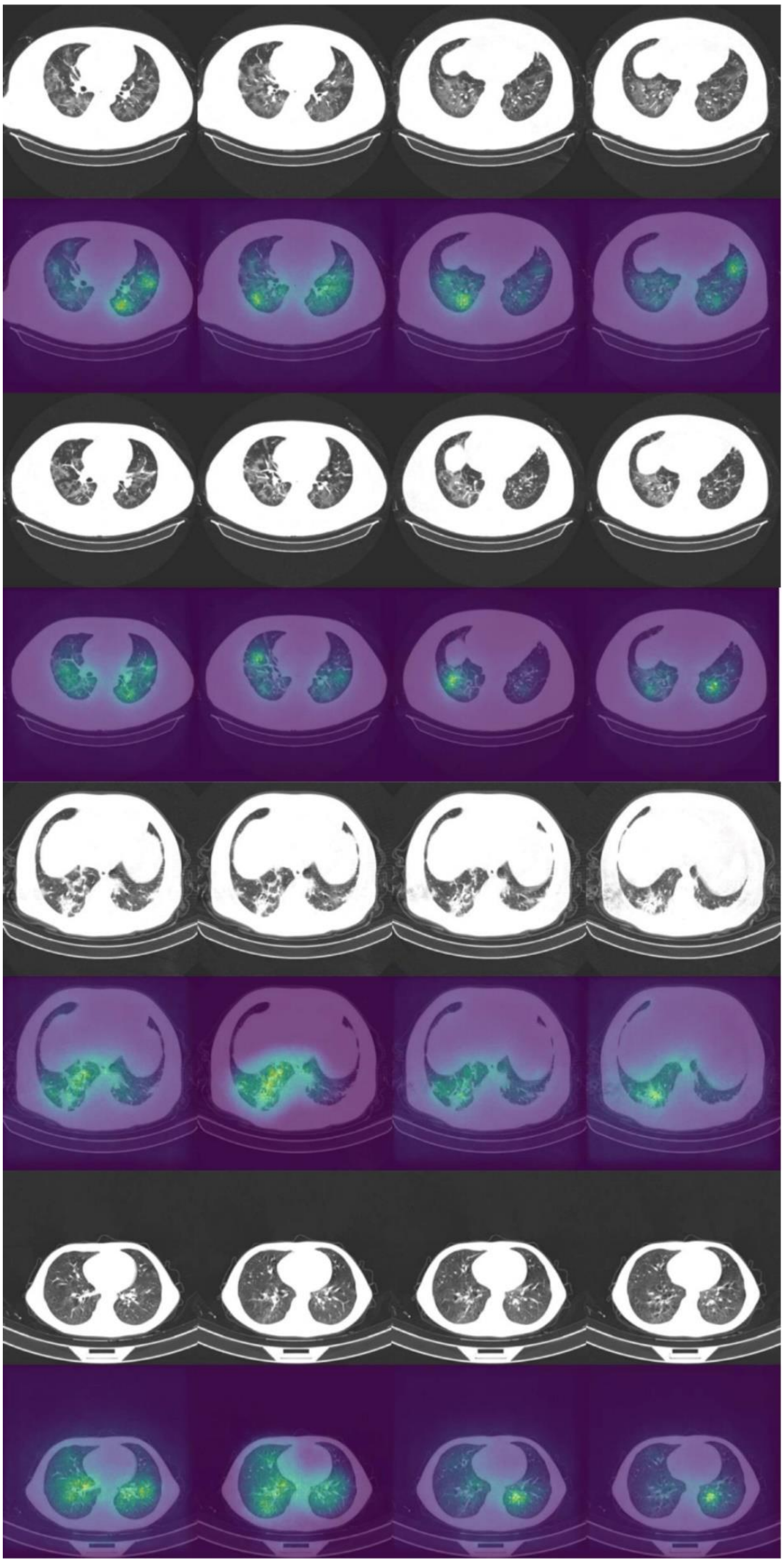


Figure 5.8: Saliency maps of four COVID-19 positive scans at four consecutive slices.

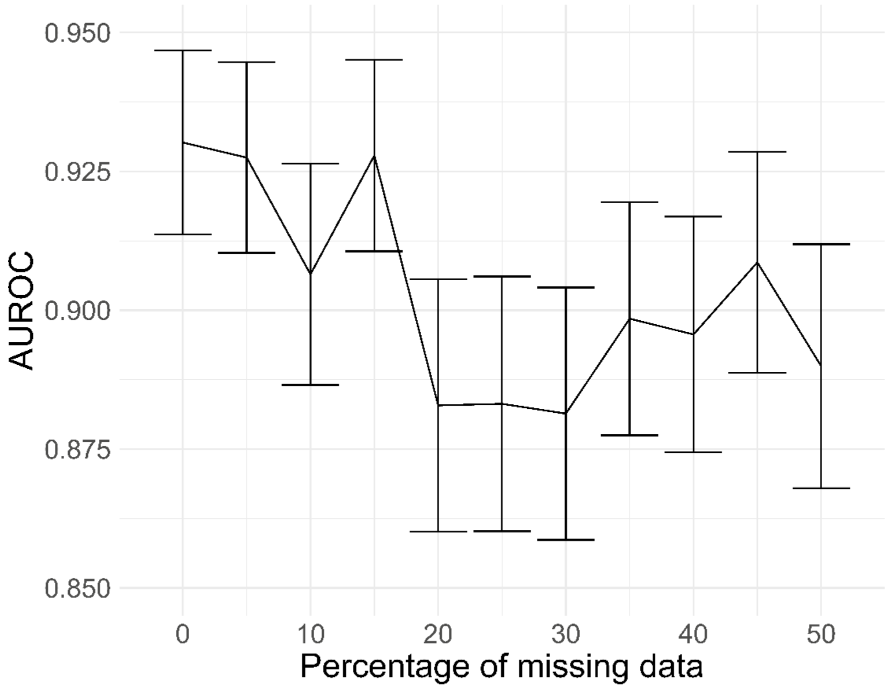


Figure 5.9: Graph showing AUROC values attained by the mixed-effects model on the external validation dataset at different levels of missingness.

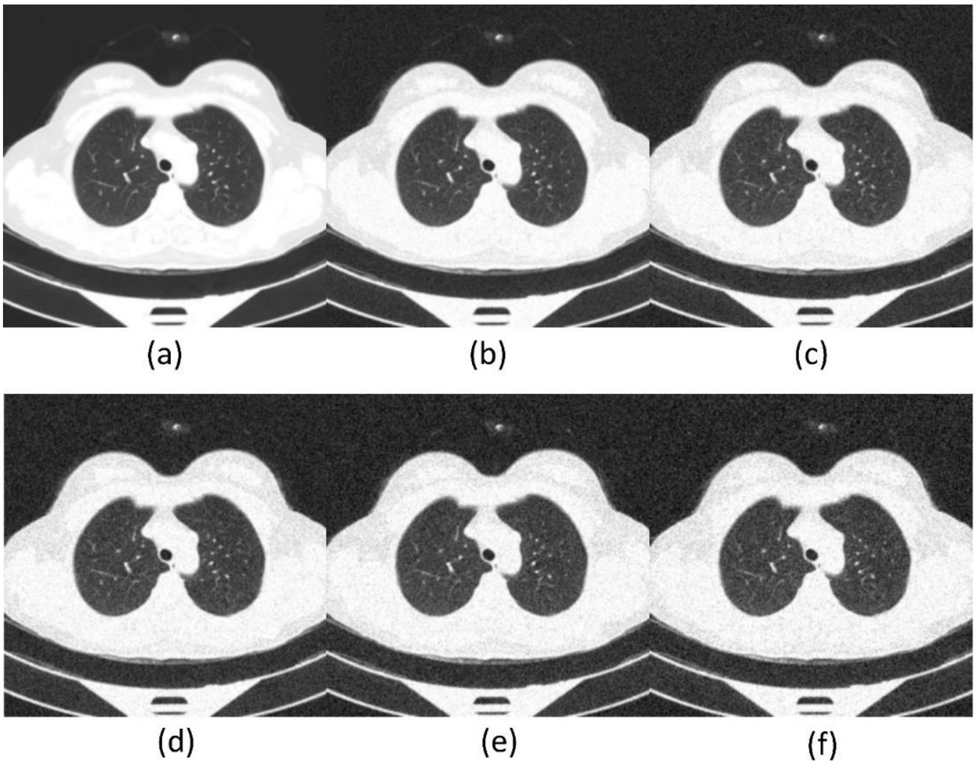


Figure 5.10: Example images from the external validation set with increasing noise levels. (a) sd = 0, (b) sd = 0.001, (c) sd=0.002 (c) sd=0.003, (d) sd = 0.004, and (f) sd=0.005.

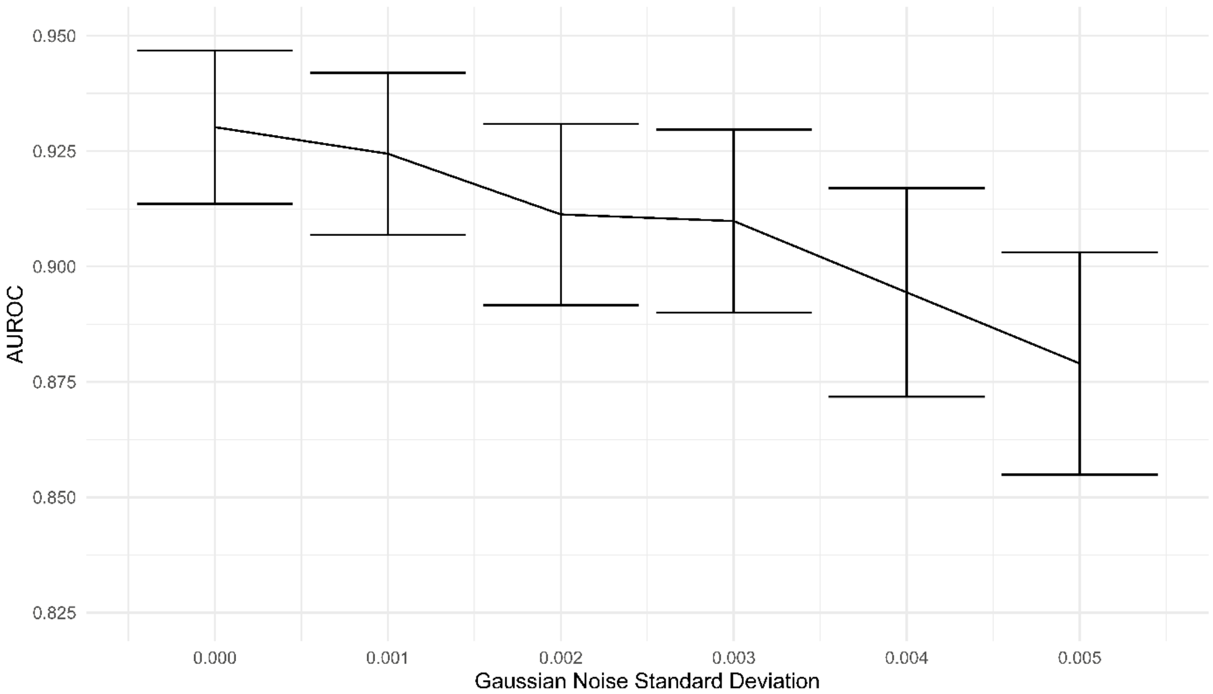


Figure 5.11: Graph showing AUROC values attained by the mixed-effects model on the external validation dataset at different levels of random noise.

5.4 Discussion

The results presented in this chapter demonstrate the ability of my mixed-effects method to distinguish between healthy and COVID-19 CT scans with an AUROC of 0.930 (0.914, 0.947) on the external validation set. The developed model could accurately detect COVID-19 on CT scans and shows some improvement over previous methods. Even when removing up to 50% of the slices, the model performed reasonably well, suggesting that the mixed-effects model could handle missing data appropriately.

This work highlights the importance of checking model calibration. Although the published models used for comparison models showed good discriminative performance, they all had poor calibration, suggesting they are unsuitable for clinical use. The calibration curve for the Bai et al. model shows that all predictions are close to 0.5. This poor calibration may be due to the combined effect of batch normalisation and dropout, which has been shown to lead to worse performance.

Two of the three comparator studies reported good performance on external validation sets; however, there were significant problems with those sets, leading to bias. The original Bai et al. study used data with mainly positive scans from one country and negative scans from another country. The algorithm may pick up minor variations in the scans between countries, and the algorithm could be classifying the country rather than the disease. In the CoviNet paper, the model was retrained on a subset of the external validation data meaning it is impossible to know how the model truly generalises to completely external data. In my work, I did not mix datasets; training/internal validation data were taken from a consortium of hospitals in Russia, while external validation datasets were taken from a consortium of hospitals in China. This helped reduce bias by keeping the samples within splits from similar hospitals taken at similar times. Additionally, the external validation data were only used for the external validation with no retraining. Therefore, the results presented in my work may give a less biased idea of how the models may perform in a completely new setting and may explain the widely different results reported in other studies.

At first, I used the Zhang et al. data for training and the MosMed dataset for external validation. This gave excellent results; however, the saliency maps showed that the algorithm was looking at the outside of the image rather than the lungs, suggesting some bias within the dataset. This shows the importance of model explainability in deep learning. Saliency maps provide an easily interpretable representation of what the algorithm looks at within the image and allow the user to check for bias within the dataset. Producing some kind of saliency map of class activation map is essential for ensuring that the algorithm works as intended.

5.4.1 Limitations

There are several ways in which my work could be improved. Firstly, the loss function to ensure the random-effects parameters follow an approximately normal distribution uses the absolute value. Using the squared value may provide better results and could be explored in future work. In this work, I weighted both the loss functions equally. Introducing a factor which weights one of the losses may provide better results or faster convergence.

As I have discussed, there is no established method of sample size calculation for deep learning models. Previous methods relied on having large amounts of similar data for similar tasks to produce learning curves. This is impractical on new tasks when it is unclear how similar the task is to others. It is possible to look at other work in similar areas to assess how much data may be needed; however, even if similar models have been developed, the model may be too different to compare. The method I used here is based on established sample size methods in statistics.[185] These methods take the number of final model parameters into account. The model parameters act differently in traditional statistical models and deep learning models. It is common for deep learning models to contain millions of parameters without suffering from the same issues that traditional statistical models may encounter. As mentioned in the methods section, using the usual sample size calculation would recommend a minimum of over half a billion samples; however, deep learning models are often trained using far fewer samples. Treating the final fully-connected layer of the model as the classification layer and using those parameters in the calculation allows for more reasonable sample size estimations. Based on using 21 parameters, an estimated disease prevalence of 80%, and a conservative expected C-statistic of 0.8, I estimated a sample size of 923 would be needed for model training; however, the sample size calculation method was created for traditional statistical models. This sample size appears reasonable in this work, and the mixed-effects model was trained sufficiently with good results on the external validation dataset. More work is needed to assess whether only using the parameters in the final classification layer provides valid sample sizes.

5.4.2 Future work

I have evaluated the trained model on a dataset from a separate dataset from a different country. This external geographical validation shows that the model can be applied to datasets outside the sample population. However, it is vital to robustly evaluate the model in every setting it is intended to be used in. The images from both datasets were collected on overlapping dates. As new variants of COVID-19 have emerged and are likely to continue, it is important to assess if these changes affect the model's performance. In other applications, advances in imaging technology or protocols may also affect the model performance. External temporal validation using more recent images is needed to assess if the model needs updating with time.

The results obtained by the algorithm may appear to show improved sensitivity over PCR testing; however, it is impossible to make this conclusion without comparing both my algorithm and the PCR testing on the same set of patients. Future work may look into this to test whether CT imaging is more appropriate than PCR. One advantage of CT imaging and a diagnostic algorithm is that some human error in extracting a sample from a patient is reduced. However, CT imaging PCR testing will likely be used in different scenarios.

5.5 Conclusions

In this chapter, I have presented my mixed-effects model for COVID-19 diagnosis. The model showed improved performance over three previously published models for COVID-19 diagnosis. I have also shown the importance of robust analysis of results, which includes assessing with various measures.

Although this work does not directly relate to my central aims, the mixed-effects method I developed will form part of my survival model in the next section. This chapter has acted as an ablation study enabling me to test the mixed-effects model. The results presented here demonstrate that the mixed-effects layer works and may be able to deal with missing data.

Chapter 6: Survival Model

In this chapter, I incorporate the mixed-effects layer from Chapter 5 into a survival model to help account for missing data. I then add a survival model to the network to predict the probability of progression at any chosen time point. This model is similar to a joint model in traditional statistics and can handle both missing and right-censored data.

6.1 Introduction

The models I presented in Chapter 4 obtain good discriminative performance and can accurately predict progression to advanced AMD. However, there are two challenges that the previous models fail to address. Firstly, there is likely to be missing data due to patients not having enough visits or images from previous visits being lost or corrupted. Secondly, as AMD is a degenerative disease, it is likely that all patients could progress to advanced AMD given enough time.

Although I have shown in Chapter 4 that a single time point may be sufficient for predicting AMD progression up to three years, this may not always be true. In applications where additional time points are useful, it is important to account for situations where they are missing.

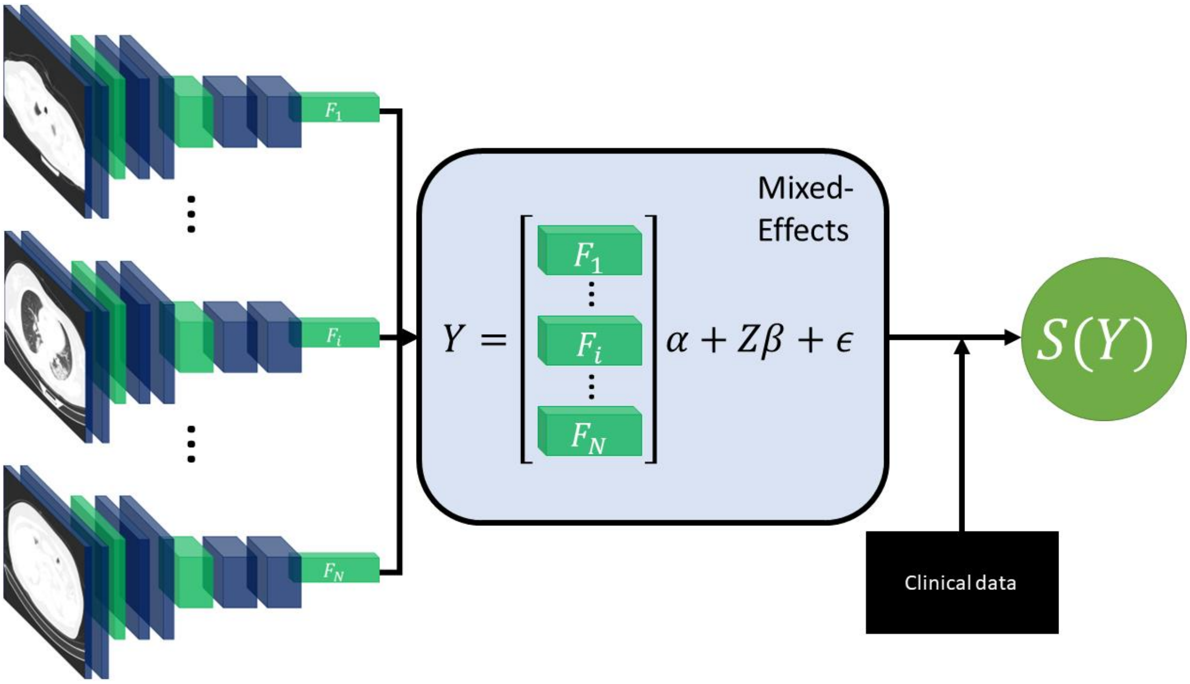
In Chapter 5, I proposed a method to deal with missing data and showed that removing 50% of the images can result in good model performance. Survival models can be used to account for right-censored data, where the patient is not observed to progress in the follow-up time. A loss function can be used to account for the right-censoring.

In traditional statistics, a joint model combines mixed-effects and survival models to create a longitudinal prognostic model. In this chapter, I present my work that combines my mixed-effects layer with a survival model similarly; however, I incorporate it all within the deep learning framework and a CNN to automatically extract features from images.

The work presented in this chapter aimed to create a model that accounts for missing visits and right-censored data. As with the models in Chapter 4, this model also accounts for uneven intervals between visits; however, the mixed-effects model accounts for the uneven intervals instead of using my interval scaling technique.

6.2 Methods

My proposed method consists of three main stages. Similar to my previous models in Chapter 4, I begin by using a CNN to extract features from each image. This results in a feature vector for each image. A mixed-effects model concatenates the feature vectors into a single vector. The mixed-effects layer accounts for the missing observations and the variable times between observations. Clinical data can be incorporated into the model at his point by appending the data onto the single vector. A survival model then estimates the survival probability. An overview of the whole method is shown in Figure 6.1.



F_1
F_i
F_N
Mixed-Effects
$Y = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix} \alpha + Z\beta + \epsilon$
$S(Y)$
Clinical data

Figure 6.1: Overview of the model architecture.

6.2.1 CNN feature extractor

Similar to the GRU and VAR models in 0 4, this method uses a CNN to extract features for each image. Image features are extracted using a CNN, with a CNN with shared weights for each image. In this work, I used Inception V3, which is highly generalisable and applicable to many different image types. I used weights pretrained on ImageNet to reduce the time taken to convergence. After the final convolution, global average pooling is applied to produce a single feature vector of length 2048 for each image. I applied a dropout of 0.6 after pooling to reduce overfitting.

6.2.2 Mixed-effects layer

The mixed-effects layer works similarly to the one I proposed in Chapter 5. In this chapter, I model the temporal relationship instead of the spatial relationship. The random-effects design matrix Z is changed so that it models the relationship between time points

$$Z = \begin{bmatrix} 1 & 0 & \frac{1}{t_1 - t_2} & \frac{1}{t_1 - t_3} \\ 1 & \frac{1}{t_2 - t_1} & 0 & \frac{1}{t_2 - t_3} \\ 1 & \frac{1}{t_3 - t_1} & \frac{1}{t_3 - t_2} & 0 \end{bmatrix}, \qquad (6.1)$$

where t_1, t_2 and t_3 are the times of the three previous observations. I rescaled the time points such that $t_1 = 0$ for the baseline time. The mixed-effects layer results in a single vector, with the relationships between time points modelled using the random-effects part.

The random-effects design matrix is more complicated than the one I used in Chapter 5 for two main reasons. Firstly, the distance between the observations is no longer uniform, and the model must now account for this. Secondly, time points may occur before others. In this design matrix, some values may be negative to account for this.

6.2.3 Survival layer

A survival model estimates the probability of an event occurring up to a certain time. The event could be death, the need for intervention, or the progression to the next stage of disease. These models can give the likely prognosis over time for a patient. In this work, I used a specific type of survival model known as a proportional hazards model[130]. In this section, I use Greek letters to indicate parameters that are to be estimated.

The probability of a patient experiencing the event at exactly time t is called the hazard function and is given by the instantaneous hazard function

$$h(t) = h_0(t)e^{\beta' x}, \tag{6.2}$$

where $h_0(t)$ is a baseline hazard function, x is a vector of covariates, and β is a vector of parameters. The hazard function gives the instantaneous death rate for a patient conditional on surviving up to time t. The cumulative hazard function can be calculated using

$$H(t) = \int_0^t h(u)\, du. \tag{6.3}$$

Then the probability of the patient surviving to time t is

$$S(t) = P(T \geq t) = exp\{-H(t)\}, \tag{6.4}$$

conversely, the probability of having failed by time t is

$$F(t) = P(T < t) = 1 - S(t). \tag{6.5}$$

There are many options for the baseline hazard function, $h_0(t)$. In this work, I consider the Exponential,[130] Weibull,[186] and Gompertz[187] distribution and compare their performance.

The Exponential survival model is the simplest; the baseline hazard function is given by

$$h_0(t) = \lambda, \tag{6.6}$$

where λ is a constant; this assumes a constant hazard rate over time.

The Weibull distribution[186] adds an additional parameter, γ,

$$h_0(t) = \lambda\gamma t^{\gamma-1}, \tag{6.7}$$

where t is time to progression. The addition of time in the baseline hazard function allows the baseline hazard to change over time.

The Gompertz mortality function was developed to model mortality rates[187], with a baseline hazard function

$$h_0(t) = \lambda e^{-\gamma t}. \tag{6.8}$$

The Weibull model becomes the Exponential model when $\gamma = 1$, and the Gompertz model becomes the Exponential model with $\gamma = 0$. Derivations of the survival functions from th baseline hazard function are shown in Appendix B.

As AMD is a disease which increases with age, it may be reasonable to assume that a time-dependent distribution, such as the Weibull or Exponential model, will be superior. However, the change may not be large enough over a few years to justify additional parameters. Comparing the Weibull and Gompertz models with the Exponential model allowed me to assess whether the extra parameter is needed.

One advantage of this approach is that I can easily concatenate the vector obtained from the mixed-effects stage with demographic data. To demonstrate how demographic data may be easily added to the model, I retrained the best performing model with age at baseline, sex, BMI at baseline, and whether the patient had ever smoked added as covariates. These covariates were identified from the previous work described in Chapter 2.

6.2.4 Loss function

Some patients may not be observed progressing due to dropout or the study ending before the patient progressed; this is right-censored data. The previous models I have developed and presented in this thesis also used right-censored data; however, there was no way to account for this. Survival models have methods for dealing with right-censored data in the loss function.

To account for the censoring, I use the negative of the proportional hazards log-likelihood function as the loss function to account for right-censored data. The log-likelihood function for censored data is

$$l = \sum_i \{\delta_i \log(f_i) + (1 - \delta_i) \log{(S_i)}\}, \tag{6.9}$$

where δ is an indicator function

$$\delta = \begin{cases} 1, & \mathit{if\ the\ event\ is\ observed}, \\ 0 & \mathit{otherwise}. \end{cases} \tag{6.10}$$

and f is the probability density function

$$f(t) = \frac{d}{dt}F(t) = h(t)S(t). \tag{6.11}$$

I also added the random-effects loss function from Chapter 5.

6.6.5 Clinical information

Clinical information can easily be added into this model by appending the information onto the output from the mixed-effects part. In this work, only baseline clinical variables were available; however, in future studies variables such as smoking and BMI may be recorded at each time point. These variables can be incorporated into the model by appending them onto the corresponding feature vectors inside the mixed-effects section ($F1, F2, F3$).

6.2.6 Data

As with Chapter 5, I used the pmsampsize package to estimate the minimum sample size required for model development. My largest model had nine candidate predictor parameters; I aimed for a shrinkage factor of 0.9 and an optimism of 0.05 in the apparent $R^2_{Nagelkerke}$ and predicted a progression rate of 0.1 per year and a mean follow-up of three years. I also aimed to predict at one, two, and three years. This gives an estimated minimum sample size of 1,575, which I needed to develop the model.

As the model used in this chapter can account for missing data, I was able to use more cases with images that may have only had one or two observations. I found 5,569 eyes from 3,032 patients that fit the criteria; 952 (17.1%) eyes progressed to advanced AMD. I used stratified sampling to split data on the patient level into 50% training, 25% validation, and 25% testing. It could be argued that progressing patients may be more likely to have missing visits as they may progress and drop out before the end of the study. Therefore, I only included patients with complete data in the testing dataset to reduce the risk of bias introduced by the model learning missing data. This also allowed me to assess how missing data affects the testing set by calculating model performance with one or two observations removed. However, patient profiles in the validation and testing sets may contain missing data.

Stratified sampling maintained approximately 17.1% progression across all three data splits. Patient demographics in each split are shown in Table 6.1, and examples of two patient profiles are shown in Figure 6.2.

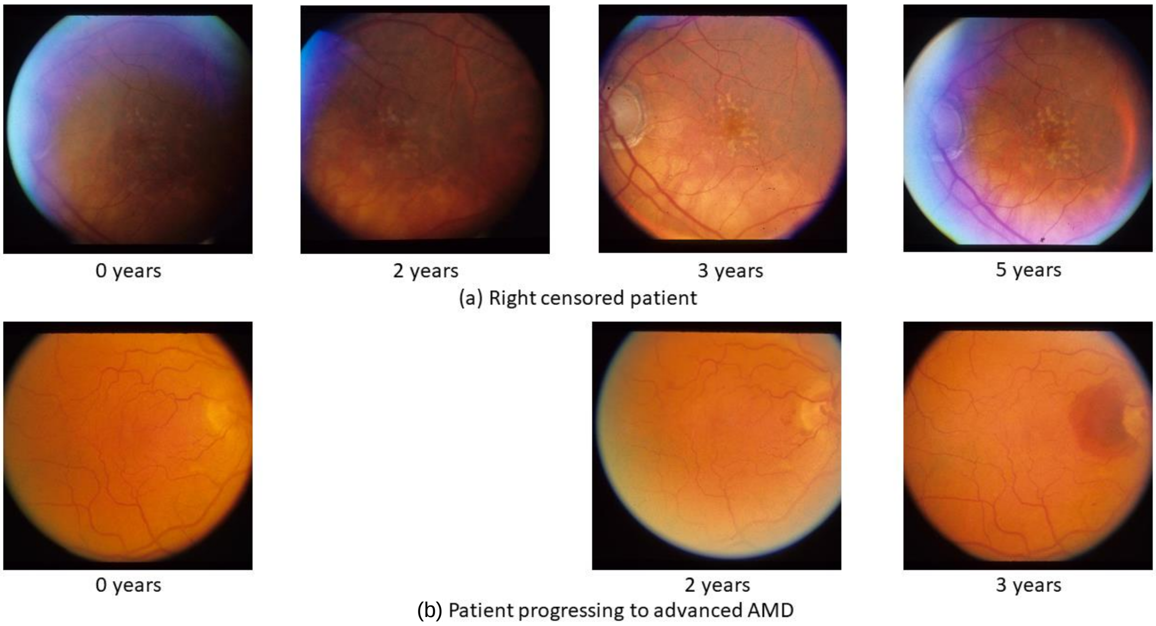
I performed the same pre-processing and online data augmentation described in Chapter 4. When a patient only had one or two visits, I used a blank image, the same method as in Chapter 5.

To assess the impact of missing images, I added in a blank image. These images can be added with any reasonable time point specified. In my work, if there was one missing image. I inserted it between the two present images and set the missing time point as the middle of the two non-missing time points. If there were two missing images, I placed one missing image between the present image and the prediction time point and the second at the same distance before the present image.

In one model, I aimed to add clinical data to assess the impact on model performance. One patient was missing BMI information, so I set the BMI to 27.5, which is the mean of the training dataset. The BMI was divided by 60 and the age by 100 to normalise the values; however, values can exceed these normalisations. The sex covariate was 0 if male and 1 if female, and the “ever smoked” covariate was 0 if the patient had never smoked and 1 if they had.

Table 6.1: Patient demographics across the data splits. Characteristics are shown to be reasonably consistent between splits.

	Training	Validation	Testing
Eyes	2785	1392	1392
Patients	1532	755	754
Female (%)	1528 (54.9%)	782 (56.2%)	794 (57.0%)
Mean baseline age (range)	74.4 (58.4, 87.9)	74.4 (56.9, 85.5)	74.7 (56.9, 87.8)
Mean follow-up Years (Range)	1.3 (0.5, 8.0)	1.3 (0.5, 12.0)	1.24 (0.5, 6.0)
Progressing (%)	476 (17.1%)	238 (17.1%)	238 (17.1%)
Mean BMI at baseline (Range)	27.5 (8.9, 58.2)	27.4 (15.5, 54.9)	27.2 (16.1, 47.1)
Ever smoked (%)	1499 (53.8%)	775 (55.7%)	689 (49.5%)



(a) Right censored patient

(b) Patient progressing to advanced AMD

Figure 6.2: Example profiles of (a) a right-censored patient that is not observed progressing to advanced AMD and (b) a patient who is observed progressing to advanced AMD; this patient also has a missing visit. The first three time points are used to predict the probability of progression at the fourth time point. As AMD is a degenerative eye condition, it can be assumed that the patient who was not observed progressing will progress at some future time point; I treat this as right-censored data.

6.3 Results

In this section, I begin by presenting the results first for the standard Exponential, Weibull, and Gompertz models at one, two, and three time points. Each of these models uses three observations. I then assess how removing one and two of the observations affects the results. Finally, I looked at how clinical information (age, sex, BMI, and whether the patient has ever smoked) can be incorporated into the model and whether this improves performance.

6.3.1 Standard models

The Exponential model had a validation C-index of 0.818 (0.794, 0.843), the Weibull model attained a C-index of 0.790 (0.763, 0.816), and the Gompertz model had 0.786 (0.762, 0.810). Model performance on the validation set suggests that the additional parameter added by the Weibull and Gompertz models are not justified, and the Exponential model may be sufficient.

On the testing data, the Exponential and Weibull models showed similar performance with C-indices of 0.796 (0.769, 0.822) and 0.796 (0.769, 0.822), respectively. However, the Gompertz model showed significantly worse performance with a C-index of 0.584 (0.541, 0.628).

Full results with dynamic AUROCs for one, two, and three year predictions are shown in Table 6.2 and Table 6.3. ROC curves for one, two, and three year predictions are shown in Figure 6.3. ROC component curves are shown in Appendix C. Log-negataive-log plots which show that the proportional hazards assumption holds, are also shown in Appendix C.

The calibration curves shown in Figure 6.4, Figure 6.5, and Figure 6.6 suggest that all models are poorly calibrated, and recalibration is required. The calibration curves for the Gompertz model in the testing dataset show particularly poor calibration, which is expected from the poor discriminative performance.

The decision curves in Figure 6.7 show that both the Exponential and Weibull models have improved net benefit over the treat-all approach. The Gompertz model has some small net benefit, even in the testing set; however, the net benefit is much lower than the Exponential and Weibull models.

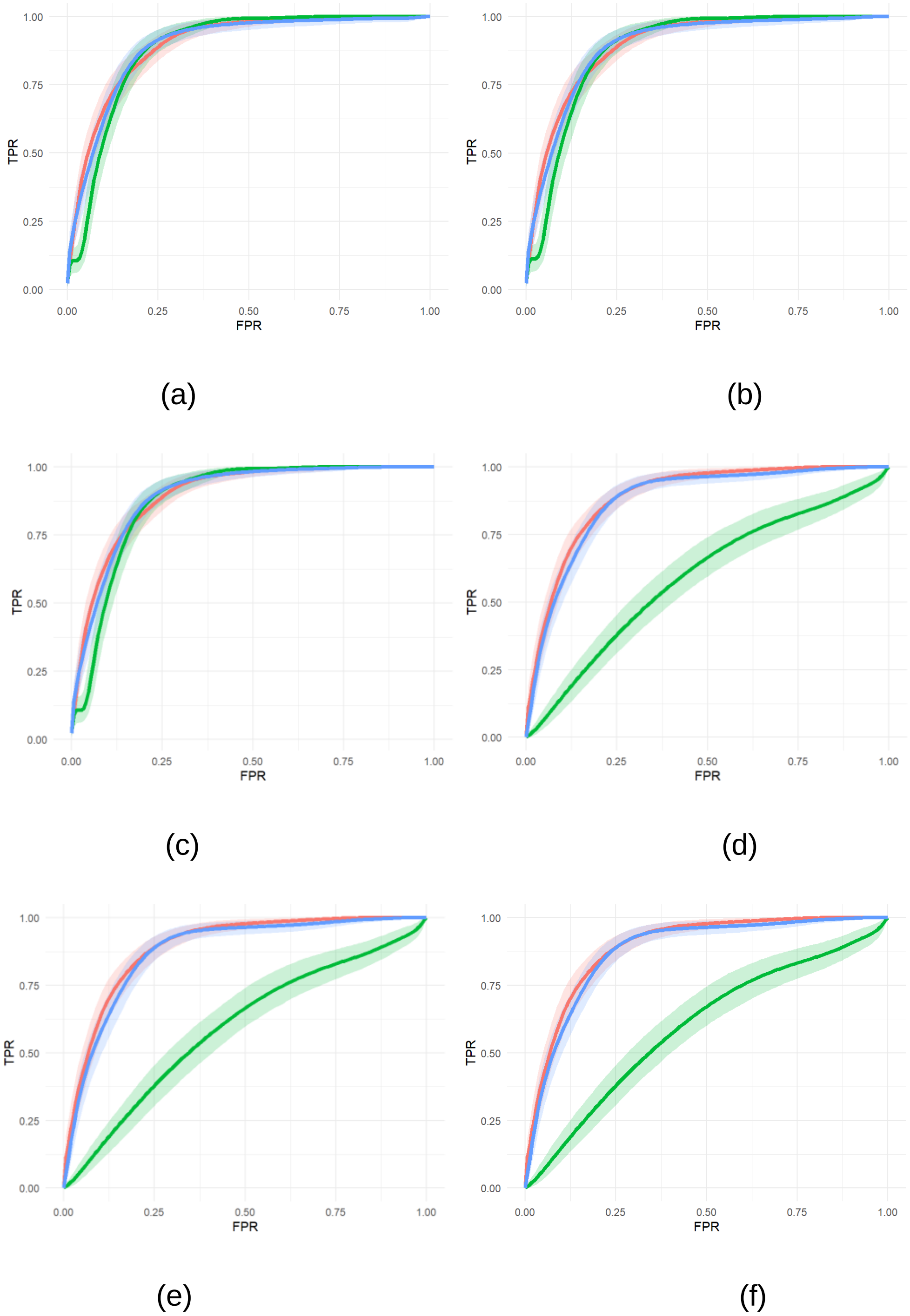
As in previous sections, I created saliency maps using SmoothGrad,[82] with 20 samples and a noise of 0,2. The saliency maps shown in Figure 6.8, Figure 6.9, and Figure 6.10 suggest that the Exponential model successfully identifies the drusen as useful in the prediction. Compared to the saliency maps shown in Chapter 4, these maps seem to be much more precise and identify specific areas of the image.

Table 6.2: C-index and AUROC at one, two, and three years for the Exponential, Weibull, and Gompertz models on the validation dataset.

Model	C-index	Years	AUROC
Exponential	0.818 (0.794, 0.843)	1	0.834 (0.810, 0.856)
		2	0.959 (0.944, 0.974)
		3	0.962 (0.947, 0.979)
Weibull	0.790 (0.763, 0.816)	1	0.812 (0.785, 0.838)
		2	0.936 (0.907, 0.960)
		3	0.958 (0.932, 0.979)
Gompertz	0.786 (0.762, 0.810)	1	0.808 (0.785, 0.831)
		2	0.937 (0.908, 0.959)
		3	0.937 (0.912, 0.963)

Table 6.3: C-index and AUROC at one, two, and three years for the Exponential, Weibull, and Gompertz models on the testing dataset.

Model	C-index	Years	AUROC
Exponential	0.813 (0.789, 0.837)	1	0.893 (0.872, 0.914)
		2	0.947 (0.916, 0.971)
		3	0.951 (0.919, 0.973)
Weibull	0.796 (0.769, 0.822)	1	0.880 (0.856, 0.903)
		2	0.922 (0.892, 0.948)
		3	0.935 (0.902, 0.965)
Gompertz	0.584 (0.541, 0.628)	1	0.595 (0.546, 0.643)
		2	0.578 (0.517, 0.636)
		3	0.613 (0.540, 0.687)



(a)

(b)

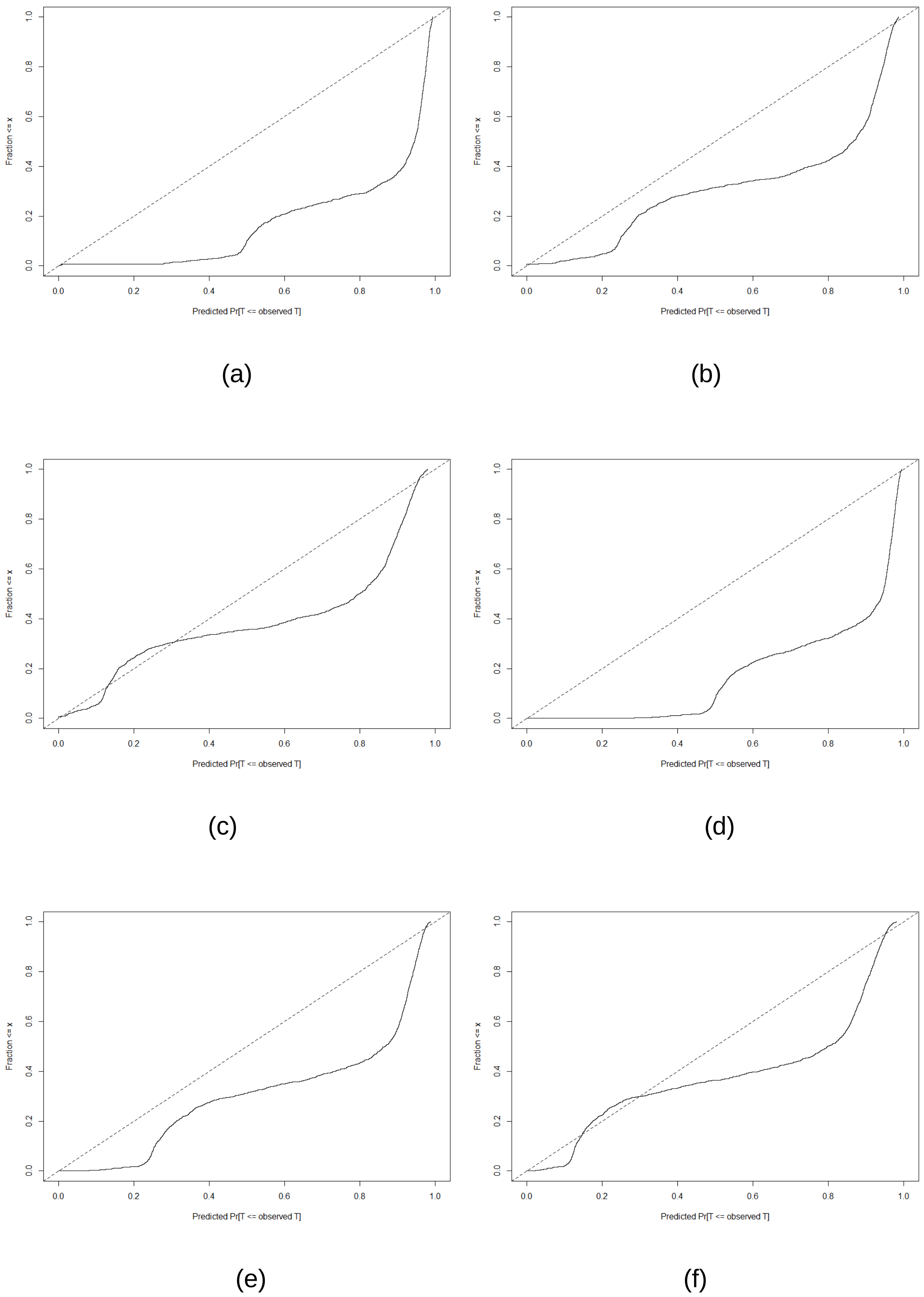
(c)

(d)

(e)

(f)

Figure 6.3: ROC curves for the Exponential (red), Weibull (blue), and Gompertz (green) models on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

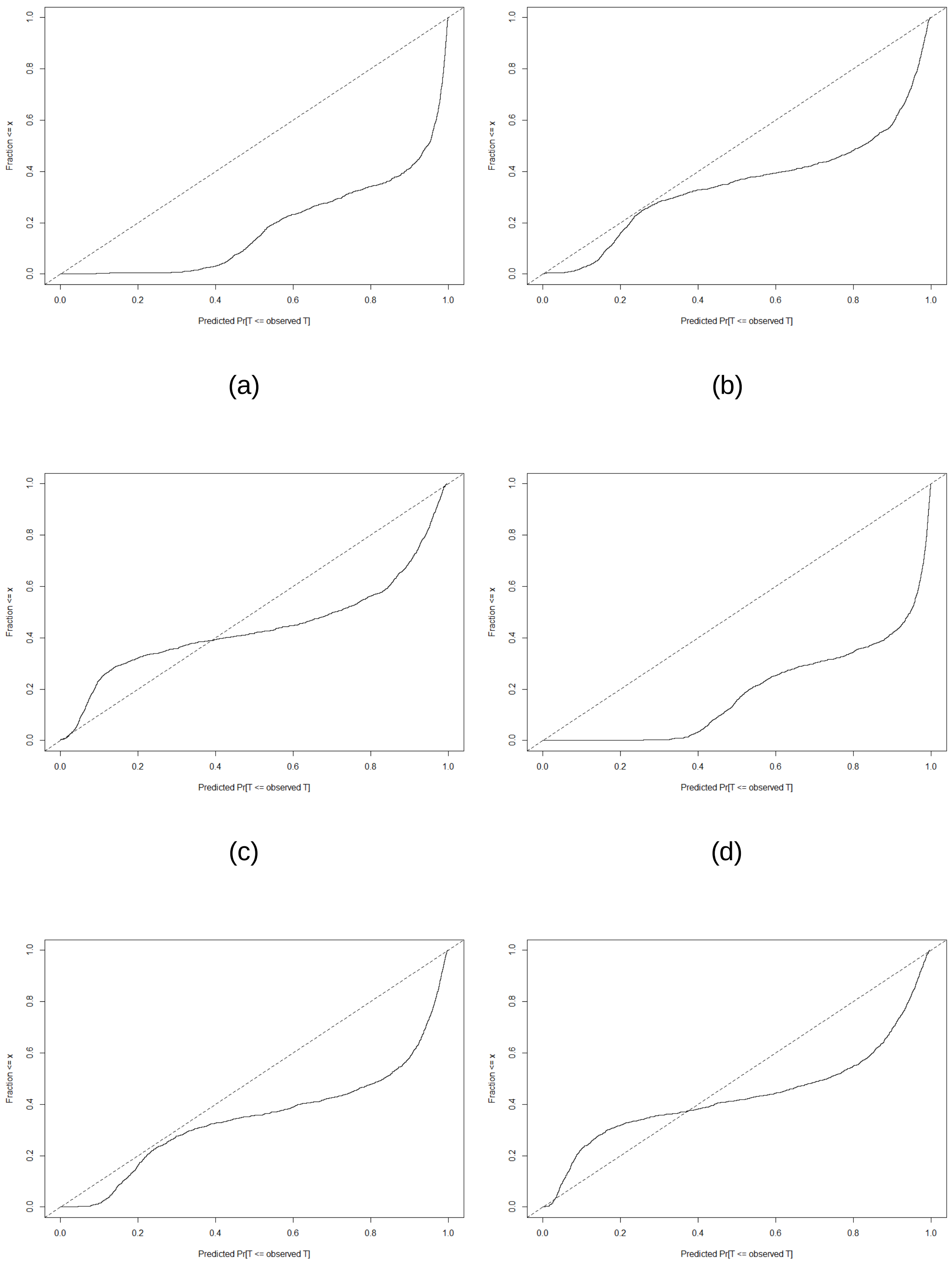
(c)

(d)

(e)

(f)

Figure 6.4: Calibration curves for the Exponential model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

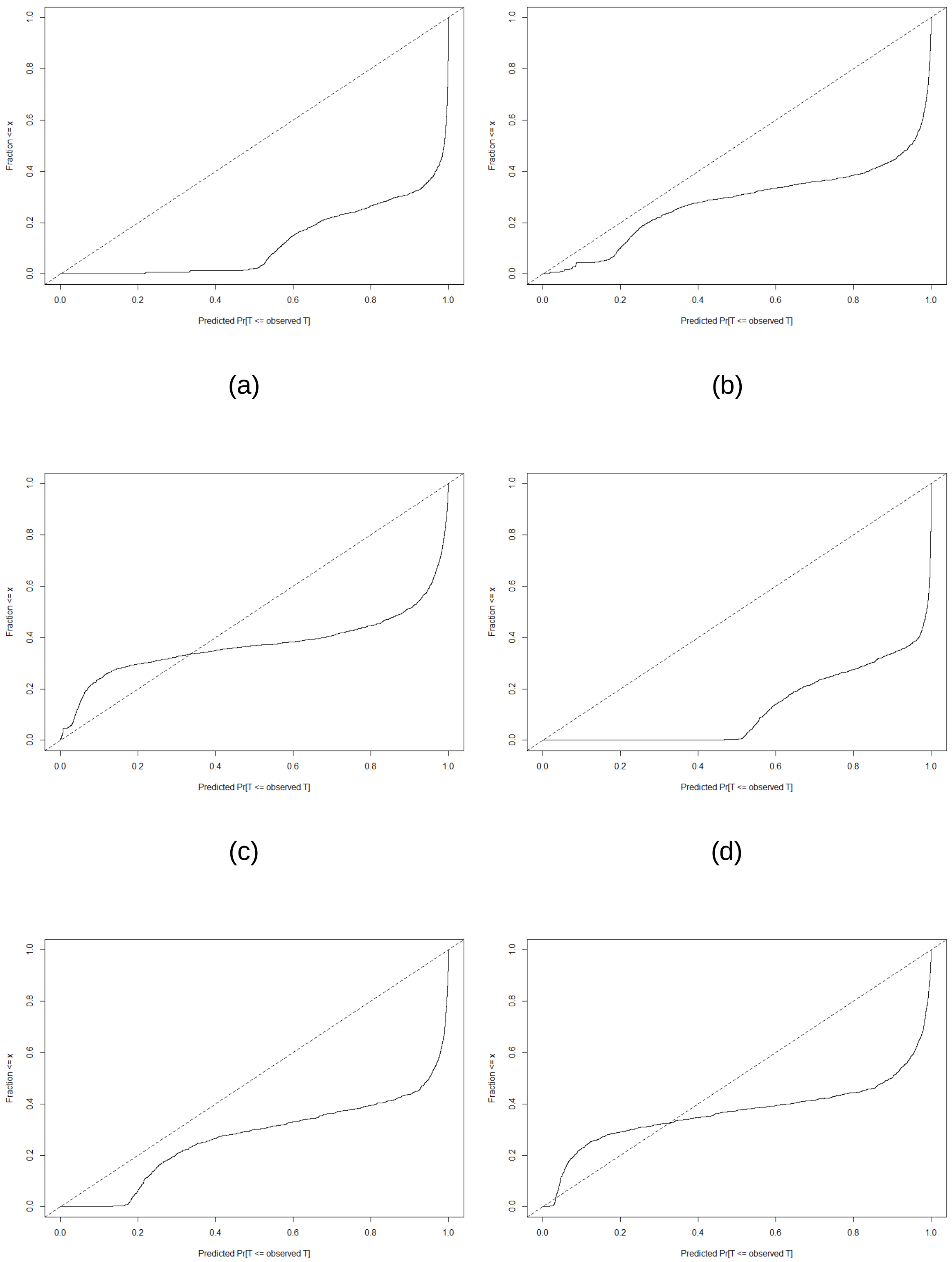
(c)

(d)

(e)

(f)

Figure 6.5: Calibration curves for the Weibull model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

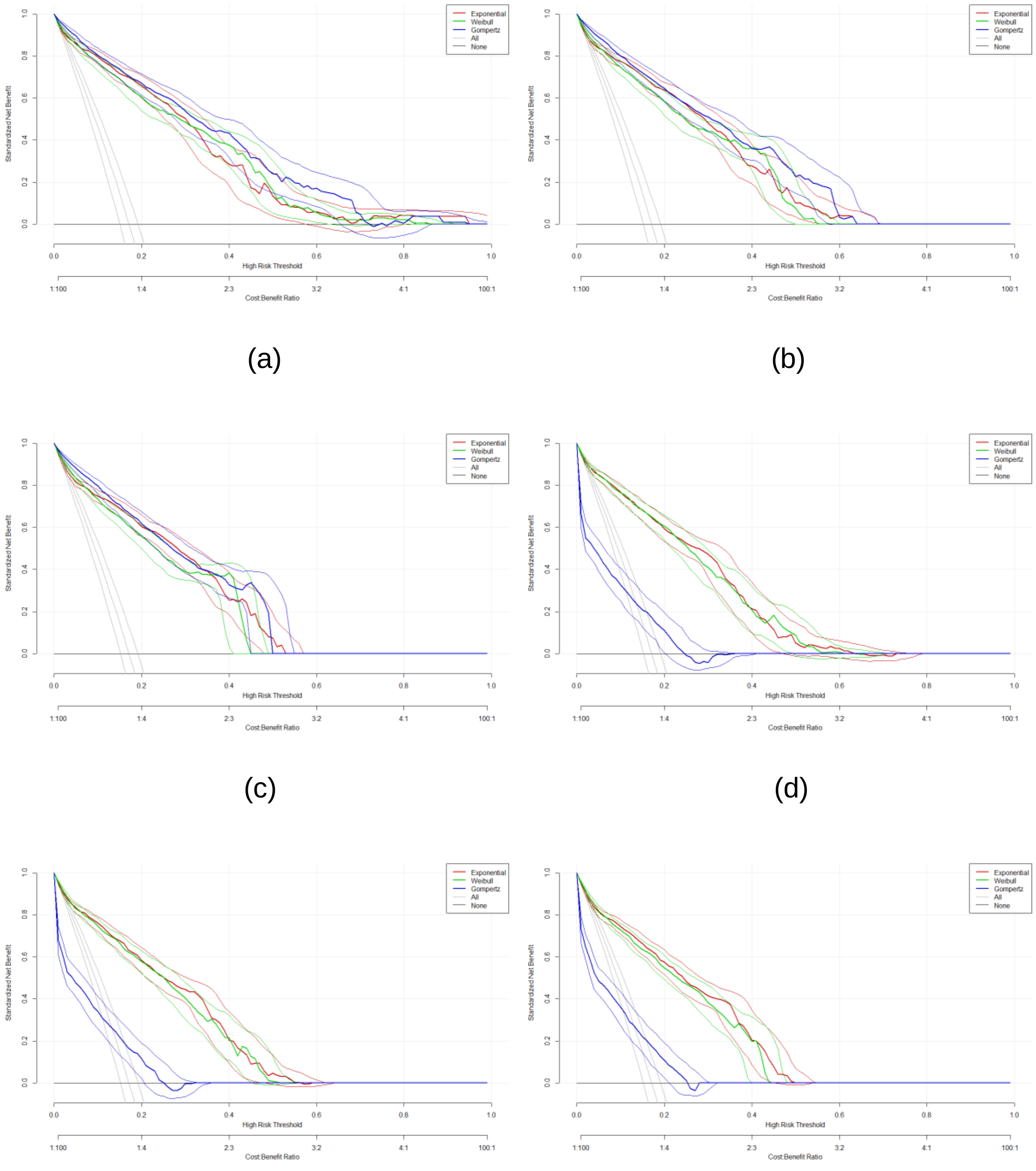
(c)

(d)

(e)

(f)

Figure 6.6: Calibration curves for the Gompertz model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

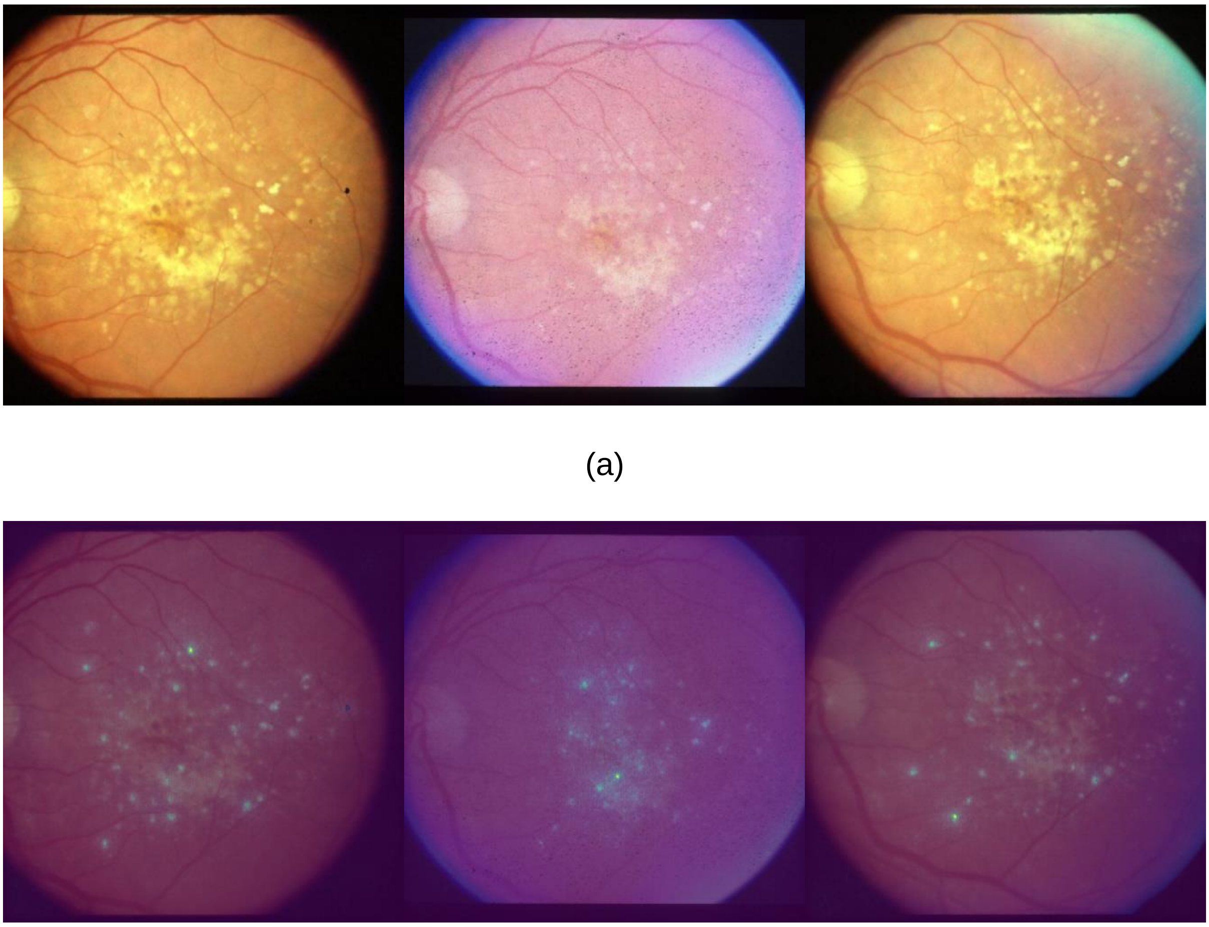
(c)

(d)

(e)

(f)

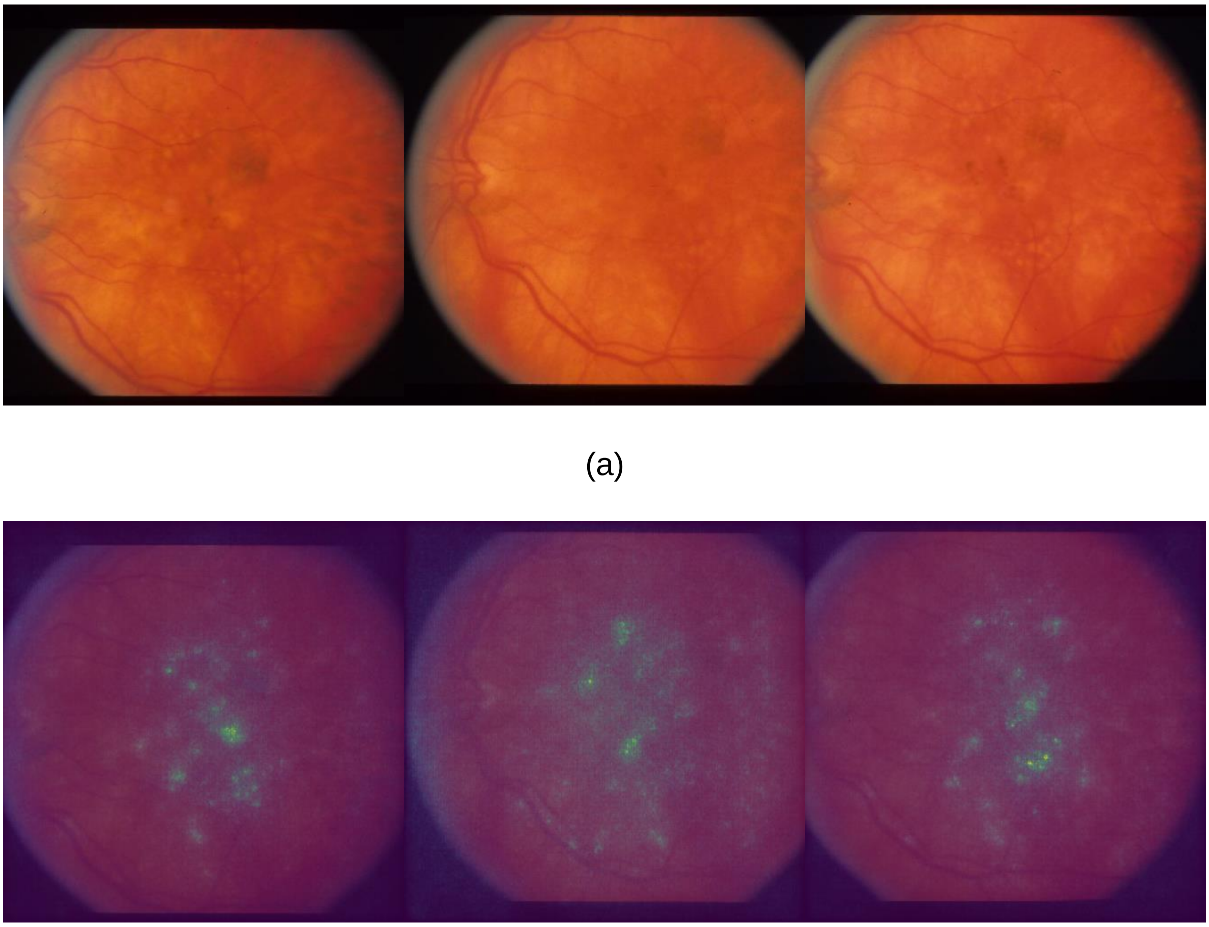
Figure 6.7: Decision curves for the Exponential, Weibull, and Gompertz models on the validation dataset at (a) one year, (b) two years, and (c) three years and on the testing dataset at (d) one year, (e) two years, and (f) three years.



(a)

(b)

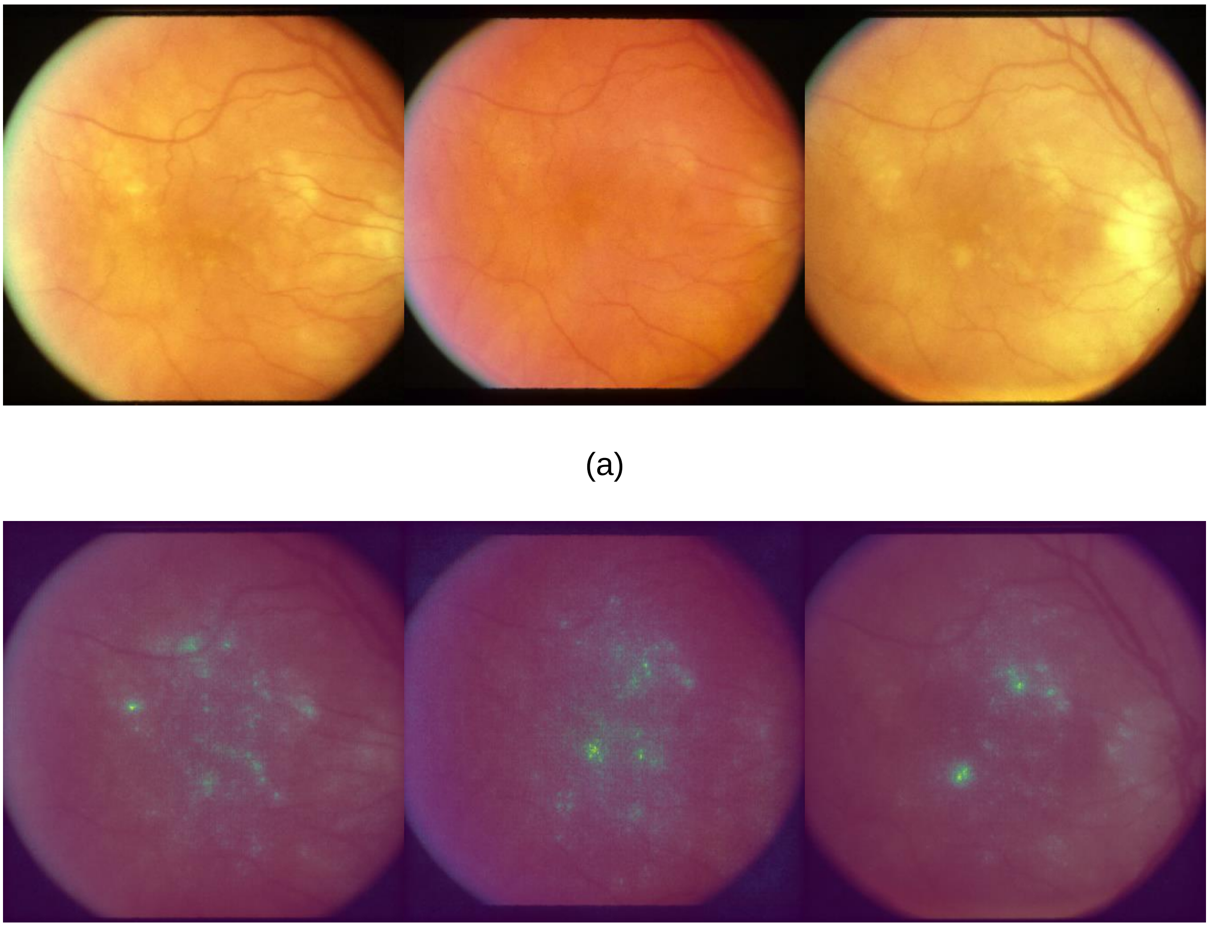
Figure 6.8: Saliency maps for a progressing patient showing (a) the original image and (b) saliency maps for the Exponential model.



(a)

(b)

Figure 6.9: Saliency maps for a progressing patient showing (a) the original image and (b) saliency maps for the Exponential model.



(a)

(b)

Figure 6.10: Saliency maps for a progressing patient showing (a) the original image and (b) saliency maps for the Exponential model.

6.3.2 Missing data

As discussed in Chapter 5, mixed-effects models are capable of dealing with missing data. This section displays results for models with one and two observations missing from the model. As there is no reason I could identify, in this application, to choose the Weibull or Gompertz model over the Exponential model, I used the Exponential model to assess the effects of missing data.

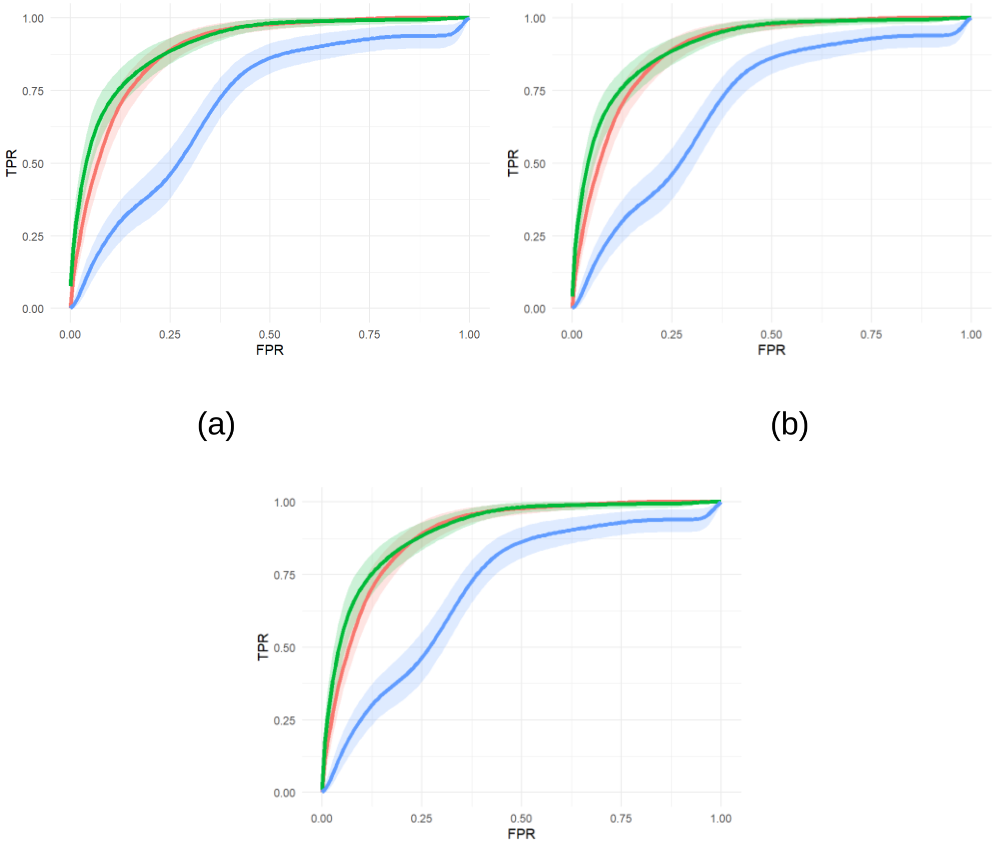
The C-indices for no missing data, one missing observation, and two missing observations are 0.813 (0.789, 0.837), 0.796 (0.772, 0.821), and 0.832 (0.805, 0.859). These results suggest that the performance is not significantly different even when two of the three observations are missing, and the mixed-effects model successfully accounts for missing data. However, the AUROC values for individual time points, displayed in Table 6.4, are significantly lower when two time points are missing. This suggests that a performance measure which mixes clinically useful and clinically useless time points and decision thresholds such as the the C-index may not necessarily indicate the best model. Practically, we are more concerned about performance at the times we are making predictions at and not all time points. ROC curves are shown in Figure 6.11, and ROC component curves are displayed in Appendix C.

Figure 6.12 shows that the calibration is similar for no missing time points and one missing time point; however, the calibration is much worse when two time points are missing. As with the model without missing data, model recalibration is needed.

The decision curves shown in Figure 6.13 suggest that even with much worse performance in terms of AUROC, there is still some net benefit over the treat-all approach, even when using two missing images. Decision curves without confidence intervals and clinical impact curves are displayed in Appendix C.

Table 6.4: C-index and AUROC at one, two, and three years for the Exponential model with complete data, one missing observation, and two missing observations on the testing dataset.

Missing observations	C-index	Years	AUROC
0	0.813 (0.789, 0.837)	1	0.893 (0.872, 0.914)
		2	0.947 (0.916, 0.971)
		3	0.951 (0.919, 0.973)
1	0.796 (0.772, 0.821)	1	0.910 (0.888, 0.930)
		2	0.935 (0.899, 0.957)
		3	0.947 (0.896 ,0.976)
2	0.832 (0.805, 0.859)	1	0.706 (0.664, 0.745)
		2	0.744 (0.698, 0.788)
		3	0.722 (0.642, 0.802)

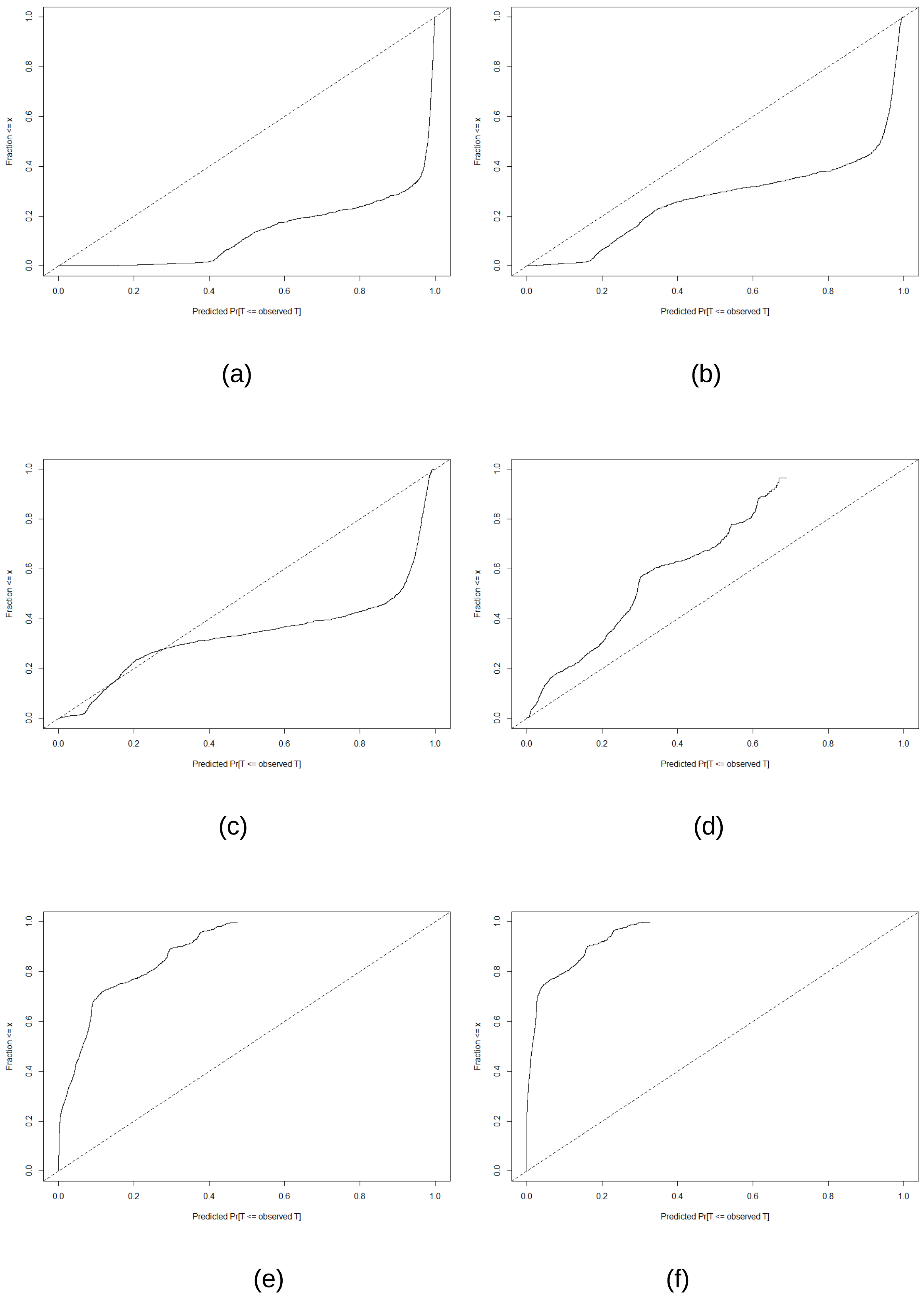


(a)

(b)

(c)

Figure 6.11: ROC curves for the Exponential model with zero (red), one (green), and two (blue) missing time points models on the testing dataset for predictions at (a) one year, (b) two years, and (c) three years.



(a)

(b)

(c)

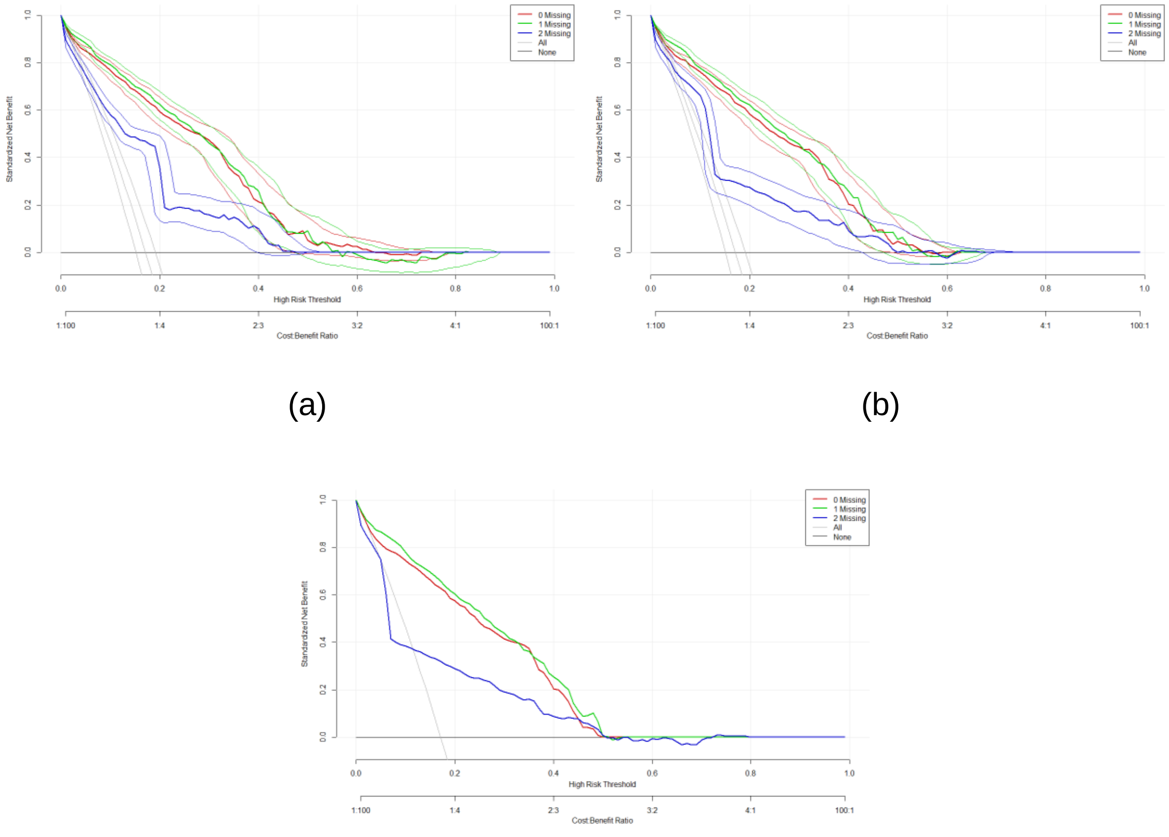
(d)

(e)

(f)

Figure 6.12: Calibration curves for the Exponential model with one missing observation for predictions at (a) one year, (b) two years, and (c) three years and

with two missing observations for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

(c)

Figure 6.13: Decision curves for the Exponential model with zero, one, and two missing observations for predictions at (a) one year, (b) two years, and (c) three years.

6.3.3 Adding clinical information

Adding age, sex, BMI, and whether the patient had ever smoked to the Exponential model gave C-indices of 0.857 (0.837, 0.877) and 0.838 (0.813, 0.863) on the validation and testing datasets, respectively. Full results, including the AUROC at one, two, and three years, are shown in Tables 6.5 and 6.6, with the Exponential model results for comparison. Adding clinical information did not significantly improve performance compared to the Exponential model alone. ROC curves are displayed in Figure 6.14, with ROC component curves shown in Appendix C. These curves display that the models with and without clinical covariates perform similarly.

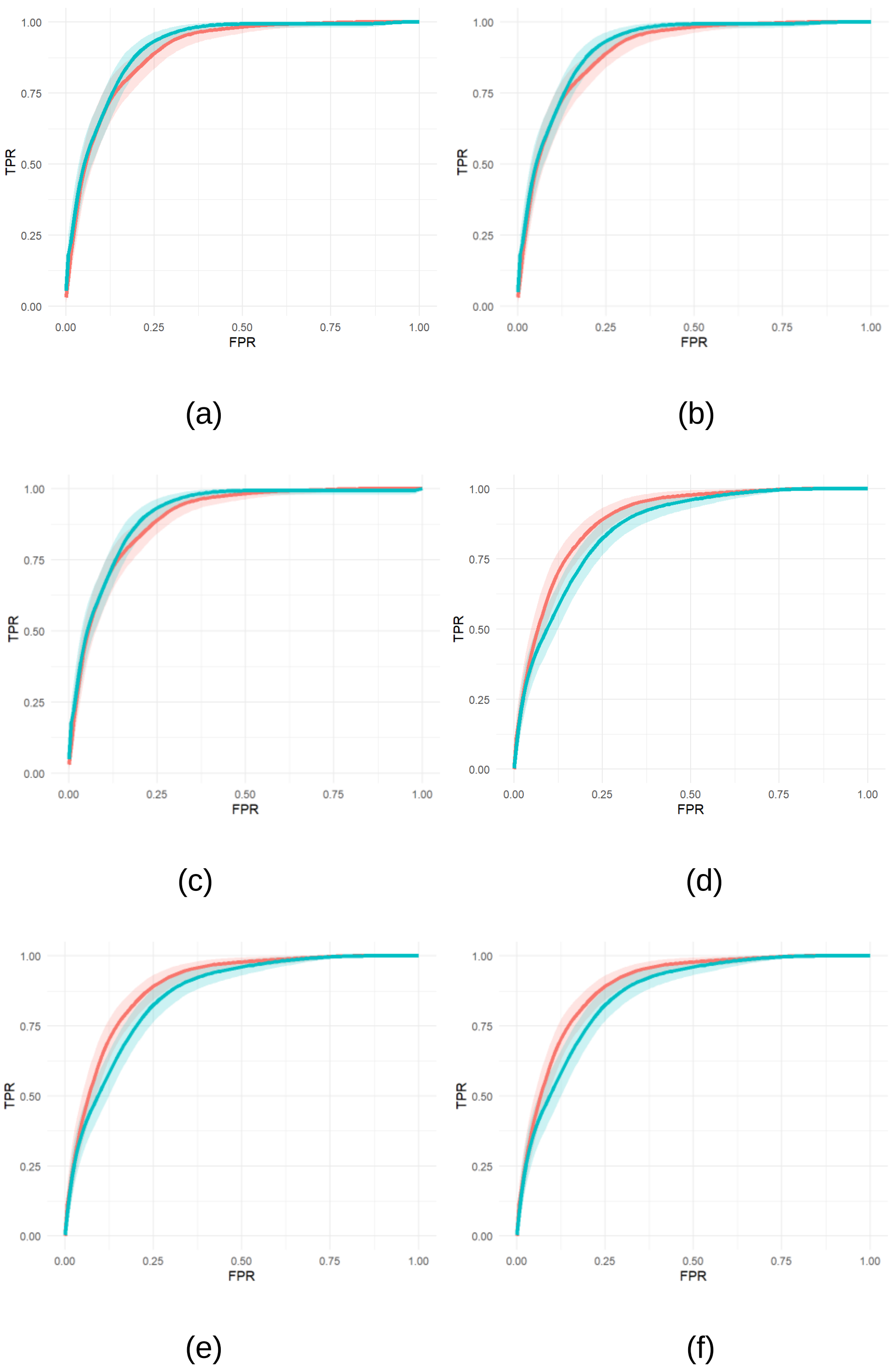
Figure 6.15 shows calibration curves which suggest that the model requires recalibration, similar to the previous models I have shown. The decision curves in Figure 6.16 suggest that adding covariates may improve net benefit, but not significantly. However, if net benefit is the primary concern irrespective of cost, the marginal increase may be worthwhile, as shown in the decision curves without confidence intervals in Appendix C. Clinical impact curves are also shown in Appendix C.

Table 6.5: C-index and AUROC at one, two, and three years for the Exponential model with and without covariates on the validation dataset.

Model	C-index	Years	AUROC
Exponential	0.818 (0.794, 0.843)	1	0.834 (0.810, 0.856)
		2	0.959 (0.944, 0.974)
		3	0.962 (0.947, 0.979)
Exponential with covariates	0.857 (0.837, 0.877)	1	0.867 (0.846, 0.886)
		2	0.960 (0.944, 0.976)
		3	0.972 (0.957, 0.985)

Table 6.6: C-index and AUROC at one, two, and three years for the Exponential model with and without covariates on the testing dataset.

Model	C-index	Years	AUROC
Exponential	0.813 (0.789, 0.837)	1	0.893 (0.872, 0.914)
		2	0.947 (0.916, 0.971)
		3	0.951 (0.919, 0.973)
Exponential with covariates	0.838 (0.813, 0.863)	1	0.865 (0.839, 0.89)
		2	0.905 (0.88, 0.927)
		3	0.94 (0.916, 0.96)

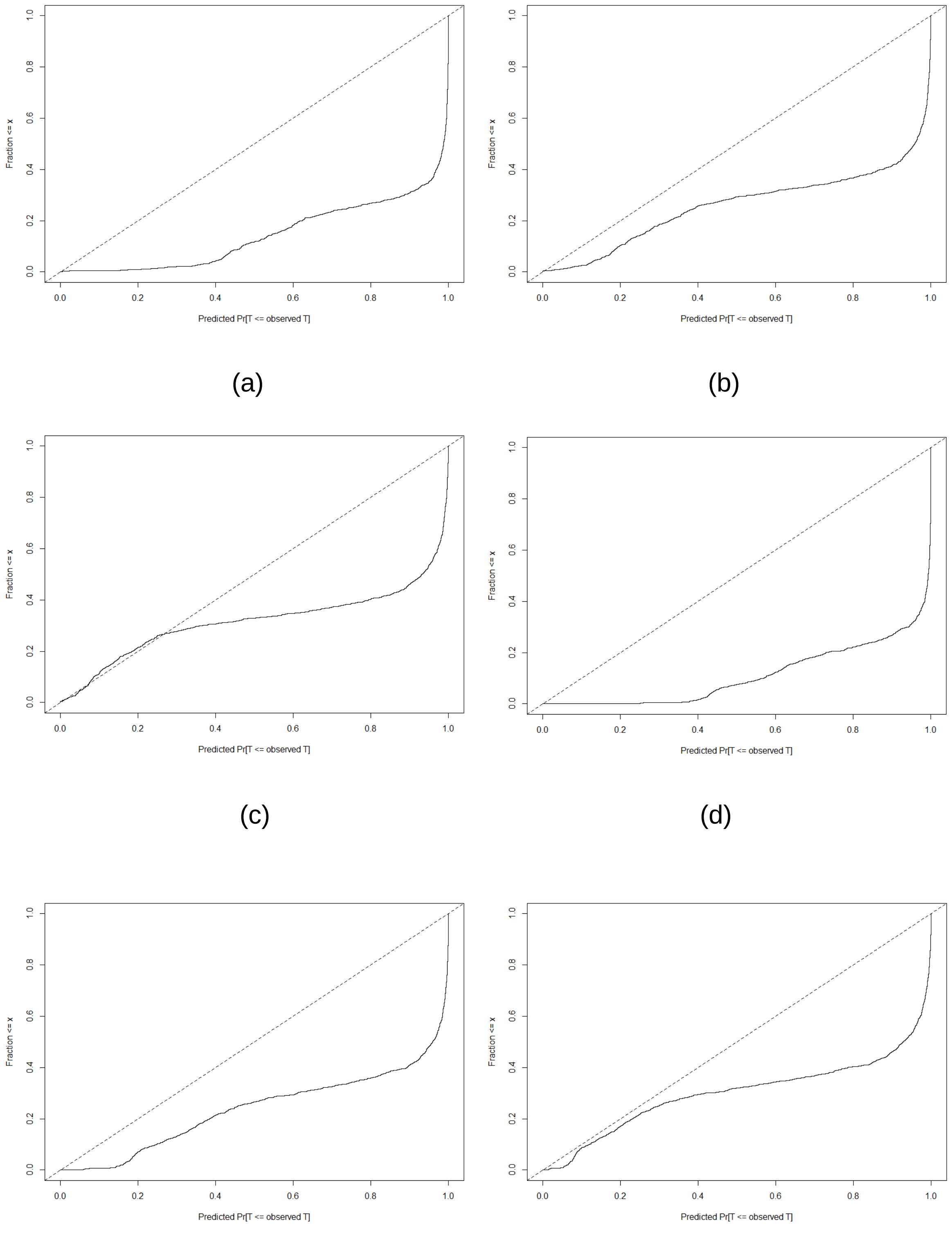


(a) (b)

(c) (d)

(e) (f)

Figure 6.14: ROC curves for the Exponential model with (blue) and without (red) covariates on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset at (d) one year, (e) two years, and (f) three years.

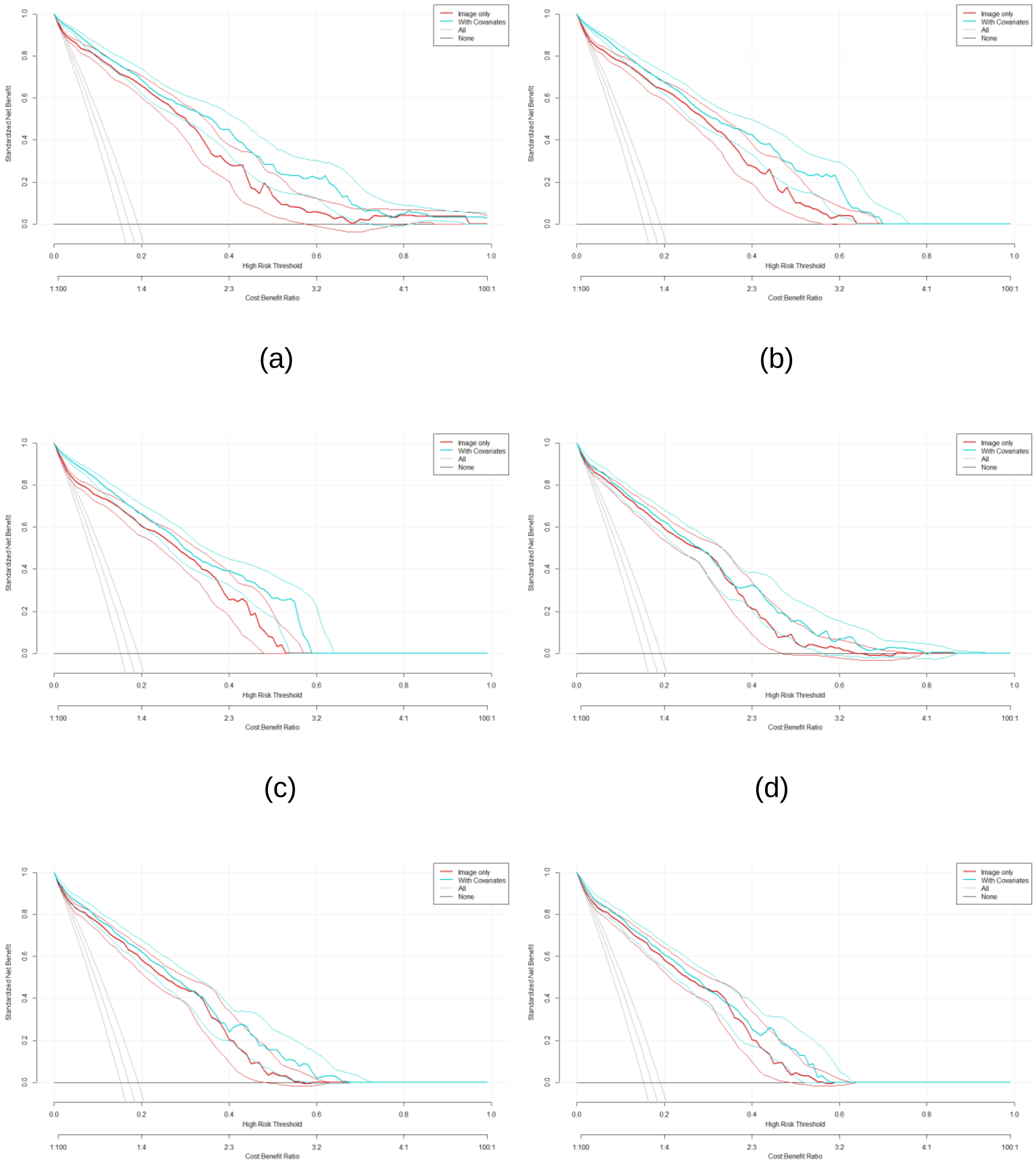


(a) (b)

(c) (d)

(e) (f)

Figure 6.15: Calibration curves for the Exponential model with covariates on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.16: Decision curves for the Exponential model with covariates on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

6.4 Discussion

In this section, I have extended the work I presented in Chapter 4 to deal with missing data. The method I proposed combines my mixed-effects model and a survival model. My method is similar to joint models in traditional statistics but uses backpropagation instead of a joint likelihood function. To the best of my knowledge, this was the first joint model in deep learning. My model's biggest novelty is the ability to take longitudinal images and covariates and produce reasonable predictions of the future outcome.

This was the final model I developed; unfortunately, I did not have enough time to solve all the limitations I wanted. The biggest limitation of this work is the surprising result that introducing a time-dependent hazard function did not result in improved performance. The proportional hazards models I used assume proportional hazards, although this assumption can often be violated without causing issues. I have been unable to check this as no method is available for images. One possible solution to this may be to check the proportionality of the feature vectors, but I was unable to implement this.

I have also not recalibrated the models as I did not have enough time. Recalibration is not a priority at this stage, as any model should be recalibrated in its intended setting anyway. One possible recalibration method is temporal calibration, which helps to account for the changes in survival rates over time.[188] Advances in treatment may improve survival rates and reduce the appropriateness of any model developed on older data. Temporal recalibration recalibrates the baseline hazard using a period analysis of a sample.

I examined three hazard functions for the survival model: the Exponential distribution, the Weibull distribution, and the Gompertz distribution. The Exponential and Weibull models showed similar performance attaining good discrimination; however, the Gompertz model showed significantly worse performance on the testing data. The Exponential model is the simplest, with the Weibull and Gompertz model containing one extra parameter. Parsimonious models are preferred in statistics, and the Exponential model should be chosen in this situation. The Weibull or Gompertz models may be expected to be better as they contain a time-dependent parameter allowing the baseline hazard to increase with time; however, over the three years that I have predicted, this increase may be negligible. Over such a short time span, the change in hazard may mean that a time-dependent hazard function is unnecessary. A time-dependent distribution may be preferred over longer predictions, such as five years.

When one image was replaced with a blank image to simulate missing data, the model still performed reasonably well, and there was not a significant loss in model performance. This is similar to the results observed in Chapter 5. However, when two of the images were replaced with blank images, the model performed much worse based on the AUROCs at one, two, and three years. This suggests that having two-thirds missing data is too much, and this particular algorithm can only handle removing one-third of the data. This may not be true for applications, and the limit at which missingness begins to affect the model performance may be higher or lower. The C-index at for two missing images appeared to be very good although the AUROCs for individual years were much lower. One explanation for this is that the C-index combines clinically useful and clinically useful time points into a single measure. It may be that the model with two missing time points performs very well for very short or very long range predictions, but not at the times that we may be interested in.

I have shown how clinical and demographic covariates can be easily added to the model in the survival part. The addition of age, sex, BMI, and whether the patient has ever smoked did not significantly improve predictive performance. This does not necessarily mean that the covariates do not affect the outcome; they do not add much prognostic value over the images alone in this situation. I could have used variable selection rather than including variables previously identified as useful; however, variable selection, especially stepwise regression, is a contentious subject. It is often better to use expert knowledge, including clinician opinion and literature searches, to choose covariates.[189]

There are several possible reasons why adding clinical factors did not significantly improve performance. Firstly, late fusion may not be the best approach. In this work, I have only assessed the impact of concatenating the covariates onto the feature vector before the survival layer. This is a simple solution; however, an early or joint approach may have been better able to model the relationship between imaging and clinical covariates. Secondly, BMI and smoking may change throughout the follow-up. BMI and smoking status are modifiable risk factors, which the patient may change after being diagnosed with early/intermediate AMD. Stopping smoking or reducing BMI may significantly reduce the chances of progression. Including these as time covariates in the mixed-effects part of the model could improve performance. Unfortunately, while baseline demographic and clinical variables were available for all patients, variables such as BMI and smoking status in AREDS were often blank at each subsequent visit. It is impossible to know whether the data is missing, zero, the same as the last visit, unknown, or not recorded. More modern studies following Good Clinical Practice would not allow for blanks in records, and this problem would be avoided. Finally, the model may be able to rely on imaging alone to make the predictions. BMI and smoking are also often unreliable, especially if they are patient-reported.

6.5 Conclusions

In this chapter, I have combined the mixed-effects model from Chapter 5 with a survival model to create a deep learning model capable of dealing with missing data and uneven intervals between visits. I found the Exponential model to perform better overall. This model was able to attain good performance, even when one observation was missing. This allows for a single model to be used, even when the patient only has images for two visits available. I have shown how other clinical variables may be added to the model; however, this did not significantly improve performance over using the images alone. Other applications may benefit from the inclusion of other variables.

Chapter 7: Discussion

In this chapter, I briefly summarise the thesis and highlight the key conclusions and contributions that can be made from the work I have performed during my PhD. I also outline ways my work could be improved and extended in the short term. I then discuss the limitations of my work and several future directions that could extend my work in the long term.

7.1 Summary

The main aim of my work was to create a deep learning prognostic model to predict disease progression using longitudinal imaging data with uneven intervals between visits and missing data. Previous models for imaging have required specific intervals between visits, for example, 30-day intervals.[147] This has greatly limited the applicability of previously published models as patients are highly unlikely to visit clinic at precisely the same interval each time. Any model would be made invalid if the patient missed a visit or had their appointment moved due to the algorithm's predictions.

The first model I developed introduced a novel interval scaling mechanism which helps to account for the different intervals between visits. I developed two models utilising my novel interval scaling mechanism, allowing for uneven intervals between time points. My scaling mechanism also allows for any prediction time point to be chosen. My GEV activation also helps to account for the class imbalance. The GEV activation is easy to implement, simply replacing the sigmoid activation. I have also developed the mGEV activation for the multiclass case. I expect that the activations have further applications outside of classification, such as in segmentation, although I have not had time to test this. Models I have developed using this showed good discrimination.

The mixed-effects layer I have developed means that spatial and temporal relationships can be modelled without assuming 3D structures. I have also shown that missing images can be appropriately handled. The novel loss function I introduced ensures that the normality of the random effects errors is enforced, which is one of the assumptions of the random effects model. When applied to COVID-19 data, my model showed improved generalisability over previously published models with good calibration maintained in external validation.

Finally, I developed models which combine mixed-effects and survival models. These models are capable of handling both missing and right-censored data. I found that an exponential model is adequate for my particular application, but other models may be more appropriate in other situations.

7.2 Strengths and Limitations

One major strength of my work is that I have followed best practice reporting guidelines throughout to better understand how my models may perform in clinic. Following these guidelines is vital to ensure that the models provide accurate, safe, and useful predictions. Although my models have been shown to have poor calibration, it is known that recalibration is needed to ensure that models do not under- or over-estimate the risk of progression.

Shortly after starting the second year of my PhD, the COVID-19 pandemic began, and I spent around half of my PhD studentship working from home. As clinicians from all areas were refocused to help with the global effort of care and research for COVID-19, data became more difficult to obtain. I could not obtain external validation datasets, and applications for honorary contracts were put on hold. Adapting to working from home and having less access to guidance was also a challenge. Due to these issues, I could not complete as much work as I would have hoped, and there are several limitations of my work resulting from this.

Longitudinal images are common in clinical settings, with patients often having images collected each time they visit a clinic. However, longitudinal imaging datasets are often very uncommon. These datasets are large and expensive to curate, requiring extensive long-term funding and planning. Collecting longitudinal images also proves problematic as imaging and storage systems may change. For this reason, I was only able to gain access to one longitudinal dataset to demonstrate my methods.

In this dataset, a single image appears sufficient for prediction, with only marginal, non-statistically significant improvements apparent by adding additional time points. For other clinical applications, the addition of longitudinal data may provide greater benefit. In Section 4.4, I gave possible reasons for this, which include the changes in AMD progression rate not changing much or at all over the three years we are predicting at. It may seem surprising that the Weibull and Gompertz models did not perform better than the exponential survival models in Chapter 6. The reason for this may be linked to the reason that multiple time points did not improve performance. The added value of the Weibull and Gompertz models is that they incorporate time into the baseline hazard meaning that they the baseline hazard can change over time. The exponential model assumes a constant baseline hazard over all time points. If the baseline hazard does not change significantly over the three years that we aim to predict at, then the expoenential model will be sufficient for prediction. Both of these surprising results can therefore be explained by the relatively short window that we are aiming to predict at.

The AREDS datasets may not be the most suitable datasets in which to develop and test my work. Although they are large-scale, there are many problems, as discussed in Chapter 3. The findings need to be treated with caution as the allocation of participants to supplement, or control may have an effect. My models did not account for the formulations as it was not the aim of my work. Adding a covariate for nutritional supplementation may improve performance; however, it may also limit the use of the model. A prospective study with data collected to externally validate the model in the intended clinical setting is needed. This also highlights why temporal validation is important; future patients diagnosed with early AMD may change diet and supplement intake based on the findings of AREDS changing their prognosis.

Other potential limitations of the dataset stem mainly from the missing data. The survival model I have developed can deal with missing data if the data is missing at random; however, there is a limit to the percentage of missingness in the data. The AREDS dataset had missing demographic or clinical data, especially for the longitudinal variables, such as smoking status, at each visit. As the missing data was left blank, it is unclear whether the data was missing, not recorded, not known, or the same as the previous visit/baseline. Current Good Clinical Practice (GCP) guidelines require the field to be filled in even if the value is not recorded or not known. Following GCP guidelines avoids the ambiguity of missing data.

The models I have developed show excellent discriminative performance; however, the models are not well-calibrated. Before deploying these models in a clinical setting, it is vital to ensure they have good calibration to avoid under- or over-prediction of risk. Models should follow best-practice reporting guidelines to ensure that all aspects of model performance are reported. Calibration curves should be used to assess moderate calibration. For survival models, temporal calibration is required to account for changes in treatment and lifestyles over time.

Advanced AMD consists of two types, atrophic and neovascular AMD. Patients who progress to advanced AMD may develop either one or both conditions. I have only considered advanced AMD in my work, but it may be useful to predict the individual types of disease. Additionally, I have only considered progression on single eyes. The fellow eye is often important in AMD progression, and progression of one eye often means that the other eye is likely to progress. In future, utilising information about the fellow eye may provide improved performance. A competing risks model may be an interesting method to achieve this.

7.3 Future directions

There are several areas in which future research could progress; with rapid advancements in both AI and disease treatment, it is impossible to know how the future will look. One possible area for future work is to use a different backbone network. Networks are increasingly becoming more efficient, which will help the accessibility of the models presented in this thesis. The CNN may even be entirely replaced by a vision transformer or another type of network. These small changes may help to improve model performance or deployability. Here, I do not focus on small changes such as network architectures; instead, I discuss how research may progress into new areas.

All the models discussed here give a probability of progression; it may be useful to produce the future image. Deep learning techniques such as generative adversarial networks (GANs) have been used previously to predict the next frames in a video sequence. It may be possible to use similar techniques to predict the next image in the sequence of visits. However, this may not be clinically useful as the images may not accurately represent future images. GANs applied to medical imaging data have sometimes added extra objects to the image that do not exist in reality.

Often patients may have more than one condition, which can potentially progress. For example, in AMD, a patient may have two eyes at risk of progressing, with either eye needing treatment if they do. A model could be developed for each of these conditions, or a single model could be used for both risks of progression. Traditional statistical models may use a competing risks model in this case. The survival models presented here can be easily extended to account for competing risks and may provide some extra clinical utility.

The final goal of developing any clinical prediction model is to have it deployed in a clinical setting. In this thesis, I have only concentrated on model development. The deployment of deep learning models in a clinical setting is a great challenge in AI. One of the biggest barriers to adoption and deployment is the lack of robust validation of models. External validation following best practice guidelines with models assessed in each intended setting is required to ensure these models are safe for clinical deployment. Ideally, a prospective study with around 200 patients observed progressing to advanced AMD would be needed to assess calibration. Patients would need to be observed at baseline, year one, and year two, followed up for three years. Assuming a cost of £400 per visit and a progression rate of 50% in those three years, this would cost around £1m alone, with additional costs possibly taking the total to £2m for a 6-year study. For this reason, validation studies are challenging to find funding for. Using retrospectively collected data can decrease the cost but tends to introduce bias.

The validation of in-home monitoring is an exciting possibility, reducing pressure on clinics and patients. Handheld colour fundus cameras are becoming more capable of providing high-quality images which can be used to diagnose diseases such as diabetic retinopathy.[190] Artificial intelligence can be incorporated into handheld systems, increasing their utility.

In my work, I have used colour fundus because that is the most available. However, OCT is now the preferred method of assessing AMD. Future predictive models will likely need to use OCT, this can create a challenge for prognostic models as it may take 10 years to collect data using a modality and this modality may become obsolete in that time. A recent protocol for the PINNACLE trial, aims to use both retrospective and prospective data to identify biomarkers of progression on AMD to develop future prognostic models.[191] The method I have presented in Chapter 6 could be adapted to OCT by utilising two mixed-effects model: one for the spatial dimensions in the OCT scan and one for the temporal dimension between time points.

One recently proposed solution to the problem of undervalidated models is to commercialise models with the money being given to the researchers.[192] Blockchain technology has been proposed to create a marketplace for risk prediction models. Each time a model is used, the clinician would pay a fee, and the model developed would receive income. This would encourage the developer to validate the model so that it could be placed on the platform. The authors suggested that a body such as Public Health England could set up such a platform.

There are several serious issues with this approach. Firstly, a complicated blockchain solution may not be necessary for such a marketplace, and a REST API may suffice. Secondly, risk prediction models are challenging to commercialise. Although the model can be identified as an IP, it is often difficult to extract value from that IP. Journals increasingly require open access or access upon request for datasets, and the mathematical algorithms are notoriously difficult to patent and defend. Many algorithms are already open source, meaning that the developer would need to develop completely novel algorithms in the hope that they are patentable. Once the model is placed behind a paywall, nothing stops someone from adding an extra parameter, a slightly different technique, or different data and calling it a new IP. For an algorithm to be commercialisable, it may not be made publishable, meaning that the developer would have to hold off publishing their work, as they do in industry. It is unclear who would be paid for the algorithm; with so many people involved in the development and validation of models, it may be impossible to decide who gets what. Also, would the developer or the validator be paid? If the goal is to encourage validation, then it would be logical to reward the validator. Many groups may see validation as an easy way to make quick money and focus solely on validating models. If groups are encouraged to both develop and validate models themselves, then it may promote falsifying results to show better performance to increase the chances of the model being used. A model is nothing without data, so there is a strong argument that the patients should be paid more than any researcher.

Finally, putting models which would usually be free to use behind a paywall will inevitably increase healthcare inequality; countries and settings which cannot afford to pay per patient would not have access to these models. Some healthcare workers may feel that this violates their oath under the Declaration of Geneva 2017[193], which states, “I WILL SHARE my medical knowledge for the benefit of the patient and the advancement of healthcare.” These problems may have solutions that can enable this idea to work. For example, the model may be made freely available, and the service made the commercialisable product. However, this would mean the app developers would likely get the majority of the revenue, leaving very little for the researchers developing the models. A simpler alternative to forcing users to pay for prediction models is for funders, publishers, and institutions to begin recognising that model validation benefits everyone.

7.4 Main conclusions and contributions

The main conclusions of my work are as follows:

1. I have introduced a novel mechanism that accounts for the uneven intervals between time points.
2. My new GEV activation function provides an alternative to conventional class imbalance corrections. Very recent work by Goorbergh et al. has highlighted the problems associated with class imbalance corrections, such as oversampling.[194] My novel solution uses a distribution better suited to long-tailed data. The GEV is made up of three separate distributions: Gumbel, Fréchet, and Weibull, with the distribution chosen by the parameter ξ. The parameters are learned in the deep learning model. This means that the model is fit to the data, rather than the data being fit to the model as is done with sampling methods. This has been shown to provide improved results over the sigmoid activation when data is highly imbalanced and similar results to the sigmoid activation when the data is balanced. The GEV activation may also be useful in other applications where the sigmoid is used, such as binary segmentation.
3. Combining both my GEV activation and my interval scaling methods, I have developed a method for developing longitudinal prognostic models using

images. These models showed good discriminative performance at one, two, and three years.

4. I developed a VAR model, which shows reduced computation time over a GRU layer with similar results. This VAR layer contains more parameters than GRU, but is a much simpler calculation and may suggest that complicated RNNs are not always necessary.
5. I have created a mixed-effects layer in deep learning, with a new loss function to enforce the normality of the random-effects errors. When applied to a dataset of CT scans, I have shown that the mixed-effects layer has improved generalisability in an external validation dataset with improved calibration over previous methods. My mixed-effects method also allows for missing images or incomplete scans, with good performance observed even with up to 50% missing data.
6. Finally, I aimed to combine my mixed-effects layer with a survival layer to produce a longitudinal survival model. This resulted in a model similar to a joint effects model. I found that the exponential hazard function may be suitable in my particular application.

References

1. Riley RD, van der Windt D, Croft P, et al. Prognosis research in healthcare: concepts, methods, and impact: Oxford University Press 2019.
2. Apgar V. A proposal for a new method of evaluation of the newborn. *Classic Papers in Critical Care* 1952;32(449):97.
3. D'Agostino RB, Vasan RS, Pencina MJ, et al. General Cardiovascular Risk Profile for Use in Primary Care. *Circulation* 2008;117(6):743-53. doi: doi:10.1161/CIRCULATIONAHA.107.699579
4. Eleuteri A, Taktak AFG, Coupland SE, et al. Prognostication of metastatic death in uveal melanoma patients: A Markov multi-state model. *Computers in Biology and Medicine* 2018;102:151-56. doi: https://doi.org/10.1016/j.compbiomed.2018.09.024
5. Department of Health. Long Term Conditions Compendium of Information Third Edition 2012 [Available from: https://www.gov.uk/government/publications/long-term-conditions-compendium-of-information-third-edition accessed 02/03/2022 2022.
6. British Medical Association. NHS diagnostics data analysis 2022 [Available from: https://www.bma.org.uk/advice-and-support/nhs-delivery-and-workforce/pressures/nhs-diagnostics-data-analysis accessed 02/03/2022 2022.
7. Steyerberg EW, Moons KGM, van der Windt DA, et al. Prognosis Research Strategy (PROGRESS) 3: Prognostic Model Research. *PLOS Medicine* 2013;10(2):e1001381. doi: 10.1371/journal.pmed.1001381
8. Longoni C, Bonezzi A, Morewedge CK. Resistance to Medical Artificial Intelligence. *Journal of Consumer Research* 2019;46(4):629-50. doi: 10.1093/jcr/ucz013
9. Hesse BW, Nelson DE, Kreps GL, et al. Trust and Sources of Health Information: The Impact of the Internet and Its Implications for Health Care Providers: Findings From the First Health Information National Trends Survey. *Archives of Internal Medicine* 2005;165(22):2618-24. doi: 10.1001/archinte.165.22.2618
10. Collins GS, Reitsma JB, Altman DG, et al. Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): the TRIPOD Statement. *BMC Medicine* 2015;13(1):1. doi: 10.1186/s12916-014-0241-z
11. Chakravarthy U, Wong TY, Fletcher A, et al. Clinical risk factors for age-related macular degeneration: a systematic review and meta-analysis. *BMC Ophthalmology* 2010;10(1):31. doi: 10.1186/1471-2415-10-31
12. Scheufele TA, McHenry JG, Edwards AO. Optic Neuropathy and Age–Related Macular Degeneration. *Investigative Ophthalmology & Visual Science* 2004;45(13):1627-27.
13. Moons KGM, Royston P, Vergouwe Y, et al. Prognosis and prognostic research: what, why, and how? *BMJ* 2009;338:b375. doi: 10.1136/bmj.b375
14. Van Calster B, McLernon DJ, van Smeden M, et al. Calibration: the Achilles heel of predictive analytics. *BMC Medicine* 2019;17(1):230. doi: 10.1186/s12916-019-1466-7
15. Van Calster B, Nieboer D, Vergouwe Y, et al. A calibration hierarchy for risk models was defined: from utopia to empirical data. *Journal of Clinical Epidemiology* 2016;74:167-76. doi: 10.1016/j.jclinepi.2015.12.005
16. Vickers AJ, Elkin EB. Decision Curve Analysis: A Novel Method for Evaluating Prediction Models. *Medical Decision Making* 2006;26(6):565-74. doi: 10.1177/0272989X06295361

17. Newson R. Confidence Intervals for Rank Statistics: Somers' D and Extensions. *The Stata Journal* 2006;6(3):309-34. doi: 10.1177/1536867X0600600302
18. Beyene KM, El Ghouch A. Smoothed time-dependent receiver operating characteristic curve for right censored survival data. *Statistics in Medicine* 2020;39(24):3373-96. doi: https://doi.org/10.1002/sim.8671
19. Austin PC, Harrell Jr FE, van Klaveren D. Graphical calibration curves and the integrated calibration index (ICI) for survival models. *Statistics in Medicine* 2020;39(21):2714-42. doi: https://doi.org/10.1002/sim.8570
20. Steyerberg EW. Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating: Springer International Publishing 2019.
21. Vickers AJ, Cronin AM, Elkin EB, et al. Extensions to decision curve analysis, a novel method for evaluating diagnostic tests, prediction models and molecular markers. *BMC Med Inform Decis Mak* 2008;8:53-53. doi: 10.1186/1472-6947-8-53
22. Heagerty PJ, Lumley T, Pepe MS. Time‐dependent ROC curves for censored survival data and a diagnostic marker. *Biometrics* 2000;56(2):337-44.
23. Young T. II. The Bakerian Lecture. On the theory of light and colours. *Philosophical Transactions of the Royal Society of London* 1802;92:12-48. doi: doi:10.1098/rstl.1802.0004
24. Maxwell JC. XVIII.—Experiments on Colour, as perceived by the Eye, with Remarks on Colour-Blindness. *Transactions of the Royal Society of Edinburgh* 1857;21(2):275-98. doi: 10.1017/S0080456800032117 [published Online First: 2013/01/17]
25. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;521(7553):436-44. doi: 10.1038/nature14539
26. Tappert CC. Who Is the Father of Deep Learning? *2019 International Conference on Computational Science and Computational Intelligence (CSCI)* 2019:343-48. doi: 10.1109/CSCI49370.2019.00067
27. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* 2010;9:249--56.
28. Géron A. Hands-On Machine Learning with Scikit-Learn & TensorFlow: O'Reilly 2017.
29. Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 1998;86(11):2278-324. doi: 10.1109/5.726791
30. Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Computation* 1997;9(8):1735-80. doi: 10.1162/neco.1997.9.8.1735
31. Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. *1999 Ninth International Conference on Artificial Neural Networks ICANN 99 (Conf Publ No 470)* 1999;2:850-55 vol.2. doi: 10.1049/cp:19991218
32. Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:14061078* 2014
33. Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:14123555* 2014
34. Su Y, Kuo CCJ. On extended long short-term memory and dependent bidirectional recurrent neural network. *Neurocomputing* 2019;356:151-61. doi: https://doi.org/10.1016/j.neucom.2019.04.044
35. Greff K, Srivastava RK, Koutník J, et al. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems* 2016;28(10):2222-32.

36. Ming L, Xiaolin H. Recurrent convolutional neural network for object recognition. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2015:3367-75. doi: 10.1109/CVPR.2015.7298958
37. Brown T, Mann B, Ryder N, et al. Language models are few-shot learners. *Advances in neural information processing systems* 2020;33:1877-901.
38. Fedus W, Zoph B, Shazeer N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:210103961* 2021
39. Hinton GE, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:12070580* 2012
40. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning* 2015:448-56.
41. Li X, Chen S, Hu X, et al. Understanding the Disharmony Between Dropout and Batch Normalization by Variance Shift. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2019:2677-85. doi: 10.1109/CVPR.2019.00279
42. Bridle JS. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. *Neurocomputing* 1990:227-36.
43. Maas AL. Rectifier Nonlinearities Improve Neural Network Acoustic Models. 2013
44. He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision* 2015:1026-34.
45. Xu B, Wang N, Chen T, et al. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:150500853* 2015
46. Clevert D-A, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:151107289* 2015
47. Ramachandran P, Zoph B, Le QV. Searching for activation functions. *arXiv preprint arXiv:171005941* 2017
48. Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 2015;115(3):211-52. doi: 10.1007/s11263-015-0816-y
49. Yuan L, Chen D, Chen Y-L, et al. Florence: A New Foundation Model for Computer Vision. *arXiv preprint arXiv:211111432* 2021
50. Pham H, Dai Z, Xie Q, et al. Meta pseudo labels. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2021:11557-68.
51. Tsipras D, Santurkar S, Engstrom L, et al. From imagenet to image classification: Contextualizing progress on benchmarks. *International Conference on Machine Learning* 2020:9625-35.
52. LeCun Y. Generalization and network design strategies. 1989
53. LeCun Y, Boser B, Denker JS, et al. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* 1989;1(4):541-51. doi: 10.1162/neco.1989.1.4.541
54. LeCun Y, Boser B, Denker JS, et al. Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems 2: Morgan Kaufmann Publishers Inc. 1990:396–404.
55. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM* 2017;60(6):84–90. doi: 10.1145/3065386
56. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:14091556* 2014

57. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* 2015:1-9.
58. Nolan C. Inception, 2010.
59. Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning. *Thirty-first AAAI conference on artificial intelligence* 2017
60. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition* 2016:770-78.
61. Learning transferable architectures for scalable image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition; 2018.
62. Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning; 2019. PMLR.
63. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. *Advances in neural information processing systems* 2017;30
64. Cbam: Convolutional block attention module. Proceedings of the European conference on computer vision (ECCV); 2018.
65. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:201011929* 2020
66. Brier GW. VERIFICATION OF FORECASTS EXPRESSED IN TERMS OF PROBABILITY. *Monthly Weather Review* 1950;78(1):1-3. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2
67. Murphy AH. A New Vector Partition of the Probability Score. *Journal of Applied Meteorology (1962-1982)* 1973;12(4):595-600.
68. Bermejo S, Cabestany J. Oriented principal component analysis for large margin classifiers. *Neural Networks* 2001;14(10):1447-61. doi: https://doi.org/10.1016/S0893-6080(01)00106-X
69. Lin T-Y, Goyal P, Girshick R, et al. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision* 2017:2980-88.
70. Rosasco L, Vito ED, Caponnetto A, et al. Are Loss Functions All the Same? *Neural Computation* 2004;16(5):1063-76. doi: 10.1162/089976604773135104
71. Kullback S, Leibler RA. On Information and Sufficiency. *The Annals of Mathematical Statistics* 1951;22(1):79-86. doi: 10.1214/aoms/1177729694
72. Sørenson T. A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons: I kommission hos E. Munksgaard 1948.
73. Dice LR. Measures of the Amount of Ecologic Association Between Species. *Ecology* 1945;26(3):297-302. doi: https://doi.org/10.2307/1932409
74. Chen X, Williams BM, Vallabhaneni SR, et al. Learning active contour models for medical image segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* 2019:11632-40.
75. Klambauer G, Unterthiner T, Mayr A, et al. Self-normalizing neural networks. *Advances in neural information processing systems* 2017;30
76. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature* 1986;323(6088):533-36. doi: 10.1038/323533a0
77. Polyak BT. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics* 1964;4(5):1-17. doi: https://doi.org/10.1016/0041-5553(64)90137-5

78. Nesterov Y. A method for unconstrained convex minimization problem with the rate of convergence o(1/k^2). 1983

79. Duchi J, Hazan E, Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J Mach Learn Res* 2011;12(null):2121–59.

80. Hinton G. Neural Networks for Machine Learning 2012 [

81. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:14126980* 2014

82. Smilkov D, Thorat N, Kim B, et al. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:170603825* 2017

83. Oramas J, Wang K, Tuytelaars T. Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks. *arXiv preprint arXiv:171206302* 2017

84. SCOUTER: Slot attention-based classifier for explainable image recognition. Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021.

85. McCoy LG, Brenna CTA, Chen SS, et al. Believing in black boxes: machine learning for healthcare does not need explainability to be evidence-based. *Journal of Clinical Epidemiology* 2022;142:252-57. doi: 10.1016/j.jclinepi.2021.11.001

86. Chakravarthy U, Evans J, Rosenfeld PJ. Age related macular degeneration. *BMJ* 2010;340:c981. doi: 10.1136/bmj.c981

87. World Health Organization. World report on vision. 2019 doi: https://www.who.int/publications/i/item/9789241516570

88. Wong WL, Su X, Li X, et al. Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: a systematic review and meta-analysis. *The Lancet Global Health* 2014;2(2):e106-e16. doi: https://doi.org/10.1016/S2214-109X(13)70145-1

89. Solomon SD, Lindsley K, Vedula SS, et al. Anti‐vascular endothelial growth factor for neovascular age‐related macular degeneration. *Cochrane Database of Systematic Reviews* 2019(3) doi: 10.1002/14651858.CD005139.pub4

90. Wormald R, Evans JR, Smeeth LL, et al. Photodynamic therapy for neovascular age‐related macular degeneration. *Cochrane Database of Systematic Reviews* 2007(3) doi: 10.1002/14651858.CD002030.pub3

91. Tierney JF, Vale C, Riley R, et al. Individual Participant Data (IPD) Meta-analyses of Randomised Controlled Trials: Guidance on Their Use. *PLOS Medicine* 2015;12(7):e1001855. doi: 10.1371/journal.pmed.1001855

92. Bhandari S, Vitale S, Agrón E, et al. Cataract Surgery and the Risk of Developing Late Age-Related Macular Degeneration: The Age-Related Eye Disease Study 2 Report Number 27. *Ophthalmology* doi: 10.1016/j.ophtha.2021.11.014

93. Jackman WT, ; Webster, J. D. On photographing the retina of the living eye. *Philadelphia Photographer* 1886;23:340-41.

94. Dimmer F. Die Photographie des Augenhintergrundes. 1907

95. Age-Related Eye Disease Study Research Group. The Age-Related Eye Disease Study (AREDS): Design Implications AREDS Report No. 1. *Controlled Clinical Trials* 1999;20(6):573-600. doi: https://doi.org/10.1016/S0197-2456(99)00031-8

96. Age-Related Eye Disease Study Research Group. A Randomized, Placebo-Controlled, Clinical Trial of High-Dose Supplementation With Vitamins C and E, Beta Carotene, and Zinc for Age-Related Macular Degeneration and Vision Loss: AREDS Report No. 8. *Archives of Ophthalmology* 2001;119(10):1417-36. doi: 10.1001/archopht.119.10.1417

97. The Age-Related Eye Disease Study 2 Research Group. Lutein + Zeaxanthin and Omega-3 Fatty Acids for Age-Related Macular Degeneration: The Age-Related Eye Disease Study 2 (AREDS2) Randomized Clinical Trial. *JAMA* 2013;309(19):2005-15. doi: 10.1001/jama.2013.4997

98. Areds Research Group, Chew EY, Clemons T, et al. The Age-Related Eye Disease Study 2 (AREDS2): study design and baseline characteristics (AREDS2 report number 1). *Ophthalmology* 2012;119(11):2282-89. doi: 10.1016/j.ophtha.2012.05.027 [published Online First: 2012/07/26]

99. Seddon JM, Silver RE, Rosner B. Response to AREDS supplements according to genetic factors: survival analysis approach using the eye as the unit of analysis. *British Journal of Ophthalmology* 2016;100(12):1731-37. doi: 10.1136/bjophthalmol-2016-308624

100. Age-Related Eye Disease Study Research Group. The effect of five-year zinc supplementation on serum zinc, serum cholesterol and hematocrit in persons randomly assigned to treatment group in the age-related eye disease study: AREDS Report No. 7. *J Nutr* 2002;132(4):697-702. doi: 10.1093/jn/132.4.697

101. Hammond BR, Jr, Renzi-Hammond LM. Perspective: A Critical Look at the Ancillary Age-Related Eye Disease Study 2: Nutrition and Cognitive Function Results in Older Individuals with Age-Related Macular Degeneration. *Advances in Nutrition* 2016;7(3):433-37. doi: 10.3945/an.115.011866

102. Age-Related Eye Disease Study Research Group. The Age-Related Eye Disease Study system for classifying age-related macular degeneration from stereoscopic color fundus photographs: the Age-Related Eye Disease Study Report Number 6. *American journal of ophthalmology* 2001;132(5):668-81.

103. Coleman HR, Chan C-C, Ferris FL, III, et al. Age-related macular degeneration. *The Lancet* 2008;372(9652):1835-45. doi: 10.1016/S0140-6736(08)61759-6

104. Law SK, Sohn YH, Hoffman D, et al. Optic disk appearance in advanced age-related macular degeneration. *American Journal of Ophthalmology* 2004;138(1):38-45. doi: 10.1016/j.ajo.2004.02.021

105. Mistry DA, Wang JY, Moeser M-E, et al. A systematic review of the sensitivity and specificity of lateral flow devices in the detection of SARS-CoV-2. *BMC Infectious Diseases* 2021;21(1):828. doi: 10.1186/s12879-021-06528-3

106. Dinnes J, Deeks JJ, Berhane S, et al. Rapid, point‐of‐care antigen and molecular‐based tests for diagnosis of SARS‐CoV‐2 infection. *Cochrane database of systematic reviews* 2021(3)

107. Griffin S. Covid-19: Lateral flow tests are better at identifying people with symptoms, finds Cochrane review. *BMJ* 2021;372:n823. doi: 10.1136/bmj.n823

108. Kortela E, Kirjavainen V, Ahava MJ, et al. Real-life clinical sensitivity of SARS-CoV-2 RT-PCR test in symptomatic patients. *PLOS ONE* 2021;16(5):e0251661. doi: 10.1371/journal.pone.0251661

109. Zu ZY, Jiang MD, Xu PP, et al. Coronavirus Disease 2019 (COVID-19): A Perspective from China. *Radiology* 2020;296(2):E15-E25. doi: 10.1148/radiol.2020200490

110. Fang Y, Zhang H, Xie J, et al. Sensitivity of Chest CT for COVID-19: Comparison to RT-PCR. *Radiology* 2020;296(2):E115-E17. doi: 10.1148/radiol.2020200432

111. Hemingway H, Croft P, Perel P, et al. Prognosis research strategy (PROGRESS) 1: A framework for researching clinical outcomes. *BMJ : British Medical Journal* 2013;346:e5595. doi: 10.1136/bmj.e5595

112. Riley RD, Hayden JA, Steyerberg EW, et al. Prognosis Research Strategy (PROGRESS) 2: Prognostic Factor Research. *PLOS Medicine* 2013;10(2):e1001380. doi: 10.1371/journal.pmed.1001380

113. Hingorani AD, Windt DAvd, Riley RD, et al. Prognosis research strategy (PROGRESS) 4: Stratified medicine research. *BMJ : British Medical Journal* 2013;346:e5793. doi: 10.1136/bmj.e5793

114. Cancer Research UK. Eye Cancer Survival 2021 [accessed 20/02/2022 2022.

115. Damato B, Heimann H. Personalized treatment of uveal melanoma. *Eye* 2013;27(2):172-79.

116. Hippocrates. Epidemics I. 400 BC

117. Holz FG, Tadayoni R, Beatty S, et al. Key drivers of visual acuity gains in neovascular age-related macular degeneration in real life: findings from the AURA study. *British Journal of Ophthalmology* 2016;100(12):1623-28. doi: 10.1136/bjophthalmol-2015-308166

118. Wilson PWF, D'Agostino RB, Levy D, et al. Prediction of Coronary Heart Disease Using Risk Factor Categories. *Circulation* 1998;97(18):1837-47. doi: 10.1161/01.CIR.97.18.1837

119. Damen JA, Pajouheshnia R, Heus P, et al. Performance of the Framingham risk models and pooled cohort equations for predicting 10-year risk of cardiovascular disease: a systematic review and meta-analysis. *BMC Medicine* 2019;17(1):109. doi: 10.1186/s12916-019-1340-7

120. Eagle KA, Lim MJ, Dabbous OH, et al. A Validated Prediction Model for All Forms of Acute Coronary SyndromeEstimating the Risk of 6-Month Postdischarge Death in an International Registry. *JAMA* 2004;291(22):2727-33. doi: 10.1001/jama.291.22.2727

121. Fox KAA, FitzGerald G, Puymirat E, et al. Should patients with acute coronary disease be stratified for management according to their risk? Derivation, external validation and outcomes using the updated GRACE risk score. *BMJ Open* 2014;4(2):e004425. doi: 10.1136/bmjopen-2013-004425

122. Age-Related Eye Disease Study Research Group. The Age-Related Eye Disease Study Severity Scale for Age-Related Macular Degeneration: AREDS Report No. 17. *Archives of Ophthalmology* 2005;123(11):1484-98. doi: 10.1001/archopht.123.11.1484

123. Landis JR, Koch GG. The measurement of observer agreement for categorical data. *biometrics* 1977:159-74.

124. Mitchell P, Foran S. Age-Related Eye Disease Study Severity Scale and Simplified Severity Scale for Age-Related Macular Degeneration. *Archives of Ophthalmology* 2005;123(11):1598-99. doi: 10.1001/archopht.123.11.1598

125. Age-Related Eye Disease Study Research Group. A Simplified Severity Scale for Age-Related Macular Degeneration: AREDS Report No. 18. *Archives of Ophthalmology* 2005;123(11):1570-74. doi: 10.1001/archopht.123.11.1570

126. Vitale S, Clemons TE, Agrón E, et al. Evaluating the Validity of the Age-Related Eye Disease Study Grading Scale for Age-Related Macular Degeneration: AREDS2 Report 10. *JAMA Ophthalmology* 2016;134(9):1041-47. doi: 10.1001/jamaophthalmol.2016.2383

127. Collins GS, Ogundimu EO, Cook JA, et al. Quantifying the impact of different approaches for handling continuous predictors on the performance of a prognostic model. *Statistics in Medicine* 2016;35(23):4124-35. doi: https://doi.org/10.1002/sim.6986

128. Altman DG. Categorizing Continuous Variables. Wiley StatsRef: Statistics Reference Online2014.

129. Seddon JM, Reynolds R, Yu Y, et al. Risk Models for Progression to Advanced Age-Related Macular Degeneration Using Demographic, Environmental, Genetic, and Ocular Factors. *Ophthalmology* 2011;118(11):2203-11. doi: 10.1016/j.ophtha.2011.04.029

130. Cox DR. Regression Models and Life-Tables. *Journal of the Royal Statistical Society: Series B (Methodological)* 1972;34(2):187-202. doi: https://doi.org/10.1111/j.2517-6161.1972.tb00899.x

131. Seddon JM, Silver RE, Kwong M, et al. Risk Prediction for Progression of Macular Degeneration: 10 Common and Rare Genetic Variants, Demographic, Environmental, and Macular Covariates. *Investigative Ophthalmology & Visual Science* 2015;56(4):2192-202. doi: 10.1167/iovs.14-15841

132. Shin KU, Song SJ, Bae JH, et al. Risk Prediction Model for Progression of Age-Related Macular Degeneration. *Ophthalmic Research* 2017;57(1):32-36. doi: 10.1159/000449168

133. Hilario A, Sepulveda JM, Perez-Nuñez A, et al. A Prognostic Model Based on Preoperative MRI Predicts Overall Survival in Patients with Diffuse Gliomas. *American Journal of Neuroradiology* 2014;35(6):1096-102. doi: 10.3174/ajnr.A3837

134. Arenja N, Andre F, Riffel JH, et al. Prognostic value of novel imaging parameters derived from standard cardiovascular magnetic resonance in high risk patients with systemic light chain amyloidosis. *Journal of Cardiovascular Magnetic Resonance* 2019;21(1):53. doi: 10.1186/s12968-019-0564-1

135. Banerjee I, de Sisternes L, Hallak J, et al. A deep-learning approach for prognosis of age-related macular degeneration disease using SD-OCT imaging biomarkers. *arXiv preprint arXiv:190210700* 2019

136. Fujimoto JG, Pitris C, Boppart SA, et al. Optical coherence tomography: an emerging technology for biomedical imaging and optical biopsy. *Neoplasia* 2000;2(1-2):9-25. doi: 10.1038/sj.neo.7900071

137. Chen Q, Leng T, Zheng L, et al. Automated drusen segmentation and quantification in SD-OCT images. *Medical Image Analysis* 2013;17(8):1058-72. doi: https://doi.org/10.1016/j.media.2013.06.003

138. Niu S, de Sisternes L, Chen Q, et al. Fully Automated Prediction of Geographic Atrophy Growth Using Quantitative Spectral-Domain Optical Coherence Tomography Biomarkers. *Ophthalmology* 2016;123(8):1737-50. doi: 10.1016/j.ophtha.2016.04.042

139. Yim J, Chopra R, Spitz T, et al. Predicting conversion to wet age-related macular degeneration using deep learning. *Nature Medicine* 2020;26(6):892-99. doi: 10.1038/s41591-020-0867-7

140. Katzman JL, Shaham U, Cloninger A, et al. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology* 2018;18(1):24. doi: 10.1186/s12874-018-0482-1

141. Zhu X, Yao J, Huang J. Deep convolutional neural network for survival analysis with pathological images. *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* 2016:544-47. doi: 10.1109/BIBM.2016.7822579

142. Arcadu F, Benmansour F, Maunz A, et al. Deep learning algorithm predicts diabetic retinopathy progression in individual patients. *npj Digital Medicine* 2019;2(1):92. doi: 10.1038/s41746-019-0172-3

143. Babenko B, Balasubramanian S, Blumer KE, et al. Predicting progression of age-related macular degeneration from fundus images using deep learning. *arXiv preprint arXiv:190405478* 2019

144. Grassmann F, Mengelkamp J, Brandl C, et al. A Deep Learning Algorithm for Prediction of Age-Related Eye Disease Study Severity Scale for Age-Related Macular Degeneration from Color Fundus Photography. *Ophthalmology* 2018;125(9):1410-20. doi: 10.1016/j.ophtha.2018.02.037

145. Yan Q, Weeks DE, Xin H, et al. Deep-learning-based prediction of late age-related macular degeneration progression. *Nature Machine Intelligence* 2020;2(2):141-50. doi: 10.1038/s42256-020-0154-9

146. Peng Y, Keenan TD, Chen Q, et al. Predicting risk of late age-related macular degeneration using deep learning. *npj Digital Medicine* 2020;3(1):111. doi: 10.1038/s41746-020-00317-z

147. Romo-Bucheli D, Erfurth US, Bogunović H. End-to-End Deep Learning Model for Predicting Treatment Requirements in Neovascular AMD From Longitudinal Retinal OCT Imaging. *IEEE Journal of Biomedical and Health Informatics* 2020;24(12):3456-65. doi: 10.1109/JBHI.2020.3000136

148. Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition* 2017:4700-08.

149. Wang T, Qiu RG, Yu M. Predictive Modeling of the Progression of Alzheimer's Disease with Recurrent Neural Networks. *Scientific Reports* 2018;8(1):9161. doi: 10.1038/s41598-018-27337-w

150. Pham T, Tran T, Phung D, et al. Predicting healthcare trajectories from medical records: A deep learning approach. *Journal of Biomedical Informatics* 2017;69:218-29. doi: https://doi.org/10.1016/j.jbi.2017.04.001

151. Patient subtyping via time-aware LSTM networks. Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining; 2017.

152. National Eye Institute. AREDS/AREDS2 Clinical Trials 2020 [Available from: https://www.nei.nih.gov/research/clinical-trials/age-related-eye-disease-studies-aredsareds2/about-areds-and-areds2.

153. Jakobsen JC, Gluud C, Wetterslev J, et al. When and how should multiple imputation be used for handling missing data in randomised clinical trials – a practical guide with flowcharts. *BMC Medical Research Methodology* 2017;17(1):162. doi: 10.1186/s12874-017-0442-1

154. Domalpally A, Danis RP, Chew EY, et al. Evaluation of Optimized Digital Fundus Reflex Photographs for Lens Opacities in the Age-Related Eye Disease Study 2: AREDS2 Report 7. *Investigative Ophthalmology & Visual Science* 2013;54(9):5989-94. doi: 10.1167/iovs.13-12301

155. Morozov SP, Andreychenko AE, Blokhin IA, et al. MosMedData: data set of 1110 chest CT scans performed during the COVID-19 epidemic. *DD* 2020;1(1):49-59. doi: 10.17816/dd46826 [published Online First: 2020-12-30]

156. Zhang K, Liu X, Shen J, et al. Clinically Applicable AI System for Accurate Diagnosis, Quantitative Measurements, and Prognosis of COVID-19 Pneumonia Using

Computed Tomography. *Cell* 2020;181(6):1423-33.e11. doi: https://doi.org/10.1016/j.cell.2020.04.045

157. Kermany DS, Goldbaum M, Cai W, et al. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell* 2018;172(5):1122-31.e9. doi: https://doi.org/10.1016/j.cell.2018.02.010
158. Cohen JP, Morrison P, Dao L. COVID-19 image data collection. *arXiv preprint arXiv:200311597* 2020
159. Societá italiana di radiologia medica e interventistica. COVID-19 Database 2020 [accessed 11 April 2020 2020.
160. Jaeger S, Candemir S, Antani S, et al. Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative Imaging in Medicine and Surgery* 2014;4(6):475-77.
161. Yang X, He X, Zhao J, et al. COVID-CT-dataset: a CT scan dataset about COVID-19. *arXiv preprint arXiv:200313865* 2020
162. Bridge J, Harding SP, Zheng Y. Development and Validation of a Novel Prognostic Model for Predicting AMD Progression Using Longitudinal Fundus Images. *arXiv preprint arXiv:200705120* 2020
163. Bridge J, Harding S, Zheng Y. Development and validation of a novel prognostic model for predicting AMD progression using longitudinal fundus images. *BMJ open ophthalmology* 2020;5(1):e000569.
164. Bridge J, Harding S, Zheng Y. End-to-End Deep Learning Vector Autoregressive Prognostic Models to Predict Disease Progression with Uneven Time Intervals. *Annual Conference on Medical Image Understanding and Analysis* 2021:517-31.
165. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition* 2016:2818-26.
166. Bridge J, Meng Y, Zhao Y, et al. Introducing the GEV activation function for highly unbalanced data to develop COVID-19 diagnostic models. *IEEE journal of Biomedical and Health Informatics* 2020;24(10):2776-86.
167. Bridge JT, Zheng Y. mGEV: Extension of the GEV Activation to Multiclass Classification. 2021
168. Platt J. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv Large Margin Classif* 2000;10
169. Avidan S, Brostow G, Cissé M, et al., eds. Long-Tailed Instance Segmentation Using Gumbel Optimized Loss. Computer Vision – ECCV 2022; 2022 2022//; Cham. Springer Nature Switzerland.
170. Li L, Wang X, Xu M, et al. DeepGF: Glaucoma Forecast Using the Sequential Fundus Images. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V* 2020:626–35. doi: 10.1007/978-3-030-59722-1_60
171. Wu Z, Wei J, Wang J, et al. Slice imputation: Multiple intermediate slices interpolation for anisotropic 3D medical image segmentation. *Computers in Biology and Medicine* 2022;147:105667. doi: https://doi.org/10.1016/j.compbiomed.2022.105667
172. Wynants L, Van Calster B, Collins GS, et al. Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. *BMJ* 2020;369:m1328. doi: 10.1136/bmj.m1328

173. Bridge J, Meng Y, Zhu W, et al. Development and External Validation of a Mixed-Effects Deep Learning Model to Diagnose COVID-19 from CT Imaging. *medRxiv* 2022:2022.01.28.22270005. doi: 10.1101/2022.01.28.22270005
174. Watson J, Whiting PF, Brush JE. Interpreting a covid-19 test result. *BMJ* 2020;369:m1808. doi: 10.1136/bmj.m1808
175. Mallett S, Allen AJ, Graziadio S, et al. At what times during infection is SARS-CoV-2 detectable and no longer detectable using RT-PCR-based tests? A systematic review of individual participant data. *BMC Medicine* 2020;18(1):346. doi: 10.1186/s12916-020-01810-8
176. Roberts M, Driggs D, Thorpe M, et al. Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans. *Nature Machine Intelligence* 2021;3(3):199-217. doi: 10.1038/s42256-021-00307-0
177. Bai HX, Wang R, Xiong Z, et al. Artificial Intelligence Augmentation of Radiologist Performance in Distinguishing COVID-19 from Pneumonia of Other Origin at Chest CT. *Radiology* 2020;296(3):E156-E65. doi: 10.1148/radiol.2020201491
178. Li L, Qin L, Xu Z, et al. Using Artificial Intelligence to Detect COVID-19 and Community-acquired Pneumonia Based on Pulmonary CT: Evaluation of the Diagnostic Accuracy. *Radiology* 2020;296(2):E65-E71. doi: 10.1148/radiol.2020200905
179. Bhuvan M, JungHwan O. CoviNet: Covid-19 diagnosis using machine learning analyses for computerized tomography images. *ProcSPIE* 2021;11878 doi: 10.1117/12.2601065
180. Zhu W, Ku JY, Zheng Y, et al. Spatial Linear Mixed Effects Modelling for OCT Images: SLME Model. *Journal of Imaging* 2020;6(6):44.
181. Bowman FD, Waller LA. Modelling of cardiac imaging data with spatial correlation. *Statistics in medicine* 2004;23(6):965-85.
182. Balki I, Amirabadi A, Levman J, et al. Sample-Size Determination Methodologies for Machine Learning in Medical Imaging Research: A Systematic Review. *Canadian Association of Radiologists Journal* 2019;70(4):344-53. doi: 10.1016/j.carj.2019.06.002
183. Cho J, Lee K, Shin E, et al. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *arXiv preprint arXiv:151106348* 2015
184. Rokem A, Wu Y, Lee A. Assessment of the need for separate test set and number of medical images necessary for deep learning: a sub-sampling study. *bioRxiv* 2017:196659. doi: 10.1101/196659
185. Riley RD, Snell KI, Ensor J, et al. Minimum sample size for developing a multivariable prediction model: PART II - binary and time-to-event outcomes. *Statistics in Medicine* 2019;38(7):1276-96. doi: https://doi.org/10.1002/sim.7992
186. Weibull W. A statistical distribution function of wide applicability. *Journal of applied mechanics* 1951
187. Gompertz B. XXIV. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. In a letter to Francis Baily, Esq. F. R. S. &amp;c. *Philosophical Transactions of the Royal Society of London* 1825;115:513-83. doi: doi:10.1098/rstl.1825.0026

188. Booth S, Riley RD, Ensor J, et al. Temporal recalibration for improving prognostic model development and risk predictions in settings where survival is improving over time. *International Journal of Epidemiology* 2020;49(4):1316-25. doi: 10.1093/ije/dyaa030

189. Rencher AC, Pun FC. Inflation of R<sup>2</sup> in Best Subset Regression. *Technometrics* 1980;22(1):49-53. doi: 10.2307/1268382

190. Midena E, Zennaro L, Lapo C, et al. Handheld Fundus Camera for Diabetic Retinopathy Screening: A Comparison Study with Table-Top Fundus Camera in Real-Life Setting. *Journal of Clinical Medicine* 2022;11(9):2352.

191. Sutton J, Menten MJ, Riedl S, et al. Developing and validating a multivariable prediction model which predicts progression of intermediate to late age-related macular degeneration—the PINNACLE trial protocol. *Eye* 2022 doi: 10.1038/s41433-022-02097-0

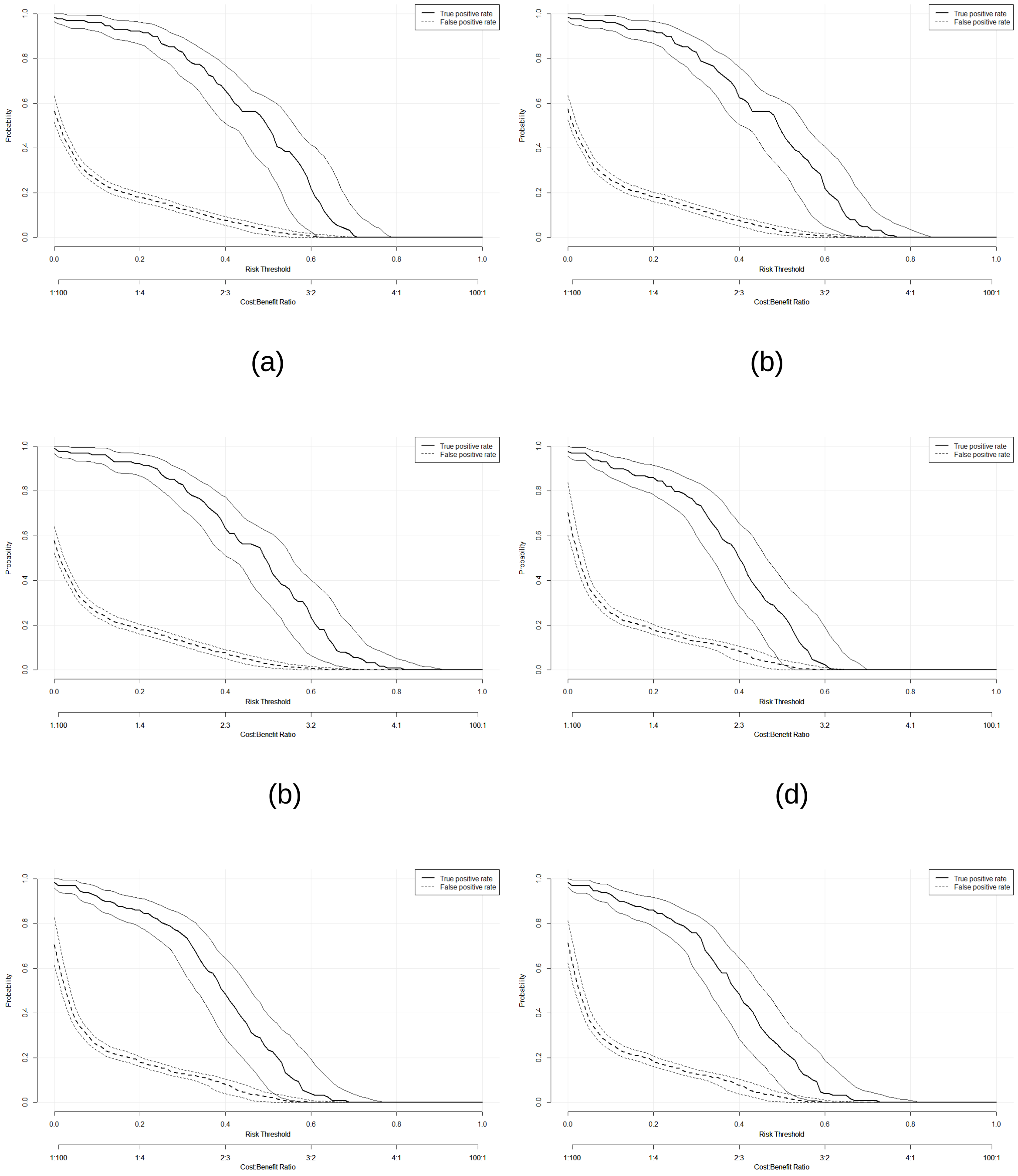
192. Sharma V, Davies A, Ainsworth J. Clinical risk prediction models: the canary in the coalmine for artificial intelligence in healthcare? *BMJ Health &amp;amp; Care Informatics* 2021;28(1):e100421. doi: 10.1136/bmjhci-2021-100421

193. World Medical Association. WMA Declaration of Geneva 2017 [

194. van den Goorbergh R, van Smeden M, Timmerman D, et al. The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression. *Journal of the American Medical Informatics Association* 2022;29(9):1525-34. doi: 10.1093/jamia/ocac093

Appendix A: Additional Figures from Chapter 4

A.1 ROC component curves



(a)

(b)

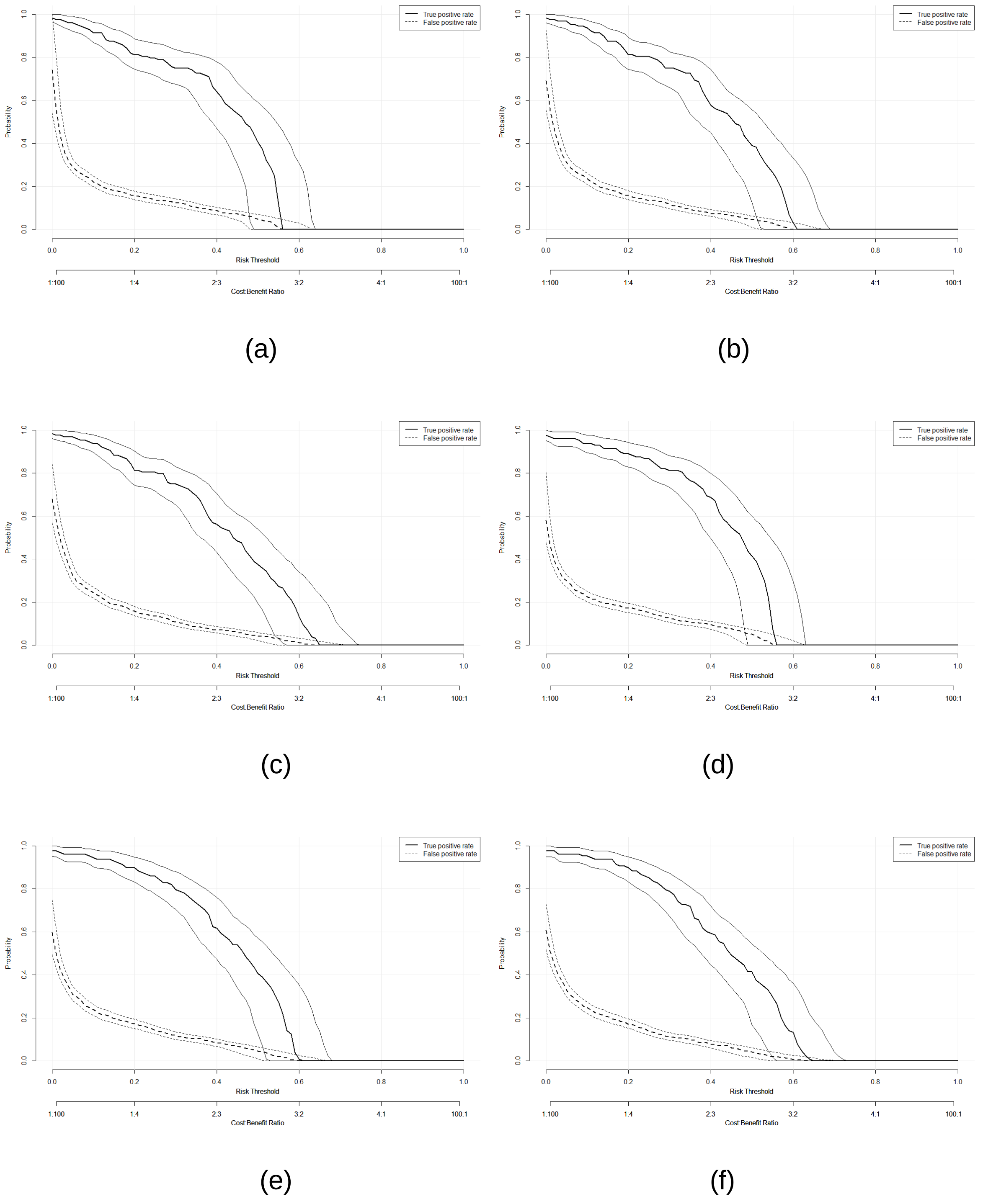
(b)

(d)

(e)

(f)

Figure A.1: ROC component curves for the single time point model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

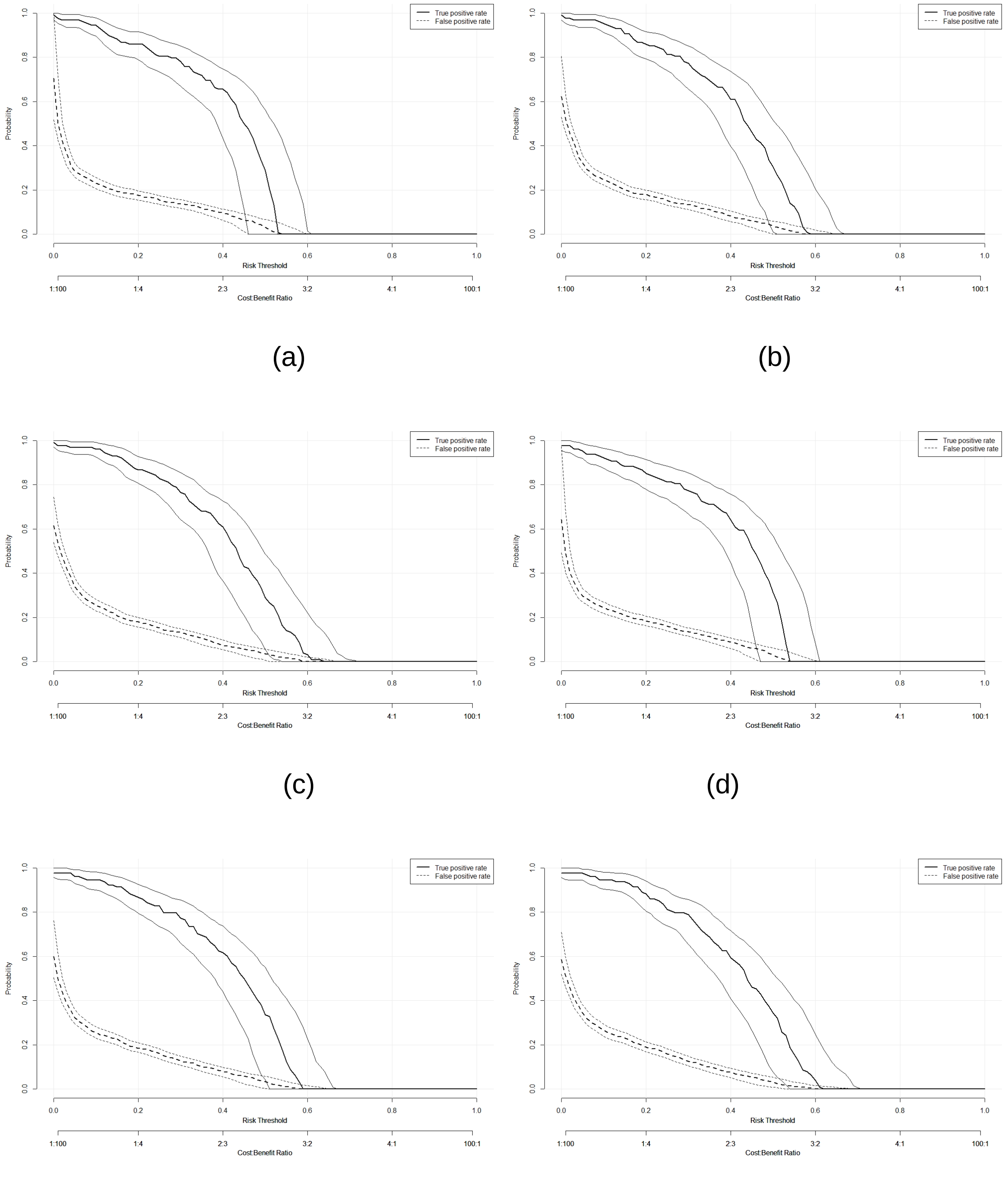
(c)

(d)

(e)

(f)

Figure A.2: ROC component curves for the GRU model with two time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

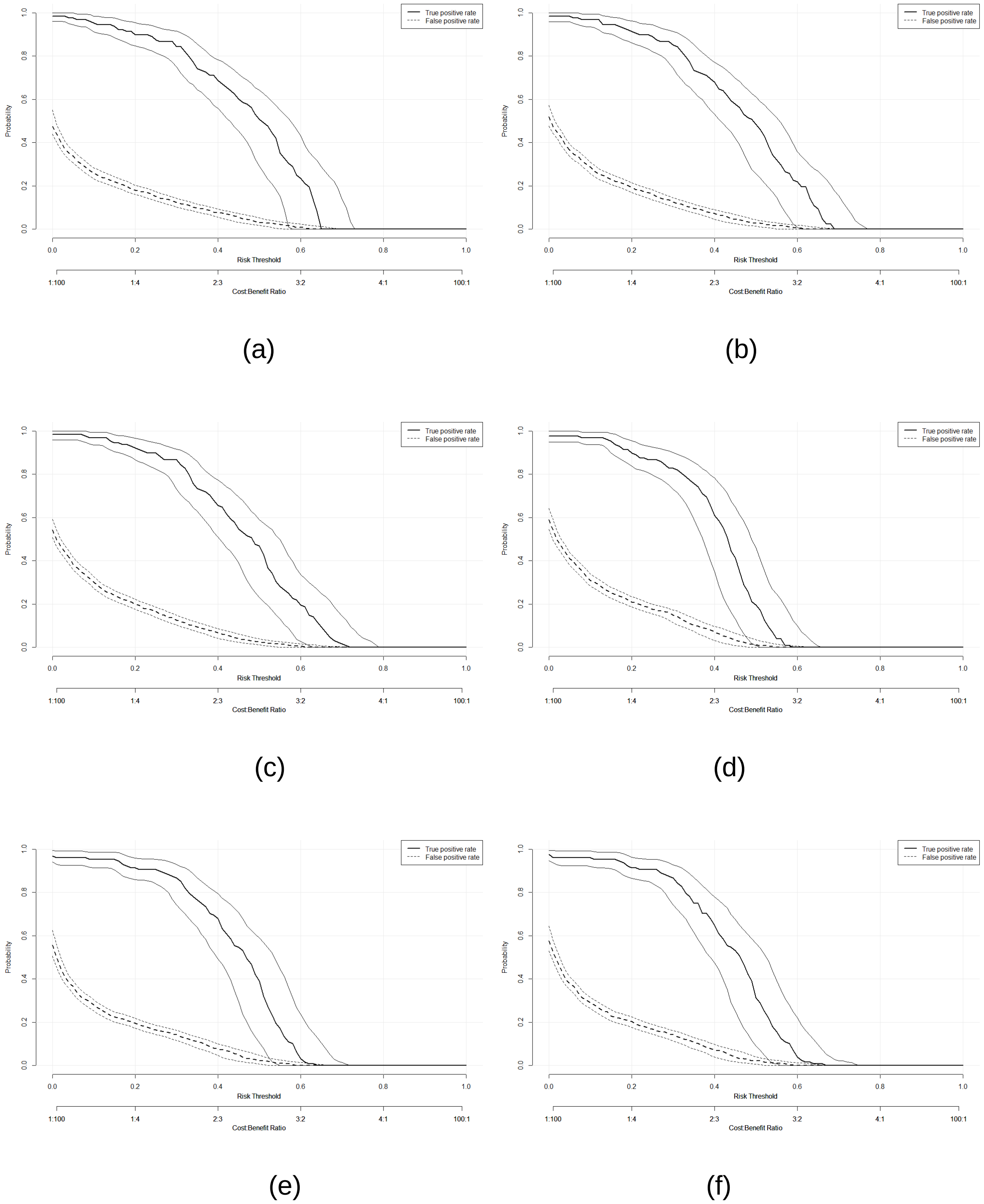
(c)

(d)

(e)

(f)

Figure A.3: ROC component curves for the GRU model with three time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

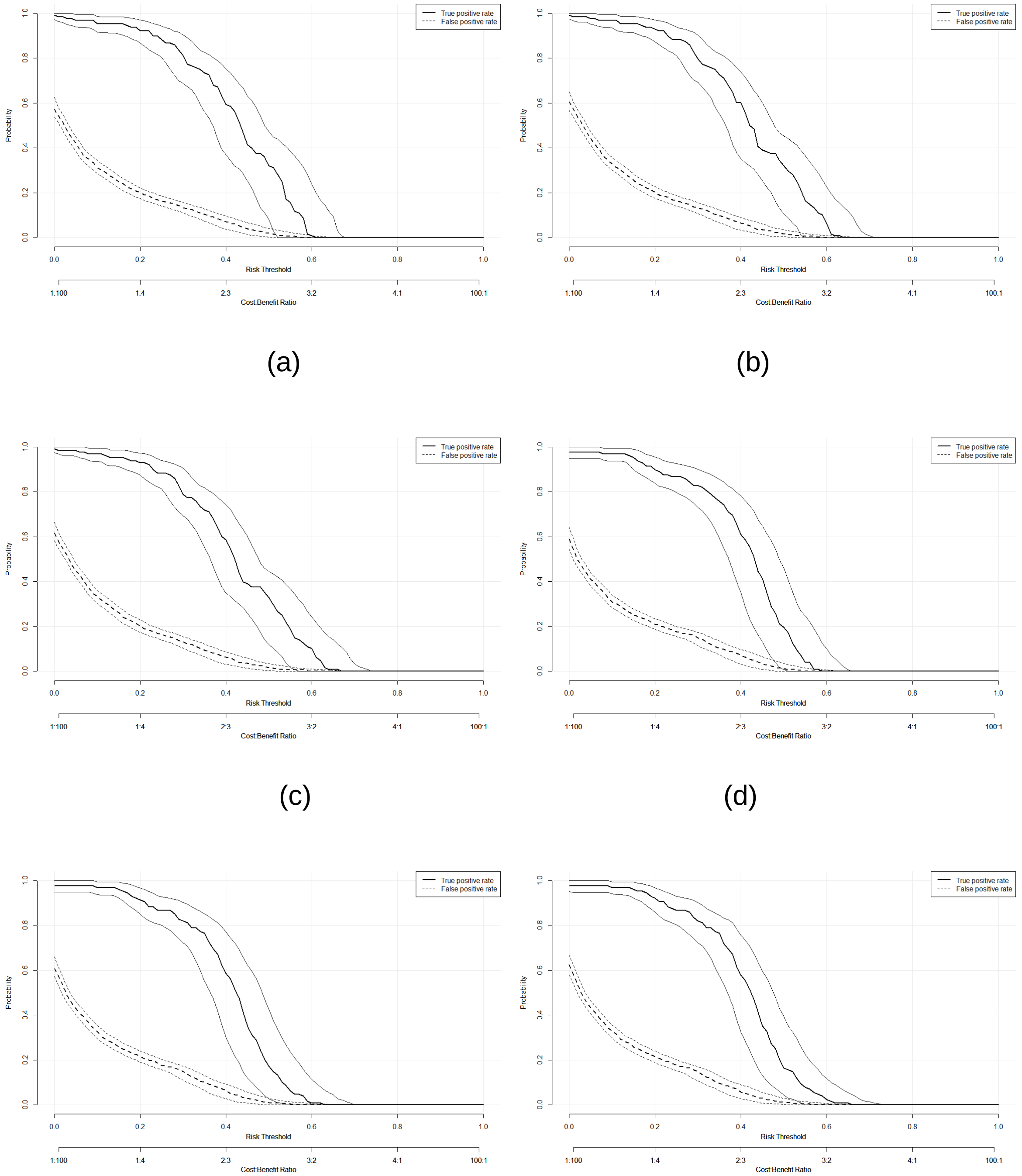
(c)

(d)

(e)

(f)

Figure A.4: ROC component curves for the VAR model with two time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, € two years, (f) and three years.



(a)

(b)

(c)

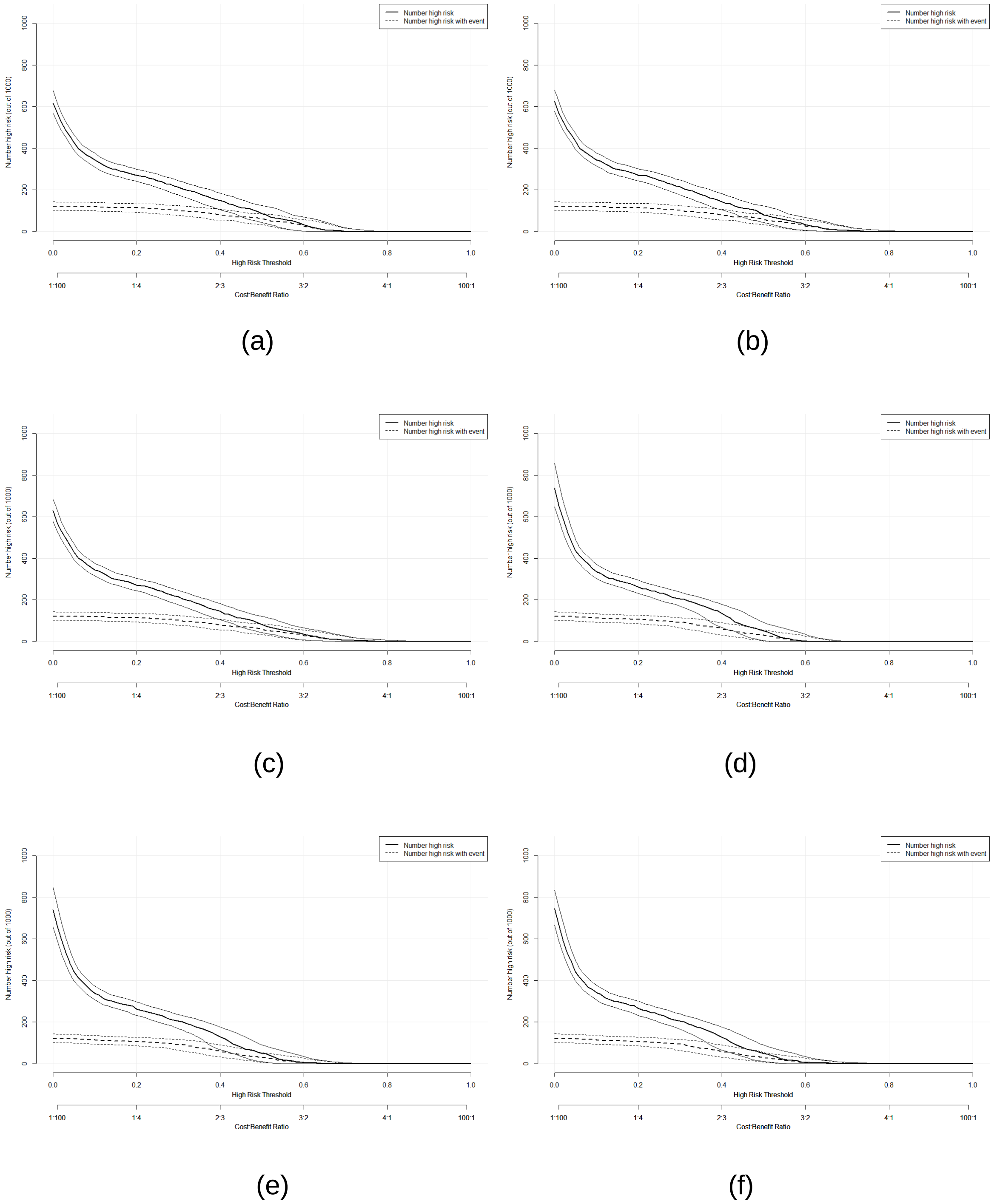
(d)

(e)

(f)

Figure A.5: ROC component curves for the VAR model with three time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

A.2 Clinical impact curves



(a)

(b)

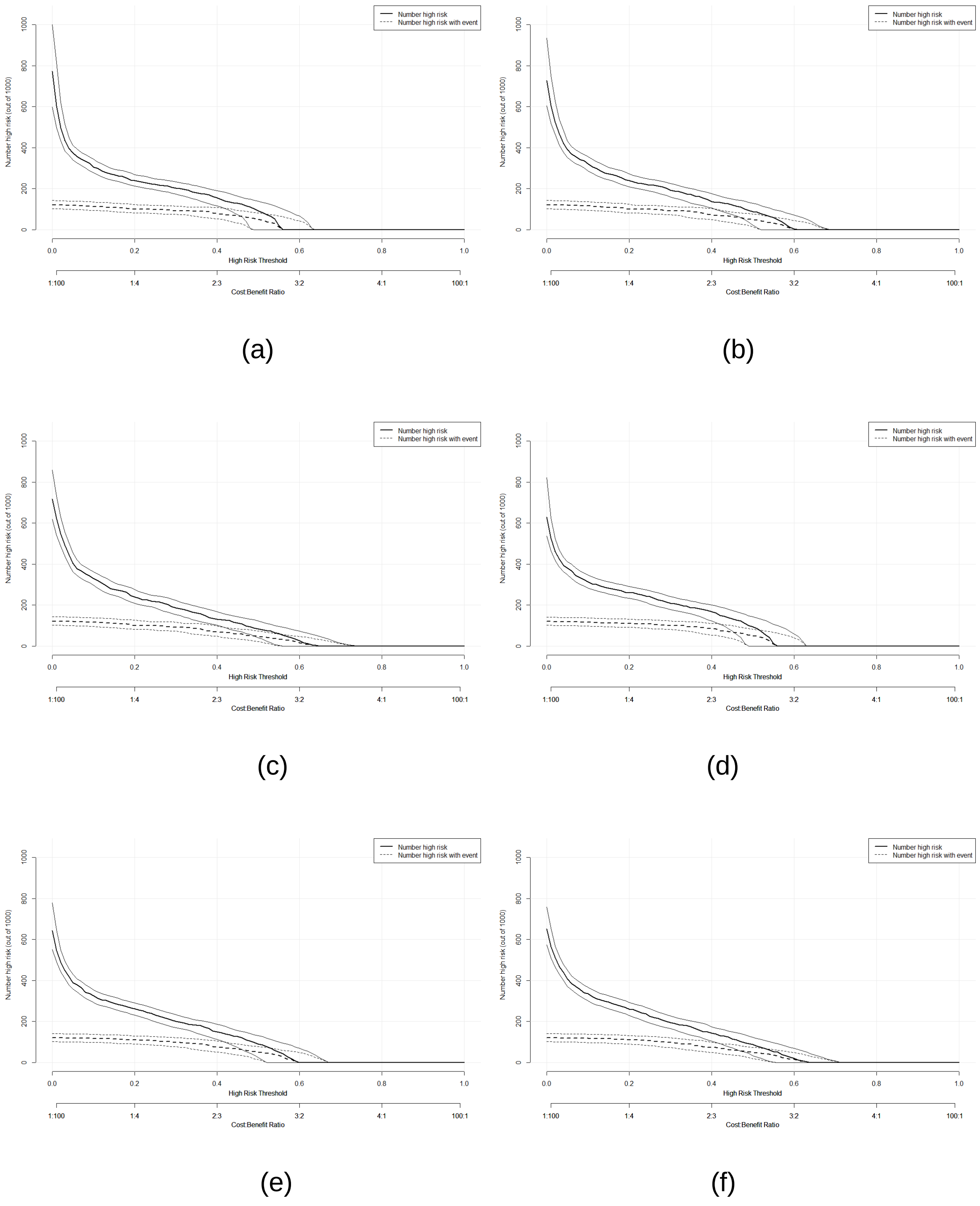
(c)

(d)

(e)

(f)

Figure A.6: Clinical impact curves for the single time point model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

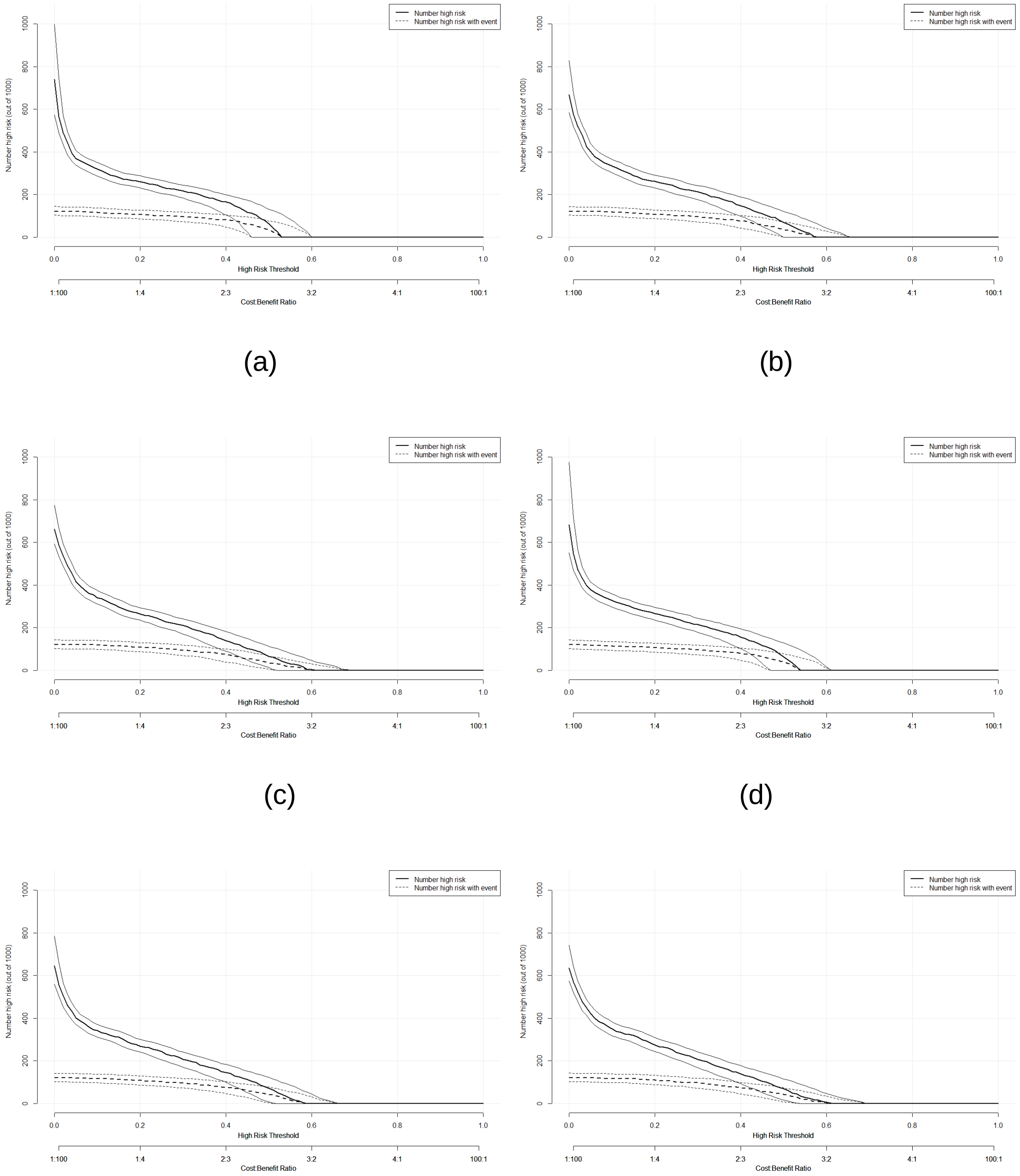
(c)

(d)

(e)

(f)

Figure A.7: Clinical impact curves for the GRU model with two time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

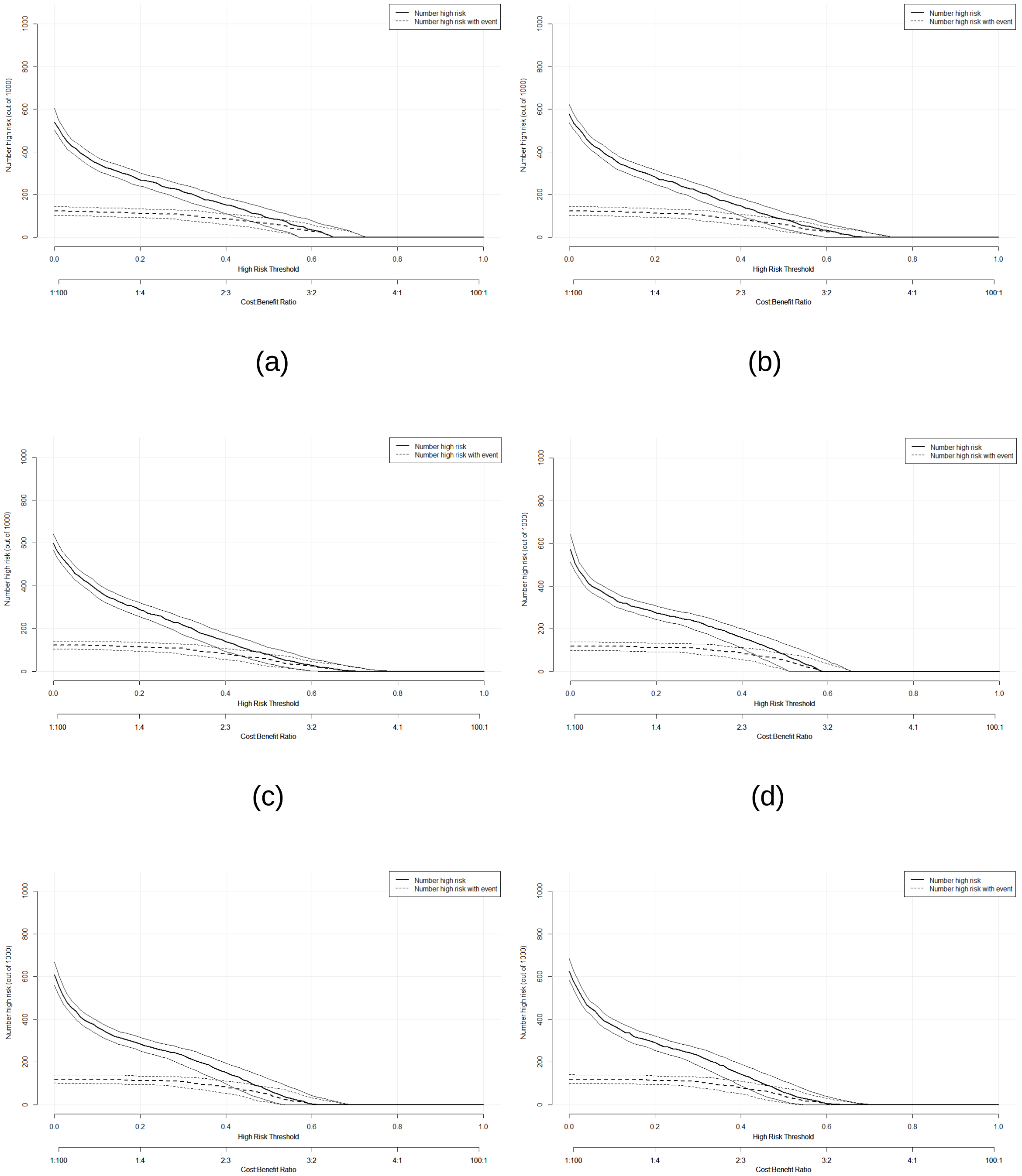
(c)

(d)

(e)

(f)

Figure A.8: Clinical impact curves for the GRU model with three time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

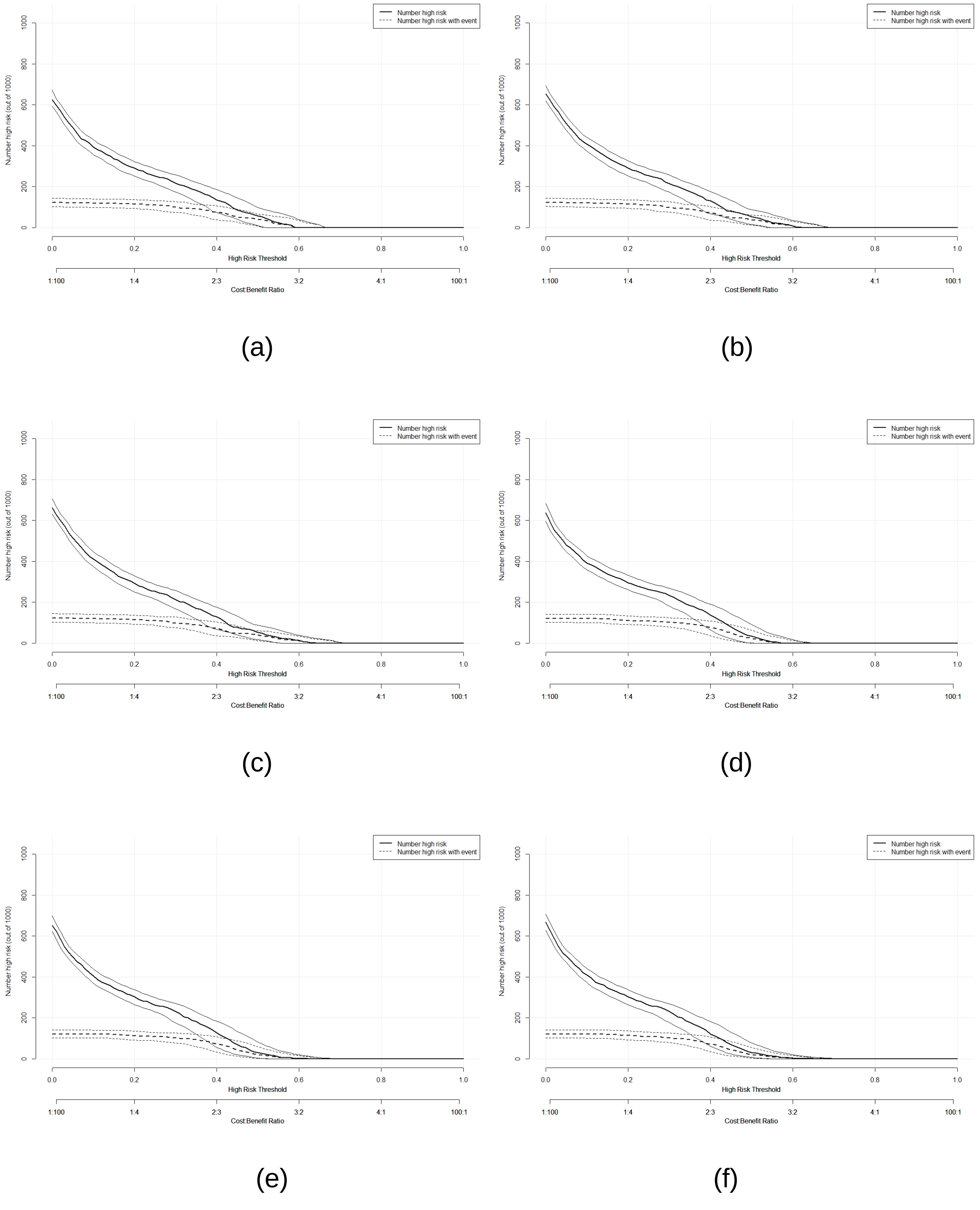
(c)

(d)

(e)

(f)

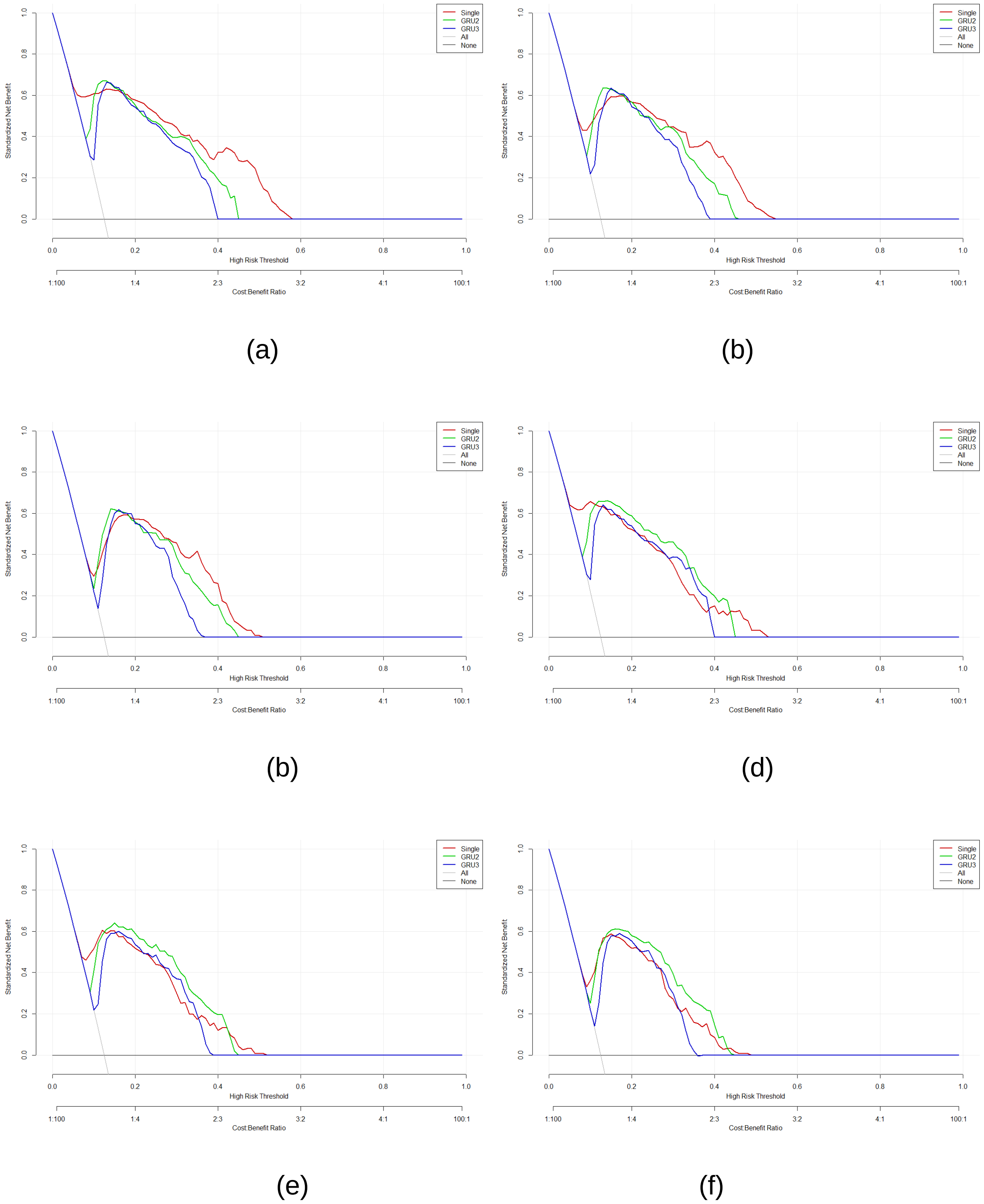
Figure A.9: Clinical impact curves for the VAR model with two time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a) (b) (c) (d) (e) (f)

Figure A.10: Clinical impact curves for the VAR model with three time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

A.3 Decision curves without confidence bands



(a) (b)

(b) (d)

(e) (f)

Figure A.11: Decision curves for the single and GRU models on the testing dataset at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

Appendix B: Derivations of the Survival Functions in Chapter 6

In this appendix, I briefly show the derivations of the survival functions from the baseline hazard functions.

B.1 Exponential

$$h_0(t) = \lambda$$

$$\Rightarrow \mathrm{h(t)} = \lambda e^{\beta x}$$

$$\Rightarrow H(t) = \lambda t e^{\beta x}$$

$$\Rightarrow S(t) = exp\{-\lambda t e^{\beta x}\}.$$

B.2 Weibull

$$h_0(t) = \lambda\gamma t^{\gamma-1}$$

$$\Rightarrow \mathrm{h(t)} = \lambda\gamma t^{\gamma-1} e^{\beta x}$$

$$\Rightarrow H(t) = \lambda t^{\gamma} e^{\beta x}$$

$$\Rightarrow S(t) = exp\{-\lambda t^{\gamma} e^{\beta x}\}.$$

B.3 Gompertz

$$h_0(t) = \lambda e^{-\gamma t}$$

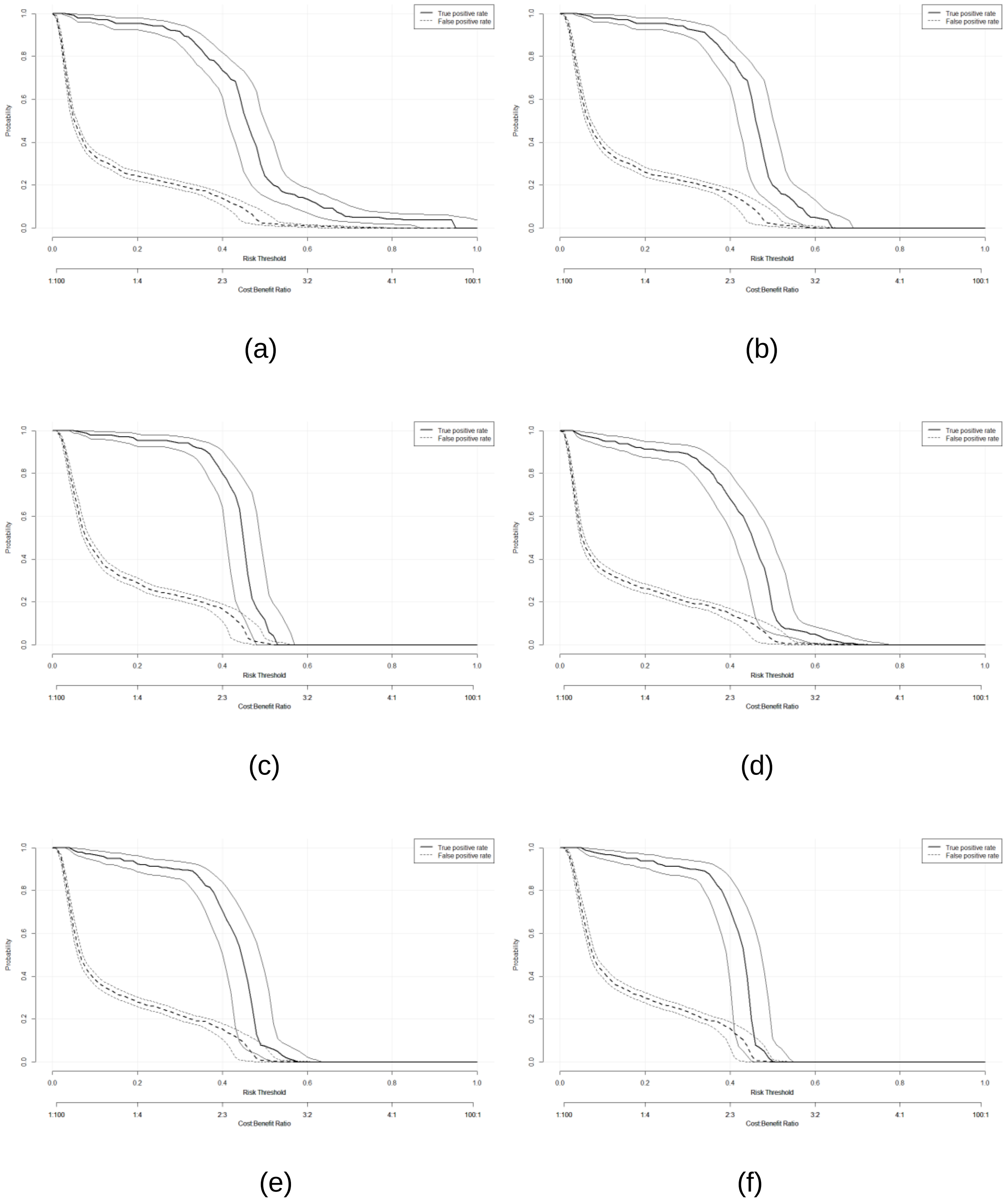
$$\Rightarrow h(t) = \lambda e^{-\gamma t} e^{\beta x}$$

$$\Rightarrow H(t) = \frac{\lambda}{\gamma}[e^{\gamma t} - 1]e^{-\gamma t} e^{\beta x}$$

$$\Rightarrow S(t) = exp\left\{\frac{\lambda}{\gamma}[e^{\gamma t} - 1]e^{-\gamma t} e^{\beta x}\right\}.$$

Appendix C: Additional Figures from Chapter 6

C.1 ROC component curves

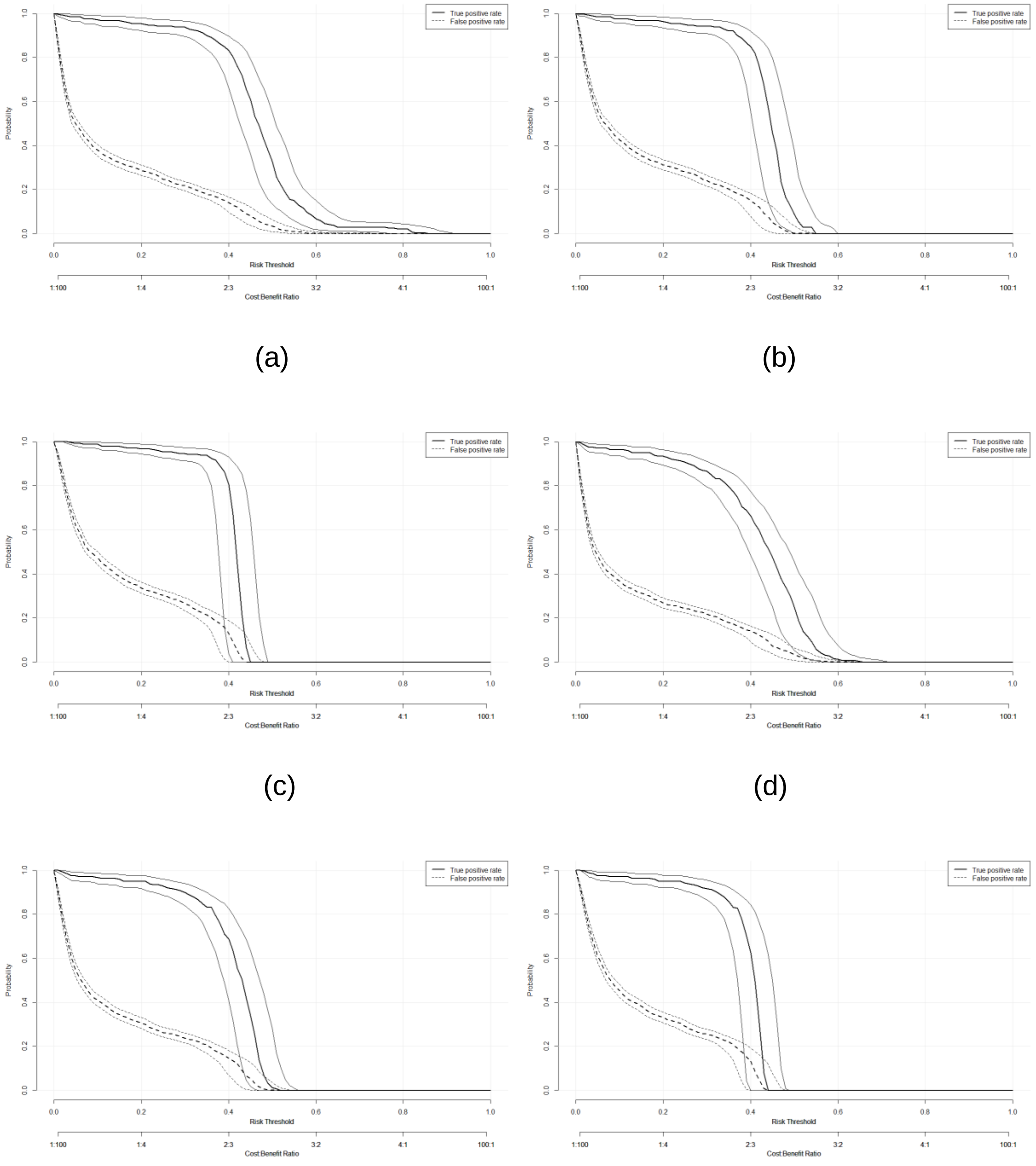


(a) (b)

(c) (d)

(e) (f)

Figure C.1: ROC component curves for the Exponential model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

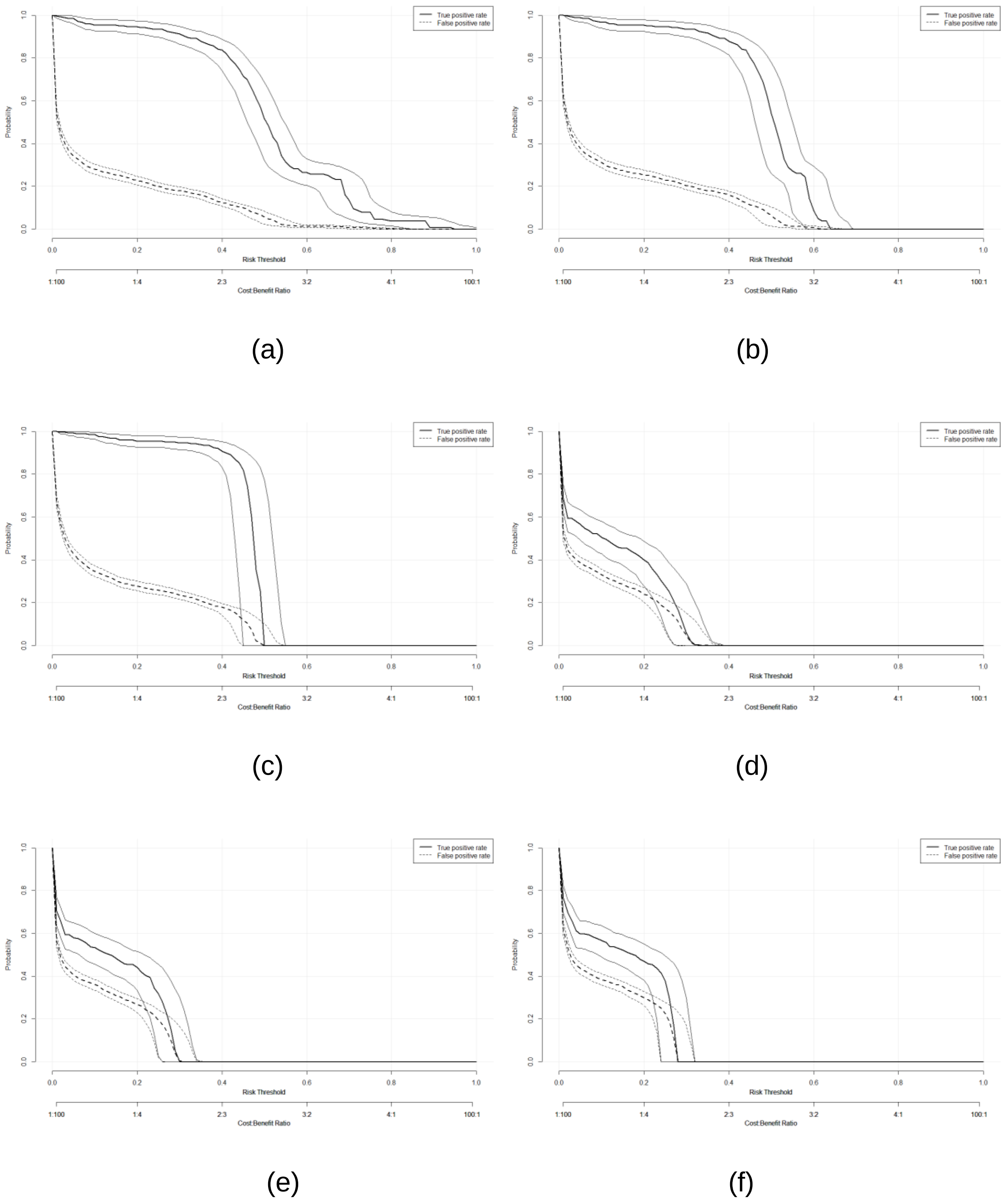
(c)

(d)

(e)

(f)

Figure C.2: ROC component curves for the Weibull model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

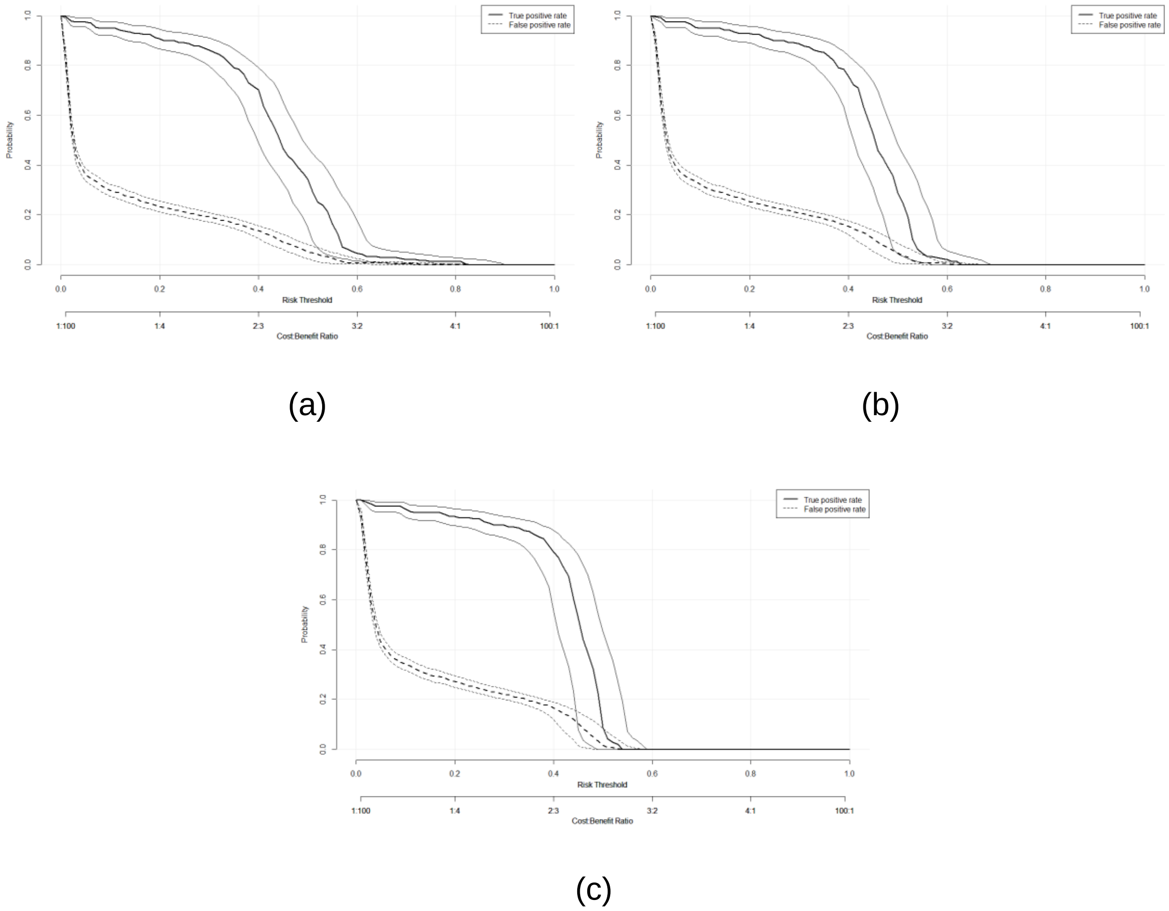
(c)

(d)

(e)

(f)

Figure C.3: ROC component curves for the Gompertz model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

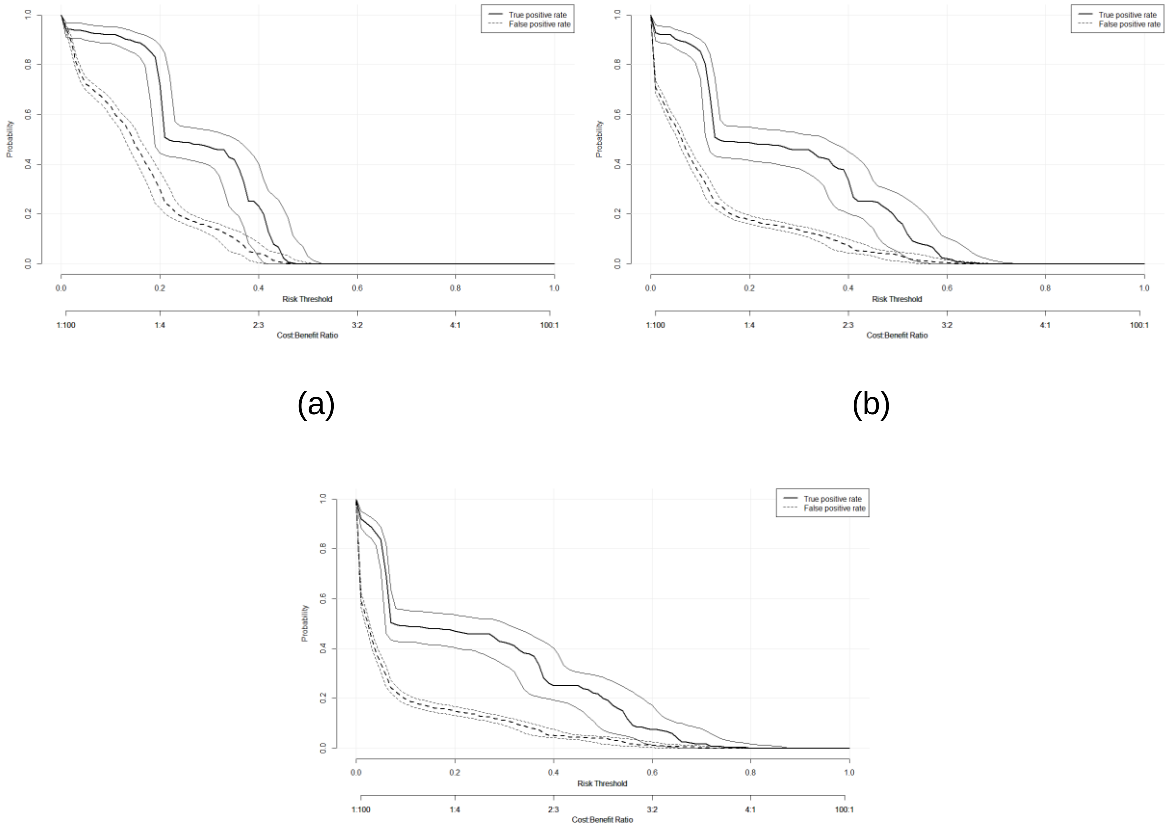


(a)

(b)

(c)

Figure C.4: ROC component curves for the Exponential model with one missing time point on the testing dataset for predictions at (a) one year, (b) two years, and (c) three years.

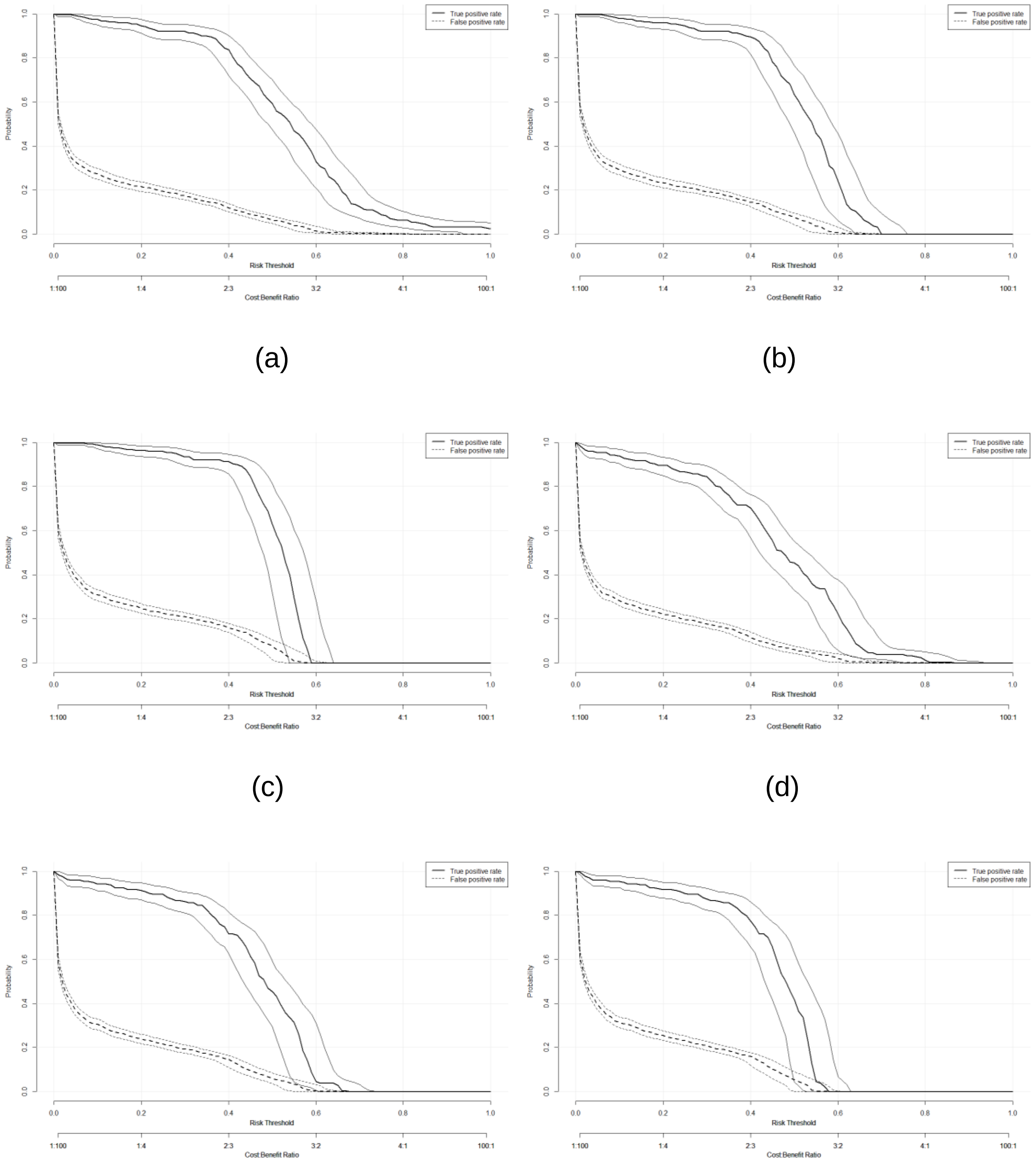


(a)

(b)

(c)

Figure C.5: ROC component curves for the Exponential model with two missing time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years.



(a)

(b)

(c)

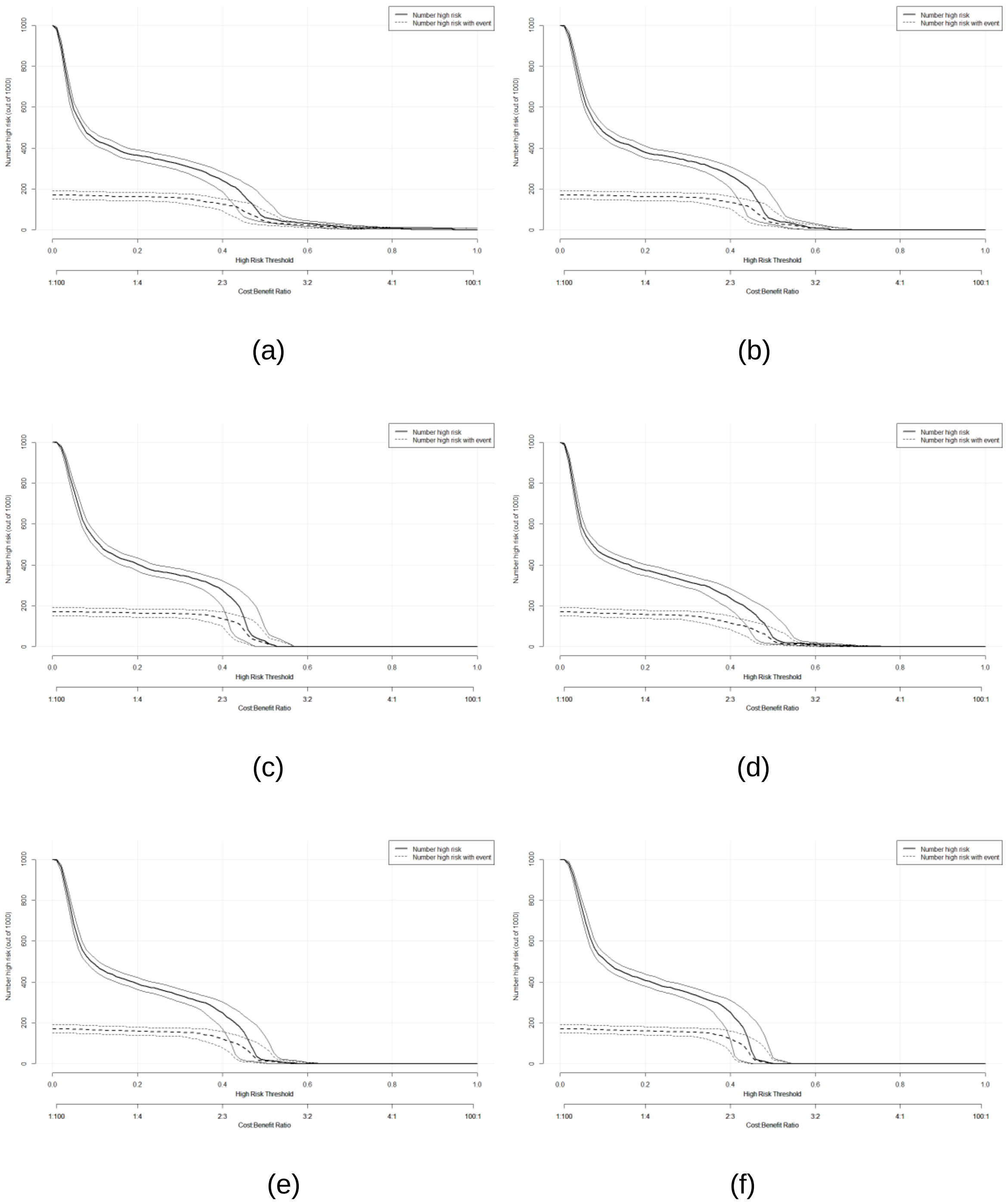
(d)

(e)

(f)

Figure C.6: ROC component curves for the Exponential model with covariates on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

C.2 Clinical impact curves

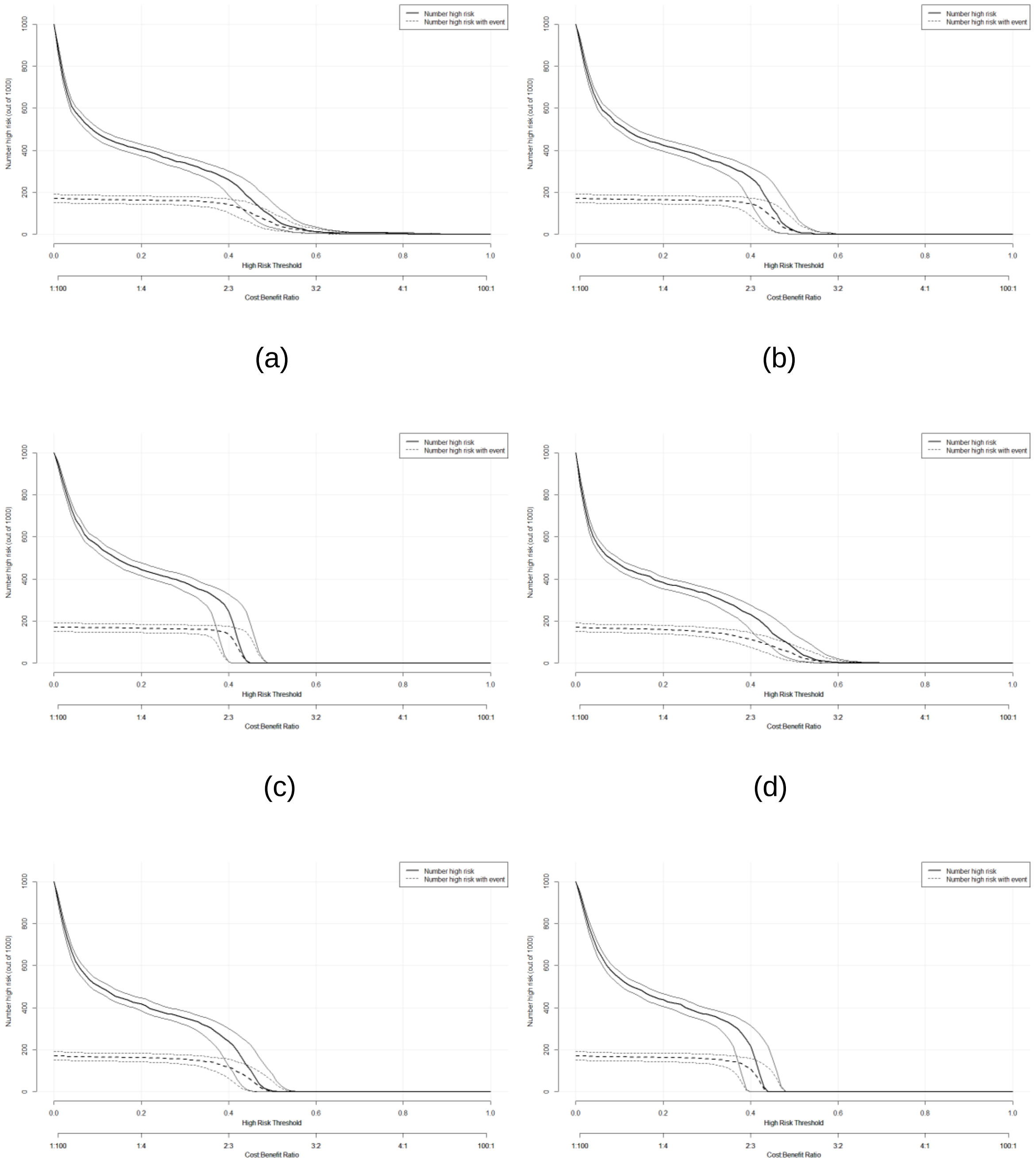


(a) (b)

(c) (d)

(e) (f)

Figure C.7: Clinical impact curves for the Exponential model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

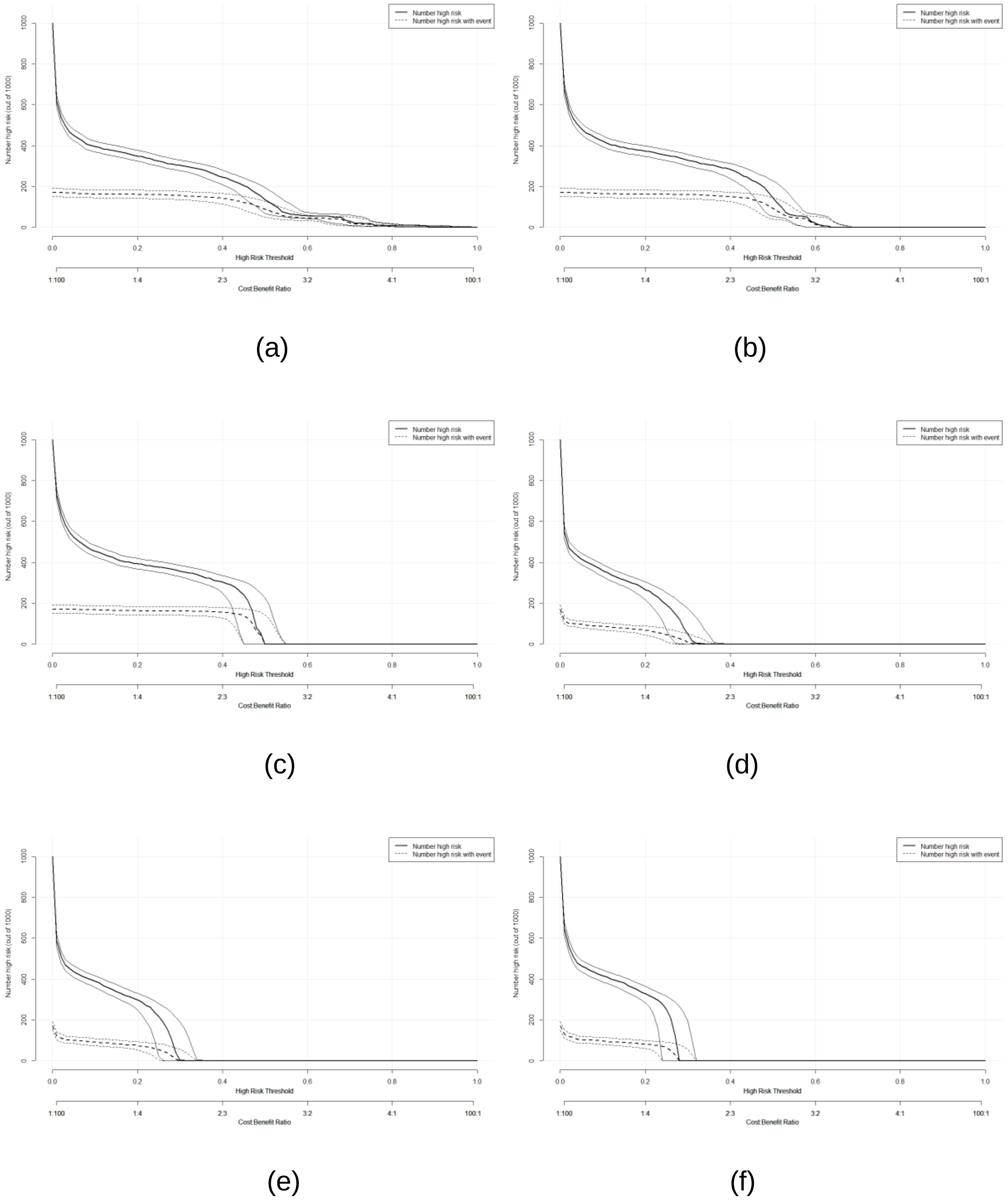
(c)

(d)

(e)

(f)

Figure C.8: Clinical impact curves for the Weibull model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.



(a)

(b)

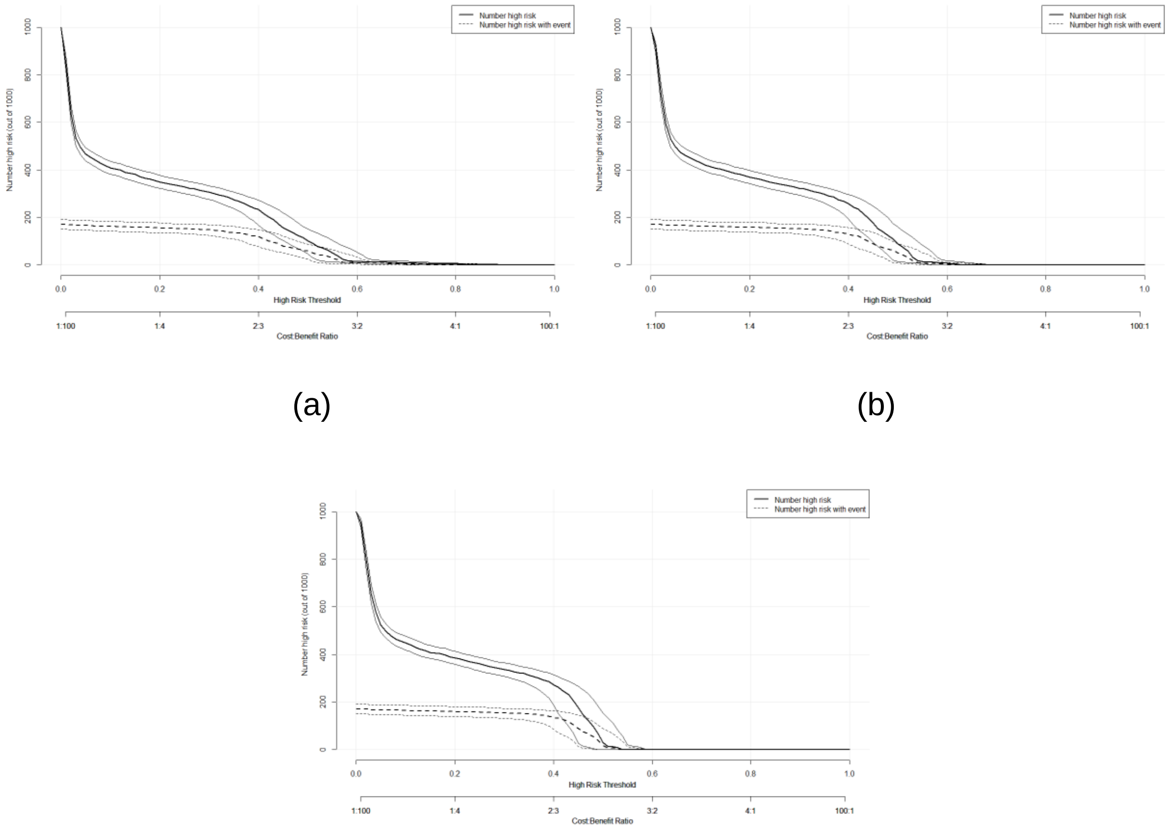
(c)

(d)

(e)

(f)

Figure C.9: Clinical impact curves for the Gompertz model on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

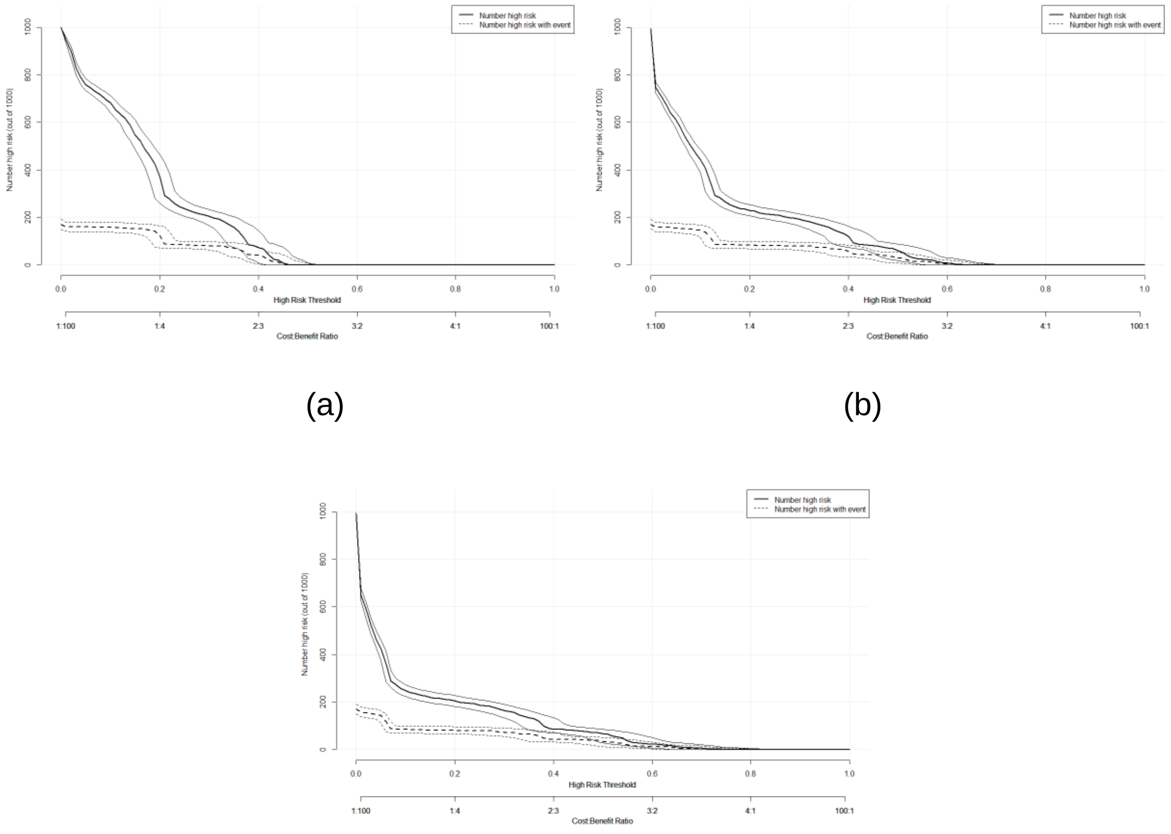


(a)

(b)

(c)

Figure C.10: Clinical impact curves for the Exponential model with one missing time point on the testing dataset for predictions at (a) one year, (b) two years, and (c) three years.

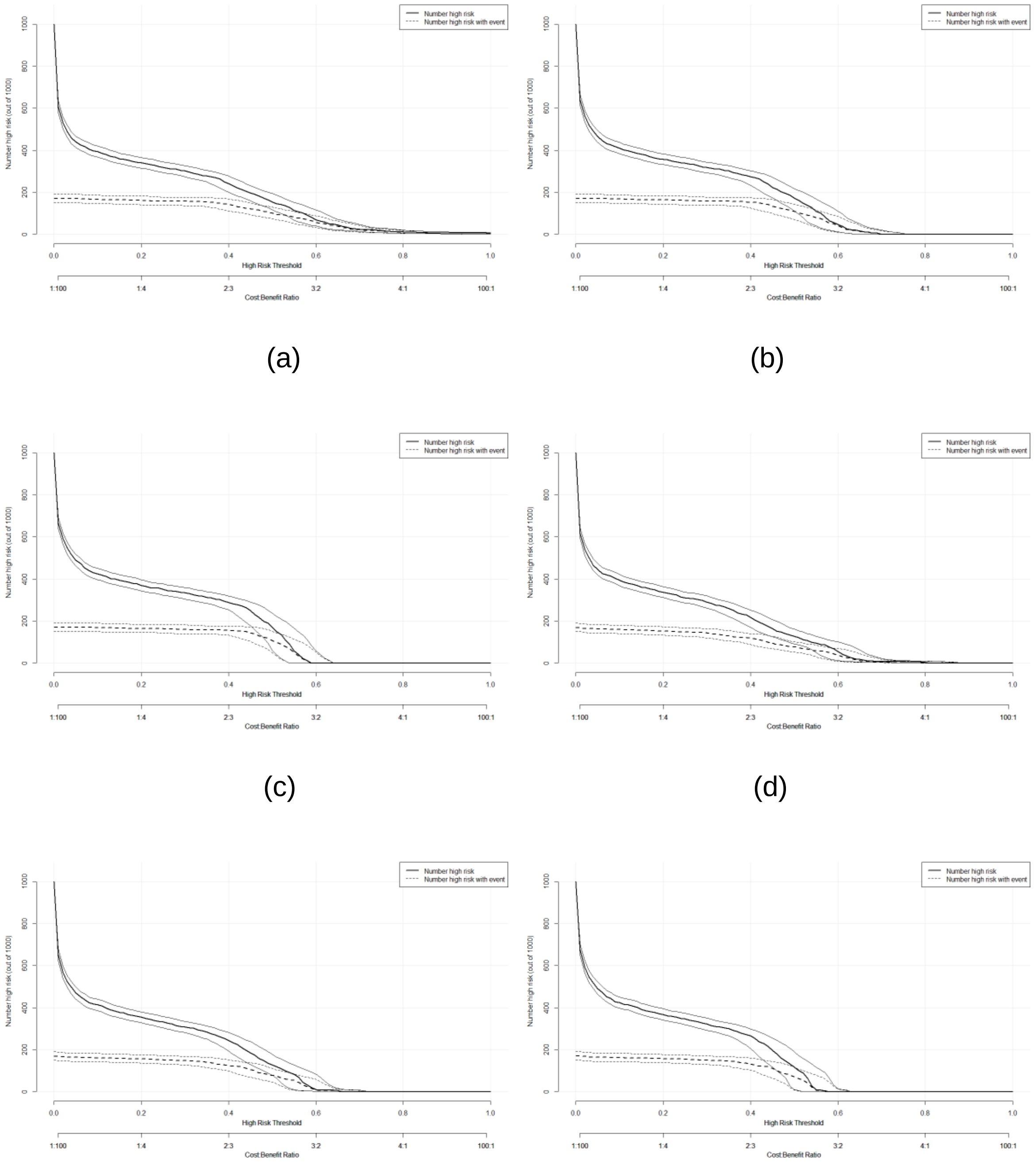


(a)

(b)

(c)

Figure C.11: Clinical impact curves for the Exponential model with two missing time points on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years.



(a)

(b)

(c)

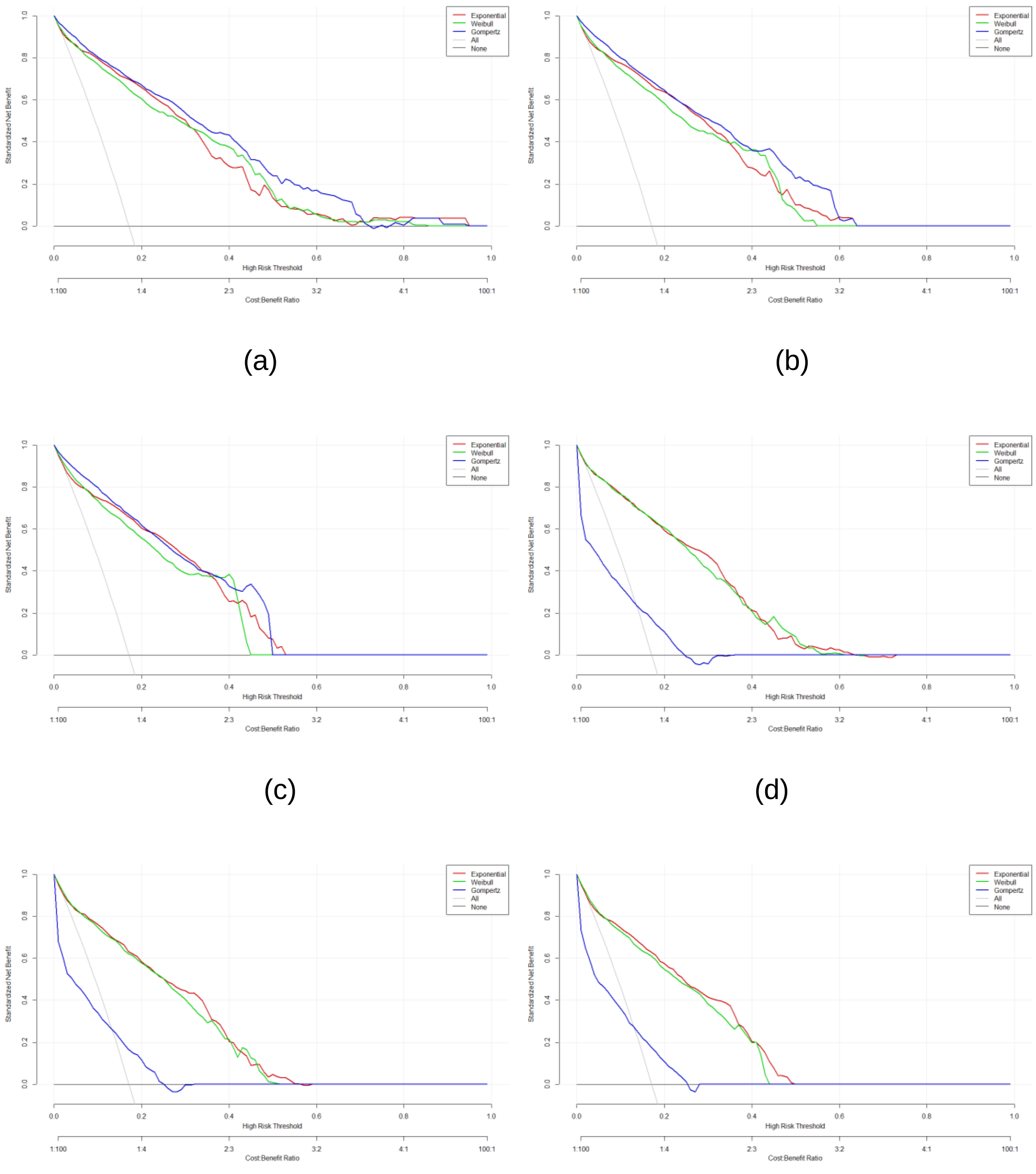
(d)

(e)

(f)

Figure C.12: Clinical impact curves for the Exponential model with covariates on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

C.3 Decision curves without confidence bands



(a)

(b)

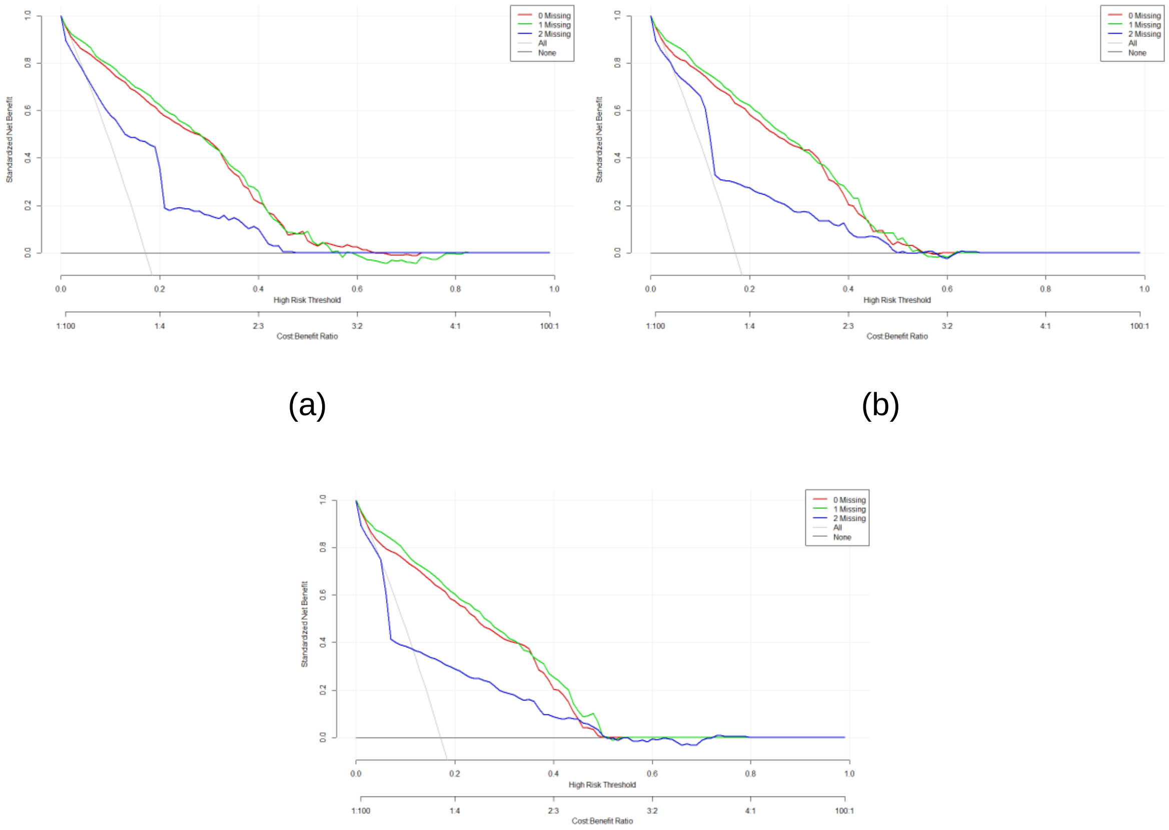
(c)

(d)

(e)

(f)

Figure C.13: Decision curves without confidence bands for the Exponential, Weibull, and Gompertz models on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

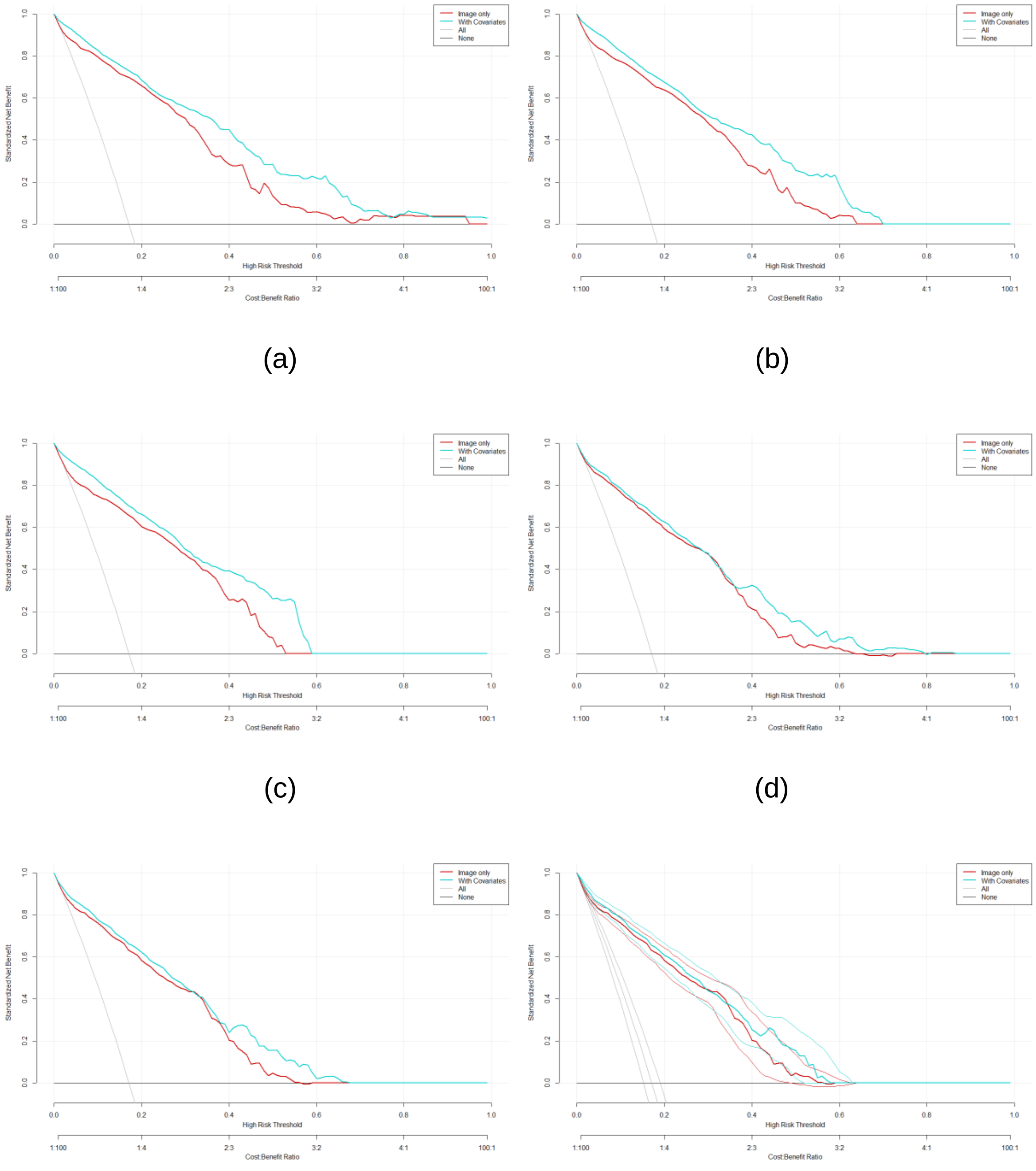


(a)

(b)

(c)

Figure C.14: Decision curves without confidence bands for the Exponential model with 0, 1, and 2 missing time points on the testing dataset for predictions at (a) one year, (b) two years, and (c) three years.



(a)

(b)

(c)

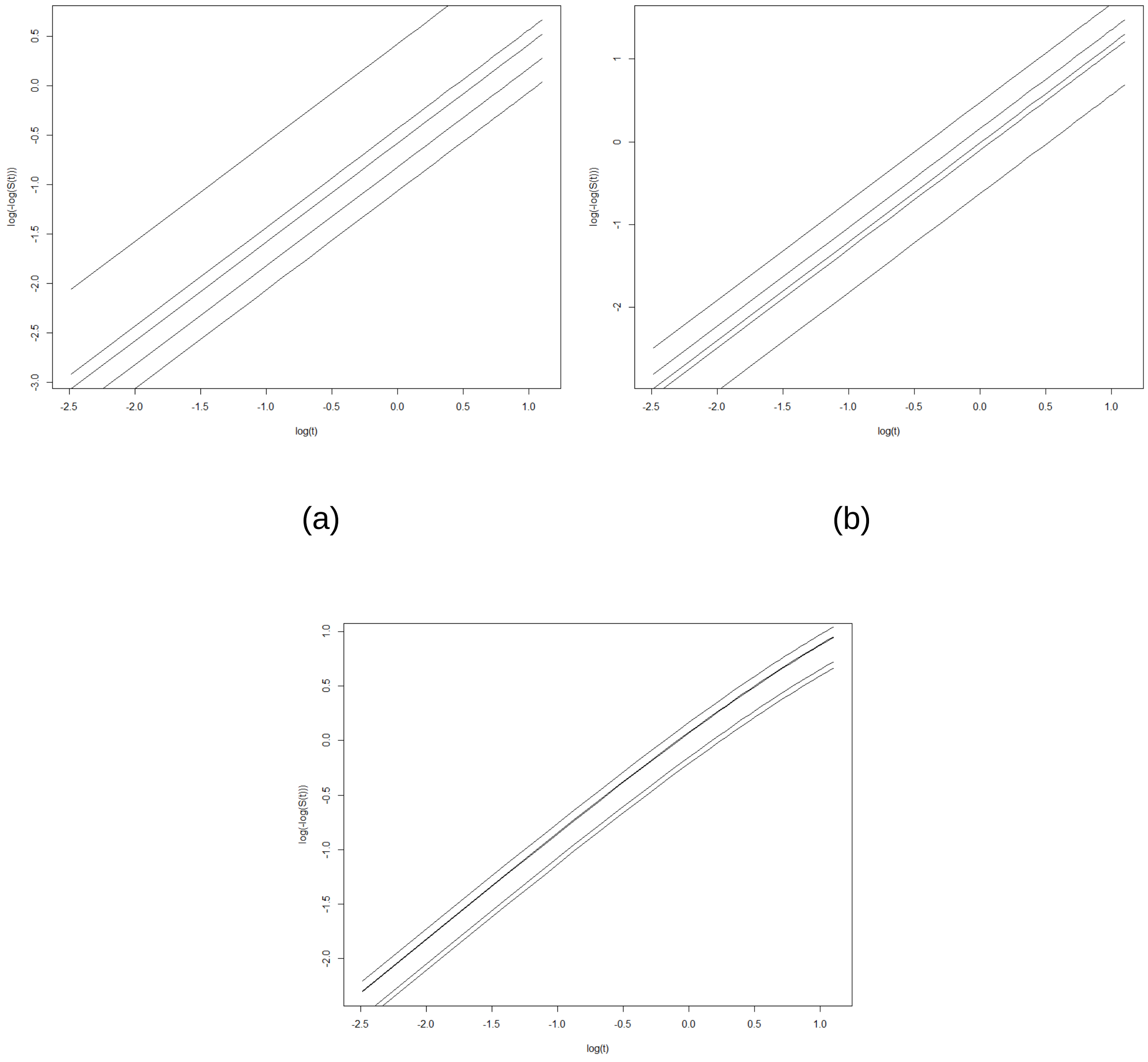
(d)

(e)

(f)

Figure C.15: Decision curves without confidence bands for the Exponential model with and without covariates on the validation dataset for predictions at (a) one year, (b) two years, and (c) three years and on the testing dataset for predictions at (d) one year, (e) two years, (f) and three years.

C.4 Log-negative-log plots for the proportional hazards assumption



(a)

(b)

(c)

Figure C.16: Log-negative-log plots used to check the proportional hazards assumption. The parallel lines indicate that the assumption holds.