Predicting Hypotensive Episodes in the Traumatic Brain Injury Domain

Rob Donald

Submitted in the fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Mathematics and Statistics
College of Science and Engineering
University of Glasgow

May 2014

Abstract

The domain with which this research is concerned is traumatic brain injury and models which attempt to predict hypotensive (low blood pressure) events occurring in a hospital intensive care unit environment. The models process anonymised, clinical, minute-by-minute, physiological data from the BrainIT consortium. The research reviews three predictive modelling techniques: classic time series analysis; hidden Markov models; and classifier models, which are the main focus of this thesis.

The data preparation part of this project is extensive and six applications have been developed: an event list generator, used to process a given event definition; a data set generation tool, which produces a series of base data sets that can be used to train machine learning models; a training and test set generation application, which produces randomly drawn training and test data sets; an application used to build and assess a series of logistic regression models; an application to test the statistical models on unseen data, which uses anonymised real clinical data from intensive care unit bedside monitors; and finally, an application that implements a proposed clinical warning protocol, which attempts to assess a model's performance in terms of usefulness to a clinical team. These applications are being made available under a public domain licence to enable further research (see Appendix A for details).

Six logistic regression models and two Bayesian neural network models are examined using the physiological signals heart rate and arterial blood pressure, along with the demographic variables of age and gender. Model performance is assessed using the standard ROC technique to give the AUC metric. An alternative performance metric, the H score, is also investigated. Using unseen clinical data, two of the models are assessed in a manner which mimics the ICU environment. This approach shows that models may perform better than would be suggested by standard assessment metrics. The results of the modelling experiments are compared with a recent similar project in the healthcare domain and show that logistic regression models could form the basis of a practical early warning system for use in a neuro intensive care unit.

Contents

List of Tables

List of Figures

Acknowledgements

Thanks to Professors Stephen Senn and Mark Girolami and Dr Ludger Evers at the University of Glasgow, who took on the task of guiding me through the process towards a PhD. Although both Professors Senn and Girolami have moved away from University of Glasgow, Professor Senn has kept a keen interest in my research and continued to provide support and advice. Dr Ludger Evers took over my supervision part way through my studies and has been especially helpful in bringing my mathematical skills up to the level required for this challenging area of research.

I would like to gratefully acknowledge the financial support provided by the University of Glasgow School of Mathematics and Statistics who paid my tuition fees. I would have been unable to embark on this academic journey without this help. I also gratefully acknowledge the support of Dr Andy Lawton, Boehringer Ingelheim United Kingdom, who provided a contribution towards my tuition fees.

Many thanks to Dr Ian Piper and all my colleagues in the BrainIT consortium who have been a constant source of encouragement and have welcomed me into their community of researchers.

I would like to thank my former colleagues at Aridhia Informatics who have taken a keen interest in the final stages of my research and listened to lots of medical detail. A special thanks to Rodrigo Barnes and Dr Andy Judson for their support.

Thanks are due to Dr John Millar for not thinking I was crazy when I first suggested doing a PhD during a golf game in Strathpeffer (2005?). I haven't played much golf since then.

Finally, as ever, thanks to my girls Jan, Laura and Emma.

Author’s Declaration

I declare that all this research work is my own.

Definition / Abbreviations

- AUC — area under the ROC curve
- BDS — base data set, a collection of *all* for a given event horizon and window size
- cases — rows of data from an intensive care unit bedside monitor
- EH — event horizon, minutes *before* an episode occurs
- Episode — a collection of events separated by a maximum amount time defined by the parameter, NewEpisodeGap (NEG)
- EUSIG — University of Edinburgh Secondary Insult Grades
- feature vector — a data row of physiological measurements
- ICU — intensive care unit
- MLE — maximum likelihood estimates
- NEG — NewEpisodeGap, the minimum time between events
- ROC — receiver operating characteristic, a curve showing the true positive v.s. false positive trade off
- TBI — traumatic brain injury
- WS — window size, amount of data (in minutes) used for a summary statistic

Throughout this thesis, a number of diagrams are used to illustrate the research processes. These processes involve the manipulation of several types of files and Figure 1 provides a key to be used in subsequent flow charts.

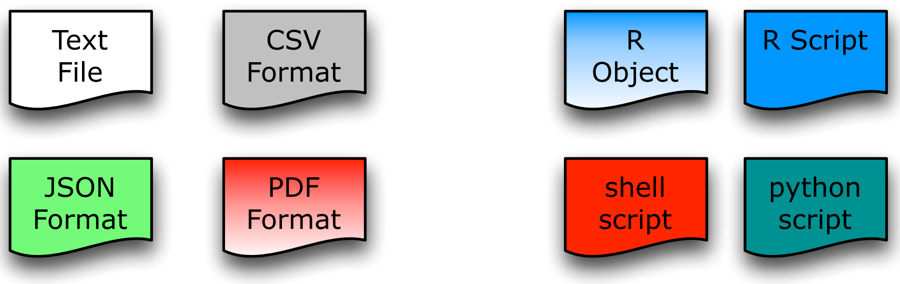


Figure 1: File Types Used

Chapter 1

Introduction

This thesis details extensive research regarding the question of whether or not early warning of hypotensive (dangerously low blood pressure) episodes can be provided for a neurosurgical team treating patients in an intensive care setting by using simple statistical models. The hypothesis is that classifier methods, specifically the well established logistic regression model, favoured by clinicians, can provide useful early warning which matches the current state of the art in the traumatic brain injury (TBI) domain.

In many areas of modern society there is an interest in predicting the probability of an event. If a system could be produced that gave early warning of a particular event then some kind of pre-emptive action might be taken. For example, it may be possible to provide early warning of a dangerous medical condition to the clinical staff in a hospital intensive care unit (ICU). An example from the commercial world, with similar safety implications, would be the prediction of the onset of jet engine malfunction. Early warning of this condition would give the pilots of the aircraft time to take the appropriate action and thereby ensure the safe completion of the journey. Many other examples of a purely financial nature can be constructed, such as the prediction that a project will go over budget. In all these examples the early warning system is providing the end user with a estimate of the *probability* that an event will occur within a specified time period. Ideally the user would prefer values of high probability along with a sufficient early warning before the event. In addition, the system must be accurate i.e. when it gives a high probability warning, the event is highly likely to happen.

These are examples of problems that are studied in the field of “Machine Learning”, a combination of computer science, statistics and mathematics which explores methods of learning from data with a view to producing cross-domain solutions. Although a relatively new discipline (mid 1990s) in its current form, the roots of the research can be traced back to early work in artificial intelligence in the 1950s. Machine learning applications have achieved widespread use and can range, for example, from medical diagnostics to in-car satellite navigation systems. The field continues to address real-world problems from a broad range of subject areas. A review of machine learning research and its relationship with science as a whole is discussed in an insightful paper by Wagstaff (2012).

In the above examples, the approach to the problem can be generalised, resulting in the following tasks:

- **Agree a definition of the event** — this can be a *lot* harder than it sounds. This process should involve domain experts who might hold very differing opinions. Another crucial point to establish is whether or not the system under study is subject to *episodes* of events. It is often the case that a steady state system will briefly enter an event condition and then return to its steady state regime. If a full fault condition is going to happen this first excursion into the event condition may be a precursor to a much longer event. If this is the case then the grouping of several associated events is called an *episode*. A useful early warning system predicts the onset of a new episode.

- **Collect relevant data** — again this can be straightforward or extremely complex. The important point is to have a good understanding of the quality of the data that is available. The task is to collect as many signals as is practical before the event, along with the times that the event actually occurred. It may, however, be the case that the creation of the event timings falls into the next step of the process.

- **Prepare the data for statistical assessment** — the raw measurements taken are rarely in the form that is required for the modelling techniques that will be used. Preparing the data can be a time-consuming step and it is essential that any steps taken to preprocess the raw measurements are fully repeatable by an independent research team. This involves careful record keeping of all the transformations together with full availability and documentation for any software tools used. As mentioned above, this step may also include the production of the event list itself. It may be the case that the events, of which the system is trying to provide some early warning, are poorly understood and not routinely identified or dealt with at the data source. In this situation the system will have to sweep through the collected data generating a list of events that match the agreed definition. Once the event list is available, a further processing step is required to provide two classes of data; one that precedes the event condition (conventionally called the positive case) and the other for when the process is in a steady state condition.

- **Construct suitable models** — several techniques are available to the researcher.

Simpler models are preferred, as this makes the validation process more straightforward and also helps when trying to interpret the model parameters. It is also easier to explain the output of the models to a non-specialist audience (who may well be the client). This topic, acceptance of the model by domain experts, will be shown to have a large influence on deciding which techniques are to be used.

- **Assess the models** — how do the models perform on unseen data? Is one model better overall or is performance improved if models are changed, depending on the current regime of the system? How should a model be assessed? There are several well established methods for checking model performance using objective measures. However, the models will hopefully be used in a practical setting and the operation and usefulness of the results, which can be harder to assess, must also be taken into account.

The project will show how the general steps required to address an event prediction problem can be applied to the specific problem of providing early warning of a dangerous complication which occurs in treating patients in an ICU.

This thesis investigates problems from the medical world, specifically from the traumatic brain injury (TBI) domain. Chapter 2 provides medical background material. In particular the research focuses on phenomena known as secondary insults. A secondary insult is a condition that occurs after a TBI patient has been stabilised by emergency treatment and transferred to a neurointensive ICU. There are several types of secondary insult however the focus of this current research is on the condition known as hypotension (dangerously low blood pressure). Hypotension is a clinical problem that affects a substantial number of TBI patients, (Chesnut et al., 1993).

Research carried out in the mid 1990s, at the Western General Hospital, Department of Clinical Neurosciences, University of Edinburgh, resulted in a published paper, (Jones et al., 1994), which defined the "Edinburgh University Secondary Insult Grades" (EUSIG). The EUSIG definition for hypotension, is systolic arterial blood pressure $\leq$ 90 mmHg or mean arterial blood pressure of $\leq$ 70 mmHg for at least 5 minutes.

The data used to demonstrate the application of these techniques is actual clinical data from TBI patients, gathered by the BrainIT Consortium (2007). This group of senior neurosurgeons and intensivists ran a European Union (EU) funded project (QLG3-CT_2002-01160) over a three year period from 2003 to 2005. This project produced a database of minute-by-minute physiological data along with treatment notes and lab results from approximately 200 patients in a multi-centre study across 22 hospitals in Europe, (Piper et al., 2010). Using the BrainIT database and the EUSIG definition for hypotension, a dataset of approximately 3000 events can be identified from 136 patients.

A definition of the problem of event prediction is provided in section 2.2 and a discussion is provided on the situation where closely spaced events form a single episode. The research focus is on predicting the start of an episode however, as most occurrences of hypotension will take the form of a single event, for readability, the two terms are often used interchangeably. A section on three possible approaches to the prediction problem is presented and this section discusses using time series analysis, hidden Markov models and classifier techniques. As well as technical aspects of how to approach the project, context is provided by considering the important problem of how the models could be used in a clinical setting which crucially requires that clinicians have some understanding of the approach being used.

Data preparation for the chosen classifier based approach is considerable and a full explanation is detailed in chapter 4. This process moves through three stages: identification of events and episodes; production of a pool of data by processing the ICU monitoring data from *all* patients; and finally construction of two randomly drawn data sets, a training set and a test set.

Using the training and test sets, this thesis will show the development of a logistic regression based model which attempts to estimate the probability of a EUSIG defined hypotensive event occurring with at least 10 minutes warning.

Model assessment is an essential part of the model building cycle and approaches to this assessment are discussed in the latter part of chapter 5 and again from a clinical perspective in chapter 7.

Six software applications have been developed to support this research:

Event Analysis Application (EAA) — an application for the identification of episodes and events given a set of general defining parameters; **Base Set Generator** (BSG) — an application which uses a list of events, and the data associated with these events, to produce base data sets containing *all* possible feature vectors corresponding to a base data set definition; **Training and Test Set Generator** (TTG) — a program to produce randomly drawn training and test sets suitable for further statistical processing; **Build LR Models** — an application to run the logisitic regression model functions provided by the R statistical framework; **ICU Data Stream** — a software tool specific to the research data source, which drives anonymised real clinical data from the ICU through the developed models; **Clinical Warning Protocol Processor** (CWPP) — a program for assessing the model's predictive capability from a clinical perspective. It is intended to make these applications available to the research community by publishing the source code under an open source agreement.

In summary, the research question for this thesis is whether or not early warning of hypotensive events can be provided for a neurosurgical team. The research hypothesis is that classifier methods, specifically logistic regression models (a simple and explainable technique readily accepted by clinicians) can provide useful early warning which matches the current state of the art in the TBI domain.

An outline of the research process is shown in Figure 1.1.

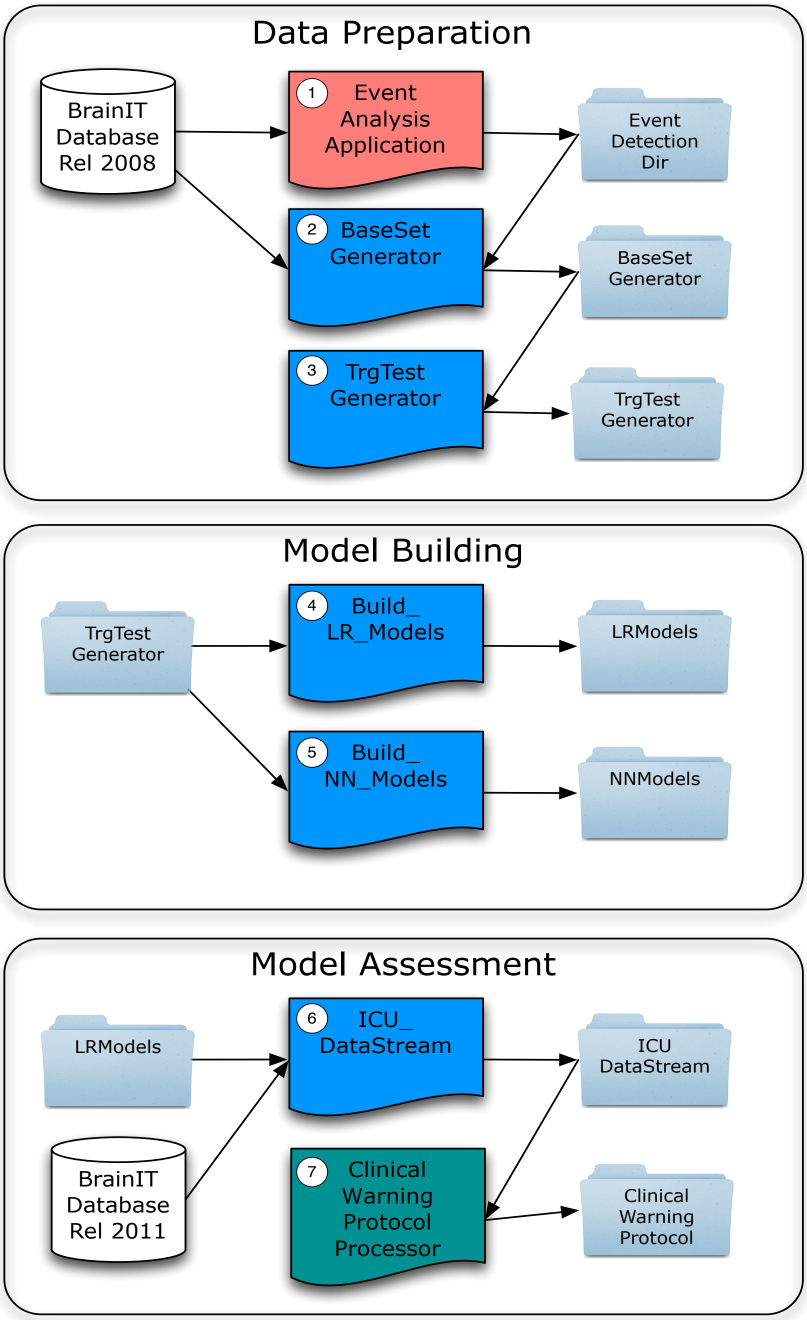


Figure 1.1: Research work flow

Chapter 2

Background — Traumatic Brain Injury

A person does not expect to suffer traumatic brain injury (TBI). This condition is often only one aspect of polytrauma resulting from an accident, and might only be recognised after the initial assessment and stabilisation of a patient, (American College Of Surgeons, 2011). TBI is usually classified as mild or severe and is a major cause of healthcare costs which are estimated at $60 billion (mild and severe) in the US in 2000, (Centers for Disease Control and Prevention, 2011), with approximately 90% of these costs being due to severe TBI, (Centers for Disease Control and Prevention, 2012) and €3 billion (hospitalisation costs only) in Europe in 2005, (Andlin-Sobocki et al., 2005). This thesis only considers severe TBI where the patient requires hospitalisation.

After the initial stabilisation, if TBI is recognised, the patient will be transferred to a neurointensive ICU. The patient will usually be in a coma and mechanically ventilated. The specialist clinical teams in the unit will subsequently be on alert for a class of conditions known as *secondary insults*. These are conditions caused by changes in brain physiology and chemistry and are not fully understood. In layman's terms, the problems arise from the fact that the body's normal methods of dealing with injury i.e. swelling and bruising, are confined within the fixed volume of the skull. In fact, these mechanisms do occur and it is this internal swelling that affects blood flow and pressures within the brain.

It is one of these secondary insults, hypotension, that is the focus of this thesis.

2.1 TBI pathophysiology, primary/secondary insults

The pathophysiology of a medical condition is the attempt to classify and understand the underlying changes, caused by that condition, to the normal processes of the body. TBI pathophysiology focuses on two areas, the primary injury — describing the external forces involved in the damage to the brain, and the secondary injuries — which examine the processes which occur over an extended time period after the initial trauma.

The primary injury falls into one of two main categories, physical contact with an object, or acceleration followed by deceleration of the head. Physical contact may result in visible injuries such as bruising and hematomas. Contact injury may also result in contusions, swollen brain tissue, and skull fractures which are typically detected by CT scans. Acceleration/deceleration injury involves considerable internal forces and can result in intracranial hematoma and damage to blood vessels and nerves within the brain. TBI will usually result in further complications, the secondary injuries.

Although references to secondary complications can be found in the literature from the 1950s, (Maciver et al., 1958), the major breakthrough in this domain is considered to have started with the work of neurosurgeons in Glasgow in the mid 1970s, (Rose et al., 1977).

Secondary insults often do not present with clinical signs. These secondary complications are usually the result of the initial injury, but can also be associated with the treatment given during the patient's stay in the ICU, so called *iatrogenic* events.

With the advent of more affordable computerised data recording techniques in the 1990s, a group led by the leading neurosurgeon Douglas Miller, (Royal Society Edinburgh, 1995), at Edinburgh's Western General Hospital published a paper, (Jones et al., 1994), which classified the main types of secondary insult and showed that they were more prevalent than previously thought. More importantly, the research showed that secondary insults could be related to patient outcome. The study identified 14 types of secondary insults and classified them into either systemic or intracranical problems. These secondary insult types are presented in Table 2.1. The systemic secondary insult hypotension — dangerously low blood pressure — was shown to be the most significant when compared to the standardised Glasgow Outcome Scale (see Table 2.2). The standardised Glasgow Outcome Scale is a measure of the patient's long term recovery (Jennett and Bond, 1975; Teasdale et al., 1998; tbi impact.org, 2013) and is best obtained by the use of a structured interview (Wilson et al., 1997, 2007).

Systemic	Intracranial
Hypoxemia	Hematoma
Arterial hypotension	Raised intracranial pressure
Anemia	Seizures
Hypocarbia	Infection
Hypercarbia	Vasospasm
Pyrexia	
Hyponatremia	
Hypoglycemia	
Hyperglycemia	

Table 2.1: Secondary insults that can contribute to hypoxic and/or ischemic brain damage. Adapted from Table 1, p5 (Jones et al., 1994)

Survival vs. death		Good vs. poor outcome	
Variable	Signif.	Variable	Signif.
Duration of hypotension	0.0064	Duration of hypotension	0.0118
Duration of pyrexia	0.0137	Pupil response O/A	0.0226
Duration of hypoxemia	0.0244	Duration of pyrexia	0.0772
Age	0.0652	Age	0.0964
Duration of raised ICP	0.1162	Duration of hypoxemia	0.1217
Duration of hypertension	0.3689	Duration of raised ICP	0.1941
ISS	0.3855	Duration of bradycardia	0.3737
GCSs postresuscitation	0.3858	ISS	0.5701
Duration of tachycardia	0.4001	Duration of hypertension	0.6133
Pupil response O/A	0.4857	Duration of tachycardia	0.6327
Duration of bradycardia	0.8733	GCSs postresuscitation	0.9051
Goodness of fit	90.00 %	Goodness of fit	83.32 %

Table 2.2: Significance of logistic regression components for GOS outcomes at 12 months. Adapted from Table 8, p10 (Jones et al., 1994). ICP, intracranial pressure; ISS, Injury Severity Score; GCS, Glasgow Coma Score; O/A, on admission.

The group also defined the Edinburgh University Secondary Insult Grading scheme, which detailed thresholds for the various physiological signals as well as time durations over which these signals must be maintained. These levels were based on previous studies and the clinical judgement of the group.

Secondary Insult	Grade 1	Grade 2	Grade 3
Raised ICP (mmHg)	≥ 20	≥ 30	≥ 40
Hypotension (mmHg) systolic	≤ 90	≤ 70	≤ 50
or mean	≤ 70	≤ 55	≤ 40
Hypertension (mmHg) systolic	≥ 160	≥ 190	≥ 220
or mean	≥ 110	≥ 130	≥ 150
CPP (mmHg)	≤ 60	≤ 50	≤ 40
Hypoxemia SaO_2 (%)	≤ 90	≤ 85	≤ 80
or PaO_2 (kPa)	≤ 8.0	≤ 7.0	≤ 6.0
Cerebral oligemia $SjvO_2$ (%)	≤ 54	≤ 49	≤ 45
Cerebral hyperemia $SjvO_2$ (%)	≥ 75	≥ 85	≥ 95
Hypercarbia (kPa)	≥ 6.0	≥ 8.0	≥ 10.0
Hypocarbia (kPa)	≤ 3.0	≤ 2.5	≤ 2.0
Pyrexia (°C)	≥ 38	≥ 39	≥ 40
Tachycardia (bpm)	≥ 120	≥ 135	≥ 150
Bradycardia (bpm)	≤ 50	≤ 40	≤ 30
Global cerebral hypoxia			
Ca-jvO_2(mlO_2/100 ml blood)	≥ 9	(one grade only)	
Global cerebral hyperemia			
Ca-jvO_2(mlO_2/100 ml blood)	≤ 4	(one grade only)	

Table 2.3: Edinburgh University secondary insult grades (EUSIG), in adults > 14 years. Adapted from Table 2, p7 (Jones et al., 1994)

The timing definitions for the secondary insults are defined on page 6 of the Jones paper as:

> "To be considered a secondary insult in this study, abnormal values had to persist for ≥ 5 min. The insult was deemed to have ended only when values returned to normal for five consecutive minutes"

Further background material on TBI pathophysiology can be found on the Medscape website; the article by Pangilinan (2012), is a good starting point. An accessible review of the topic of secondary insults in TBI can be found in the PhD thesis by Elf (2005) of Uppsala University Hospital.

2.2 Hypotensive event definition

There are many definitions of hypotension and the definition varies across the clinical domains, (Berry et al., 2011; Bijker et al., 2009; Physio Net, 2009). In the TBI domain, a well respected example is the Brain Trauma Foundation (2007) “Guidelines for the Management of Severe Traumatic Brain Injury, 3rd Edition” which defines hypotension as arterial systolic pressure < 90 mmHg. However, this document also states, “The importance of mean arterial pressure, as opposed to systolic pressure should also be stressed, ...”. Research undertaken as part of the AvertIT (2008) project, showed that both systolic and mean arterial blood pressure play a role in current clinical management. This thesis therefore uses the EUSIG definitions, which require monitoring of both systolic and mean arterial pressures. It can be shown, when using the EUSIG definition for hypotension, that most hypotensive events are in fact triggered by a breach of the mean arterial blood pressure level of 70 mmHg, (Donald et al., 2012a).

The general definition of an event is presented in Section 2.2.1. A section is then provided on the question of groups of events, which are called episodes, and is discussed fully in Section 2.2.2 .

2.2.1 Event definition

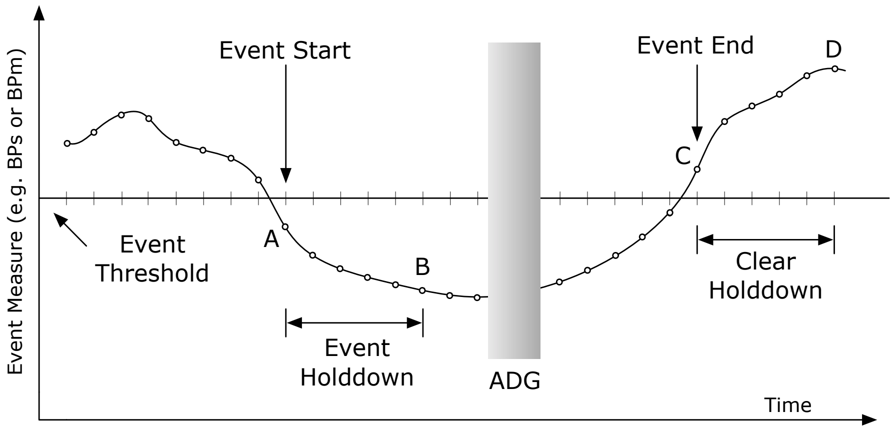


Figure 2.1: General event definition

Consider a trace of a measured signal being scanned, the sequence of measurements that are being looked for are detailed in Table 2.4.

Position	Description
A	Trace falls below the “Event Threshold”.
B	Trace remains below the event threshold for the defined “Event Holddown” period. The event is now “Active” starting from point “A”.
ADG	It is possible to tolerate some small data gaps; Allowable Data Gap (ADG). This is an area for possible future work (see Section 9).
C	Trace rises above the event threshold.
D	Trace remains above the event threshold for the “Clear Holddown” period. At this point the event is declared complete and subsequently the event is defined from point “A” to “C”.

Table 2.4: Event characteristics

2.2.2 Episode definition

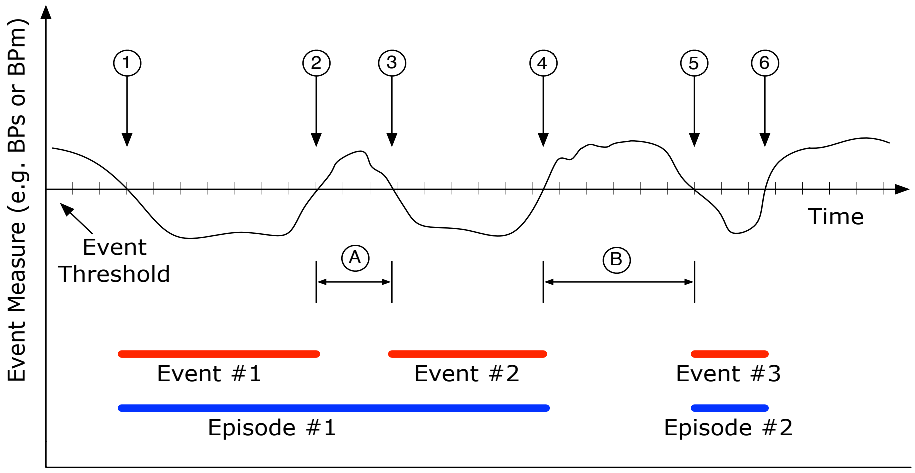


Figure 2.2: General episode definition

In conversations with clinical staff, anecdotal evidence suggests that short hypotensive events often precede a main prolonged event. This gives rise to the concept of an *episode* whereby a group of events, separated by a small gap, are combined to form a single episode. Although every hypotensive event is considered dangerous, it is the prediction of the first event in an episode that would be of most clinical use.

Figure 2.2 shows two episodes made up from three events. Consider the case where the tick markers on the time axis represent 5 minutes and the "new episode gap" (NEG) is defined as ≤ 15 minutes. The initial dip below the event threshold at point 1 starts the first event *and* the first episode. This event lasts for 35 minutes until it clears at point 2. A second event begins 14 minutes later at point 3. The NEG (line A) is < 15 minutes therefore the first episode remains active. The second event lasts for just over 25 minutes before clearing at point 4. The measurement continues above the event threshold for some 25 minutes (line B) and therefore the first episode can be declared complete 15 minutes after point 4. The trace again drops below the threshold at point 5, which starts off event 3, and also constitutes the start of episode 2. The signal then rises above the threshold at point 6 to complete the event. It stays above this level for more than 15 minutes thereby completing the episode.

2.3 BrainIT database

The primary source of data for this research comes from the BrainIT Consortium (2007). Through a series of EU funded projects, the most recent of which was QLG3-CT_2002-01160, this group has produced a database of clinical measurements on patients suffering from TBI. Crucially, the time series data in the form of minute by minute readings from the ICU monitors is available.

The database is not strictly public domain but is available for academic research by contacting the coordinator, Dr Ian Piper, by email at ian.piper@brainit.org. The ethos of the group is one of information sharing and therefore any results derived from the database are expected to be published via the academic community.

The database contains three classes of information: demographic data — giving one off information about the patient such as age, gender, time and type of trauma, along with initial clinical readings collected at admission to both the nearest emergency hospital and the specialist neurosurgery unit; physiological data — minute-by-minute readings of clinical vital signs for example, intracranial pressures, blood pressures, heart rate and oxygen saturation; and episodic data — notes regarding any treatments given to the patient, laboratory results from the regular blood samples taken, and regular Glasgow Coma Scores (GCS) which are used to assess a patient's neurological condition. The GCS is a scale ranging from 3 to 15 which is used by clinical staff to monitor the state of the patient's conciousness (Teasdale and Jennett, 1974). It consists of three tests which check a patient's eye, verbal and motor responses. Episodic data also contains notes on any nursing care which has occurred during the patient's stay in the ICU.

For this research program, two releases of the BrainIT database were used. Using the 2008 release, a cohort was constructed by examining the database for patients who were over 15 years of age and had more than 24 hours of data available for the signals described in Section 2.4. This reduced the number of suitable patients from 199 to 136. The demographic characteristics of this first cohort are described in Table 4.1. Using the 2011 release, a cohort of 30 patients was identified using the same criteria and this was used to test the developed models in a manner which simulates the actual stay in the ICU by having the model process the minute-by-minute data. The demographic characteristics of this validation cohort are described in Table 7.1.

2.4 Data used for research

A modern neurointensive ICU contains a wide range of sophisticated equipment to help the clinical teams look after the patient. Central to this range of equipment is the physiological monitoring system which records minute-by-minute values of the patient's vital signs. A typical system will capture values for heart rate, breathing, blood pressures and body temperatures. This research uses only heart rate and blood pressure signals.

Figure 2.3: Modern neuro intensive care unit

The data source from the BrainIT Consortium (2007), contains the four signals that will be used during this research: heart rate (HRT), systolic arterial blood pressure (BPs), diastolic arterial blood pressure (BPd) and mean arterial blood pressure (BPm). An example of these traces is shown in Figure 2.4. The plots are from BrainIT patient 73704046, a 54 year old male. This series of plots show the signal trace for each variable in the 40 minutes leading up to a hypotensive event. The grey hatched area covers the 10 minutes just before the event which is not used in the training of the models.

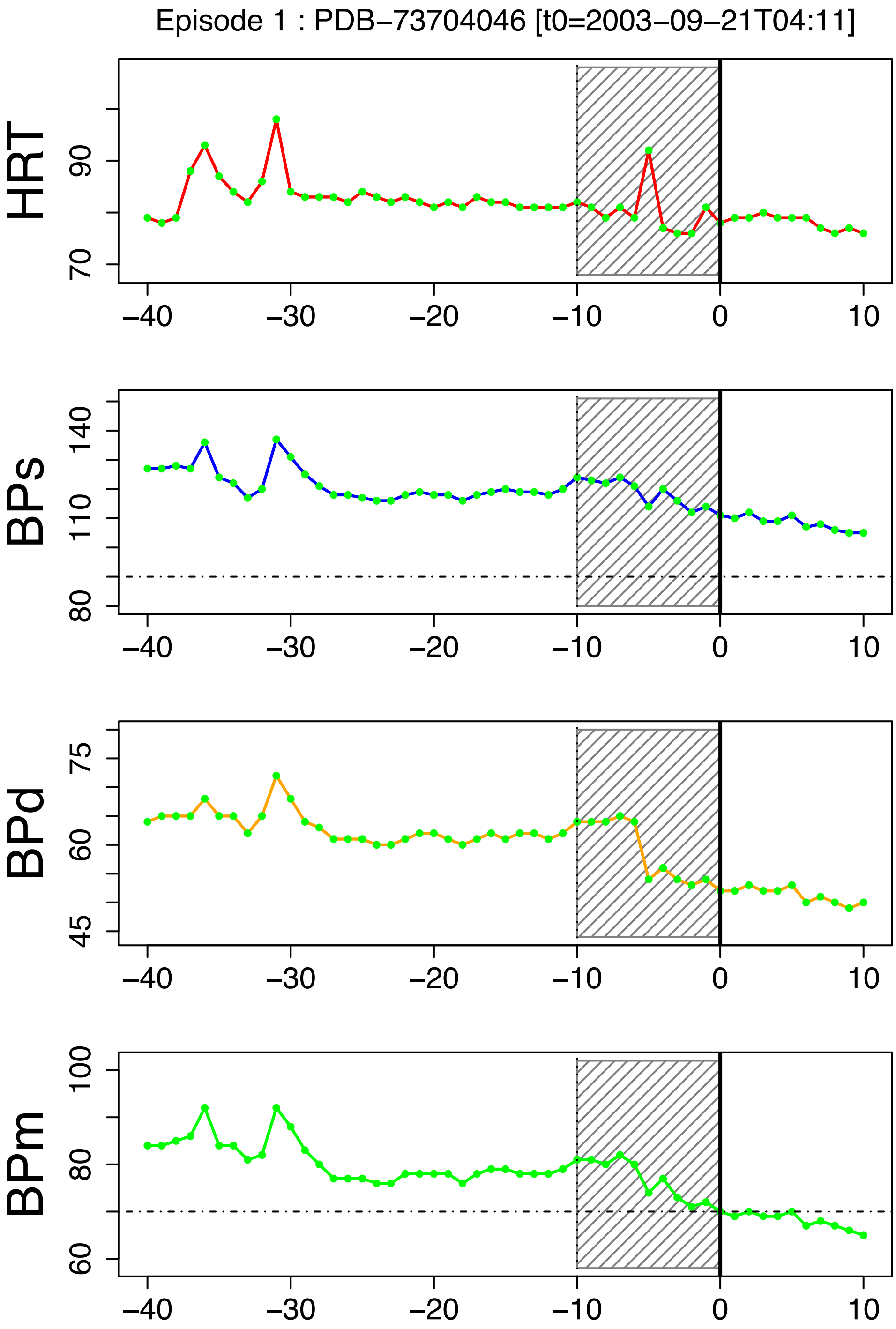


Figure 2.4: Physiological signals associated with episode 1 from patient 73704046 in the BrainIT database; y-axis pressures in mmHg, HRT in beats/min, x-axis in minutes

Figure 2.4 shows the four physiological signals that are being tracked. In each trace, $t = 0$ is the point where a hypotensive event occurred, in this case a breach of the BPm threshold of 70 mmHg. The trace between $t - 40$ and $t - 10$ is the data on which the model will operate, as the aim is to provide at least 10 minutes of warning of increased risk of an event. The horizontal lines on the BPs and BPm traces are the EUSIG threshold values of 90 and 70 mmHg respectively. Details of the measurements are given below:

Signal	Description	Units	Comments
HRT	heart rate	beats/min	Measured by electrocardiogram.
BPs	systolic arterial blood pressure	mmHg	The maximum blood pressure measured by intravenous (IV) cannulation (permanent needle) in one of the patient's main arteries.
BPd	diastolic arterial blood pressure	mmHg	The minimum blood pressure measured in the same line as BPs.
BPm	mean arterial blood pressure	mmHg	The average blood pressure computed by the bedside monitoring device. This is not a simple arithmetic average due to the shape of the cardiac cycle.

Table 2.5: ICU signals used for research

The statistical properties of blood pressures in TBI patients have been identified in a paper by Mitchell et al. (2007). A brief description of the signal characteristics is detailed below. A full discussion on the processing applied to these signals is given in Chapter 4, Section 4.2.2.

One limitation imposed on the models is that they must operate on a minimal amount of data, ideally 30 mins (this will be discussed in more detail in Section 4.2.6). All the traces in Figure 2.4 show a disturbance in the section between $t - 40$ to $t - 20$ period. The question must be asked — is this what the data looks like all the time? The next plot shows that this patient was, in fact, in a relatively stable condition before the 40 minutes shown in Figure 2.4.

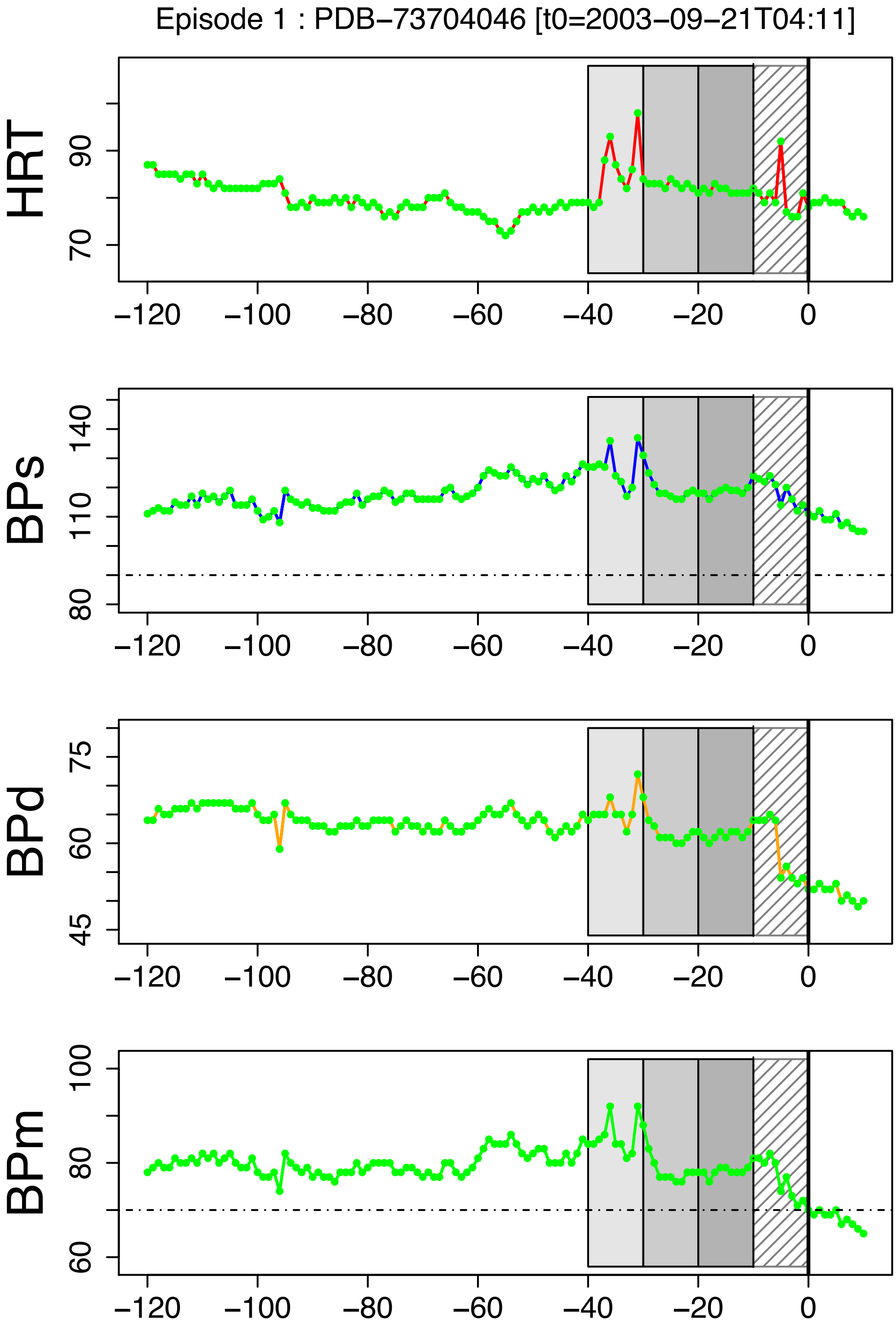


Figure 2.5: Physiological signals associated with episode 1 from patient 73704046 in the BrainIT database, stable pattern moving to episode, y-axis pressures in mmHg, HRT in beats/min, x-axis in minutes

Figure 2.5 is provided to show that the patient was in a stable condition prior to the event. Notice that the traces do not contain as much variation as the -40 to -20 section of

Figure 2.4 however, there are definite positive and negative trends.

There are, of course, several ways of attacking this problem; the increasing trend pattern shown by BPs in Figure 2.5 would suggest that some measure of long term trend would be a useful characteristic to assess. However, discussions with clinical colleagues suggest that a practising clinical team would tend to focus on the most recent 30 minutes and be less trusting of earlier data. It can, of course, be argued that with modern computing hardware it is easier to monitor longer term trends, and indeed some patterns may be developing much earlier than current clinical practice can spot. This is the nature of cutting edge research — a balance between acceptance by clinical teams resulting in a willingness to try new techniques and novel research to identify unknown patterns. This topic will be addressed in Chapter 9 with regard to future work.

2.4.1 Signal characteristics

The following pages showing kernel density estimates and box plots for each of the four signals that are used in the thesis. The plots are produced by combining the raw clinical data from the bedside monitors with a database of episode start times, which have been constructed as part of the research (see Section 4.1.1). The first plot on a page shows the kernel density plot of measurements at a point n minutes before an event. This period of n minutes is known as the event horizon (EH) and the technique for characterising the data will be fully explained in Section 4.2.2. Each plot shows two traces; a density curve for measurements that subsequently became an event n minutes later (the red trace) along with a density curve of measurements that represent the stable state of the patient (the black trace). This stable state measurement means that n minutes after the measurement was taken, the patient was still in a stable condition. Construction of the density plots allows a value for *mode* separation to be calculated. The second plot on the page uses a box plot to characterise the data. In this plot the *median* separation has been calculated.

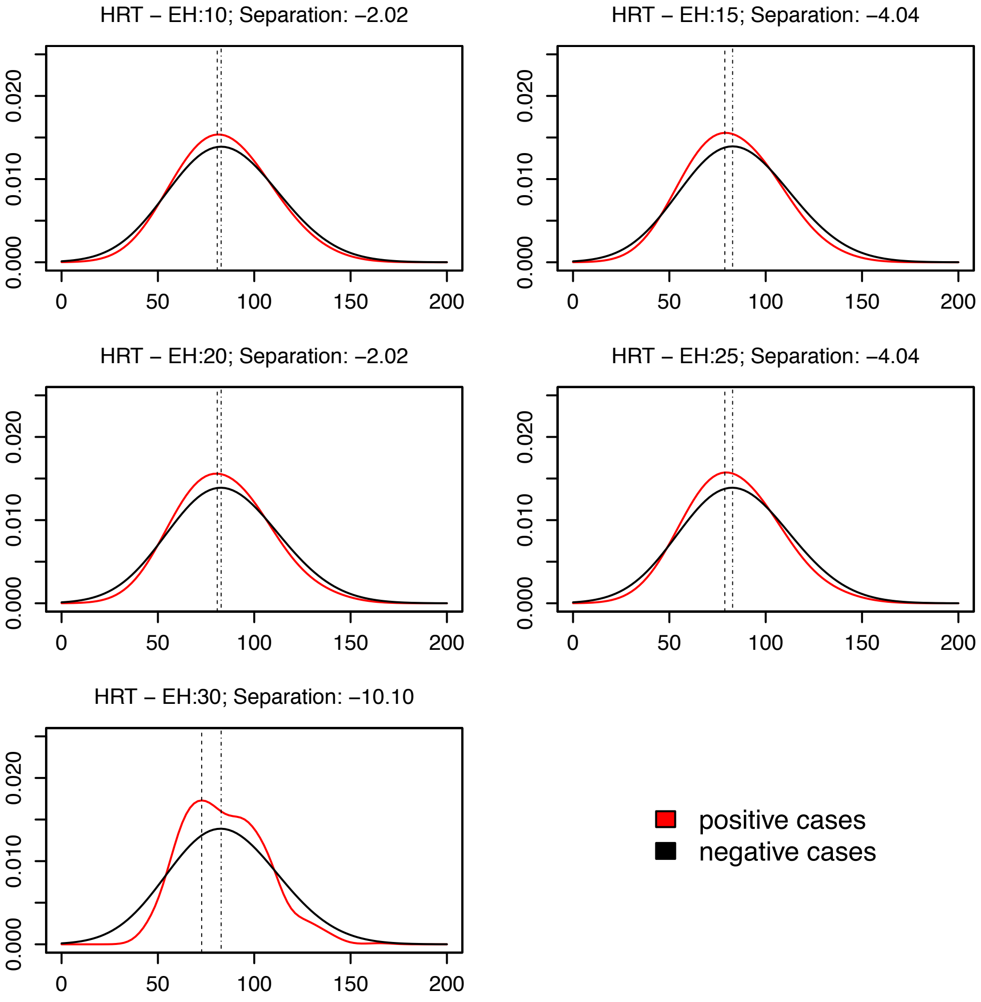


Figure 2.6: HRT — positive and negative cases; mode separation, x-axis beats per minute

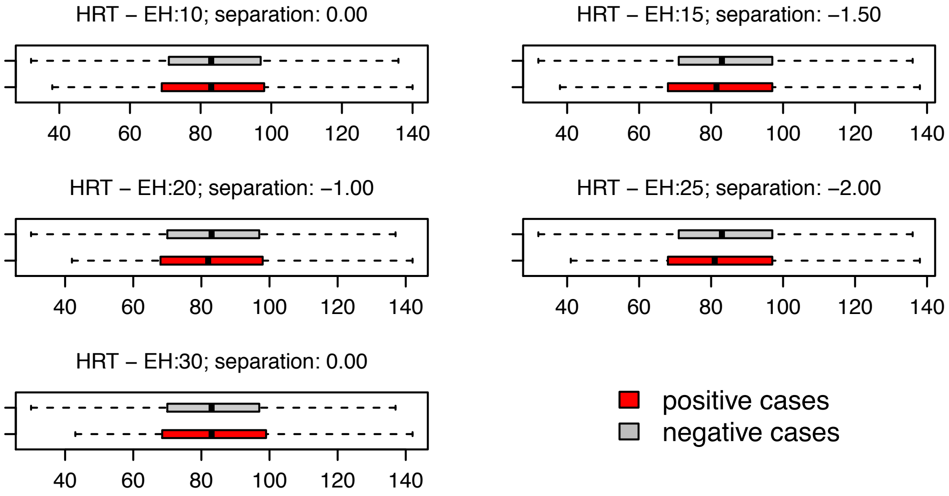


Figure 2.7: HRT — positive and negative cases; median separation, x-axis beats per minute

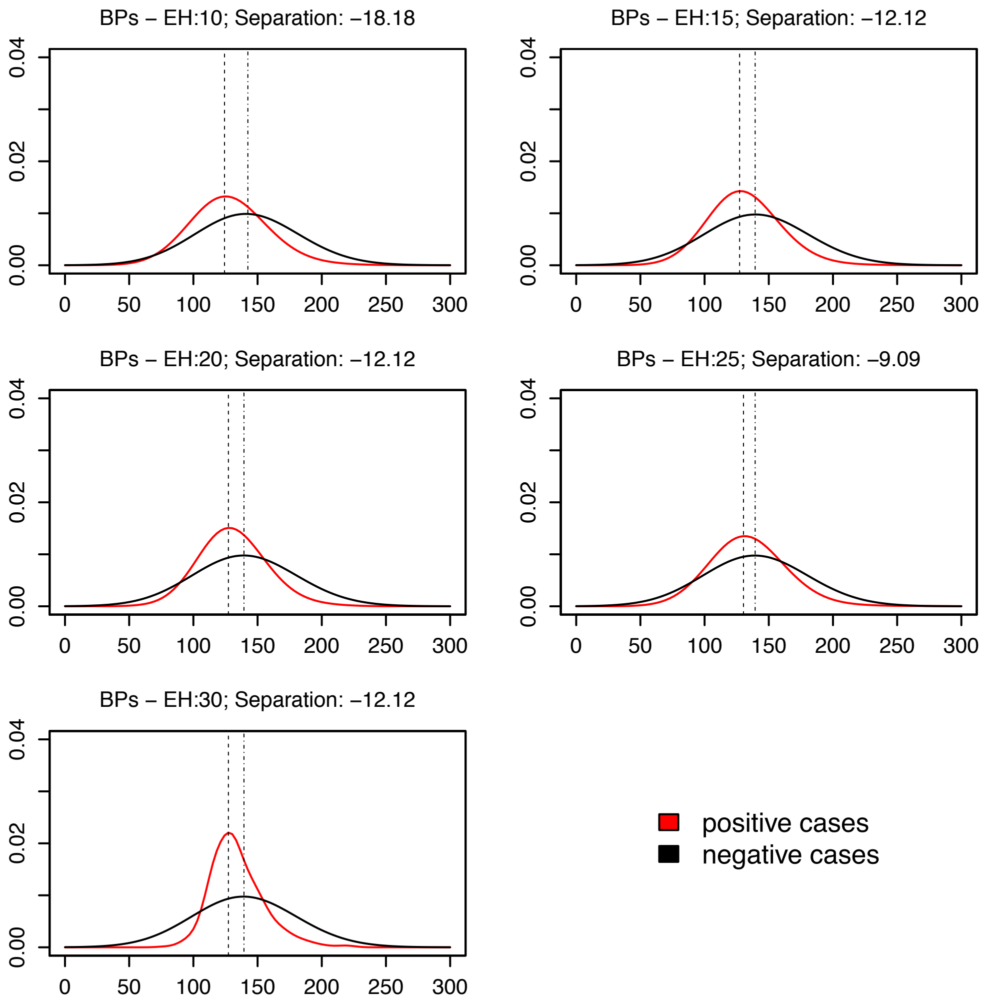


Figure 2.8: BPs — positive and negative cases; mode separation, x-axis mmHg

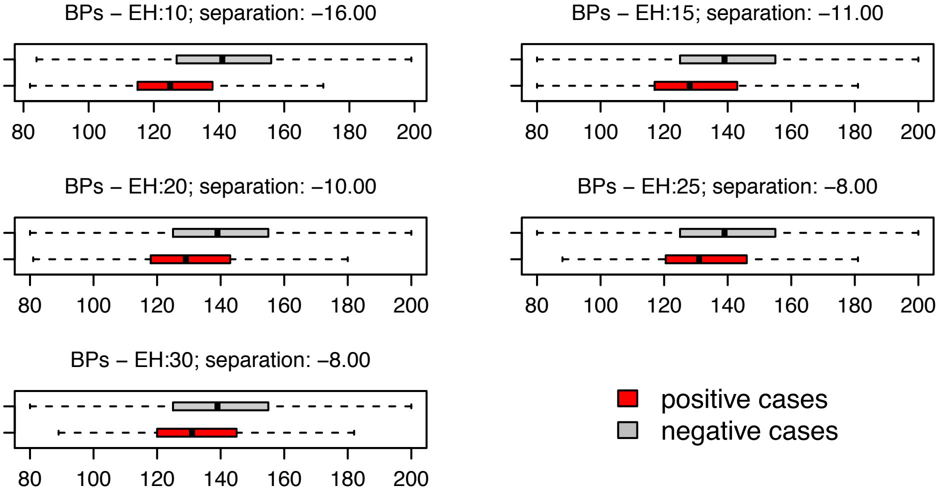


Figure 2.9: BPs — positive and negative cases; median separation, x-axis mmHg

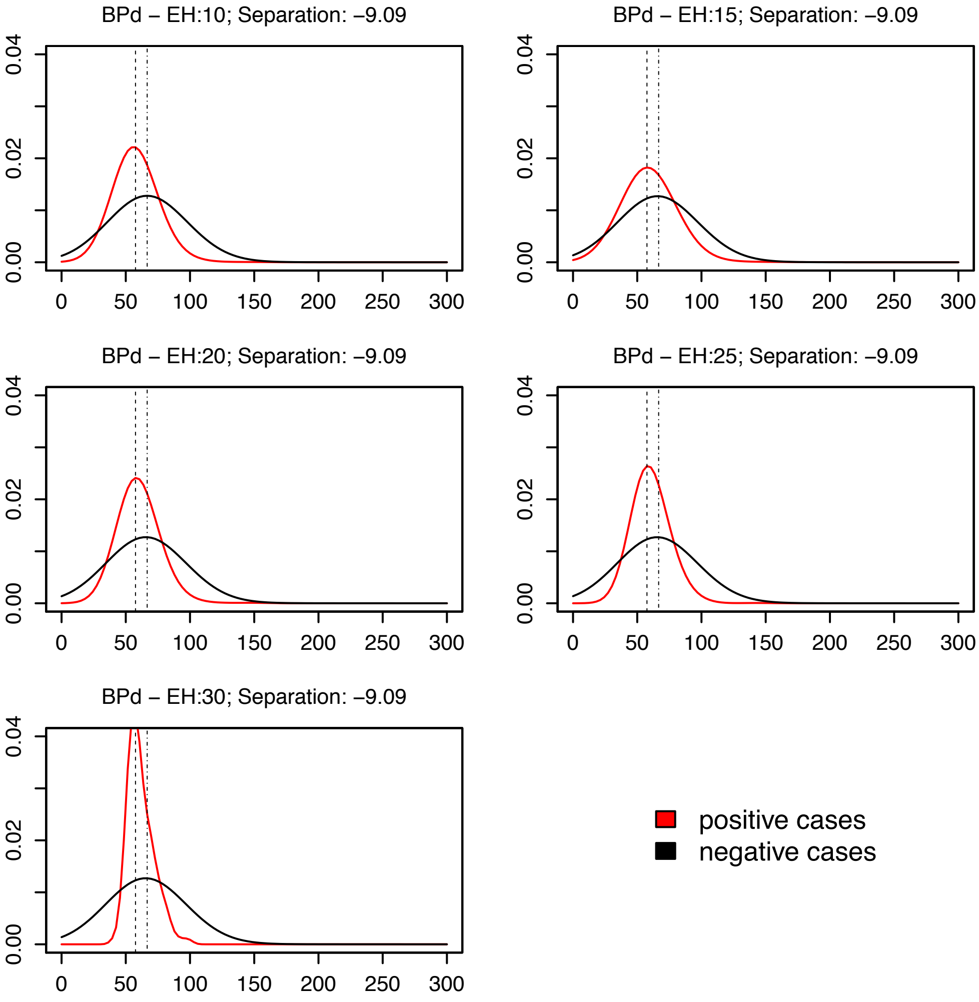


Figure 2.10: BPd — positive and negative cases; mode separation, x-axis mmHg

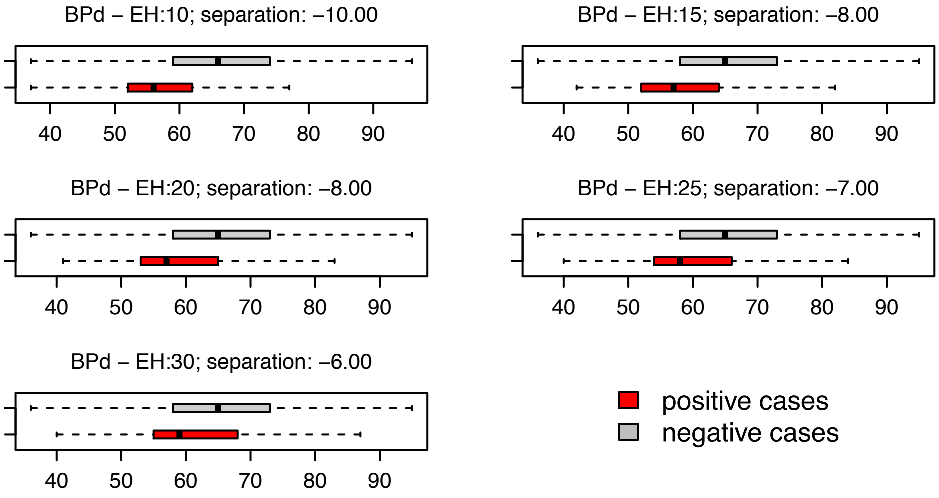


Figure 2.11: BPd — positive and negative cases; median separation, x-axis mmHg

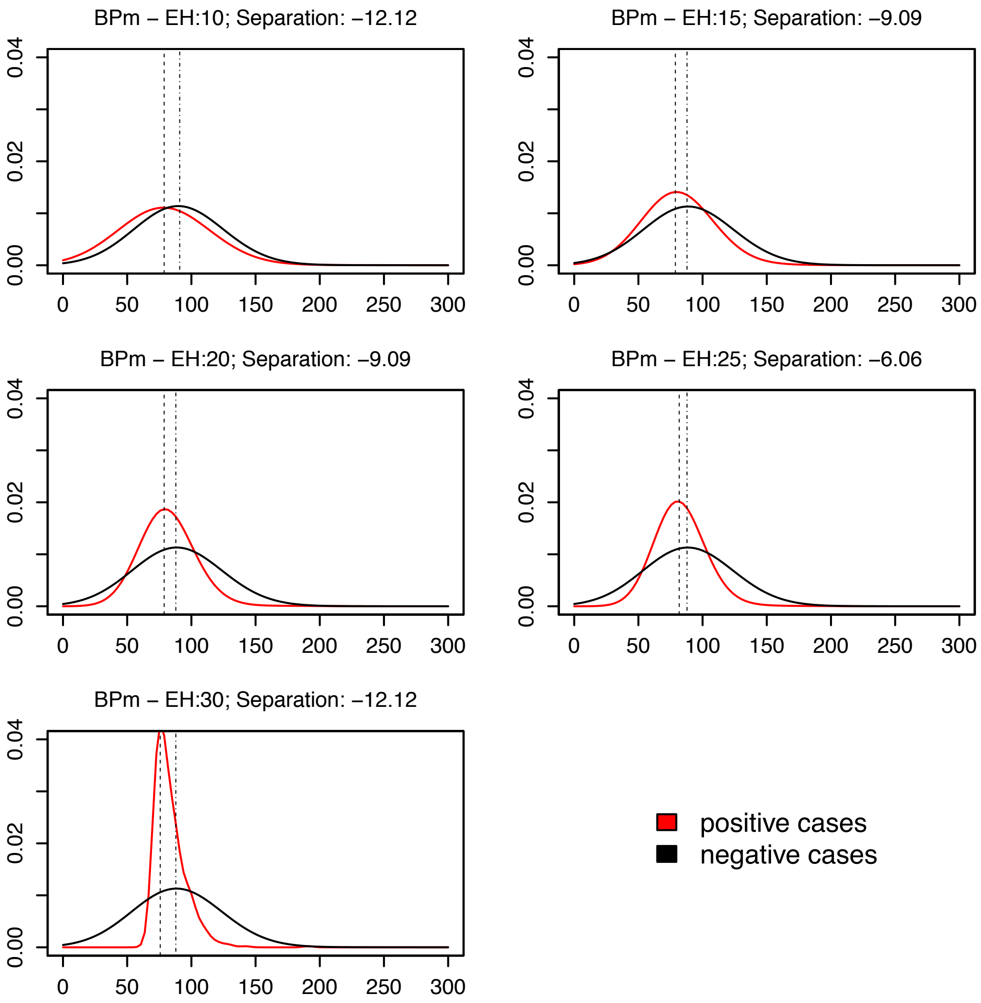


Figure 2.12: BPm — positive and negative cases; mode separation, x-axis mmHg

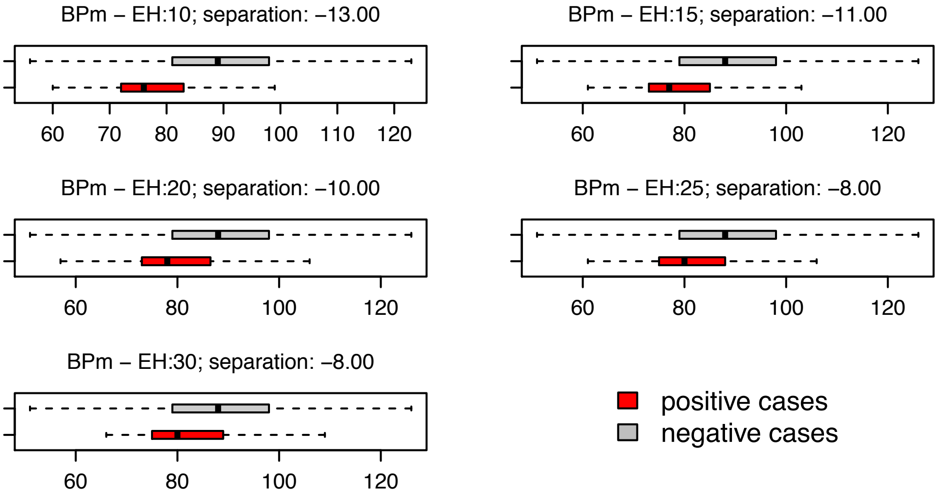


Figure 2.13: BPm — positive and negative cases; median separation, x-axis mmHg

2.4.2 Correlation assessment

It is expected that the available signals will show some correlation. Tables 2.6 to 2.8 examine the correlation between available physiological signals.

	HRT	BPs	BPd	BPm
HRT	1.00	-0.03	0.11	0.05
BPs	-0.03	1.00	0.49	0.73
BPd	0.11	0.49	1.00	0.91
BPm	0.05	0.73	0.91	1.00

Table 2.6: Correlation between physiological measures, stable condition

	HRT	BPs	BPd	BPm
HRT	1.00	-0.15	0.13	0.05
BPs	-0.15	1.00	0.23	0.58
BPd	0.13	0.23	1.00	0.87
BPm	0.05	0.58	0.87	1.00

Table 2.7: Correlation between physiological measures, unstable condition

	HRT	BPs	BPd	BPm
HRT	0.00	0.11	-0.02	0.00
BPs	0.11	0.00	0.26	0.15
BPd	-0.02	0.26	0.00	0.03
BPm	0.00	0.15	0.03	0.00

Table 2.8: Difference between stable and unstable correlations between physiological measures, i.e. the difference between Table 2.6 and 2.7

In order to assess the correlation between physiological signals, the data from 50 random draws of the 10_5 base data set was averaged to give Tables 2.6 and 2.7. Table 2.6 show the correlation when the patient is in a stable condition. It can be seen that there is considerable correlation between the values for systolic arterial blood pressure (BPs) and mean arterial blood pressure (BPm). Table 2.7 show the correlation when the patient is in a unstable condition i.e. the patient will start a hypotensive episode in 10 minutes time. This correlation is to be expected as the signals are all derived from the same intravenous line and BPm is the geometric mean of the aortic pressure trace (see Figure 2.14) over the last minute of heartbeats (typically 72).

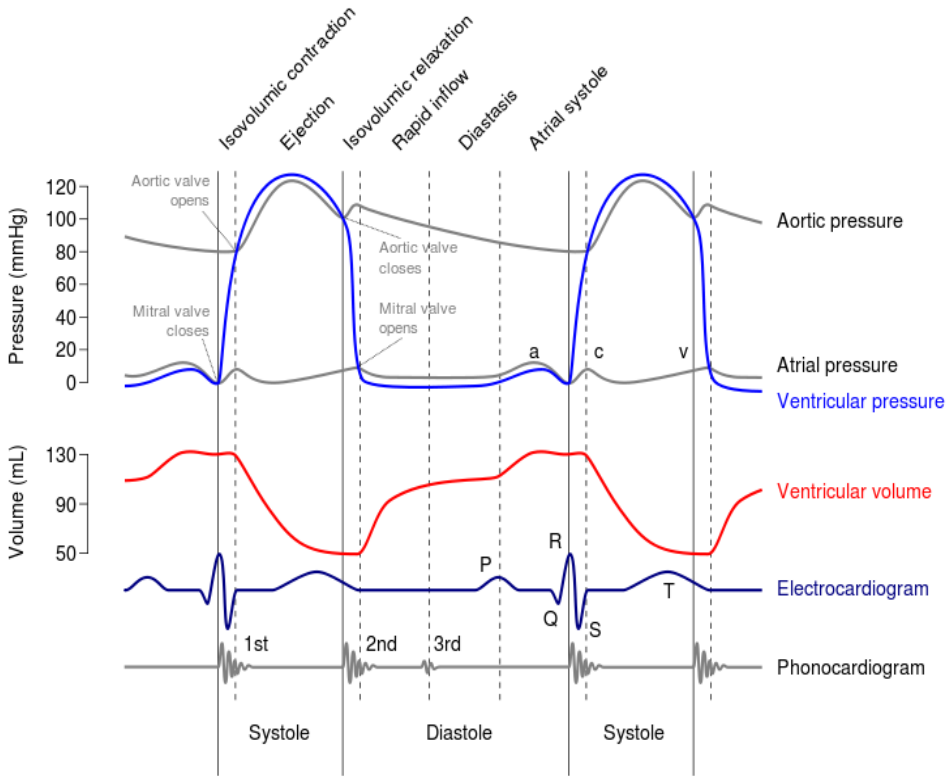


Figure 2.14: Wiggers diagram showing the cardiac cycle which consists of the Systole phase followed by the Diastole phase (Wikipedia, 2012). The trace of particular interest is the aortic pressure wave (top grey trace).

The BPm value is *approximated* by the equation (Klabunde, 2005, Chapter 5):

$$BPm = \frac{1}{3}BPs + \frac{2}{3}BPd \tag{2.1}$$

Eqn 2.1 results from the fact that a complete cardiac cycle (see Figure 2.14) is made up from the systolic phase which takes approximately $1/3$ of the cycle followed by the diastolic phase which accounts for the remaining $2/3$ of the cycle. This ratio is maintained if the patient is in a *stable* state. If the patient is unstable the shape of the cardiac cycle may change and the ratio $1:2$ may change. This is known as tachycardia if the heart rate is too fast and bradycardia if the heart rate is too slow. If either of these complications become severe they are deemed secondary insults in their own right (see Table 2.3).

Because of the correlation between signals, normal modelling practice would be to leave one of the signals out of any models under investigation. However because BPm is acting as a measure of instability the research will also investigate a model using all three blood pressure signals. This will be discussed in more detail in Section 5.2.

2.4.3 Signal characteristics by injury type

A classification of the episodes by injury type is given in Figure 2.15. This plot clearly shows that the bulk of the episodes fall into three injury categories: road traffic accidents (RTA); falls (Fl); and pedestrian (Pd).

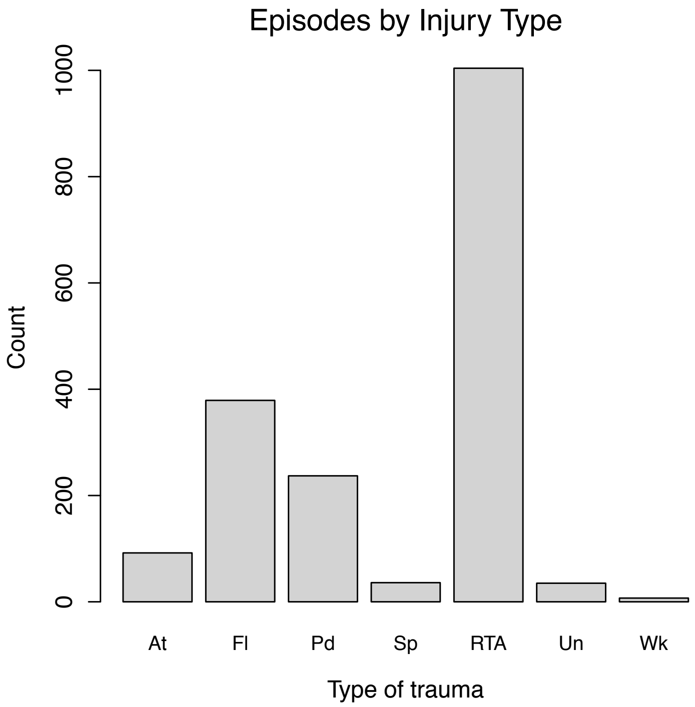


Figure 2.15: Episodes by injury types: At = Assault; Fl = Fall; Pd = Pedestrian; Sp = Sport; RTA = Road Traffic Accident; Un = Unknown; Wk = Work.

Taking the data from the three main injury types, Figure 2.16 details the median values of each physiological signal over the 30 minutes of available data used for modelling purposes. Each strip plot shows green markers indicating the median value when the patients are in a stable state. The red markers shows the median value for the unstable state. This data comes from the BDS_10_30_all_data.csv data set (See Section 4.2.4 for a description of the data sets and their naming conventions). The patterns for the blood pressure signals are similar for all three types of injuries. The stable and unstable medians for the blood pressure signals tend to be higher for patients who have suffered a fall. The pattern for heart rate is not so clear. For patients admitted with injuries classified as RTA or Pedestrian, the heart rate before an episode is higher than the median stable heart rate. Interestingly, for patients suffering from falls the heart rate is lower than the median stable heart rate.

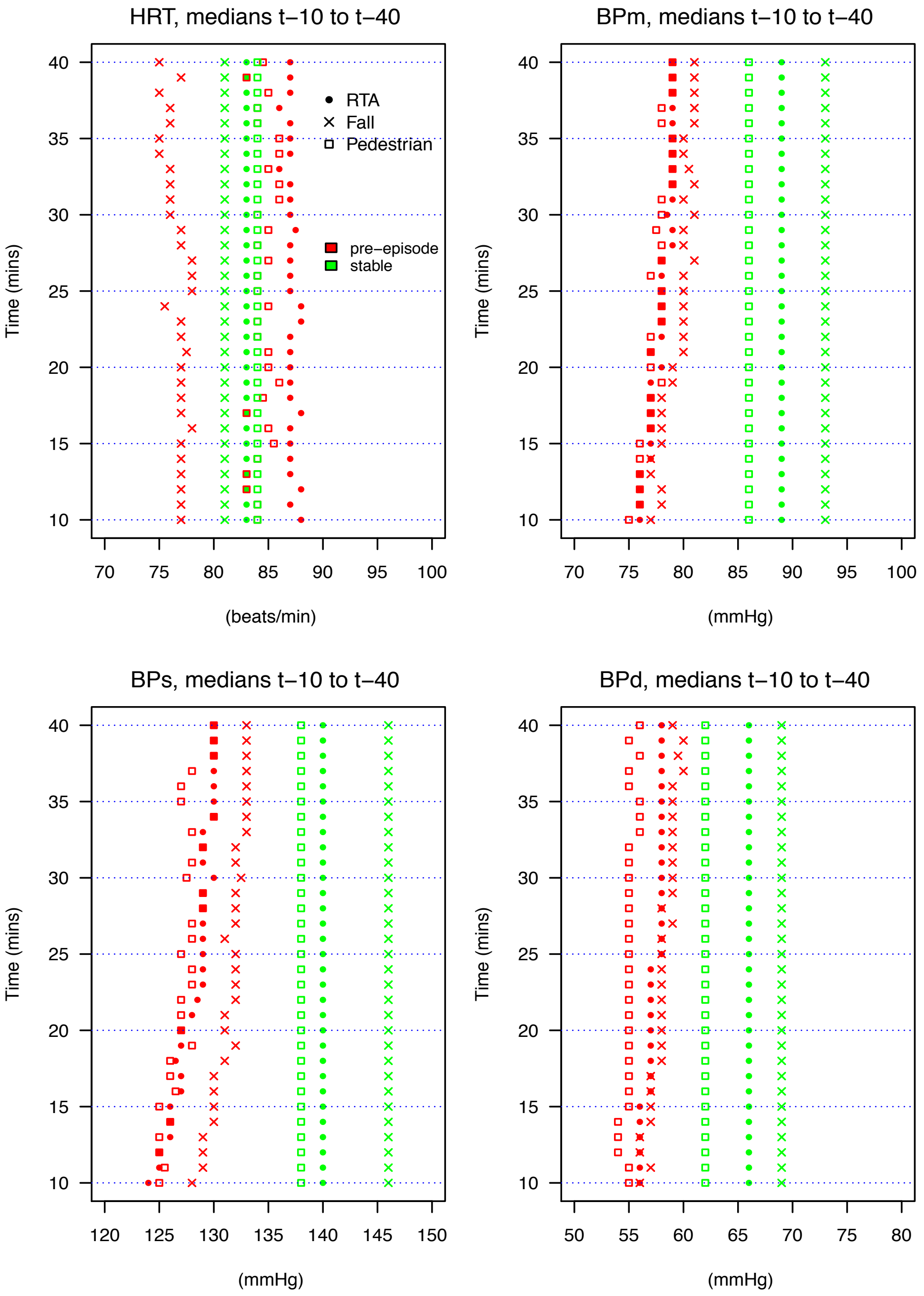


Figure 2.16: Signal medians by injury types. The green markers are from the stable state of the patients. The red markers indicate values just before a hypotensive episode is about to start.

Chapter 3

Methods Review

This chapter provides a review of three modelling approaches: classic time series analysis; a hidden Markov model based approach; and finally, classifier systems.

3.1 Approaches to modelling episodes

Although there are various approaches to the analysis of the problem of detecting the onset of hypotensive episodes, it is useful to restate the aim of the project which is to produce a system for predicting *increased probability* of a risk of a hypotensive episode, i.e. it would not be expected to exactly predict the *start time* of the episode. Indeed, clinicians would not accept such a system. The system should, however, ideally predict the possible onset of hypotension as far in advance as possible and be suitable for incorporation into an online decision support system.

The available data is essentially a collection of time series profiles with associated meta data (age, gender, type of injury, etc.). A first approach could be to produce a "typical" time series profile and attempt to use the techniques of classic time series analysis. This approach is discussed in Section 3.2. Another approach would be to use a method which accepts that the observed measurements are symptomatic of the underlying state of the patient and that the real processes contributing towards the onset of the hypotensive state are unobservable, at least in the context of during treatment in an ICU. This approach is discussed in Section 3.3. Finally, one could take the approach that the patient is either in one of two states: stable or unstable and approaching hypotension. The signals from the ICU monitors can then be classified as being from either one or the other of these states. If these assumptions are used, then a group of techniques called pattern classification can be used. This is the main approach taken in this thesis and is discussed in considerable detail in Section 3.4.

3.2 Classic time series analysis

As can be seen from the typical data described in Section 2.4, the identification of hypotensive episodes could be considered as an exercise in time series analysis. The data is collected at regular intervals i.e. $X_t = (HRT_t, BPs_t, BPd_t, BPm_t)$ the time interval index

t in this case being one minute, and it is likely that an individual vector of measurement, X_t, will have some correlation with previous vectors of measurement $(\ldots, X_{t-2}, X_{t-1})$.

Time series analysis is a well studied field and standard approaches to the analysis of time series signals have been developed. In particular, the techniques described by Box et al. (1994) are considered the classic method of building time series models. A useful text, with many examples and supporting analysis code using the R statistical language (R Development Core Team, 2008) is provided by Shumway and Stoffer (2006). Another accessible text is provided by Chatfield (2004) and the following description of the methods used during a time series analysis use the structure from chapters two to five of this book.

An examination of a given series should always start with exploratory data analysis of the data. This involves plotting the data and visually checking for trends and patterns that may be present. This initial approach is discussed in Section 3.2.1. Various probability models can then be tried and the approaches to building these models are discussed in Section 3.2.2. Once a suitable model has been chosen, the next step in the analysis is to fit the model to the observed time series data set, this procedure is detailed in Section 3.2.3. Finally, having chosen and fitted a model to the time series under investigation, the model can be used to make predictions regarding the future path of the time series signal. This is discussed in Section 3.2.4 with respect to both simple univariate models and more complex multivariate models that would be required in any attempt to predict the onset of a hypotensive episode.

In any time series analysis it is possible to examine the series from a time domain or frequency domain perspective. The frequency domain is not a natural area that clinical teams are familiar with and therefore the following discussions are confined to techniques developed in the time domain. Although time series analysis is not the focus of this thesis, there are examples of researchers using this type of approach in medical domains for example Christini et al. (1995) who investigated heart rate dynamics, and Vandenhouten et al. (2000) who examined brainstem signals using these techniques.

3.2.1 Descriptive procedures for time series data sets

A fundamental difference between a time series data set and a more “standard” data set obtained from some experimental procedure is the existence of correlation between successive measurements in the data set. Typically the major assumption made when examining a data set and beginning an analysis is that the observed measurements for a given variable are independent of each other. This is *not* the case in a typical time series data set. It is one of the main tasks of any time series analysis to assess and quantify the correlation between successive measurements for a given signal. It is important to realise that the normal measures that are used to characterise a data set e.g. mean and standard deviation are inappropriate for a time series.

A general model for a time series data set is often represented by the equation

$$x_t = \mu_t + \gamma_t + \epsilon_t \tag{3.1}$$

where x_t are the observations, μ_t is a trend component, γ_t is a seasonal effect and ϵ_t is the random error or unexplained component. A more detailed analysis may also include terms c_t for a cyclic effect, i_t for intervention effects controlled by an indicator variable and cv_t for covariate effects.

Note that this is a relatively simple form of a time series model. In practice, each of the terms on the right hand side can consist of further terms.

Eqn. 3.1 describes an additive model where the components of the model act independently. Another possible model uses the same components however they act multiplicatively as in Eqn. 3.2.

$$x_t = \mu_t \gamma_t \epsilon_t \tag{3.2}$$

This results in the seasonal effect or variation depending on the size of the trend effect. If this structure is suspected it is best to transform the data set using, for example, the natural logarithm on both sides of Eqn. 3.2 to give

$$\begin{aligned} log(x_t) &= log(\mu_t \gamma_t \epsilon_t) \\ &= log(\mu_t) + log(\gamma_t) + log(\epsilon_t) \end{aligned} \tag{3.3}$$

which can be analysed as an additive model. Other transformations are often used e.g. square root. Both natural logarithms and square root transformations are examples of the general Box-Cox transformation (Box and Cox, 1964).

As time series are often considered over a yearly period, there is frequently a “seasonal” aspect to the data e.g. higher temperatures in summer. For the the physiological signals that are being studied, the seasonal effect γ_t can usually be ignored. Cyclic effects can however be found even in shorter duration time series. In the context of the ICU measurements being studied, a possible cyclic effect would be the rise in blood pressure due to the regular breathing cycle. This may be particularly of concern if the patient is being mechanically ventilated. The more likely systematic variation would be in the form of a trend component which can be described as a long-term change in the mean value of the signal. An analysis would start with an investigation for a trend component and consideration of two cases. The first case sets μ_t to a constant value e.g. BPm in a patient who has been stabilised with the observed variation being modelled solely by the ϵ_t term. The second case assumes μ_t is changing over time i.e. a trend effect. This would be observed, for example, in the case where BPm begins to rise due to the administration of fluids by the clinical staff. In this situation the slope of the trend, positive or negative, is of interest.

Once any cyclic and trend components have been removed the signal should consist of a series of "residual" measurements. It is this residual signal that is modelled using the techniques and models to be discussed in subsequent sections. It is important however to stress that *all* of the components of the time series signal may be of use during the treatment of the patient.

An important concept in time series analysis is the idea of a "stationary" signal. This will be defined formally in Section 3.2.2.1. Conceptually a stationary process is one where there is no systematic trend away from the mean value of the signal, no systematic change in the variance of the data and all identifiable periodic variations have also been removed. In other words the residual time series is more or less the same across the entire data set. This property is unlikely to be strictly present in real world signals however as the probabilistic models are based on this assumption it is essential that the data is checked to judge if it is reasonable to assume this property is close to being achieved.

Perhaps the most important tool in the initial exploratory phase of the analysis is the "time plot". The two plots in Figure 3.1 are examples of time plots and show real blood pressure data from a patient. It is possible to see from these simple plots whether or not there is a possible trend and the general nature of the variation about a mean value.

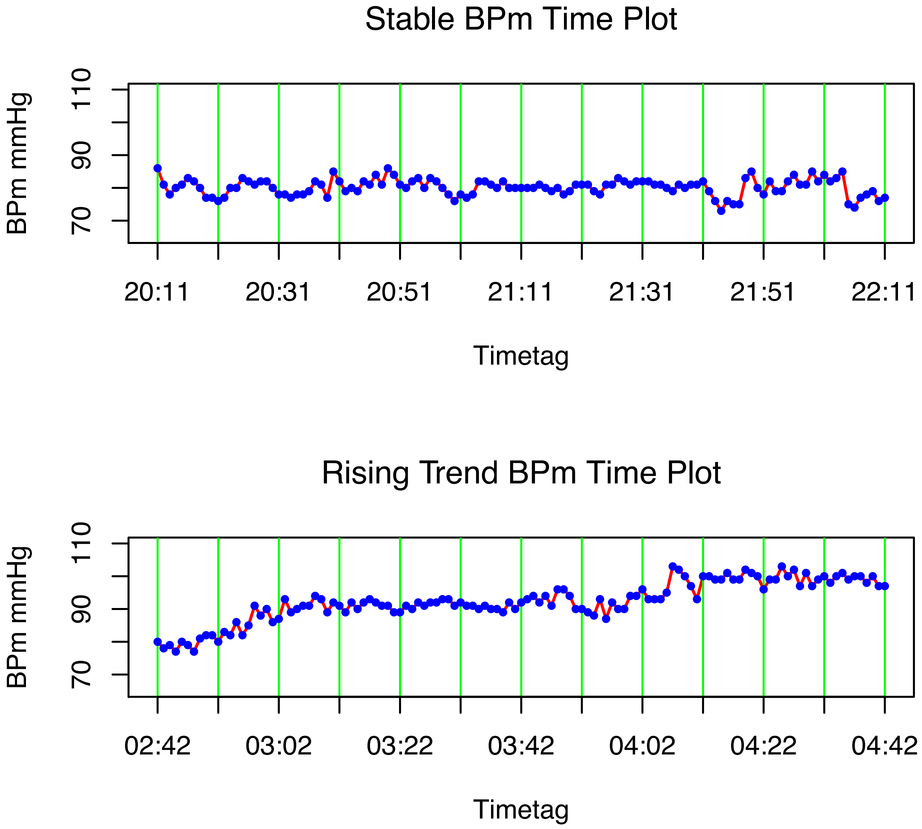


Figure 3.1: Two traces of mean blood pressure data from BrainIT patient 7157372, a 46 year old female. The top plot (from 2003-09-05) shows a stable mean arterial blood pressure of approximately 80 mmHg across a two hour period. The bottom plot (from 2003-09-01) shows a rising trend in mean arterial blood pressure over a different two hour period. The blood pressure rises from about 80 mmHg to about 98 mmHg. A visual examination suggests a linear trend may be present across the entire range of data from 02:42 to 04:42.

3.2.1.1 Characterising trends within a time series

If a trend component is suspected, there are typically three ways it can be removed: regression analysis; differencing; or moving average smoothing.

The trend component μ_t from Eqn. 3.1 is often modelled using a standard curve fitting technique. Techniques that may be used are: simple linear regression, a quadratic or higher order polynomial, natural splines or a logistic curve. Regression analysis is the most useful because part of the procedure obtains an estimate for the slope of the trend. A common analysis would start by using the familiar equation for a straight line with an added zero mean Gaussian noise term. This is shown in Eqn. 3.4 for a realisation x_t of a random variable X_t.

$$x_t = \alpha + \beta t + \epsilon_t \tag{3.4}$$

with constants α and β. ϵ_t represents the zero mean Gaussian noise. The parameters would typically be estimated using a maximum likelihood procedure. By subtracting the original series from the estimated trend a stationary residual series should be produced. The correlation (if any) in this residual series can then help to chose a suitable model for the stationary stochastic process generating the residuals (see Section 3.2.2). It should be noted that when using these curve fitting techniques to estimate the trend component, assumptions such as independence of observations and normality of the data may not be strictly valid and may explain problems with the resulting residual series.

Another common technique is differencing the series, which involves working with a series which is achieved by taking the difference between successive observed samples. A first order difference would be $z_t = x_t - x_{t-1}$.

The third technique is to use a *filter* to smooth the observations and thereafter proceed to analyse the residual series. A moving average is a form of linear filter which converts the original time series $\{x_t\}$ into another series $\{y_t\}$ using the linear operation

$$y_t = \sum_{r=-q}^{+s} a_r x_{(t+r)} \tag{3.5}$$

with $\{a_r\}$ equal to a set of weights with the property that $\sum a_r = 1$. The usual approach is to make the filter symmetric with $s = q$ and $a_j = a_{-j}$.

The simple moving average would define all $a_r = 1/(2q+1)$ to give

$$y_t = \frac{1}{2q+1} \sum_{r=-q}^{+s} x_{(t+r)} \tag{3.6}$$

It remains for the analyst to pick a value for q, the smoothing order, which is done using empirical methods. This also results in the removal of some of the data from the analysis. Moreover this approach does not provide a value for the trend gradient.

A more popular filter, which preserves the length N of the data, is the exponential smoother which is defined as

$$y_t = \sum_{j=0}^{N} \alpha(1-\alpha)^j x_{(t-j)} \tag{3.7}$$

with α a constant defined as $0 < \alpha < 1$.

3.2.1.2 Auto correlation

It is useful to assess the degree of correlation between measurements within a time series. The measurements can be taken at different distances within a series and provide useful properties called the sample autocorrelation coefficients. The interpretation of these coefficients can be used to choose a suitable probability model (see Section 3.2.3).

From standard statistics, given N pairs of observations $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ the sample correlation coefficient r is defined as

$$r = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{[\sum(x_t - \bar{x})^2 \sum(y_t - \bar{y})^2]}} \tag{3.8}$$

The value for r can vary between +1 (complete positive correlation) through zero (no correlation) to -1 (complete negative correlation).

For a time series with N observations $x_1, x_2, \ldots, x_N$, $N-1$ pairs of observations can be formed by taking each pair to be separated by one time interval, in other words, $(x_1, x_2), (x_2, x_3), \ldots, (x_{N-1}, x_N)$ by comparison with Eqn 3.8, this gives the autocorrelation coefficient at lag 1, r_1, as

$$r_1 = \frac{\sum_{t=1}^{N-1}(x_t - \bar{x})(x_{t+1} - \bar{x})}{(N-1)\sum_{t=1}^{N}(x_t - \bar{x})^2/N} \tag{3.9}$$

where $\bar{x} = \sum_{t=1}^{N} x_t/N$ is the overall mean. For large N the factor $N/N-1$ can be dropped and Eqn 3.9 becomes

$$r_1 = \frac{\sum_{t=1}^{N-1}(x_t - \bar{x})(x_{t+1} - \bar{x})}{\sum_{t=1}^{N}(x_t - \bar{x})^2} \tag{3.10}$$

The autocorrelation coefficient can also be defined at an arbitrary lag k, where the observations are at k steps apart from each other. This gives from Eqn 3.10

$$r_k = \frac{\sum_{t=1}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{N}(x_t - \bar{x})^2} \tag{3.11}$$

Note that, in practice, the autocorrelation coefficients, $\{r_k\}$, are often calculated by first forming the autocovariance coefficents, $\{c_k\}$, using the formula

$$c_k = \frac{1}{N}\sum_{t=1}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x}) \tag{3.12}$$

then computing

$$r_k = c_k/c_0 \tag{3.13}$$

for $k = 1, 2, \ldots, M$ with $M < N$

A useful graphical representation of the $\{r_k\}$ autocorrelation coefficients is called the correlogram, also known as the autocorrelation function (ACF) plot. The correlogram plots the sample autocorrelation coefficients against the lags k for $k = 0, 2, \ldots, M$ with M typically being 20 or 30. Note that r_0 is by definition 1 but it is usually plotted to give a reference for the other coefficients. Fig 3.2 gives an example of a correlogram for the blood pressure time series depicted in Fig 3.1. The dashed lines indicate approximate 95% confidence limits (see Section 3.2.3), values of r_k below these limits are considered to be not significantly different from zero.

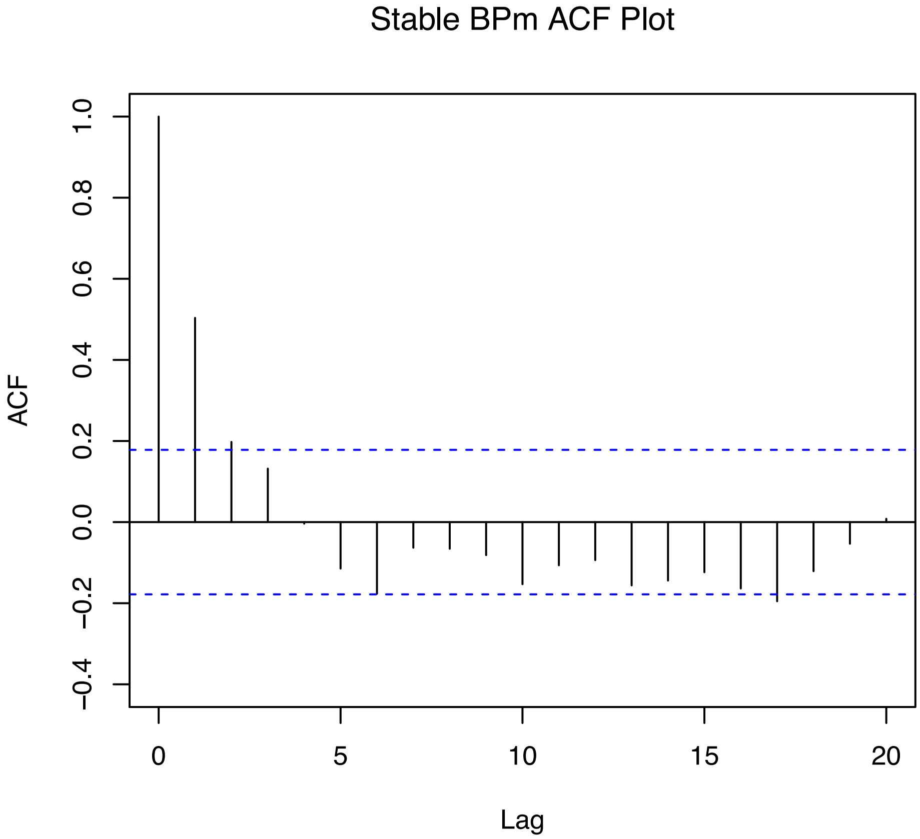


Figure 3.2: The correlogram for the stable blood pressure (BPm) time series plot shown in Fig 3.1. After three minutes the autocorrelation coefficients have decayed to below the significant level.

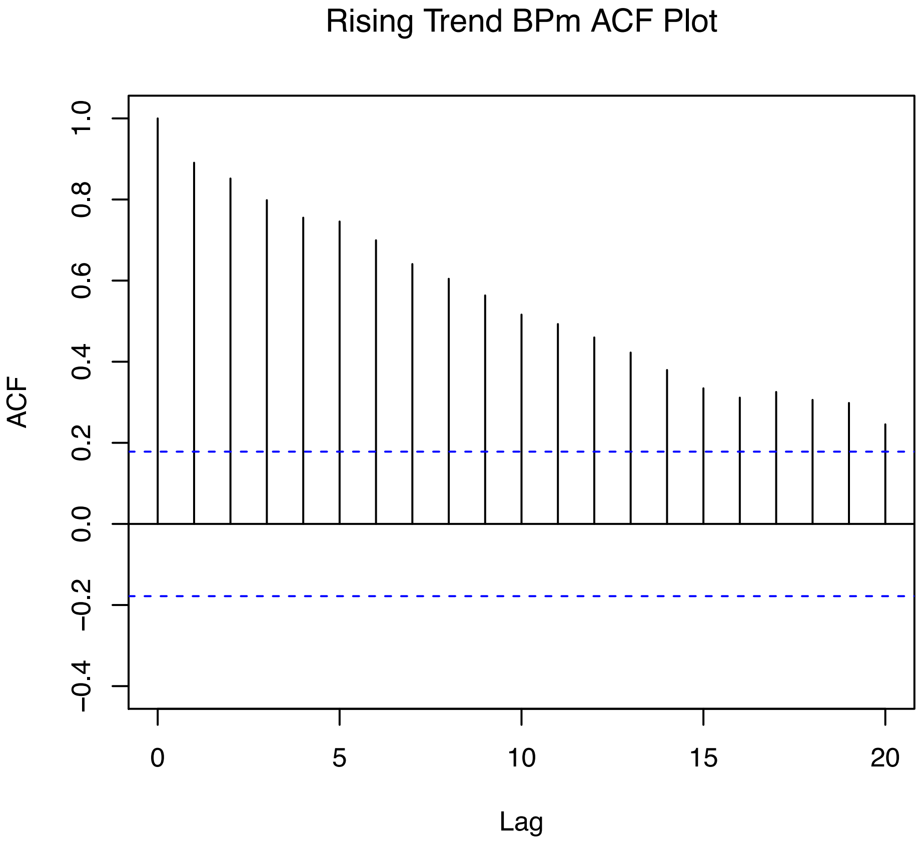


Figure 3.3: The correlogram for the rising trend blood pressure (BPm) time series plot shown in Fig 3.1. The effects of the trend are shown in the slowly deceasing values of the autocorrelation coefficients at increasing lags.

3.2.2 Probability models for time series data sets

A class of models known as “stochastic processes” can be defined for a collection of observed time points which constitute a time series. Chatfield (2004) defines a stochastic process as ‘a statistical phenomenon that evolve in time according to probabilistic laws’.

Each time point $t = 0, \pm 1, \pm 2$ within a discrete time series is considered to be a random variable X_t and the observed time series itself, x_t, with $t = 1, \ldots, N$ is considered to be a single realisation of an infinite set of possible time series from an assumed stochastic process.

A stochastic process is characterised by its *moments* with the first and second moments being used in practice although higher moments are theoretically possible. The first moment is called the mean and is defined for all t as:

$$\mu(t) = E[X(t)] \tag{3.14}$$

The second moment is called the autocovariance function (acv.f) is defined for all t as

$$\gamma(t_1, t_2) = E[X(t_1) - \mu(t_1)][X(t_1) - \mu(t_1)] \tag{3.15}$$

A useful special case of the acv.f is the variance function which is defined for $t_1 = t_2$ as

$$\sigma^2(t) = Var[X(t)] \tag{3.16}$$

3.2.2.1 Stationarity in a time series data set

This section formally defines stationarity which was discussed in Section 3.2.1. A time series is defined as *strictly stationary* if the joint distribution of $X(t_1), X(t_2), \ldots, X(t_k)$ is the same as the joint distribution of $X(t_{1+\tau}), X(t_{2+\tau}), \ldots, X(t_{k+\tau})$ for all $t_1, \ldots, t_k$ and τ. This implies that it does not matter which point is used for the origin within the series i.e. the origin can be shifted $\pm\tau$ with no effect. The joint distribution therefore only relies on the interval between the k values of the series. The special case of $k = 1$ yields that the $X(t_k)$ must be identically distributed, including $\mu(t) = \mu$ and $\sigma^2(t) = \sigma^2$ i.e. constants which do not depend on t. For the case $k = 2$, the joint distribution of $X(t_1)$ and $X(t_2)$ only depends on the time interval $(t_2 - t_1) \equiv \tau$ which is know as the *lag*. This leads to Eqn 3.17 for the autocovariance function $\gamma(t_1, t_2)$ as

$$\begin{aligned}\gamma(\tau) &= E\{[X(t) - \mu][X(t+\tau) - \mu]\} \\ &= Cov[X(t), X(t+\tau)] \end{aligned} \tag{3.17}$$

Eqn 3.17 is usually normalised using $\gamma(0)$ to give the autocorrelation function

$$\rho(\tau) = \gamma(\tau)/\gamma(0) \tag{3.18}$$

This strict definition of stationarity is, in practice, often replaced with a requirement for *second-order stationarity* also know as *weakly stationary*. In this case the mean must be constant and the autocovariance function must only depend on the lag τ. This gives

$$E[X(t)] = \mu \tag{3.19}$$

and

$$Cov[X(t), X(t+\tau)] = \gamma(\tau) \tag{3.20}$$

Although the autocorrelation function (acf) is unique to a given stochastic process it should be noted that the opposite is not true. In other words, a given acf may represent many different stochastic processes and care must be taken when interpreting the plot of the output of an acf.

3.2.2.2 Pure random model

A pure random model consists of a series of random variables, $\{Z_t\}$ which must be mutually independent and identically distributed. It is usual to assume that the random variables are drawn from a Gaussian distribution with mean zero and variance of σ_Z^2. These assumptions therefore give a constant mean and variance and it follows that

$$\gamma(k) = Cov(Z_t, Z_{t+k}) = \begin{cases} \sigma_Z^2 & k = 0 \\ \\ 0 & k = \pm 1, \pm 2, \ldots \end{cases} \tag{3.21}$$

also, as different values of the series are uncorrelated, the autocorrelation function is given by

$$\rho(k) = \begin{cases} 1 & k = 0 \\ \\ 0 & k = \pm 1, \pm 2, \ldots \end{cases} \tag{3.22}$$

A pure random model produces a time series that is both second-order stationary and strictly stationary. This model is often used as a building block for other more complicated models such as the moving average (MA) model and the autoregressive (AR) model.

3.2.2.3 Random walk model

A random walk model is defined as

$$X_t = X_{t-1} + Z_t \tag{3.23}$$

where Z_t is drawn from a random (usually Gaussian) model $\{Z_t\}$ with zero mean and variance σ_Z^2 as described in Section 3.2.2.2. It is usual to start the model at zero for $t = 0$ giving $X_1 = Z_1$ and $X_t = \sum_{i=1}^{t} Z_i$.

A random walk model is non-stationary as the mean, $E(X_t) = t\mu$, and variance, $\text{Var}(X_t) = t\sigma_Z^2$, are both functions of t.

3.2.2.4 Autoregressive (AR) model

The definition of a autoregressive model of order p (written as AR(p)) is

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \cdots + \alpha_p X_{t-p} + Z_t \tag{3.24}$$

this has a similar form to a multiple linear regression model with X_t regressed against past values of itself. Z_t is drawn from a random model $\{Z_t\}$ with zero mean and variance σ_Z^2.

3.2.2.5 Moving average (MA) model

The definition of a moving average model of order q (usually written as MA(q)) is

$$X_t = \beta_0 Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q} \tag{3.25}$$

with the β_i coefficients as constants and $Z_t, \ldots, Z_{t-q}$ drawn from a random model $\{Z_t\}$ with zero mean and variance σ_Z^2.

3.2.2.6 Autoregressive moving average (ARMA) model

It can be useful to combine the AR and MA models to give an ARMA(p,q) process defined as

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \cdots + \alpha_p X_{t-p} + Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q} \tag{3.26}$$

An ARMA model may be found to describe a stationary time series with fewer parameters than either a pure AR or pure MA model. In these circumstances an ARMA model is preferred.

3.2.3 Fitting probability models to a given time series data set

Having discussed the properties of possible stochastic processes in Section 3.2.2, this section describes the steps necessary to chose a model and estimate the parameter values of that model. The main tool used to identify a possible model for a given time series is a plot of the autocorrelation function for increasing values of lag. This plot is known as the *correllogram* and was introduced in Figures 3.2 and 3.3. The patterns produced by this plot and a related plot the *Partial Autocorrelation Function* (PACF, to be detailed later), can guide the modelling process. An essential part of these two plots are the 95% confidence bands which, for N observations, have values approximated by $\pm 1.96/\sqrt(N)$.

Assuming that trend effects have been removed, the analysis proceeds by estimating characteristics of the remaining stationary series. The first parameter to assess is the autocovariance, a measure of how much "memory" of past values is contained within the current observation. The derivation of the normalised value of autocovariance, the autocorrelation, of a time series was presented in Section 3.2.1.2. The partial autocorrelation function (PACF) at lag τ, written as π_τ, gives a value for the excess correlation that has not been quantified by the correlation by the $\tau - 1$ lags. The π_τ value is equal to the α_τ value when an AR(τ) model is fitted to the time series.

After obtaining a correlogram for the time series under investigation, it can be compared against correlograms generated from one of the standard types of models: autoregressive (AR); moving average (MA); or a combination of both autoregressive and moving average (ARMA). First consider an MA(q) process. The correlogram for such a process will cut off at lag q. If such a distinct pattern is not present, the process may be characterised by an AR(p) or ARMA(p,q) process. In both cases the lags greater than zero will tend to decay gradually in an exponential manner.

The following sections provide more detail on each of these model types in particular the use of the correlogram to aid model choice.

3.2.3.1 Fitting an AR model

There are two tasks required to fully specify an AR(p) model. Firstly, the order p must be decided, this is followed by the estimation of the, α_i parameters. The order of the process can be found by starting with an AR(1) process and comparing the theoretical acf and pacf plots to those obtained from the time series under investigation. The order p is increased until the PACF plot cuts off at lag τ. The estimation of the parameters can be done by either the least squares method or by using the sample autocorrelation coefficients in the first p Yule-Walker equations (Chatfield, 2004, §3.4.4).

3.2.3.2 Fitting an MA model

The same two tasks are required to specify a MA(q) model. The order of an MA(q) process can be easily obtained from the acf plot as the plot cuts off at lag q. In other words the autocorrelation values at lags greater than q are within the 95% confidence markers. The estimation of parameters is carried out using an iterative numerical technique as direct least squares estimates are not available.

3.2.3.3 Fitting an ARMA model

It is possible that neither and AR(p) or MA(q) will adequately fit the time series under investigation. In these cases it is possible to try a combination of both techniques. However, the orders p and q are difficult to estimate from the acf and pacf plots. The estimation of the parameters will continue to require numerical optimisation techniques due to the the MA(q) components of the model.

3.2.3.4 Model checking

Finally, once a model has been chosen and the order and parameters have been estimated, the model should be checked to ensure that it provides a reasonable estimate of the time series under investigation. The most straightforward technique is to generate a *residual* time series by calculating the one-step-ahead forecast and subtracting it from the observed data. This provides the one-step-ahead forecast error. As an example, consider an AR(1) model, $X_t = \alpha X_{t-1} + Z_t$. The fitted value at t is given by $\hat{\alpha} x_{t-1}$. The residual, z_t for the observed value x_t is $\hat{z}_t = x_t - \hat{\alpha} x_{t-1}$. If the model is adequate a plot of the residuals should be close to zero and be random, i.e. with no discernible pattern.

Rather than carry out each of the above steps separately, these approaches can be combined by building a SARIMA $(p, d, q) \times (P, D, Q)_s$ model, (Box et al., 1994, Chapter 9). In this model a linear regression model is constructed to handle the seasonal and trend aspects of the series and an ARIMA model deals with the residual part of the observations.

3.2.4 Prediction using time series probability models

The final task is to use the model in practice to provide a prediction of the quantity of interest. Assuming several variables are involved, as is the case in this research, two approaches can be tried.

The first approach is to use univariate analysis for each of the variables separately. This is a basic technique and uses a model fitted only on currently available observations from a single variable. Therefore the predicted value x_{t+1} depends only on some function of $x_t, x_{t-1}, x_{t-2} \ldots$. For example in this project the two primary variables of interest are the systolic arterial blood pressure (BPs) and the mean arterial blood pressure (BPm).

The second approach is called multivariate analysis and uses the information from other available measurements in an attempt to provide a more accurate forecast for the values of the variables of interest. In this research other time series variables could be considered such as the patient's Glasgow Coma Score (GCS) (see Section 2.3) which is assessed at regular intervals by the clinical staff. In a multivariate setting, in addition to time series variables, static variables such as the demographic variables age and gender could be used. It is also possible to add in more subjective information from the clinical staff into the model.

If a system based on time series forecasting were to be used in a clinical setting, it is likely that a combination of both the above techniques would be used. As the univariate approach relies on regularly collected observations it can easily be automated and tables and plots can be regularly updated. The multivariate approach can also possibly be automated (or at least a portion of it can be automated) however, the final result may come from a manual review of all the component parts of the overall model.

Another consideration is whether the model will only be used to provide a point forecast, i.e. a single value, or whether there is an additional requirement to provide an assessment of the uncertainly around the point forecast.

In a univariate setting the model can be used to calculate the n-step ahead predictions. An n-step prediction is simply the model evaluated at points in time $1, \ldots, n$ beyond the currently measured values. Using this approach for this project would require building models for both the BPs and BPm signals as the EUSIG definition for a hypotensive event has a threshold for both measurements. As each new minute vector of information is received, the n-step predictions need to be carried out to determine whether or not the EUSIG threshold has been breached. If a breach has occurred, a further n-step process is required to check if the hysteresis requirements are likely to be met. The output from this process is a prediction that an episode *will* occur in EH minutes time and unless this prediction is extremely accurate, the system is be unlikely to be accepted. These classic techniques are also capable of computing confidence intervals around the n-step prediction, albeit at the expense of more processing power. Therefore a system could be developed which would give a quantitative indication of the uncertainty in the predicted signal trajectory. The above steps constitute a univariate analysis.

A multivariate approach would jointly model more than one variable e.g. heart rate (HRT) and diastolic arterial blood pressure (BPd). This makes the analysis much more complex. Given that an autoregressive model can be identified for the individual time series components of the overall model a vector autoregressive model (VAR) could be constructed. The complexity in these types of models comes from the need to consider serial correlation within an individual component and the correlation *between* each univariate series. This correlation between series is carried out by using the cross correlation function. Given a multivariate time series with m components $\{\boldsymbol{X}_t\}$ where $\boldsymbol{X}_t^N = (X_{1t}, X_{2t}, \ldots, X_{mt})$, the cross correlation function can be used to construct the cross covariance matrix of $\boldsymbol{X}_t$ and $\boldsymbol{X}_{t+k}$.

A simple form for a VAR(1) model can be seen by considering $m = 2$ which gives two equations where the values of X_{1t}, X_{2t} are linearly dependent at time $(t-1)$. This can be written as two equations

$$\begin{aligned} X_{1t} &= \phi_{11} X_{1,t-1} + \phi_{12} X_{2,t-1} + \epsilon_{1t} \\ X_{2t} &= \phi_{21} X_{1,t-1} + \phi_{22} X_{2,t-1} + \epsilon_{2t} \end{aligned} \tag{3.27}$$

Eqn 3.27 in vector form is

$$\boldsymbol{X}_t = \Phi \boldsymbol{X}_{t-1} + \epsilon_t \tag{3.28}$$

with $\epsilon_t^T = (\epsilon_{1t}, \epsilon_{2t})$ and

$$\Phi = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}$$

3.3 Hidden Markov models

Another approach would be to use a hidden Markov model (HMM), a technique suggested in the 1970s by Baum et al. (1970), with the leading reference in the literature being the papers by Rabiner and Juang (1986) and Rabiner (1989). This approach has the advantage of trying to make use of the temporal structure that is intuitively present in the physiological measurements. The model combines observed measurements (demographic and physiological variables) in a vector $\boldsymbol{x}$ with a set of unobserved states in a vector $\boldsymbol{z}$.

This form of model can be illustrated using a Directed Acyclic Graph (DAG), (Bishop, 2006, Chapter 8), that shows the factorisation of the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{z}$. Figure 3.4 shows this graphical model for an HMM with the shaded nodes representing the observed clinical variables. The unshaded nodes are the "hidden" variables and can take on one of the $1 \dots K$ states representing, in this case, the patient's "clinical" state i.e. stable, hypotensive risk, hypotensive, recovering from hypotension. This is the state that would be displayed to the clinical team in an attempt to give them warning of the "hypotensive risk" condition.

A critical assumption made by the HMM is that given the current hidden state, the past is conditionally independent of the future. Ignoring estimated parameters, the current hidden state therefore contains all information required to predict the future. This assumption simplifies the model and thus the calculations. However, it could be argued that this is unrealistic in the context of this research.

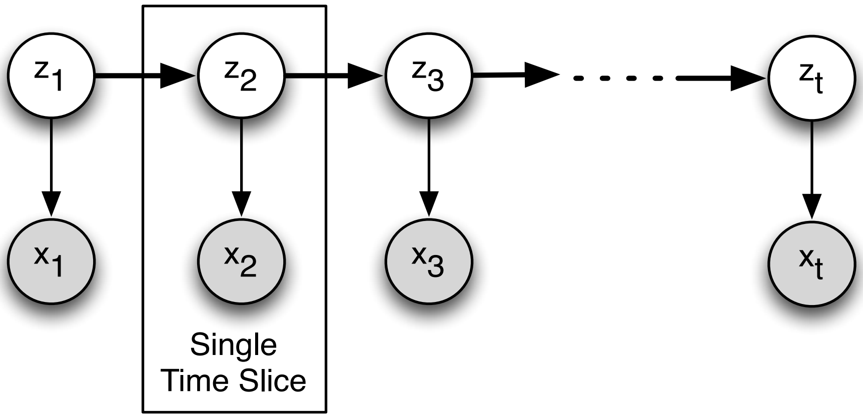


Figure 3.4: HMM graphical model

The following description is based on Bishop (2006, §13.2). Considering a single time slice, t, the distribution $p(\boldsymbol{x}_t)$ is a mixture distribution with component-wise densities given by $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$. The variable z uses a 1-of-K coding scheme where $z_{tk} = 1$ if the observations come from state k at time t. As signified by the arrow on the left hand side of the unshaded node, the probability distribution of $\boldsymbol{z}_t$ depends on the previous state $\boldsymbol{z}_{t-1}$ through the conditional distribution $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$.

The hidden variable $\boldsymbol{z}_t$ is modelled using a transition matrix $\boldsymbol{A}$ of dimension $K \times K$, where A_{jk} is the probability of the hidden variable moving from state j to state k during a single time slice,t. The transition probabilities are given by $A_{jk} \equiv p(\boldsymbol{z}_{tk} = 1|\boldsymbol{z}_{t-1,j} = 1)$ with $0 \leq A_{jk} \leq 1$ and $\sum_k A_{jk} = 1$.

This leads to the conditional distribution

$$p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1,\boldsymbol{A}}) = \prod_{k=1}^{K}\prod_{j=1}^{K} A_{jk}^{z_{t-1,j}z_{tk}} \tag{3.29}$$

The starting node for the model, $\boldsymbol{z}_1$ has no previous node and has a marginal distribution, $p(\boldsymbol{z}_1)$ represented by a vector of probabilities π with elements $\pi_k \equiv p(z_{1k} = 1)$. This gives

$$p(\boldsymbol{z}_1|\pi) = \prod_{k=1}^{K} \pi_k^{z_{1k}} \tag{3.30}$$

with $\sum_{k=1}^{K} \pi_k = 1$

The next step is to define the conditional distribution of the observed variables $p(\boldsymbol{x_t}|\boldsymbol{z}_t, \phi)$, with ϕ a set of parameters controlling the distribution. These are known as the *emission probabilities*. Given that $\boldsymbol{x}_t$ is observed, the distribution $p(\boldsymbol{x_t}|\boldsymbol{z}_t, \phi)$ is, for a given value of ϕ, a vector of K values corresponding to the K possible states of the binary vector $\boldsymbol{z}_t$ This allows the emission probabilities to be written as

$$p(\boldsymbol{x}_t|\boldsymbol{z}_t, \phi) = \prod_{k=1}^{K} p(\boldsymbol{x}_t|\phi_k)^{z_{tk}} \tag{3.31}$$

Using Eqns 3.29 to 3.31 allows the joint probability distribution over both latent variables ,$\boldsymbol{Z}$ and observed variables, $\boldsymbol{X}$ to be given by

$$p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta}) = p(\boldsymbol{z}_1|\pi)\left[\prod_{t=2}^{N} p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1,\boldsymbol{A}})\right]\prod_{m=1}^{N} p(\boldsymbol{x}_m|\boldsymbol{z}_m, \phi) \tag{3.32}$$

with $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$, and $\boldsymbol{\theta} = \left\{\pi, \boldsymbol{A}, \phi\right\}$ being the parameters governing the model.

3.3.1 Parameter estimates using maximum likelihood

In general, given an observed data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, the model parameters can be determined by the maximum likelihood method. The required likelihood function can be obtained from Eqn 3.32 by marginalising over the latent variables, $\boldsymbol{z}$, using

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{Z} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \tag{3.33}$$

A difficulty with Eqn 3.33 is that it does not factorise over t. The more usual approach used to determine values for the emission and transition matrices would be to use a variant of the EM algorithm proposed by Dempster et al. (1977), frequently referred to in the literature as the Baum Welch algorithm. This algorithm starts with an initial set of parameters, $\boldsymbol{\theta}^{old}$. The E step of the algorithm uses these initial parameters to find the posterior distribution of the latent variables, $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old})$. The posterior distribution is then used to calculate the expectation of the logarithm of the complete-data likelihood as a function of the parameters $\boldsymbol{\theta}$ giving the function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ as

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old}) \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \tag{3.34}$$

The EM algorithm is first used to develop an expression for the quantities π_k and A_{jk} in terms of the marginal posterior distribution of a latent variable,$\boldsymbol{z}_t$ designated by

$$\gamma(\boldsymbol{z}_t) = p(\boldsymbol{z}_t|\boldsymbol{X}, \boldsymbol{\theta}^{old}) \tag{3.35}$$

and an expression for the joint probability of two steps of a latent variable given by

$$\xi(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t) = p(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t|\boldsymbol{X}, \boldsymbol{\theta}^{old}) \tag{3.36}$$

this leads to values for π_k and A_{jk} as

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})} \tag{3.37}$$

$$A_{jk} = \frac{\sum_{t=2}^{N} \xi(z_{t-1,j}, z_{tk})}{\sum_{l=1}^{K} \sum_{t=2}^{N} \xi(z_{t-1,j}, z_{tl})} \tag{3.38}$$

Having obtained values for π_k and A_{jk}, a further algorithm known as the *forward-backward* algorithm (Rabiner, 1989) or Baum Welch algorithm (Baum, 1972) is used to evaluate $\gamma(\boldsymbol{z}_t)$ and $\xi(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t)$.

However, given that *labelled* training data $\{x_t, z_t\}$ for $t = 1, \ldots, N$ is available, training for the HMM can be carried out assuming the observed clinical values can be described with a multi-variate Gaussian distribution. The maximum likelihood parameters, μ_j and Σ_j for each state $j = 1, \ldots, K$, are the calculated sample means and covariances of the observed signals x_t at each state. Estimates for the transition probabilities, can be carried out using

$$p(z_t = j | z_{t-1} = k) = \frac{n_{j,k}}{\sum_{k=1}^{N} n_{j,k}} \tag{3.39}$$

where $n_{j,k}$ is the number of transitions from state j to state k in the training data.

To use the HMM in the clinical setting, as each new vector of information arrives from the ICU monitors, the task is to infer z_t. This calculation task must consider the possibility of the observations having been in each state at every time point. A useful property of HMMs (making them attractive in online-prediction settings) is that only the current hidden state z_t, the current parameter estimates of the transition matrix, and the parameters (mean, variance) of $p(x_t|z_t)$ must be stored. Assuming initial conditions of $p(z_0 = j), j = 1, \ldots, K$, inference proceeds as

$$p(z_t = i | x_{1:t}) \propto p(x_t | z_t = j) \; p(z_t = i | z_{t-1} = j) p(z_{t-1} = j | x_{1:t-1}) \tag{3.40}$$

normalised such that $\sum_{j=1}^{N} p(z_t = j | x_{1:t}) = 1$.

The above calculations assume that the training and test data can be labelled into one of the four previously described states. This could be particularly difficult for the “hypotensive risk” state.

3.3.2 Extensions to the Hidden Markov Model

There are assumptions which make this modelling technique difficult to implement. For example, the assumptions regarding the number of states within a hidden variable and the number of observations, N, used during the minute-by-minute calculations performed during real-time operation. The model can be computationally expensive and various extensions have been proposed. Ghahramani (2001) details three extensions: Factorial HMMs; Tree Structured HMMs; and Switching state space models. However even these extensions can often prove intractable and approximations may be required.

Above all, this is not an easy model to explain to clinical teams who prefer to have at least a minimal grasp on the mathematics behind a technique. However, a recent paper by Singh et al. (2010), using the HMM approach reports good results in providing early warning of hypotension albeit using the Physio Net (2009) definition of hypotension which is markedly different to the EUSIG definition. This group also took the approach of training only two HMMs, one for the unstable state and the other for the stable state of the patient. This is in contrast to the four states mentioned above.

3.4 Classifiers

A third approach, a classifier model, like the HMM also takes advantage of the fact that the available data can be labelled to produce distinct classes of information.

The three most common techniques are: logistic regression; support vector machines; and neural networks. The first two are examples of linear models, the neural network is a non linear technique.

Logistic regression is a well established technique that is readily accepted by the medical community. The technique is an extension of the well known linear regression procedure. The extension allows for a binary response and non-normal residual values. This technique will be examined in detail in Section 3.4.1.

Support vector machines are a more recent technique (1995) which attempt to find a separating hyperplane between classes. Vapnik, working in the Soviet Union, originally proposed the technique in 1965 to deal with linearly separable classes (Vapnik, 1982). The method in common use today is able to deal with non-linearly separable classes (Cortes and Vapnik, 1995). The support vector machine is a powerful technique and an attractive feature of the method is that it only requires data around the boundary area of a group of observations. The support vector machine is certainly worthy of further study however, due to time constraints, it will not be examined further in this thesis.

The most flexible approach to classification is to use a neural network. A key component of a neural network, the perceptron, was first proposed in the 1960s by Rosenblatt (Rosenblatt, 1962) and considerable research was carried out during the first wave of enthusiasm regarding “artificial intelligence”. However, limited computing power at the time and concerns published by Minsky and Papert (Minsky and Papert, 1969) resulted in the technique dropping out of favour and it became difficult to fund research in this area. Renewed interest within the machine learning community started again in the 1980s with the publication of details regarding the back propagation algorithm (Rumelhart et al., 1986). The neural network technique is considered in detail in Section 3.4.2.

Examples of using classifiers are to be found in the AvertIT (2008) project in TBI, and the work by Christensen et al. (2012) regarding the classification of EEG data. The

AvertIT project is cited (Singh et al., 2010; Jousset et al., 2009; Stell et al., 2009) as the current state of the art with respect to attempts to predict hypotensive events and will be discussed further in Section 8 as it is a project which also used a sophisticated Bayesian Artificial Neural Network (BANN) model.

3.4.1 Logistic Regression

The logistic regression model was first proposed by Berkson (1944) and is one type of generalized linear model, a framework developed in the 1970s by Nelder and Wedderburn (1972). It has been used extensively in the medical literature, (Bland and Altman, 2000). An attractive feature of this model is that the interpretation of the coefficients of the model can be easily converted to an odds ratio, a concept with which most clinicians are familiar.

A comprehensive treatment of the model is provided by Harrell (2001, Chapter 10) and Hastie et al. (2009, Chapter 4). The following description follows the text of Zumel (2011).

The logistic regression model assumes that two classes define a binary response variable Y. For this research the classes are coded 1 for "hypotensive episode started" and 0 for the "stable condition". The model also uses the concept of "odds" where the odds of an event occurring is defined as

$$\frac{P(event\,occurs)}{P(event\,does\,not\,occur)} \tag{3.41}$$

with $P(event\,does\,not\,occur)$ being equal to $1 - P(event\,occurs)$.

For a series of observations indexed by $i, i = 1 \ldots N_{obs}$, the model assumes that the log odds of a binary response variable $\boldsymbol{Y}$ with a vector $\boldsymbol{X}$ of J predictor variables , $j = 1 \ldots J$ is a linear function of the J input variables which, for the i^{th} observation, is given by,

$$\log\left(\frac{P(y_i)}{1 - P(y_i)}\right) = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} \tag{3.42}$$

with x_j being a component of $\boldsymbol{X}$ and $P(y_i)$ defined as the probability that the response variable Y for the i^{th} observation is equal to 1 (by convention called "success"). The left hand side of Eqn 3.42 is called the logit transform of $P(Y)$.

In Eqn 3.42 the intercept term β_0 can be included in the main summation by setting $x_0 = 1$ giving $J + 1$ terms as follows

$$\log\left(\frac{P(y_i)}{1-P(y_i)}\right) = \sum_{j=0}^{J} \beta_j x_{ij} \tag{3.43}$$

Consider taking the exponent of both sides of Eqn 3.43 to give

$$\frac{P(y_i)}{1-P(y_i)} = exp\left(\sum_{j=0}^{J} \beta_j x_{ij}\right) \tag{3.44}$$

As the right hand side of Eqn 3.44 is the exponential of a sum, this can be rewritten as a product of exponentials to give

$$\frac{P(y_i)}{1-P(y_i)} = \prod_{j=0}^{J} exp\left(\beta_j x_{ij}\right) \tag{3.45}$$

Eqn 3.45 shows that logistic models are multiplicative in their inputs (compared with additive as in a standard linear model). This also provides a way to interpret the coefficients. If all variables are held constant apart from x_j, then the odds ratio for a unit increase in $\beta_j x_j$ is given by

$$\frac{\text{odds}\{y_i = 1|x_{i1}, x_{i2}, \ldots, (x_{ij}+1)\}}{\text{odds}\{y_i = 1|x_{i1}, x_{i2}, \ldots, x_{ij}\}} = \exp(\beta_j) \tag{3.46}$$

Therefore, from Eqn 3.46, a simple exponentiation of a predictor's coefficient provides an estimate of the multiplicative increase or decrease in odds ratio. Despite this simplicity, the odds ratio concept must be used with caution and several authors have noted inconsistencies its use, (Moss et al., 2003; Davies et al., 1998).

Taking the inverse of Eqn 3.43 and using the notation $P_i = P(y_i = 1)$ gives a direct expression for P_i.

$$P_i = \frac{exp(\sum_{j=0}^{J} \beta_j x_{ij})}{1 + exp(\sum_{j=0}^{J} \beta_j x_{ij})} \tag{3.47}$$

the right hand side of Eqn 3.47 is called the logistic function of $\sum_{j=0}^{J} \beta_j x_{ij}$.

The standard method of calculating the values for the $\boldsymbol{\beta}$ coefficients is to use the maximum likelihood approach. To proceed, it is required to develop the likelihood function for the model. Consider the likelihood function with data observations = X. Because this research only uses two classes, the likelihood, $L(X|\beta)$, is binomial and for the y_i case is given by

$$L(X|\beta) = P(x_i)^{y_i} \times (1 - P(x_i))^{(1-y_i)} \tag{3.48}$$

with $y_i = 1$ where a hypotensive episode started *event horizon* minutes after the x_i measurements were recorded and $y_i = 0$ for the situation where *event horizon* minutes after the x_i measurements were taken the patient was still in a stable condition. For this research *event horizon* is set to 10 (see Section 4.3).

Using Eqn 3.47 and Eqn 3.48 gives the likelihood for the i^{th} row of observations from the ICU monitors

$$L(x_i|\beta_i) = \left(\frac{exp(\sum_{j=0}^{J}\beta_j x_{ij})}{1+exp(\sum_{j=0}^{J}\beta_j x_{ij})}\right)^{y_i} \times \left(1 - \left(\frac{exp(\sum_{j=0}^{J}\beta_j x_{ij})}{1+exp(\sum_{j=0}^{J}\beta_j x_{ij})}\right)\right)^{(1-y_i)} \tag{3.49}$$

Assuming independence between observations in a training set containing N_{obs} rows of data this gives

$$L(X|\beta) = \prod_{i=1}^{N_{obs}} \left\{ \left(\frac{exp(\sum_{j=0}^{J}\beta_j x_{ij})}{1+exp(\sum_{j=0}^{J}\beta_j x_{ij})}\right)^{y_i} \times \left(1 - \left(\frac{exp(\sum_{j=0}^{J}\beta_j x_{ij})}{1+exp(\sum_{j=0}^{J}\beta_j x_{ij})}\right)\right)^{(1-y_i)} \right\} \tag{3.50}$$

To avoid problems with overflow and underflow during calculations it is recommended that logarithms are used and this leads to the well known *log-likelihood* form of Eqn 3.50 as

$$\mathscr{L}(X|\beta) = \sum_{\substack{i=1 \\ y_i=1}}^{N_{obs}} logP(x_i) + \sum_{\substack{i=1 \\ y_i=0}}^{N_{obs}} log(1 - P(x_i)) \tag{3.51}$$

In order to find the maximum likelihood estimates, it is required to take the derivative of Eqn 3.51 with respect to the coefficients β and then set this to zero to find the maximum. This will require an expression for $P(x_i)'$ in terms of $P(x_i)$ which, starting from Eqn 3.47, is given by

$$\begin{aligned}
P(x_i) &= \frac{exp(z)}{1+exp(z)} \\
&= exp(z) \times (1+exp(z))^{-1} \\
&\text{Using the chain rule gives} \\
P(x_i)' &= (exp(z)) \cdot (1+exp(z))^{-1} \\
&+ (exp(z)) \cdot (-1) \cdot (1+exp(z))^{-2} \cdot (exp(z)) \\
&= \frac{exp(z)(1+exp(z))}{(1+exp(z))^2} - \frac{(exp(z))^2}{(1+exp(z))^2} \\
&= \frac{exp(z)}{(1+exp(z))^2} \\
&= \frac{exp(z)}{1+exp(z)} \cdot \frac{1}{1+exp(z)} \\
&= P(z)(1-P(z)) \qquad (3.52)
\end{aligned}$$

with $z = \sum_{j=0}^{J} \beta_j x_{ij}$

The derivative of Eqn 3.51 can then be obtained using the following two steps

$$\frac{d\mathscr{L}(X|\beta)}{d\beta} = \sum_{\substack{i=1 \\ y_i=1}}^{N_{obs}} \frac{P(x_i)'}{P(x_i)} x_i + \sum_{\substack{i=1 \\ y_i=0}}^{N_{obs}} \frac{P(x_i)'}{1-P(x_i)} x_i \qquad (3.53)$$

Using Eqn 3.52 in Eqn 3.53 gives

$$\begin{aligned}
\frac{d\mathscr{L}(X|\beta)}{d\beta} &= \sum_{\substack{i=1 \\ y_i=1}}^{N_{obs}} \frac{P(x_i)(1-P(x_i))}{P(x_i)} x_i + \sum_{\substack{i=1 \\ y_i=0}}^{N_{obs}} \frac{P(x_i)(1-P(x_i))}{1-P(x_i)} x_i \\
&= \sum_{\substack{i=1 \\ y_i=1}}^{N_{obs}} (1-P(x_i))x_i - \sum_{\substack{i=1 \\ y_i=1}}^{N_{obs}} P(x_i)x_i \\
&= \sum_{i=1}^{N_{obs}} [y_i(1-P(x_i)) - (1-y_i)(P(x_i))]x_i \qquad (3.54)
\end{aligned}$$

Cancelling terms and setting Eqn 3.54 to zero gives a maximum at

$$\sum_{i=1}^{N_{obs}} y_i x_i - P(x_i)x_i = 0 \qquad (3.55)$$

This can be written in matrix notatation as

$$\boldsymbol{X}(\boldsymbol{y}-\boldsymbol{P})=0 \tag{3.56}$$

To obtain the estimates for the coefficients $\boldsymbol{\beta}$ the iteratively re-weighted least squares (IRLS) algorithm is used. This algorithm typically uses an iterative Fisher scoring method which is a generalised version of a Newton-Raphson procedure. To illustrate this algorithm consider a Newton-Raphson procedure which takes a vector valued function $\boldsymbol{y}=\boldsymbol{f}(\boldsymbol{b})$ and determines an optimal value, $\boldsymbol{b}_{opt}$. such that $\boldsymbol{f}(\boldsymbol{b}_{opt})=0$. The algorithm starts with an initial guess $\boldsymbol{b}_0$ and constructs a Taylor expansion of $\boldsymbol{f}(\boldsymbol{b}_0)$ as

$$\boldsymbol{f}(\boldsymbol{b}_0+\boldsymbol{\Delta})\approx\boldsymbol{f}(\boldsymbol{b}_0)+\boldsymbol{f}'(\boldsymbol{b}_0)\boldsymbol{\Delta} \tag{3.57}$$

As $\boldsymbol{f}$ is a vector valued function $\boldsymbol{f}'$ is a matrix. This matrix is known as the *Jacobean* and is defined as the matrix of first derivatives of $\boldsymbol{f}$ with respect to $\boldsymbol{b}$. Solving for $\boldsymbol{\Delta}$ by setting the left hand side of Eqn 3.57 to zero gives

$$\boldsymbol{\Delta}_0=-\boldsymbol{f}(\boldsymbol{b}_0)\cdot[\boldsymbol{f}'(\boldsymbol{b}_0)]^{-1} \tag{3.58}$$

The algorithm proceeds by updating each estimate $b_n=b_{n-1}+\Delta_{n-1}$ until convergence.

For this research, the vector valued function $\boldsymbol{f}$ is the derivative of the log-likelihood, Eqn 3.55. The Jacobean matrix required for Eqn 3.57 is therefore the *Hessian* matrix of the log-likelihood function, Eqn 3.51, where a Hessian is defined as the matrix of second partial derivatives with respect to the coefficients $\boldsymbol{\beta}$.

An expression for the Hessian matrix is

$$\begin{aligned} H &= \frac{\partial}{\partial\beta}\frac{d}{d\beta}\mathscr{L} \\ &= -\sum_{i=1}^{N_{obs}} x_i \frac{d}{d\beta} P_i \\ &= -\sum_{i=1}^{N_{obs}} x_i P_i (1-P_i) x_i^T \end{aligned}$$

which written in matrix notation becomes,

$$= \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^T \tag{3.59}$$

Where $\boldsymbol{W}$ is the diagonal matrix of derivatives $\boldsymbol{P}_i'$.

Using Eqn 3.59 in Eqn 3.58 gives an expression for each Newton-Raphson update as

$$\boldsymbol{\Delta}_n = (\boldsymbol{X}\boldsymbol{W}_n\boldsymbol{X}^T)^{-1}\boldsymbol{X}(\boldsymbol{y}-\boldsymbol{P}_n) \tag{3.60}$$

with $\boldsymbol{W}$ being the current matrix of derivatives, $\boldsymbol{y}$ the response vector, and $\boldsymbol{P}_n$ the vector of probabilities calculated using the current estimate of $\boldsymbol{\beta}$.

By comparing Eqn 3.60 with the standard matrix solution to a linear regression model (Hastie et al., 2009, §3.2) which is given by

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{X}^T\boldsymbol{\beta} \\ \boldsymbol{X}\boldsymbol{y} &= \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{\beta} \\ \boldsymbol{\beta} &= (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{y} \end{aligned} \tag{3.61}$$

it can be seen that each iteration of the IRLS algorithm gives a weighted, by $\boldsymbol{W}_n$, solution to a least squares problem where the usual response vector $\boldsymbol{y}$ is replaced by a response, $\boldsymbol{y}-\boldsymbol{P}_n$, i.e. the difference between the observed response and its current estimated probability of being true.

3.4.1.1 Penalised Logistic Regression

The derivation of the logistic regression technique presented in Section 3.4.1, in particular Eqn 3.60 for the estimation of the model's parameters, show that problems can occur if inputs to the model are highly correlated. Correlated inputs can lead to a Hessian matrix which is ill-conditioned or possibly even singular if the two variables within the data are collinear. A technique to reduce these effects is called *penalised logistic regression.* A further benefit of penalised logistic regression is that it can model linearly separated data, for which standard logistic regression diverges.

Penalised logistic regression adds a penalty term proportional to the some combination of the coefficients of the model. There are many different penalty schemes, popular schemes are a ridge penalty (Hoerl and Kennard, 1970) or LASSO penalty (Tibshirani, 1996). The ridge penalty technique shrinks the regression coefficients towards zero, by constraining the sums of squares of the model coefficients (the L_2 norm). The LASSO (least absolute shrinkage and selection operator) constrains the model parameters by using the L_1 norm of the coefficients. A useful introduction to several forms of penalised logistic regression can be found in Krol (2013).

For this research, a penalty term was constructed by taking the squared second order difference between coefficients. This is defined as a centre point at coefficient β_j and calculates a forward difference $\delta_f = \beta_{j+1} - \beta_j$ and a backward difference $\delta_b = \beta_j - \beta_{j-1}$ the penalty term is then defined as the squared difference between these two points i.e. $(\delta_f - \delta_b)^2$, the coefficient λ controls the amount of penalty applied to the system.

$$\mathscr{L}(\boldsymbol{\beta})_{penalised} = \mathscr{L}(\boldsymbol{\beta}) - \lambda \sum_{j=3}^{J} (\delta_f - \delta_b)^2 \tag{3.62}$$

Eqn 3.62 can be reformulated as

$$\mathscr{L}(\boldsymbol{\beta})_{penalised} = \mathscr{L}(\boldsymbol{\beta}) - \lambda(\boldsymbol{D}\boldsymbol{\beta}) \tag{3.63}$$

where the $\boldsymbol{X}$ is the usual design matrix and the $\boldsymbol{D}$ matrix form is,

$$D = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & -2 & 1 & 0 & 0 & \cdots \\ \vdots & 0 & 1 & -2 & 1 & 0 & \cdots \\ \vdots & \vdots & 0 & 1 & -2 & 1 & \cdots \\ \vdots & \vdots & \vdots & 0 & 1 & -2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & 0 & 1 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Note that the first three terms of the model: intercept; Age; Gender are not penalised and therefore the three columns of zeros are added to the left of the $\boldsymbol{D}$ matrix.

There remains the task of determining the best value for the penalty control parameter λ. This is done by maximising the log-likelihood given by Eqn 3.62. This research takes a two pass vector search approach which is discussed fully in Section 5.3.1.

The results of using the squared difference penalty described above are detailed in Section 5.3. The results of using the LASSO penalty are reported in Section 5.4.3.

3.4.2 Neural Networks

Neural networks are non-linear classifier models that can be used for complex problems. Research on these models was originally motivated by a desire to mimic the functioning of the human brain. The requirement to analyse large dimensionality problems suggested the use of an array of parallel processing units similar to the basic processing unit of the brain which is the “neuron”. The field was started by research by McCulloch and Pitts (1943) in the 1940s enabled by the emerging electronic computing machines. Their work used threshold switches to model the neuron. By the 1960s Widrow and Hoff. (1960) had developed the adaptive linear neuron which was used used in analog telephone equipment. Rosenblatt (1962) proposed the perceptron which is the basis for the current mathematical models. After initial excitement in the field a critical paper by Minsky and Papert (Minsky and Papert, 1969) resulted in a drop off in research for nearly 15 years although a key technique, that of “back propagation of errors” was proposed by Werbos (1974). It was not until the mid 1980s when Rumelhart et al. (1986) published details regarding an algorithm for back propagation that the field as it currently exists began.

The following section describes the neural network model in mathematical detail, the approach is adapted from Bishop (1995, Chp. 4) and Bishop (2006, Chp. 5). The model consists of a series of inputs which are combined to form the input to a group of processing nodes. In the general case, the output of the first group of processing nodes can form the input into a further series of processing nodes. Each group of processing nodes is know as a *hidden layer*. The last group of processing nodes output values which provide the inputs to a final set of nodes which are known as the *output units*. The processing nodes can be any function however it is typical to use logistic sigmoid, tanh or step functions. This architecture is known as a *feed forward* network. For this research a neural network is constructed using a single hidden layer of logistic sigmoid processing nodes which form the input to one logistic sigmoid output unit.

The neural network model is depicted in Figure 3.5.

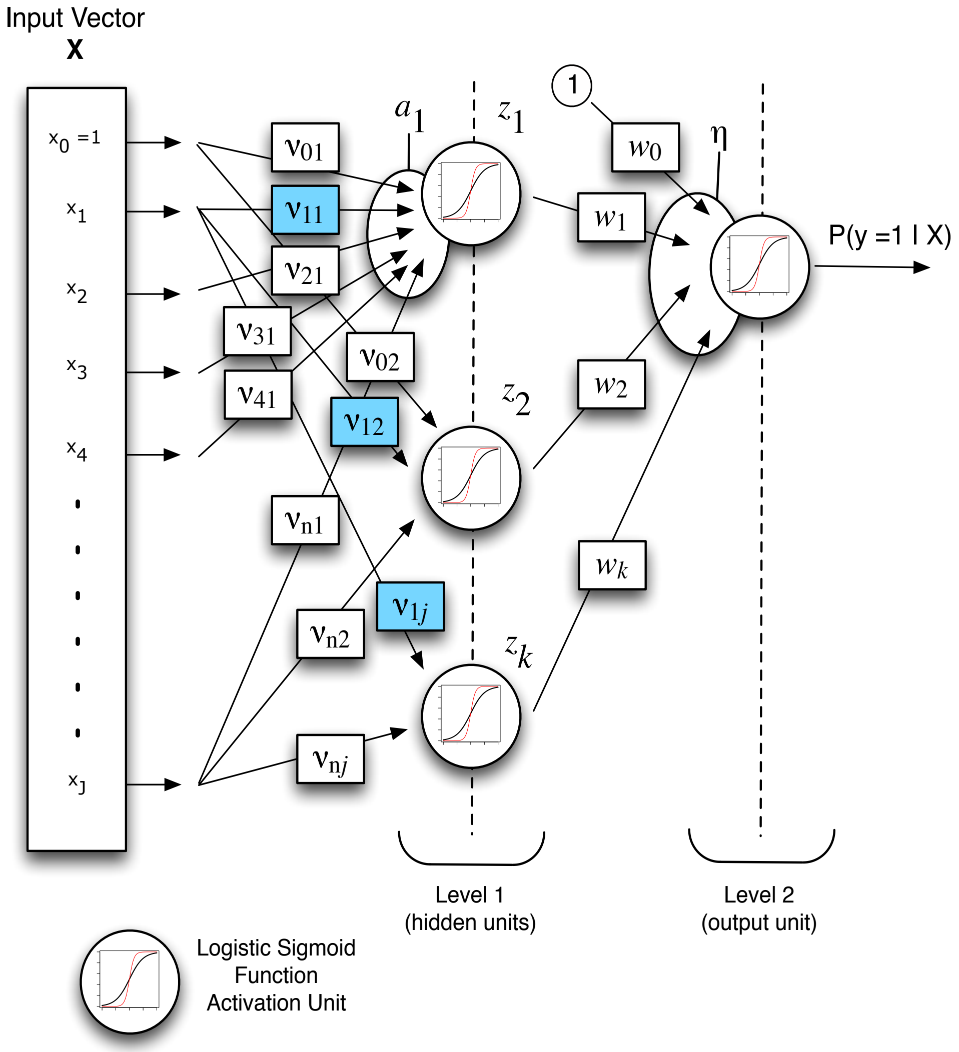


Figure 3.5: Generic single hidden layer neural network

Using the index notation from Eqn 3.42, with reference to Figure 3.5, starting on the left hand side of the diagram, the general form of the model takes a series of variables with values $x_{i1},\ldots,x_{iJ}$ for the i-th observation and constructs K linear combinations in the form

$$a_{ik} = \sum_{j=1}^{J} v_{jk} x_{ij} + v_{0k} \tag{3.64}$$

where $j = 1,\ldots,J$ are the indices of the input variables and $k = 1,\ldots,K$ are the indices of the hidden units. Note that the input x_0 is given the fixed value 1. This is known as the "bias" input and a connection is taken from this input to each linear combination, a_k known as an "activation". Each activation forms the input to a logistic sigmoid function activation unit, z_k. The series of activation units $z_1, \ldots, z_k$ form the single hidden layer. For the i^{th} observation, the output of a single activation unit is given by

$$z_{ik} = \frac{exp(a_{ik})}{1 + exp(a_{ik})} \tag{3.65}$$

In Figure 3.5, the blue weights $v_{11}, v_{12}, \ldots, v_{1k}$ are the weights for input variable x_1.

The output from the K activation units are combined to form the input, $\eta_i = \sum_{k=1}^{K} w_k z_{ik} + w_0$, to a single output unit which is again a logistic sigmoid function. This gives a final output, which is the probability of $y = 1$ given the inputs as

$$P(x_i) = P(Y_i = 1|x_i) = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)} \tag{3.66}$$

The next task is to "train" the network i.e. determine values for all the weights and biases in the model. This can be achieved using a maximum likelihood or Bayesian approach. In order to obtain a fully Bayesian model we assume that the weights $w = (v_{11}, \ldots, v_{JK}, v_{01}, \ldots, v_{0K}, w_1, \ldots w_K, w_0)$ have a normal distribution with mean 0 and variance $\sigma^2 I$ as prior distribution. Samples from the posterior distribution of the model parameters, denoted by $\theta = (\boldsymbol{w}, \sigma^2)$, are then obtained using Sequential Monte Carlo.

The loglikelihood, as a function of η_i, is identical to the likelihood used in logistic regression and given by Eqn 3.51. The difference between the neural network and logistic regression is that the neural networks assumes that the z_{ij} are logistic functions of linear combinations of the covariates, whereas logistic regression simply uses the covariates themselves as the z_{ij}'s.

For completeness, a brief overview of the maximum likelihood approach to estimating the model parameters is now given. In the likelihood framework, the first step consists of finding a weight vector $\boldsymbol{w}$ which minimises Eqn 3.51. This requires that the gradient of the negative loglikelihood be evaluated and typically, this gradient will be determined using the *error back propagation* technique (Rumelhart et al., 1986), (Bishop, 1995, §4.8). The error back propagation algorithm aids the first stage of the training of the network i.e. the calculation of the derivatives of the error function with respect to weights. The second stage of network training is to use these derivatives to adjust the weights.

The error back propagation algorithm can be summarised as

- Apply input $\boldsymbol{x}_n$ to the network and forward propagate using Equations 3.64 and 3.65 to determine the activations for all the hidden nodes and the output node.
- Evaluate $\delta_k \equiv \frac{\partial \mathscr{L}}{\partial a_j}$
- Backpropagate δs to given δ_j
- Evaluate the derivatives $h'(\cdot)$ using $\delta_j = h'(a_j) \sum\limits_k w_{kj}\delta_k$

Finally the use of regularisation is often recommended when using maximum likelihood to estimate the model parameters. This takes the form

$$\mathscr{L}_{reg}(\boldsymbol{w}) = \mathscr{L}(\boldsymbol{w}) - \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} \tag{3.67}$$

and is known as "weight decay" in the literature.

3.5 Bayesian techniques for parameter estimation

In the discussions for both the logistic regression model, Section 3.4.1 and neural network model, Section 3.4.2, equations were provided which contain parameter values as part of the model specification. Typically the parameter values are obtained by classical "frequentist" methods and provide the maximum likelihood estimation (MLE) of the parameters. An alternative approach to parameter estimation is to use Bayesian techniques.

Bayesian techniques make use of Bayes Theorem (Bayes and Price, 1763), first proposed by Thomas Bayes (1701 - 1761) but published, after his death, by his friend Richard Price. Bayesian techniques for practical problems usually require considerable numerical calculations and have only become feasible in recent times with the advent of commodity computing power.

A Bayesian analysis can be described in words as

$$Posterior \propto Likelihood \times Prior \tag{3.68}$$

A Bayesian analysis can be used to find the credible ranges (known as the highest density intervals, HDI) for the parameters of the logistic regression model. To recap from Section 3.4.1, the logistic regression model is given by

$$P(y_i) = \frac{exp(\sum_{j=0}^{J} \beta_j x_{ij})}{1 + exp(\sum_{j=0}^{J} \beta_j x_{ij})} \tag{3.69}$$

with the likelihood function for this model as

$$L(X|\beta) = \prod_{i=1}^{N_{obs}} \left\{ \left(\frac{exp(\sum_{j=0}^{J} \beta_j x_{ij})}{1 + exp(\sum_{j=0}^{J} \beta_j x_{ij})} \right)^{y_i} \times \left(1 - \left(\frac{exp(\sum_{j=0}^{J} \beta_j x_{ij})}{1 + exp(\sum_{j=0}^{J} \beta_j x_{ij})} \right) \right)^{(1-y_i)} \right\} \tag{3.70}$$

The prior term of Eqn 3.68 is constructed by the following logic. For this analysis the data will be normalised to have zero mean. This normalisation technique supports the a priori assumption that the coefficients (the βs) of the model are likely to be small and close to zero. The prior for the Bayesian analysis is therefore chosen to be a multivariate Gaussian distribution with a mean vector $\mu = 0$ and covariance matrix Σ to indicate a non-informative prior. This is written as

$$\beta \sim N_k(0, \mathbf{\Sigma}) \tag{3.71}$$

There are many possible choices for Σ. One can use a ridge prior by using a diagonal matrix for Σ. Another possibility would be to use the differencing matrix D from Section 3.4.1.1 to construct a prior covariance $\Sigma = \sigma^2(D^T D)^{-1}$. An inverse gamma distribution is used as the prior distribution for σ^2.

Eqn 3.71 gives a density function for mean vector $\mu = 0$ and covariance matrix Σ, as

$$f(\beta_1, \beta_2, \ldots \beta_k) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} exp\left(-\frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{\beta} - \boldsymbol{\mu}) \right) \tag{3.72}$$

however $\boldsymbol{\mu} = 0$ giving

$$f(\beta_1, \beta_2, \ldots \beta_k) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} exp\left(-\frac{1}{2} \boldsymbol{\beta}^T \Sigma^{-1} \boldsymbol{\beta} \right) \tag{3.73}$$

Combining Eqns 3.70 and 3.73 gives the full posterior as

$$Posterior \propto \prod_{i=1}^{n}\left\{\left(\frac{exp(\sum_{j=0}^{J}\beta_j x_{ij})}{1+exp(\sum_{j=0}^{J}\beta_j x_{ij})}\right)^{y_i}\times\left(1-\left(\frac{exp(\sum_{j=0}^{J}\beta_j x_{ij})}{1+exp(\sum_{j=0}^{J}\beta_j x_{ij})}\right)\right)^{(1-y_i)}\right\}$$
$$\times\frac{1}{\sqrt{(2\pi)^k|\Sigma|}}exp\left(-\frac{1}{2}\boldsymbol{\beta}^T\Sigma^{-1}\boldsymbol{\beta}\right) \qquad (3.74)$$

Eqn 3.74 has no closed form solution due to the logistic functions of $\boldsymbol{\beta}$ therefore a numerical technique will be required to obtained the parameter estimates.

3.5.1 Sequential Monte Carlo techniques for Bayesian analysis

As discussed at the end of the introduction to Section 3.5, (Eqn 3.74), a method is required to obtain parameter estimates from the posterior distribution. The most common method would be to use a Monte Carlo Markov chain (MCMC) algorithm (Metropolis and Ulam, 1949; Metropolis et al., 1953). The roots of the MCMC technique came from the Manhattan Project during World War II. The technique is computationally intensive, can be complex to set up and requires careful attention to avoid problems with respect to the convergence of the Markov chains. There have been many variants of the original technique with the papers by Hastings (1970) and the Gibbs sampler (Geman and Geman, 1984) being two frequently used modifications. See Johansen and Evers (2007) for a comprehensive series of lecture notes on the topic. However, the popularity of the technique is largely due to the availability of affordable computing power and several standard software packages — WinBUGS (Lunn et al., 2000) and its cross platform variants OpenBUGS (Lunn et al., 2009) and JAGS (Plummer, 2003) being the most used.

A recent alternative approach is called "Sequential Monte Carlo" (SMC) (Doucet et al., 2000). Although this approach was originally used for dynamic state space models the application of the technique to static models was recognised in the paper by Chopin (2002). The SMC method is a generalisation of the importance sampling method which also incorporates elements of the MCMC technique (Moral et al., 2006). The following description of the technique is taken from Fan et al. (2008, §3 and Appendix).

Inference is performed by obtaining samples from the joint posterior density of the parameters, $\boldsymbol{\theta}$, of the logistic regression model. Let $\pi(\boldsymbol{\theta})$ be the unnormalised density of this distribution. SMC techniques, when solving static problems, introduce a series of auxiliary distributions $\pi_0, \pi_1, \pi_2, \ldots, \pi_{S-2}, \pi_{S-1}$ to *gradually* move from the initial distribution, π_0, to $\pi_S = \pi(\boldsymbol{\theta})$ the final target distribution. At each stage $s, s = 0 \ldots S$ of the sampler algorithm a weighted sample from π_{s-1} is used to form a weighted sample from

π_s. It is the controlled manner of the transition along the series of distributions which overcomes the well known problem of particle depletion. Particle depletion in an SMC sampler is the case when a small number of particles carry all the weight in the final target distribution. A single iteration of the SMC algorithm uses weighted particles from π_{s-1} to produce particles from π_s as it proceeds through the steps of: reweighting; potentially resampling; and MCMC move. The following sections provide more detail on: the construction of π_0; the sequence of auxiliary distributions used to move from π_0 to π_S; and the individual steps of the algorithm.

3.5.1.1 Initial distribution π_0

The key element of the SMC technique is a smooth transition between auxiliary distributions. Given that a non-informative prior from the description of the Bayesian analysis (Eqn 3.71) may be too far from the target distribution, the initial distribution π_0 is constructed with a mean vector and covariance matrix obtained from the previous experiments using penalised logistic regression (See Section 3.4.1.1).

The SMC algorithm is initialised with N samples (aka “particles” hence the other term “particle filters” often found in the literature) from π_0. Let $\boldsymbol{\theta}_i^0$ be the i^{th} particle at initial stage $s = 0$ with weight $w_i^0 \equiv 1$ given to each of the N particles, thus $\{\boldsymbol{\theta}_i^0, w_i^0\}$ forms a weighted sample from π_0.

3.5.1.2 Auxiliary distributions

There are many ways that the auxiliary distributions can be constructed, this research will use the technique suggested in the paper by Fan et al. (2008) which is based on a method initially described in Moral et al. (2006). The technique uses

$$\begin{aligned} \pi_s \propto \pi_0^{1-\gamma_s} \times \pi^{\gamma_s}, \quad with \\ 0 = \gamma_0 \leq \gamma_1 \leq \ldots \leq \gamma_S = 1 \end{aligned} \tag{3.75}$$

3.5.1.3 Reweighting

With N weighted particles $\{\boldsymbol{\theta}_i^{s-1}, w_i^{s-1}\}$ from π_{s-1}, set

$$w_i^s = w_i^{s-1} \times \left(\frac{\pi_s(\boldsymbol{\theta}_i^{s-1})}{\pi_{s-1}(\boldsymbol{\theta}_i^{s-1})} \right) \tag{3.76}$$

The weights are normalised using $w_i = w_i / \sum_{j=1}^{N} w_j$. This gives $\{\boldsymbol{\theta}_i^{s-1}, w_i^s\}$ as a weighted sample from π_s

3.5.1.4 Resampling

Resampling is conditional on the value of the effective sample size (ESS). ESS is defined as

$$ESS = \frac{\left(\sum_{i=1}^{N} w_i^s\right)^2}{\sum_{i=1}^{N} \left(w_i^s\right)^2} \tag{3.77}$$

and provides an estimate of the number of independent random samples from the target distribution that are required to give an equivalent Monte Carlo variation using the N weighted particles. The aim of resampling is to discard particles with low weight and replicate particles with high weight i.e. the particles that are likely to contribute to the final estimate of the target distribution. After resampling particle weights are reset to $w_i^s \equiv 1$.

The literature suggests that if the ESS is below cN, where c is typically taken as $1/2$, then resampling should be performed. There are many resampling procedures (See Hol et al. (2006) for a review) for this research a simple residual resampling technique is used.

3.5.1.5 MCMC Move

The final part of the algorithm is to move the particles within the current distribution π_s in order to increase particle diversity. In this section two possible updates are proposed, the Metropolis-Hastings update and the Gibbs update. In the logistic regression models and the neural networks the Metropolis-Hastings update is used to update the regression coefficients (weights for the neural network) and the Gibbs sampler is used to update the prior variances.

Let $\{\boldsymbol{\theta}_i^s, w_i^s\}, i, \ldots, N$ be the samples (and weights) in this distribution after reweighting and potential resampling. As described above split $\boldsymbol{\theta}_i^s = (\boldsymbol{\beta}_i^s, \sigma_i^s)$. $\boldsymbol{\beta}_i^s$ is updated using a (symmetric) Metropolis transition kernel. For each $\boldsymbol{\beta}_i^s$ draw a new proposed value

$$\boldsymbol{\beta}_i^{s*} \sim K_s(\boldsymbol{\beta}_i^s, .) \tag{3.78}$$

and calculate the acceptance ratio

$$\alpha_i^s = \min\left\{1, \frac{\pi_s(\boldsymbol{\beta}_{\boldsymbol{i}}^{\tilde{\boldsymbol{s}}*}, \sigma_i^s)}{\pi_s(\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{s}}, \sigma_i^s)}\right\} \tag{3.79}$$

With probability α_i^s we set $\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{s}}$ to $\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{s}*}$. In this research a Gaussian transition kernel is used

$$\boldsymbol{\beta}_i^{s*} \sim N(\tilde{\boldsymbol{\beta}}_s^i, \tau\boldsymbol{\Sigma}) \tag{3.80}$$

with τ being a tuning parameter, controlling the acceptance rate. Due to high dimension of the covariate space, it is important that the matrix $\boldsymbol{\Sigma}$ is appropriately scaled. For the logistic regression models used in this thesis it is chosen proportional to the asymptotic covariance of the estimates from penalised logistic regression model. The goal is to achieve an acceptance rate of approximately 0.23 as suggested by Roberts et al. (1997). For this research τ was set to a range of values between 0.2 and 0.3 depending upon the model being used.

The prior variance $(\sigma_i^s)^2$ is updated using a Gibbs sampler, by replacing $(\sigma_i^s)^2$ with a draw from the full conditional distribution of $(\sigma_i^s)^2|\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{s}}$.

For every iteration of the SMC algorithm the Metropolis update of the regression coefficients and the Gibbs update of the prior variance of the coefficients is performed once for each particle.

In summary, the elements of the SMC technique that must be configured are

- the initial distribution π_0
- the total number of distributions S which span the transition from π_0, to $\pi_S = \pi(\boldsymbol{\theta})$
- the γ_s sequence γ_0 to γ_S used to move between the auxiliary distributions
- the transition kernel K_s used to move the particles within a single π_s distribution
- the number of particles N within a single π_s distribution

The SMC technique is used for both the penalised logistic regression (Section 5.3) and neural network models (Section 6.1). For the penalised logistic regression models each "particle" contains *all* the settings for the model coefficients. The 500 particles make up a single "s" auxiliary distribution π_s from the total of 300 distributions used. For the neural network models each "particle" contains a complete neural network consisting of the biases and weights for the inputs, hidden nodes and output node. The neural network models use 1000 particles and 400 auxiliary distributions.

3.5.2 Example SMC for Bayesian analysis

The following "toy" problem illustrates using SMC to move from a known (i.e. prior) distribution to a target distribution. Consider the need to draw samples from the exponential distribution however a technique is unavailable. It is known how to draw from a lognormal (draw from a normal and exponentiate). SMC can be used to turn the sample from the lognormal into a sample from the exponential.

In this example each particle is a number, initially drawn from the lognormal, but updated along the way. Instead of moving from the lognormal immediately to the exponential, construct a "bridge" of distributions of the form

$$f_{exponential}(x)^{\gamma} \times f_{lognormal}(x)^{1-\gamma} \tag{3.81}$$

for $\gamma = 0$, this is just the lognormal (for which it is known how to sample from), for $\gamma = 1$ this is the exponential (the target distribution). For $0 < \gamma < 1$ it some "blend" between the two distributions. Moving slowly across the "bridge" (i.e. from $\gamma = 0$ to $\gamma = 1$) between the two distributions, the algorithm does three things:

- Update the weights
- If the weights are too degenerate, resample
- MCMC move to perturb the particles

For this example, the settings were π_0 = lognormal, $S = 50$, $N = 100$. Figure 3.6 shows the movement of particles across the 50 distributions.

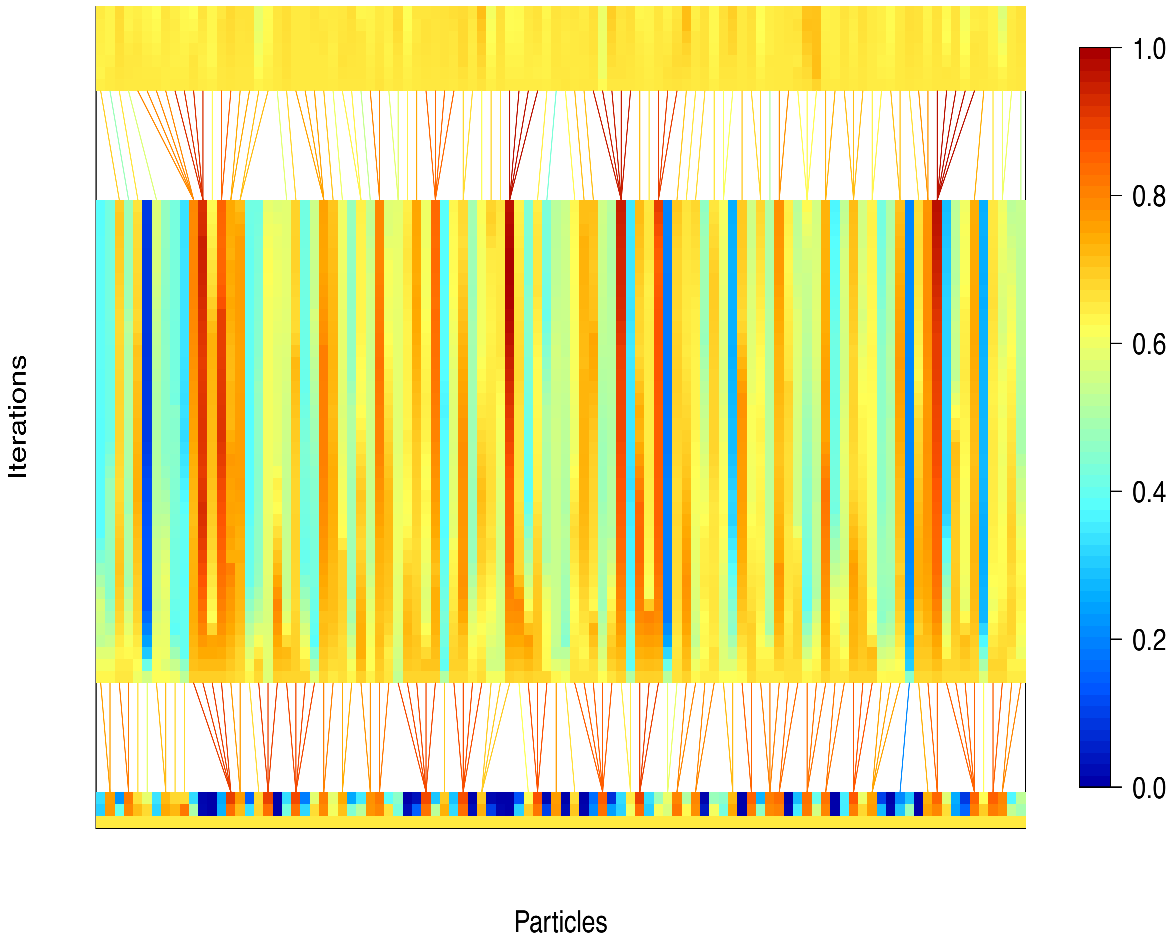
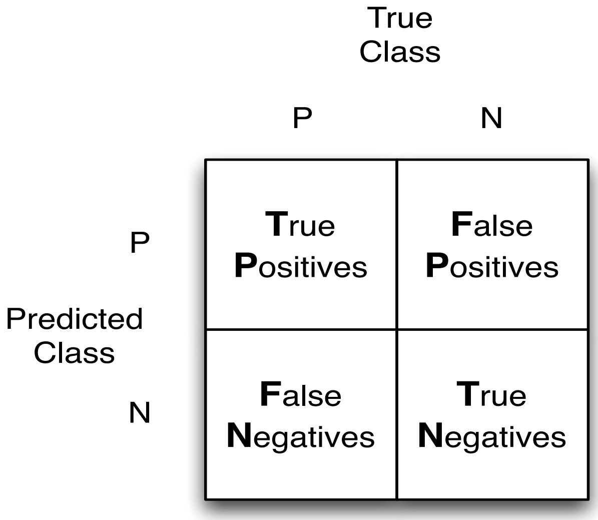


Figure 3.6: SMC, particle sampling visualisation. The initial draw from the lognormal is shown at the bottom of the plot. The colour key is a relative scale with the range being from dark blue for low weights to red for high weights. Each row of the image is a set of calculations (update of weights and MCMC move) across all 100 particles. The gaps indicate where a resample procedure occurred. The aim is to arrive at the top of the plot with an even spread of mid weight particles i.e. no particles dominating the distribution.

3.6 Assessment of classifier performance

Once the classifier has produced a probability prediction there is still a task to decide at which point to raise an alarm to the clinical team. This is ultimately the real performance that the system will be judged on. This task resolves to a trade off between the number of true positive identifications of episode starts and the number of times a false alarm is raised.

For the binary classifiers studied in this research, the following definitions apply: for each instance, I, within a test set, the classifier model maps I to one of four possible outcomes. If I is positive and the classifier predicts a positive class label, this is deemed a *True Positive (TP)*. If I is positive and the classifier predicts a negative class label, this is deemed a *False Negative (FN)*. If I is negative and the classifier predicts a negative class label, this is declared a *True Negative (TN)*. If I is negative and the classifier predicts a positive class label, this is declared a *False Positive (FP)*. These definitions are shown in Figure 3.7 known as the "Confusion Matrix" and are used in the equations 3.82 to 3.86



		True Class	
		P	N
Predicted Class	P	**T**rue **P**ositives	**F**alse **P**ositives
	N	**F**alse **N**egatives	**T**rue **N**egatives

Figure 3.7: Classifier Confusion Matrix

$$tp\ rate = \frac{TP}{TP+FN} \tag{3.82}$$

where $TP+FN$ is the column total of positives from the confusion matrix. The term $tp\ rate$ is also known as the *sensitivity*, *hit rate* or *recall.*

$$fp\ rate = \frac{FP}{FP+TN} \tag{3.83}$$

where $FP+TN$ is the column total of negatives from the confusion matrix. This term is also often called the*false alarm rate*.

$$\begin{aligned} specificity &= \frac{TN}{FP + TN} \\ &= 1 - fp\ rate \end{aligned} \tag{3.84}$$

$$precision = \frac{TP}{TP + FP} \tag{3.85}$$

$$accuracy = \frac{TP + TN}{(TP + FN) + (FP + TN)} \tag{3.86}$$

3.6.1 ROC curves

A standard visual technique to assess the trade off between true positive identifications of episode starts and false alarms is to use a receiver operating characteristic (ROC) curve. This technique was first developed in the 1940s during the development of radar detection equipment. Early academic references are attributed to Hanley and McNeil (1982, 1983), with a through treatment being available in the papers by Fawcett (2004, 2006).

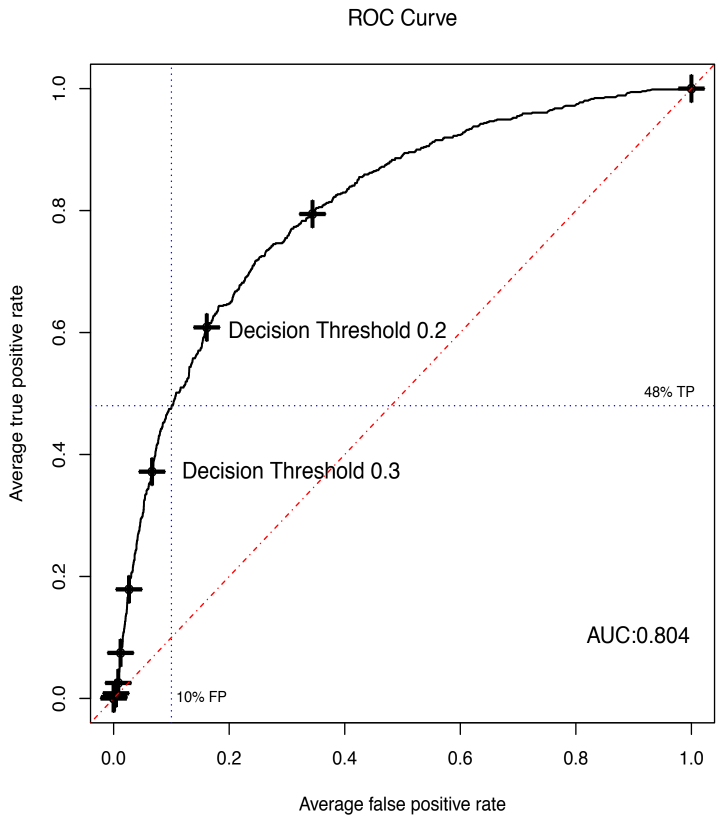


Figure 3.8: ROC Curve

The ROC curve (Figure 3.8) plots the specificity (false positive rate) on the x-axis and the sensitivity (true positive rate) on the y-axis. The cross markings on the plot indicate probability decision thresholds and move from a decision setting of equal to or greater than probability zero at the top right of the plot, which would accept all predictions from the classifier, to must be probability one at the bottom left of the plot which would reject all predictions from the classifier. The dotted blue line shows that for a decision threshold of approximately 0.26, the classifier could be expected to detect 48% of positive cases whilst producing a false positive rate of 10%. The red dash dot $y = x$ line indicates the *random guess* line. All useful classifiers must produce a curve that lies above this line. A random guess classifier could be constructed from any data set whereby the classifier randomly predicts a positive case 50% of the time. This would be expected to be correct in 50% of the cases and incorrect for the other 50% of cases. The produces a point on the ROC graph of (0.5, 0.5). If the classifier randomly predicts a positive case 90% of the time, it would identify 90% of the true positives however it would also suffer from a false positive rate of 90% giving the ROC point of (0.9, 0.9). Therefore the performance of a random guess classifier is determined by the proportion of instances it randomly guesses a positive case leading to the $y = x$ diagonal line. An attractive property of the ROC curve is that it is also invariant to class distribution i.e. the proportion of positive cases to negative cases in the test set.

3.6.2 Area under the ROC curve (AUC)

The ROC curve gives rise to a metric called the area under the ROC curve which is usually abbreviated to AUC. The AUC value of a classifier gives the probability that the classifier will score a randomly chosen positive instance higher than a randomly chosen negative instance. As the ROC graph comprises a unit square, the value of AUC must lie between 0 and 1. As described in Section 3.6.1 all useful classifiers must be above the diagonal $y = x$ and therefore all AUC values should be above 0.5. The AUC gives a convenient single measure that can be used to compare classifier models however it must be examined with care in the case where ROC curves from different classifiers cross. In this situation, Figure 3.9, classifier B has a higher AUC at 0.804 but performs poorer with respect to false positive rates below 30%. Clearly if false positive performance is the major requirement classifier A is better.

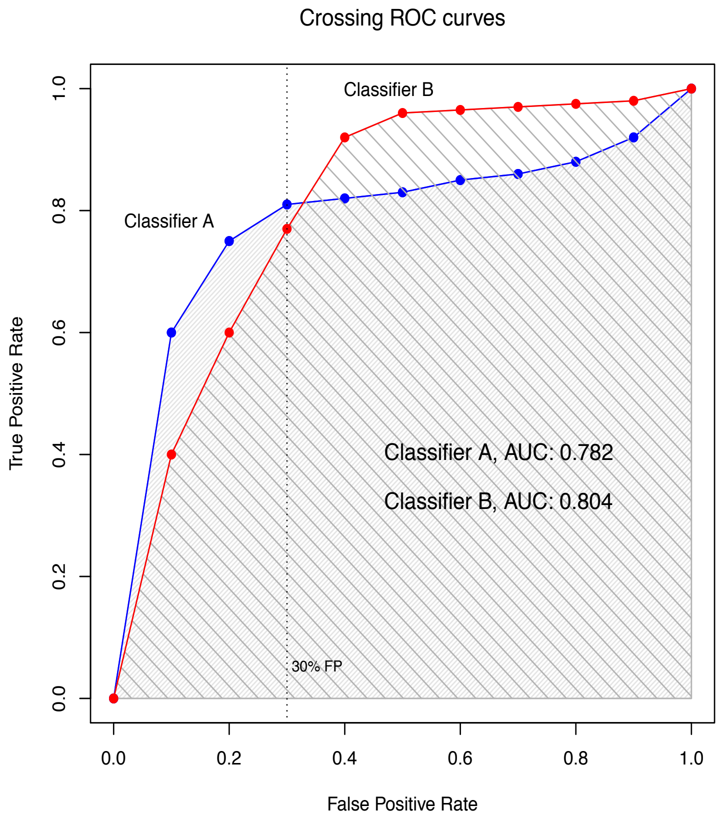


Figure 3.9: Crossing ROC curves. Judged on AUC scores alone, classifier A would be considered poorer. However, if system requirements called for a false positive rate below 30%, it is clear that classifier A is superior to classifier B.

3.6.3 H score

As prevalent as the AUC metric is in the literature on classifier performance, it is noted by researchers (Figini and Maggi, 2014; Hand, 2012) and described in Section 3.6.2, that there are problems in comparing classifier performance particularly when the ROC curves cross. Hand (2009) proposed an alternative metric, the H score, which attempts to deal with the issue of differing costs associated with class “true” vs. class “false”.

The H score assumes that a Beta distribution can be parametrised to reflect the relative costs of the two classes being investigated. The measure is then defined (Hand, 2012) as “ … the mean loss from a classifier when the distribution of relative costs is taken to be a beta($1 + \pi_1$, $1 + \pi_0$) distribution”. Hand (2009) provides an R function to calculate the H score from a series of classifier predictions and known class labels. This research has investigated the use of the H score on the logistic regression model described in Section 5.5. It can be seen from Figure 5.36 and Tables 5.9 and 5.10 that, for this research, the H score does not provide any additional insight into the classifier performance.

3.7 Approaches to input data

By using the episode list produced by the EAA application, it is possible to identify over 3000 events from 136 patients which match the EUSIG definition (see Section 4.1.3). The physiological measurements which are available each minute from the ICU monitors can be split into two classes. One class, representing an *unstable* condition, containing vectors of readings that occurred at n minutes before the event, where the n minutes is called the *event horizon* (see Figure 4.4), and another class which contains vectors of readings that represent the *stable state* of the patient, i.e. no event occurred n minutes after the readings were taken. The classifier model provides a probability estimate as to how likely the new vector of information, which arrives on a minute-by-minute basis, belongs to the unstable class, i.e. "will have an event in n minutes" where n is the event horizon used to build the training data.

For this thesis, it is assumed that the available patients have been randomly split into two independent groups. One group will be used for training the model with the other group forming a test set for model validation. There is also an assumption during model training that each row of data is independent of any other row. It is assumed in the case of the positive vectors, that each episode start is a new situation for the patient. In the case of the negative vectors, which are assumed to represent the stable state of the patient, this is more difficult as the vast majority of rows available for training are in fact negative vectors separated by a single minute. This single minute separation will often result in high correlation between negative vectors. Therefore it is required to perform a random selection of these negative vectors with a similar proportion to that found in the real ICU monitor data stream. For this research, a value of 10% positive cases to 90% negative cases has been chosen to represent the ICU data. This proportion allows for the use of *all* the positive cases along with a random selection of negative cases whilst allowing the training file to be a small enough size for model processing. This 10:90 ratio will only affect the intercept in the logistic regression modelling however, the intercept has no meaning in this particular context and can be ignored.

3.7.1 Using each minute of the data buffer

Another decision that must be made regards how to use the time series data available for each of the physiological processes. The standard approach from a statistical modelling point of view is to use as much data as possible to train the models. To use this approach, each minute of data that is available becomes an input variable into the model. If we consider a training set using an event horizon of 10 minutes, given a constraint of 30 minutes of data, this method is represented in Figure 3.10.

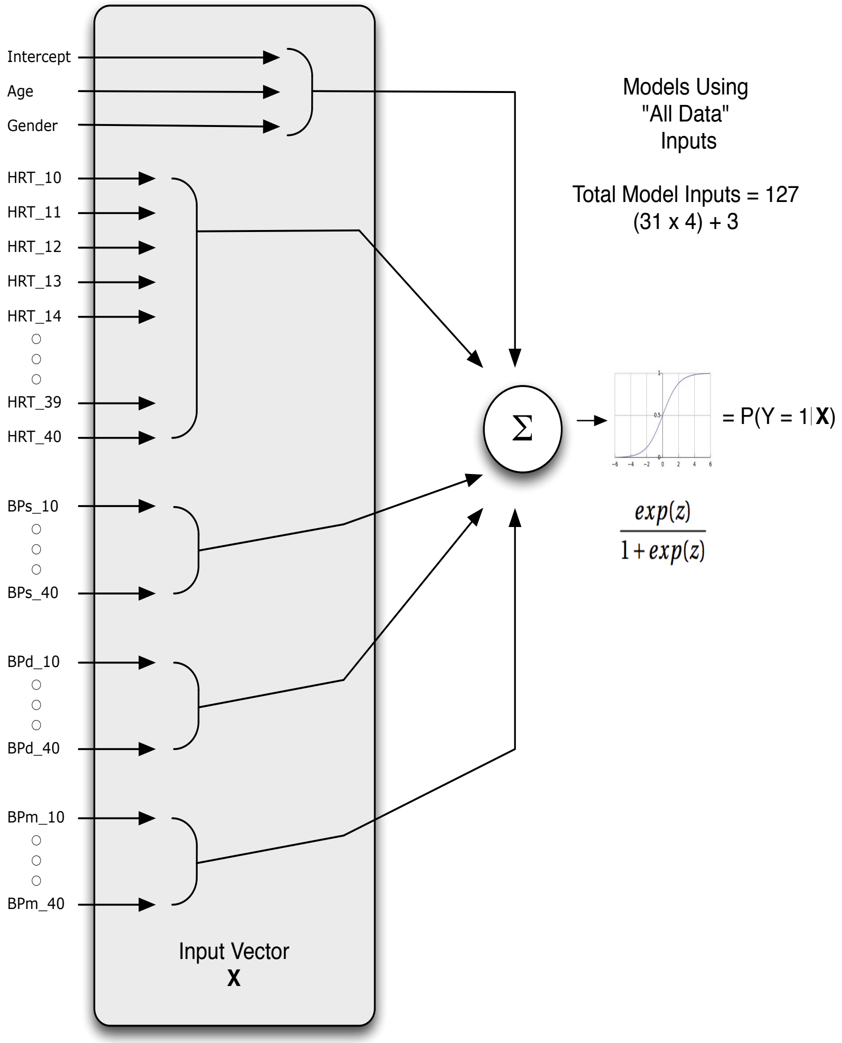


Figure 3.10: Each minute as an input to the model. Each of the 127 inputs is controlled by a model coefficient $\beta_0 \ldots \beta_{126}$. This leads to the expression for $z = \sum_{k=0}^{K} \beta_k x_k$

.

3.7.2 Using statistical measures of the data buffer

An alternative approach would be to summarise the data over suitable window sizes before the event horizon and then use these summary statistics as inputs to the model. This approach is possibly more intuitive to clinicians as they effectively carry out this process in their daily work looking at the ICU monitors. Again, using a training set with an event horizon of 10 minutes, this is shown diagrammatically in Figure 3.11.

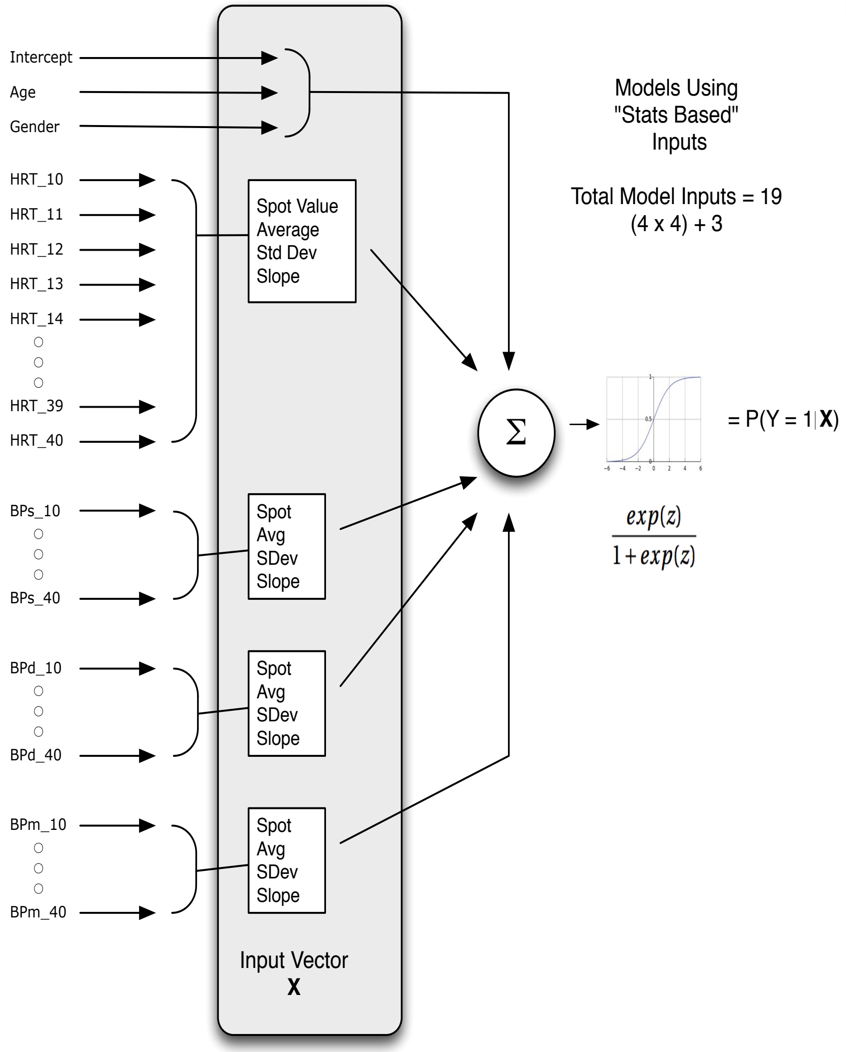


Figure 3.11: Summary statistics as input to the model. Each of the 19 inputs is controlled by a model coefficient $\beta_0 \dots \beta_{18}$. This leads to the expression for $z = \sum_{k=0}^{K} \beta_k x_k$

.

A valid criticism of both the above techniques is that the data contains multiple episodes from each patient and would be more comprehensively examined by using a mixed effects modelling approach. In this type of approach an additional term is introduced into the model to account for the variability between patients. Literature on this approach can be found in Pinheiro and Bates (2000) and Laird and Ware (1982) and the topic of mixed effects models is further discussed in Section 9.4 on future work.

Both of these input architectures are examined in Chapter 4 on data preparation and in the two Chapters 5 and 6 on model building.

3.8 Using the model in practice

The aim of this thesis is to provide further evidence to the community of TBI researchers and clinicians that, using statistical techniques, it *is* possible to develop a system which provides warning of hypotensive episodes. It is only recently, with the publication of results from the AvertIT project which has shown early warning of hypotension in approximately 40% of cases, (Donald et al., 2012b; Stell et al., 2012), that TBI practitioners have been willing to engage with statistical modelling researchers to better understand this area. The clinical community recognises the need for a monitoring system, however, they did not think that it was possible to provide any useful warning using software and statistics. What has been shown by the AvertIT group, is that it is possible to provide early warning, albeit with a very sophisticated technique (Bayesian Artificial Neural Network — BANN), and that this general approach, i.e. using software and statistics, is worth supporting.

However, a significant problem with the BANN approach is the difficulty in explaining what is producing the prediction in terms of the input signals, i.e. it is a black box and clinicians do not like something that they have to take on trust. The central aim of this thesis is about exploring the possibility that a simple approach, based on logistic regression models — which clinicians understand and accept — can actually perform as well, or nearly as well, as the BANN approach. This will be a significant contribution to the domain knowledge in TBI. If it turns out that logistic regression models can give useful early warning, it will show that the problem is not quite as non linear as perceived wisdom suggests. If not, then the research will have shown that more sophisticated models *are* required.

Chapter 4

Data Preparation

The data preparation for this research is extensive. This chapter describes the process of producing data sets, from the raw ICU monitoring data, which can be used to investigate ways of detecting possible hypotensive episodes. The tasks involved in this process are illustrated by the flow chart in Figure 4.1.

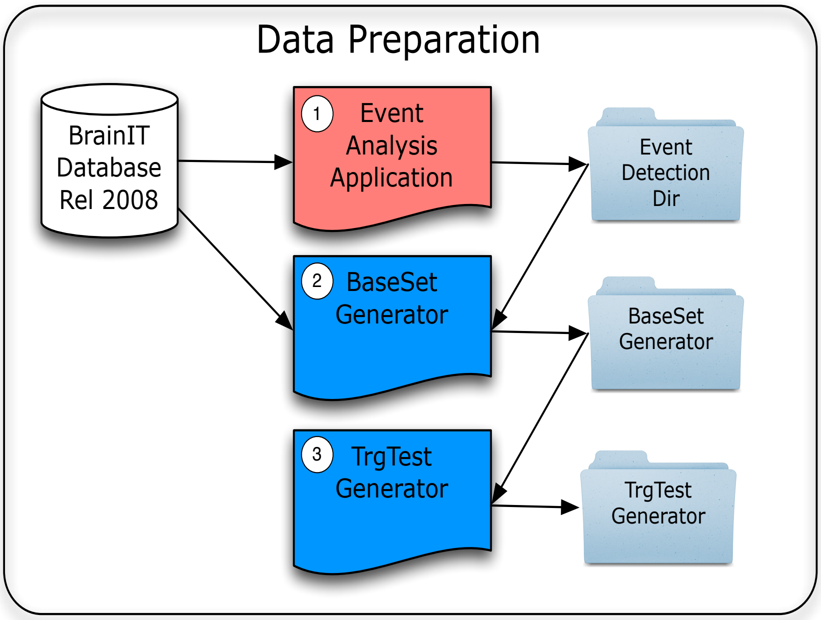


Figure 4.1: Data preparation research tasks

The first task is to produce the list of events and episodes which match the agreed definition for a hypotensive event. Section 4.1.1 provides a brief outline of the Event Analysis Application (EAA) program used to produce this list, along with the characteristics of the events and episodes obtained from the ICU data in the BrainIT database.

The second task is to produce *base data sets*. Each base data set contains minute-by-minute values from all patients. Each row constitutes a single *vector* consisting of the episode start marker and *all* the available predictor data at suitable time points before the event occurred. These vectors of information detail both the positive cases, i.e. signal values that were present just before an episode starts, and the negative cases which contain the information regarding the steady state or “normal”condition of the patient.

For this research two types of base data set are produced to support the two different input architectures used by the models (see Section 3.7). The first type supports both the “All Data” and “Stats Based” models and allows comparison between all the different model types. Each vector (i.e. row) in the file contains demographic and hospital admission data along with a value for each minute of each physiological signal from the ICU monitors. A single file called BDS_10_30_all_data.csv was produced and this file was used to produce all the training and test sets used to compare the modelling techniques. The second type of base data set used preprocessing methods to create a group of 30 files that were used to support the research into varying event horizons and window sizes described in Section 5.5.

Section 4.2.6 describes a software application, Base Set Generator (BSG), that can produce this initial list of all possible candidate vectors from the available data. A base data set typically contains approximately 1000 examples of positive vectors along with 1.4 million examples of negative vectors obtained from 136 patients.

The third and final data preparation task is to produce a series of training and test data sets randomly drawn from the base data sets. The random draw of results from the time series is used to minimise the correlation between each set of inputs, thereby allowing the use of standard statistical techniques. The Training Test Generator (TTG) software described in Section 4.3.1 performs this function ensuring that once a patient has been chosen for the training set it cannot be used in the test set. The TTG application was used on the BDS_10_30_all_data.csv base data set file to produce 50 randomly drawn training and test set pairs of files. These 100 files were used to develop and test all the models described in Chapters 5 and 6.

The following sections describe each one of the above tasks in more detail.

4.1 Event and Episode Analysis

This part of the data preparation cycle takes the definitions discussed in Chapter 2, Section 2.2 and processes the minute by minute data, building up a list of events and episodes. Due to the requirements for holddown and clear times, this task is performed by a custom application that has been specifically written for this thesis. The custom program Event Analysis Application (EAA) has been written with a view to being a more general purpose tool and is described in Section 4.1.1. Whilst the EAA produces comprehensive output, only the list of *episode* start and stop times is required for the project.

4.1.1 Event Analysis Application (EAA)

The EAA is a general purpose application, written in Java, that initially processes a configuration file to obtain the definition of (1) event trigger and clear thresholds and (2) holddown and clear times. It also reads the value for the "new episode gap" which is used to coalesce closely spaced events into a single episode. The program then runs through a finite state machine pattern for each minute by minute row of ICU data for a patient. The output of the application is a structured file in JSON format, (JSON, 2011), which contains details of all the events and episodes found that meet the definitions in the configuration file. This application is being made publicly available under an open source licence and can be obtained in due course from the web site, `http://www.statsresearch.co.uk`. Further details of obtaining and configuring the EAA can be found in Appendix A.2.

4.1.2 EUSIG hypotensive event definition

Medical opinion is not settled on the definition of hypotension, as discussed in the medical background in Section 2.2. In order to make progress, this thesis uses the Edinburgh University Secondary Insult Grades (EUSIG). The reasoning to support this decision is discussed in the work of Donald (2008), carried out during the AvertIT (2008) project. In summary, the EUSIG definitions are from published work, (Jones et al., 1994), and require the monitoring of both systolic and mean arterial blood pressure.

The EAA is configured with the EUSIG definitions (see Appendix A.2) and the application is used to process 136 patients from the BrainIT database.

4.1.3 Episode analysis of BrainIT database

The data used to illustrate the techniques described in this thesis comes from the BrainIT database. This data source is described in Section 2.3. Applying the EAA to the BrainIT database (2008 release) provides the following characteristics.

Total Patients	136					
Females	24	(17.6%)		Median age 39		Range 16 — 82
Males	112	(82.4%)		Median age 34		Range 16 — 83
Total Episodes				Total Events		
2382(ERF=76.7%)				3105		
			Type of Trauma			
Assault	Fall	Pedestrian	Sport	Traffic Accident	Unknown	Work
10	39	13	2	63	7	2

Table 4.1: Training and test cohort — demographic summary

Table 4.1 and Figure 4.2 show the demographic characteristics of the data and plots of statistical measures of the hypotensive episodes that occurred within the cohort. The Episode Reduction Factor (ERF), shows the reduction from EUSIG events to the more relevant *episodes* and it these episodes which cause the clinical teams most concern.

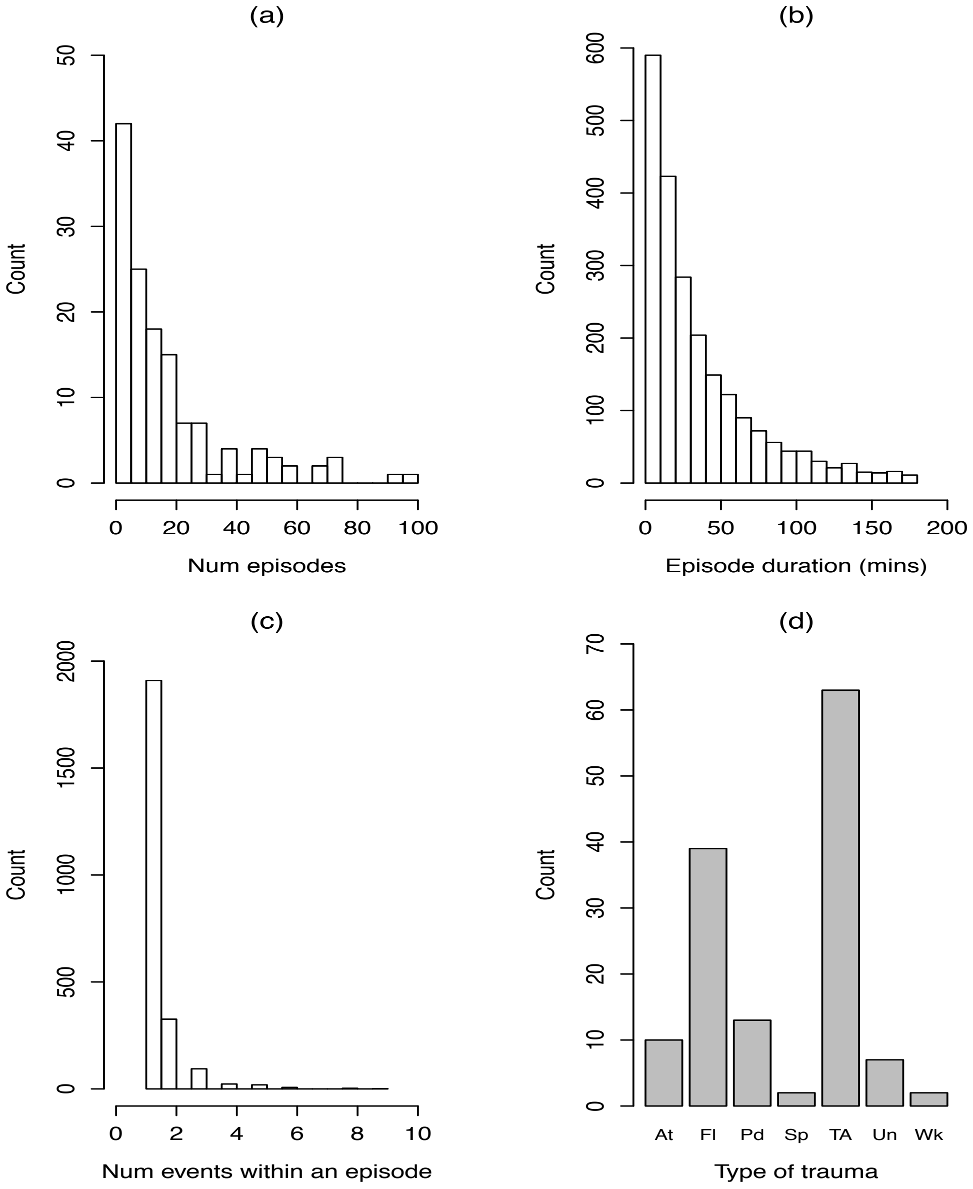


Figure 4.2: Training and test cohort summary plots: (a) histogram showing the number of patients (Count) having a given number of episodes during their stay in the ICU; (b) the spread of episode durations across the cohort, limited to durations under three hours; (c) the number of events contained within an episode; (d) types of trauma: At = Assault; Fl = Fall; Pd = Pedestrian; Sp = Sport; TA = Road Traffic Accident; Un = Unknown; Wk = Work.

4.2 Base data sets

This section describes a system that generates base data sets from a combination of patient ICU monitoring measurements, which contains possible predictor variables, and a list of episode start and stop times produced by the EAA program of Section 4.1.1. This is a general purpose requirement for building a classifier based system. The starting point for any such system is a collection of labelled data which details the state of the predictor variables at a given point in time, and the class label to which these measurements belong. In the case of this project, we are dealing with binary classification, i.e. the class choice is (a) did these measurements precede an episode start by n minutes, where n is the *event horizon* or (b) was the patient in a stable state n minutes after these readings were taken. Class (a) is given the value 1 and class (b) is given the value 0.

4.2.1 Base data sets for hypotension prediction

The central aim of this project is to produce an early warning of a hypotensive episode that is clinically useful. What is considered *clinically useful* is, of course, a subjective view of a clinical team. This thesis takes as its starting point the values of approximately 10 minutes of warning with a sensitivity of at least 30% and a false positive rate below 10%. These values come from a paper by the AvertIT team, (Stell et al., 2012). Discussion with clinical colleagues suggests this low sensitivity target is still acceptable because at the moment there are no clinically available systems which give early warning of these dangerous episodes.

Another constraint imposed on a useful system is that the system does not require large amounts of past data to perform its calculations. This constraint is to ensure that once an event has occurred, and then cleared, the system does not need to wait too long before it is operating on "clear" data. These two constraints are further described below with reference to Fig. 4.3.

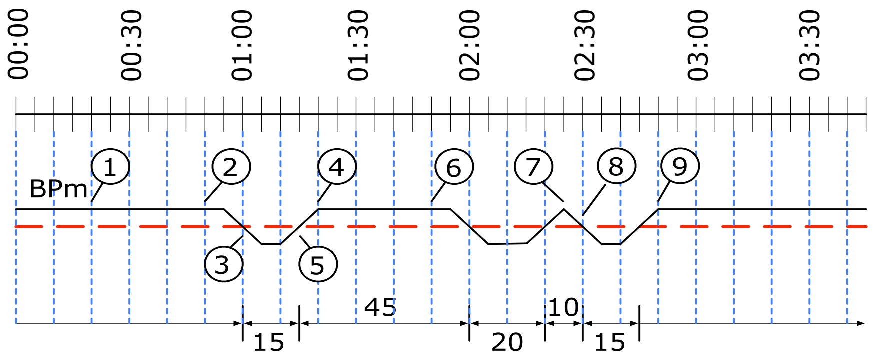


Figure 4.3: Event run

From Figure 4.3, consider point 3 where the event actually occurred. The last time to usefully predict the event is at point 2. Assuming a model runs on a 10 minute event horizon, the predictions at point 2 could be using the 30 minutes of data from point 1 up to point 2.

The event clears at point 4, although the clearance signal does not occur until point 5. At point 5, assuming a minimum window size of five minutes, the model can restart. Note however, it is not until point 6 that enough data is collected to run models which require a 30 minute window size.

The 10 minute gap between the second and third events is an example of a situation where there is limited ability to give an early warning of the third event due to lack of data. Point 7 is the first time that models using a window size of five minutes can be restarted. However, the best the model can do is give an indication of the probability of an event which may occur in the next 10 minutes. The model, therfore may not accurately forecast the event which occurs at point 8. It is not until point 9 that predictions can be restarted.

4.2.2 Characterising physiological time series measurements

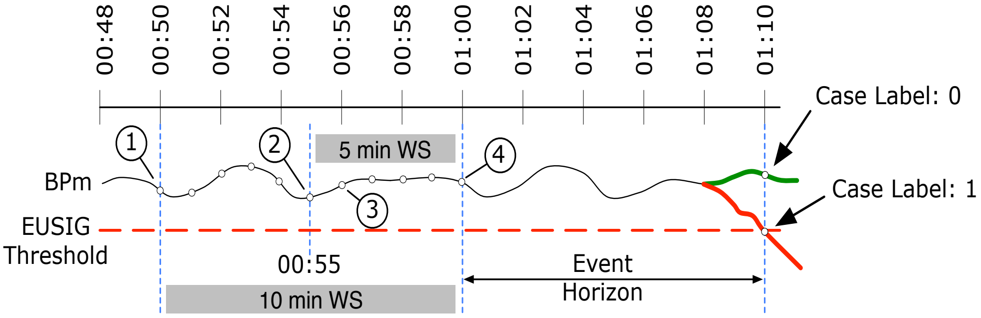


Figure 4.4: Event horizon and window size

This section defines terms used in subsequent sections which describe how the time series signals are processed to provide values for statistical measures concerning the signal. Figure 4.4 defines the concept of event horizon and window size (WS).

The period of n minutes before an event is known as the event horizon (EH). From point 4, a set of measurements can subsequently become an event n minutes later (the red trace) or represent the stable state of the patient (the green trace).

The information that is being assessed is contained in the window size (WS) collection of measurement values. In the case of Figure 4.4, two window sizes are shown;

measurements from point 1 to 4 constitute a 10 minute window size, measurements from point 2 to 4 constitute a five minute window size.

4.2.3 Statistical measures

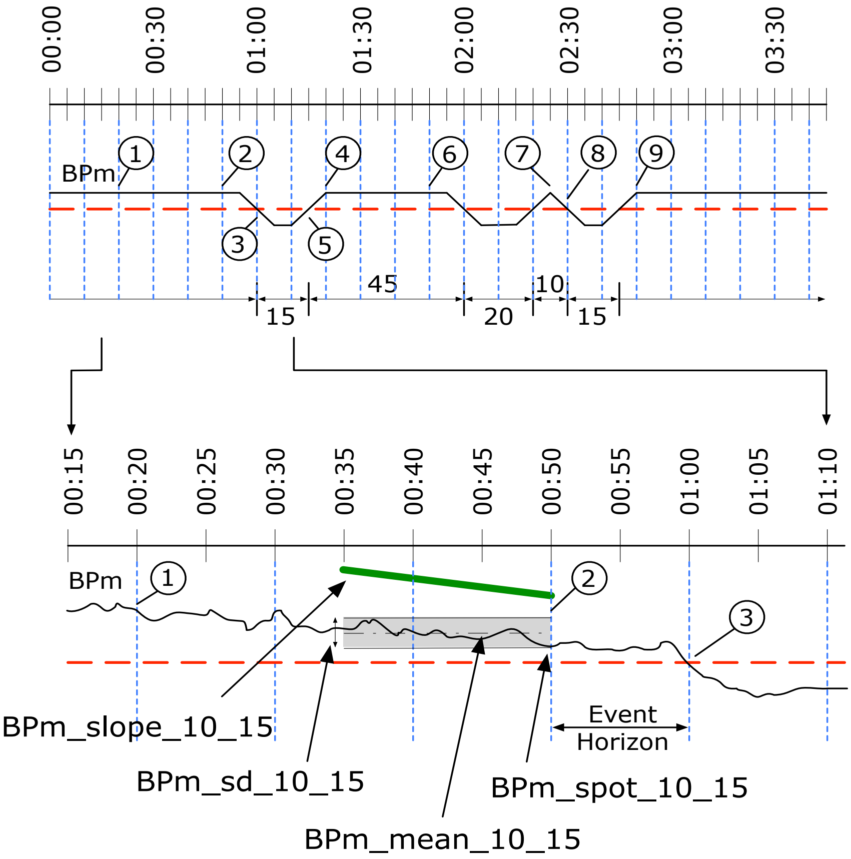


Figure 4.5: Event run expanded

Having defined a window size of data for the time series, a decision is required regarding which statistical characteristics are to be obtained from this window. Each of the characteristics will become one variable in the vector of information from each minute's readings from the ICU monitors. A good starting point is to use the standard statistical measures which describe a sample from a distribution i.e. mean and standard deviation. There is also clinical evidence in the literature suggesting that heart rate variability is a strong factor predicting hypotension, (Smiley, 2005). In addition, it is considered that the slope of the trace may contain information on the trend of the data over the window time period. Another interesting approach is to use only the spot value at the start of each window, where *start* is the measurement closest to the event horizon.

Consideration could also be given to frequency domain characteristics such as spectral density and FFT coefficients. Frequency domain analysis relies on an assumption that there are periodic signals within the data. Although there is a strong periodic signal associated with heart rate, and hence blood pressure, this would normally require higher resolution data, typically at millisecond level, for accurate analysis. There could also be some periodic element associated with breathing and circadian rhythms. However, the availability of respiration rate data tends to be limited and it is rare for the heart rate and blood pressure data to be complete over more than 24 hours — the timescale required to investigate circadian rhythms.

From discussions with clinical colleagues, and taking into account the above considerations, this thesis will only use information from the time domain. The BSG application, Section 4.2.6, will store the spot value and calculate and store the mean, standard deviation and slope for each measurement from a given window. These four statistical measures are shown in Figure 4.5 for the signal BPm using an event horizon of 10 minutes and a window size of 15 minutes. Note from Figure 4.4, considering the window size of 5 minutes, that the mean and standard deviation calculations are carried out on the *five* values between point 3 and 4, whilst the slope calculation is carried out on the *six* values from point 2 to 4.

4.2.4 Data set contents

As described in the medical background, Section 2.4, the research uses data sets which contain the following information: two demographic variables, age and gender; four raw continuous physiological variables, heart rate (HRT), systolic blood pressure (BPs), diastolic blood pressure (BPd) and mean blood pressure (BPm). The blood pressure measurements are from an invasive arterial line.

The base data set is produced by working through *all* the data from a patient file on a minute-by-minute basis. Each row consists of variables taken at points in time before an event. Each variable name takes the following format:

<measurement>_<process>_<event horizon>_<window size>

giving for example, BPm_slope_10_15. This represents the slope (as calculated by linear regression) of the mean arterial blood pressure starting at 10 minutes before the event and extending backwards in time by 15 minutes from the start point, i.e. the slope from 25 minutes before the event to 10 minutes before the event. This is shown in Figure 4.5.

4.2.5 Measurement processing

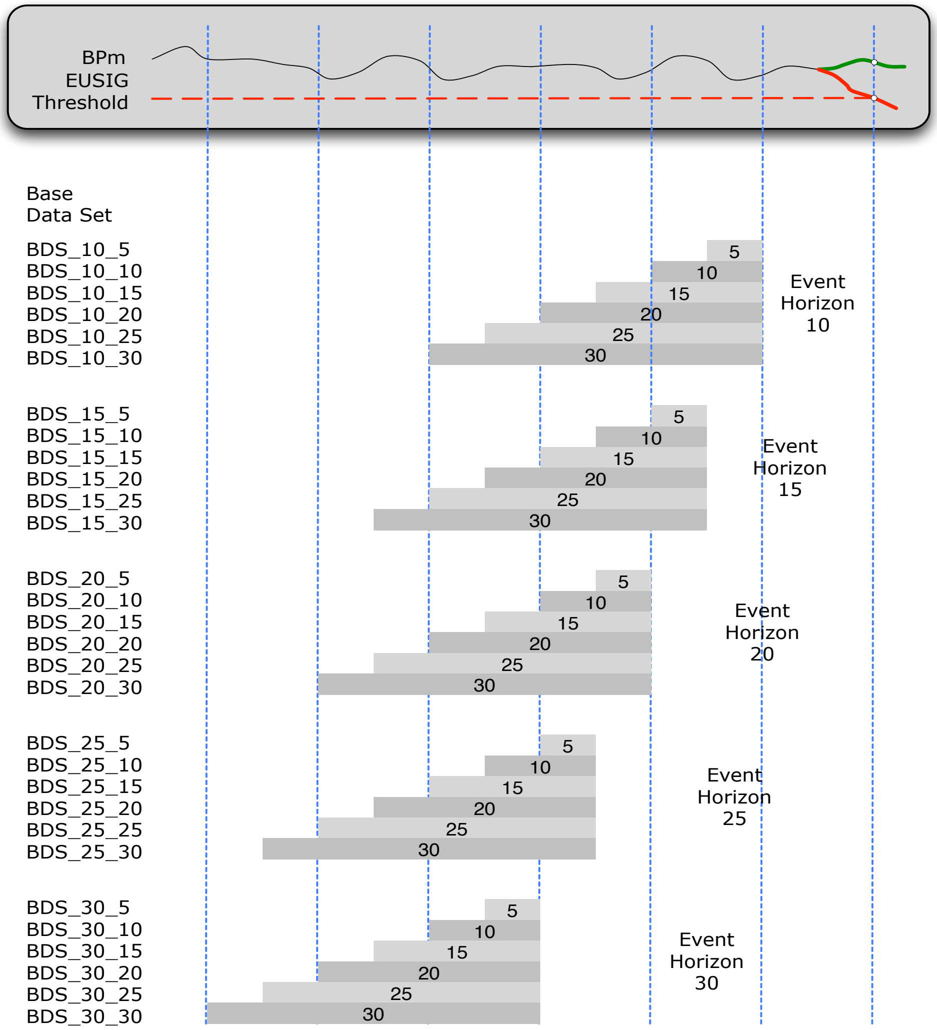


Figure 4.6: Base data set grid

Given that a useful system will predict episodes at least 10 minutes in advance and assuming that a practical window step size is five minutes, a total of 30 data sets have been constructed requiring a maximum of 30 minutes of data in an operational system. These base data sets (BDS) use event horizons at times 10, 15, 20, 25, 30 minutes before an episode and window sizes of 5, 10, 15, 20, 30 minutes. These are shown as the grey bars in Figure 4.6. Note that although Figure 4.6 seems to indicate that 50 minutes of data are required, in a real system operating in an ICU only 30 minutes of data are required before a model trained on WS=30 data can run. This is because in a real system 30 minutes of data are collected, at which point any one of the models, e.g. trained on 10_30, 15_30,

… 30_30 data sets, can be run *on the same 30 minutes of data*. The probability result that is obtained from a model indicates how likely *this 30 minute data set* is EH minutes before the start of an episode.

4.2.6 Base data set generator software

Although the preceding description is specific to the clinical data that is being used, the software application written for this task is a general purpose tool that could be used for other application domains given a database in the required format. Similar to the other tools written for this thesis, this application will be made publicly available under an open source licence and can be obtained from the website `http://www.statsresearch.co.uk`. Further details of obtaining and configuring the base set generator (BSG) can be found in Appendix A.3.

As detailed in Section 4.2.5, there are 30 base data sets that are being used to develop models for the prediction of hypotensive events. It is useful to have an overall picture of the process of creating an individual base data set. Fig. 4.7 provides a flowchart of the BSG tasks.

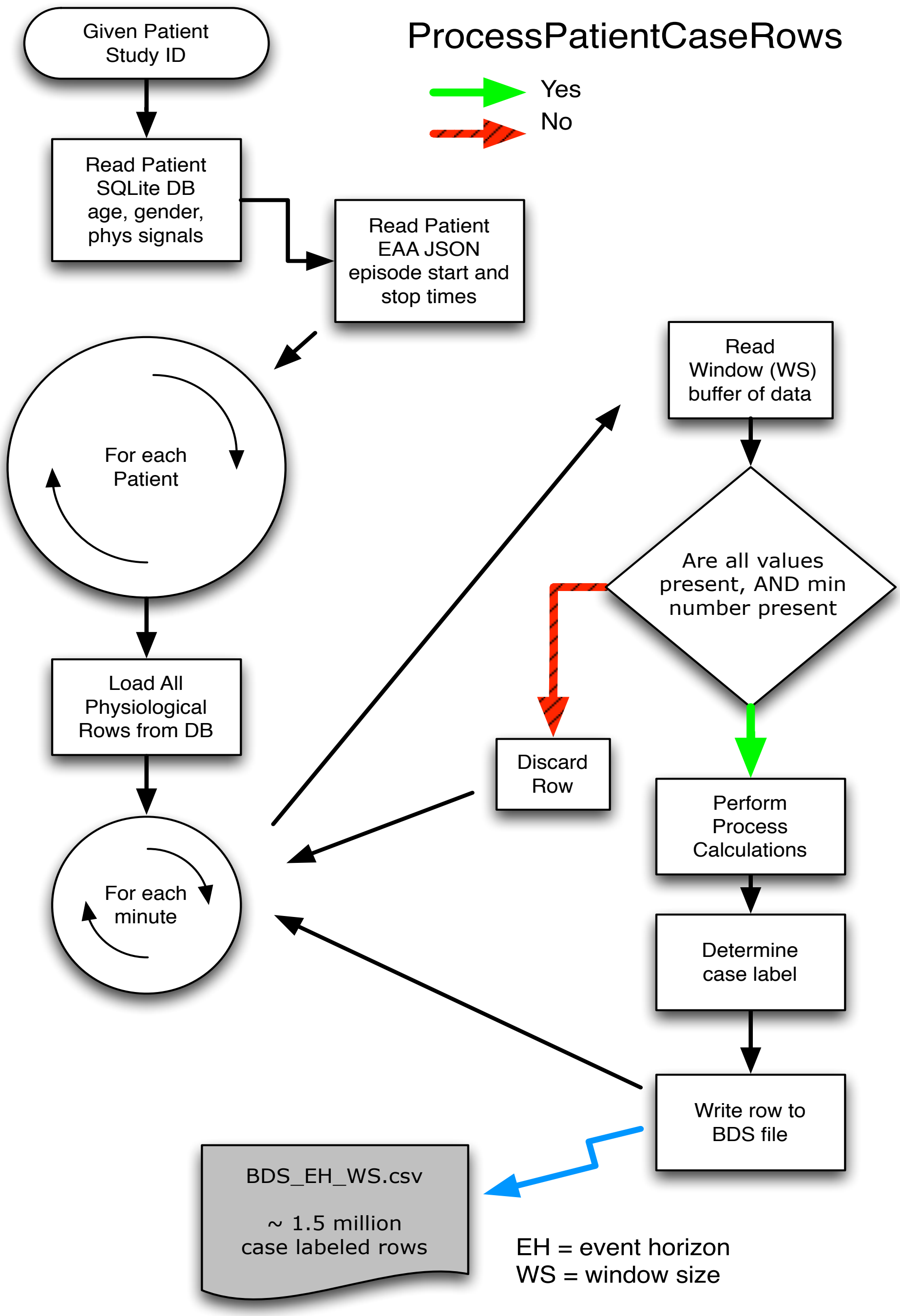

ProcessPatientCaseRows
Yes
No
Given Patient Study ID
Read Patient SQLite DB age, gender, phys signals
Read Patient EAA JSON episode start and stop times
For each Patient
Load All Physiological Rows from DB
For each minute
Read Window (WS) buffer of data
Are all values present, AND min number present
Discard Row
Perform Process Calculations
Determine case label
Write row to BDS file
BDS_EH_WS.csv
~ 1.5 million case labeled rows
EH = event horizon
WS = window size

Figure 4.7: Overview of the data preparation cycle for a single base data set

4.2.7 Example data set row calculation

Each base data set consists of rows in the following form:

<timestamp><Series of Variables><hypotensive case label = 1 or 0>

These rows are built up by processing all the physiological data from a patient's stay in the ICU on a minute by minute basis. This process is repeated for all the patients in the database. The algorithm is detailed below:

- Load current minute values
- Load window_size previous values
- Are there minimum number of values?
    - Yes — Perform <process> calculations
    - No — Move to next minute
- Add event horizon to current minute
- Was there an event at this new time?
    - Yes — Mark as positive vector, case label = 1
    - No — Mark as negative vector, case label = 0
- Write row to BDS .csv file
- Move to next minute

Consider an example using actual clinical data. Figure 4.8 plots 70 minutes of data from an ICU monitor for the signals HRT, BPs, BPd, BPm and Table 4.2 gives the actual values, in two columns.

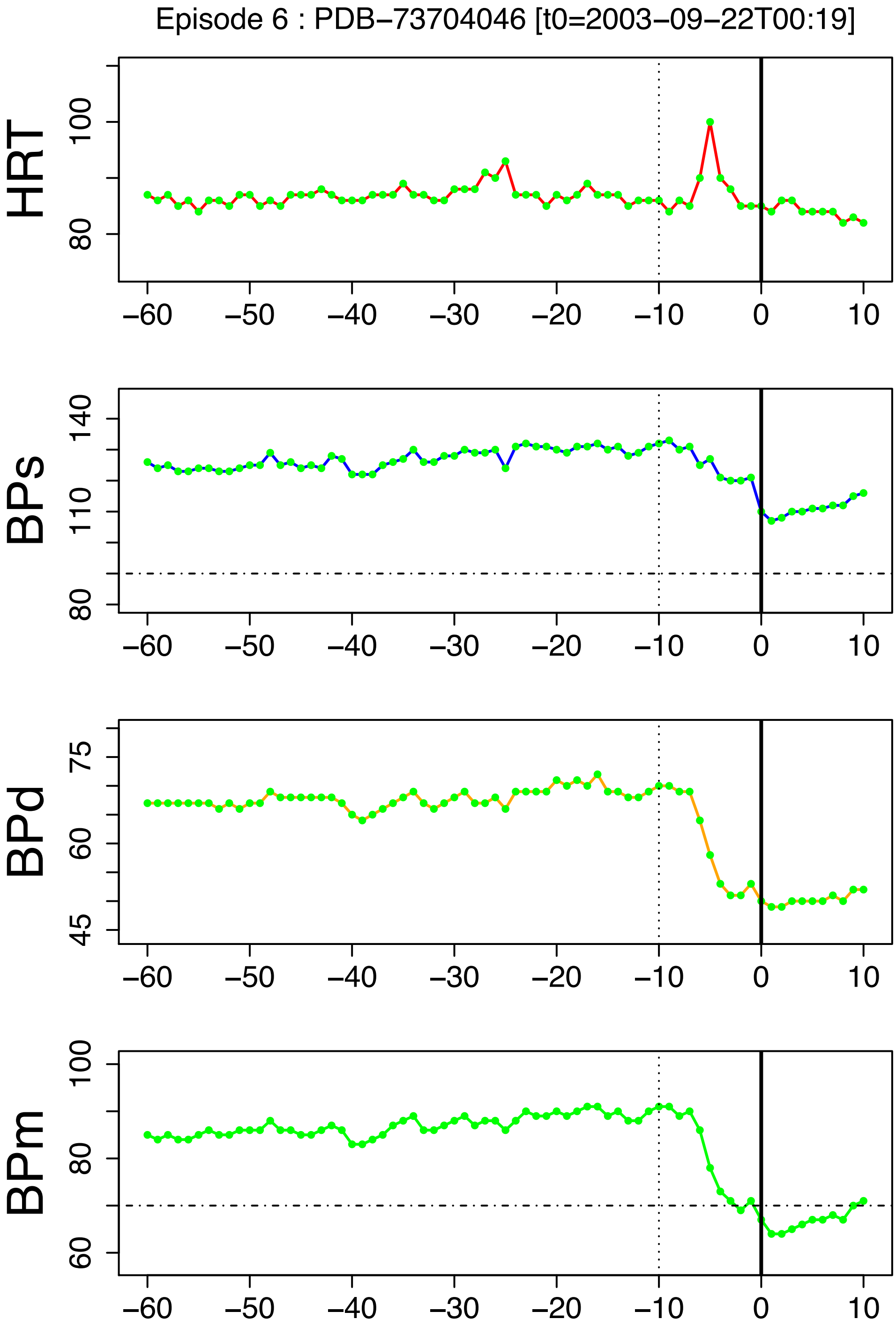


Figure 4.8: Clinical data, patient 73704046, episode 6

Time Stamp	HRT	BPs	BPd	BPm	Time Stamp	HRT	BPs	BPd	BPm
2003-09-21 23:19	87	126	67	85	2003-09-21 23:54	93	124	66	86
2003-09-21 23:20	86	124	67	84	2003-09-21 23:55	87	131	69	88
2003-09-21 23:21	87	125	67	85	2003-09-21 23:56	87	132	69	90
2003-09-21 23:22	85	123	67	84	2003-09-21 23:57	87	131	69	89
2003-09-21 23:23	86	123	67	84	2003-09-21 23:58	85	131	69	89
2003-09-21 23:24	84	124	67	85	2003-09-21 23:59	87	130	71	90
2003-09-21 23:25	86	124	67	86	2003-09-22 00:00	86	129	70	89
2003-09-21 23:26	86	123	66	85	2003-09-22 00:01	87	131	71	90
2003-09-21 23:27	85	123	67	85	2003-09-22 00:02	89	131	70	91
2003-09-21 23:28	87	124	66	86	2003-09-22 00:03	87	132	72	91
2003-09-21 23:29	87	125	67	86	2003-09-22 00:04	87	130	69	89
2003-09-21 23:30	85	125	67	86	2003-09-22 00:05	87	131	69	90
2003-09-21 23:31	86	129	69	88	2003-09-22 00:06	85	128	68	88
2003-09-21 23:32	85	125	68	86	2003-09-22 00:07	86	129	68	88
2003-09-21 23:33	87	126	68	86	2003-09-22 00:08	86	131	69	90
2003-09-21 23:34	87	124	68	85	2003-09-22 00:09	86	132	70	91
2003-09-21 23:35	87	125	68	85	2003-09-22 00:10	84	133	70	91
2003-09-21 23:36	88	124	68	86	2003-09-22 00:11	86	130	69	89
2003-09-21 23:37	87	128	68	87	2003-09-22 00:12	85	131	69	90
2003-09-21 23:38	86	127	67	86	2003-09-22 00:13	90	125	64	86
2003-09-21 23:39	86	122	65	83	2003-09-22 00:14	100	127	58	78
2003-09-21 23:40	86	122	64	83	2003-09-22 00:15	90	121	53	73
2003-09-21 23:41	87	122	65	84	2003-09-22 00:16	88	120	51	71
2003-09-21 23:42	87	125	66	85	2003-09-22 00:17	85	120	51	69
2003-09-21 23:43	87	126	67	87	2003-09-22 00:18	85	121	53	71
2003-09-21 23:44	89	127	68	88	2003-09-22 00:19	85	110	50	67
2003-09-21 23:45	87	130	69	89	2003-09-22 00:20	84	107	49	64
2003-09-21 23:46	87	126	67	86	2003-09-22 00:21	86	108	49	64
2003-09-21 23:47	86	126	66	86	2003-09-22 00:22	86	110	50	65
2003-09-21 23:48	86	128	67	87	2003-09-22 00:23	84	110	50	66
2003-09-21 23:49	88	128	68	88	2003-09-22 00:24	84	111	50	67
2003-09-21 23:50	88	130	69	89	2003-09-22 00:25	84	111	50	67
2003-09-21 23:51	88	129	67	87	2003-09-22 00:26	84	112	51	68
2003-09-21 23:52	91	129	67	88	2003-09-22 00:27	82	112	50	67
2003-09-21 23:53	90	130	68	88	2003-09-22 00:28	83	115	52	70

Table 4.2: Data values, patient 73704046, episode 6

Consider the base data set 10_15 built up, as described above, by processing each row of physiological data for each patient. With reference to Table 4.2, consider processing the row of data starting at 2003-09-21 23:40 (light grey highlight, left hand column) and the calculations required for the variables BPm_Slope_10_15, BPm_Mean_10_15, BPm_SD_10_15. The 15 values previous to this timestamp are available (blue highlight, left hand column) therefore the calculations required for mean and standard deviation can be performed. Calculation of the slope requires 16 values (light green and blue highlights, left hand column). The next part of the algorithm looks ahead for event horizon minutes. In this example, this would be EH = 10, giving the timestamp 2003-09-21 23:50 (darker grey highlight, left hand column) which is not an event. This row, therefore, is marked as a negative vector, case label = 0. The algorithm proceeds, minute by minute, until the timestamp 2003-09-22 00:09 (light grey highlight, right hand column).

The 15 values previous to this timestamp are available (blue highlight, right hand column) therefore once more the calculations required for mean and standard deviation can be performed. Again, calculation of the slope requires 16 values (light green and blue highlights, right hand column). As before, the next part of the algorithm looks ahead for event horizon minutes. In this example, for EH = 10, this gives the timestamp 2003-09-22 00:19 (red highlight, right hand column). In this case, the timestamp *is* the start of an episode (which is known from the EAA JSON file of episode information) therefore this row is marked as a positive vector, case label = 1.

4.3 Training and test data sets

This section describes how the files produced by the BSG application from Section 4.2 are transformed into training and test files that can be used by the model building process described in chapter 5.

To expand on the description given in Section 3.4 regarding classifier models, at a first level, there is a need to separate the available patients into two groups; one that will be used for training a model and one that will form a test set for model validation. This is done by randomly splitting the total cohort into two groups using a standard random number generator function.

An assumption used during the model training process is that each row of data in a training file is reasonably independent of any other row. This is straightforward in the case of the positive vectors as each episode start is a new situation for the patient and the process of coalescing close events into a single episode, see Section 2.2.2, minimises the effect of closely spaced events. In the case of the negative vectors, which are assumed to represent the stable state of the patient, this is more difficult as the vast majority of rows in a BDS file are, in fact, negative vectors separated by a single minute. This single minute separation would suggest that there will often be high correlation in a contiguous group of negative vectors. What is required is a random selection of these negative vectors in a similar proportion to that found in the real ICU monitor data stream. In the case of this research, a mix of 10% positive cases to 90% negative cases was chosen to represent the ICU data. This proportion allows for the use of *all* the positive cases together with a random selection of negative cases whilst still keeping the training file to a manageable size for model processing. As only a small proportion of negative cases are being used, it is unlikely that overlapping sequences are chosen. A further important aspect is that the selection is very sparse.

The training and test set generator application (TTG), Section 4.3.1, carries out this process on the two groups of patients separated during the first pass through the data. Because of the 10:90 ratio, the file size is manageable, typically 500 KB, and this allows the process to be repeated a number of times, giving a bootstrap effect. For this research, the number of training/test file repeats is set to 50.

4.3.1 Training and test data set generator software

An R application was written to carry out the task of producing training and test files for the model building process. As before, this application will be made publicly available under an open source licence and can be obtained from the website `http://www.statsresearch.co.uk`. Further details of how to obtain and configure the training and test set generator application (TTG) can be found in Appendix A.4.

The application writes 50 training, test and information files for a given BDS. The information file is included as a result of the random pick process carried out to select patients for either the training or test set. This random pick phase results in a varying number of episodes each time due to the varying number of episodes each patient suffers. This means that each training/test file in the 1 to 50 repeat sequence will have a varying number of positive vectors. This of course will result in a varying number of negative vectors because of the fixed 10:90 ratio.

Figure 4.9 shows a flow chart of the TTG operation.

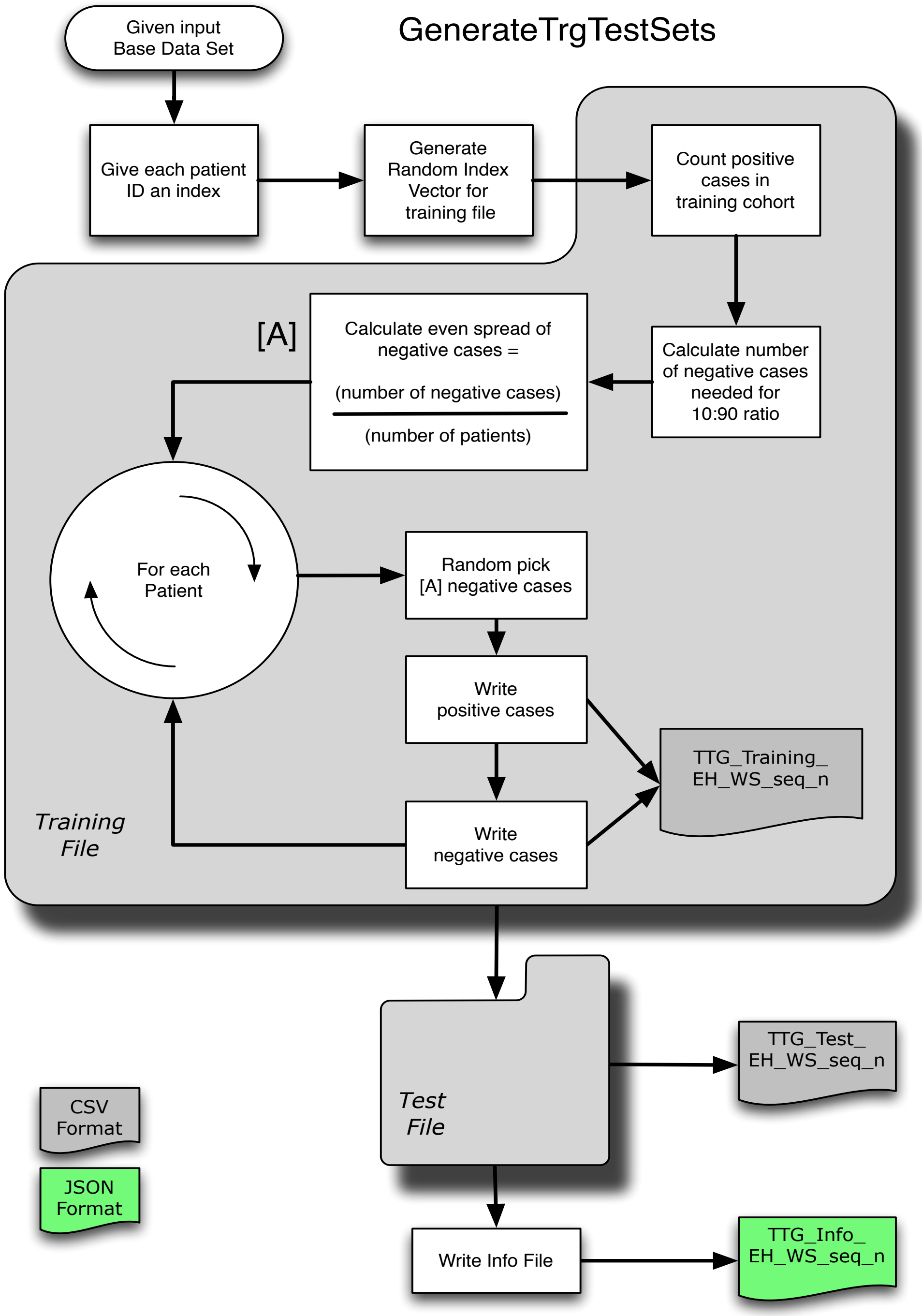

GenerateTrgTestSets
Given input Base Data Set
Give each patient ID an index
Generate Random Index Vector for training file
Count positive cases in training cohort
Calculate number of negative cases needed for 10:90 ratio
[A]
Calculate even spread of negative cases = (number of negative cases) / (number of patients)
For each Patient
Random pick [A] negative cases
Write positive cases
Write negative cases
TTG_Training_ EH_WS_seq_n
Training File
Test File
TTG_Test_ EH_WS_seq_n
Write Info File
TTG_Info_ EH_WS_seq_n
CSV Format
JSON Format

Figure 4.9: Overview of a single repeat of the training and test set generation process

From Figure 4.9 above, the generation of the training and test files moves through four stages. Given a base data set, the first stage is to split the total cohort into two groups of patients. This is achieved by assigning each unique patient ident from the study a numerical index from the sequence 1 to the maximum number of patients. Half the total is then picked by a random number generator and marked as the training set; the remaining half is marked as the test set.

The second stage is detailed in the “Training File” grey shape. This process involves counting the number of positive vectors associated with the training idents. Once the number of positive vectors is known, a calculation can be performed to obtain the number of negative vectors required to maintain a 10:90 ratio of positive to negative cases. This total number of negative vectors is randomly picked from the several thousand available for each patient. All the positive vectors and the randomly selected negative vectors are subsequently written out to the training file.

The third stage, the “Test File” grey shape, repeats the actions from the Training File but uses the test set patients, with the results being written out to the test file.

The final stage is to write out the summary statistics gathered during the formation of the training and test sets into a JSON format information file. This entire four stage process is repeated 50 times.

Chapter 5

Logistic Regression Models

This chapter describes in detail a series of logistic regression models that can provide early warning of a hypotensive event. It will be shown that a model built with this relatively simple technique can produce predictive results in line with current state of the art research.

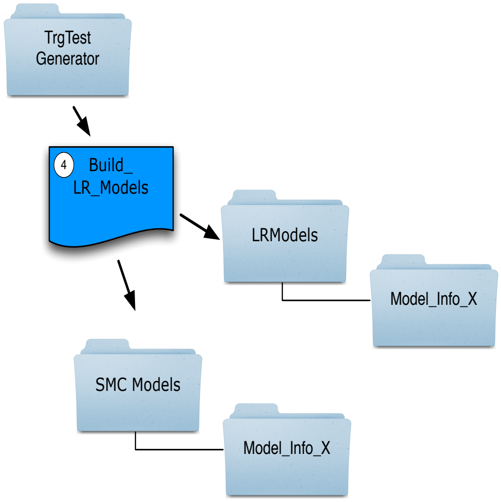


Figure 5.1: Logistic regression modelling research tasks

5.1 Chapter overview with summary plots

There are six logistic regression models detailed in this section. These six models use two different architectures for their input structure, these architectural structures were introduced in Section 3.7. The first set of two models use each minute of each physiological signal as inputs to the model (Figures 3.10 and 5.6). The second set of four models performs a series of statistical pre-calculations on each minute of each physiological signal and the results of these calculations form the inputs to the models (Figures 3.11 and 5.25). The first section of the chapter proposes a series of models. For each proposed model full details of the model parameters and their estimates are provided.

Taking each proposed model structure, the R statistical package, (R Development Core Team, 2008), is used to train the model on a randomly drawn training set. For the two models based on each minute of the available data, parameter estimation is carried out using a penalised maximum likelihood technique and a Bayesian model using Sequential Monte Carlo (Section 3.5). Both of these approaches are coded from first principles. The parameter estimates for the models based on statistical measures of the inputs are obtained by using packages available within the R framework.

A model, using the parameter estimates from the training phase, is then tested against a group of unseen patients in a test set. The predictions obtained from applying the model to the test set are used to generate an ROC curve. This process is repeated 50 times. A standard measure of classifier performance, the area under the ROC curve (AUC) is used to assess the model. The ROC plot shows the median ROC curve along with a 90% credible limit around the median. This provides a measure of the variability of model performance. This assessment method is used to decide whether or not to proceed with a model to the next stage of testing using clinical data.

Details of the R code used to provide the results presented in this chapter are provided in Appendix A.

Figures 5.2 to 5.5 are provided as an executive summary of the six logistic regression models.

5.1.1 Summary of models using each minute as input

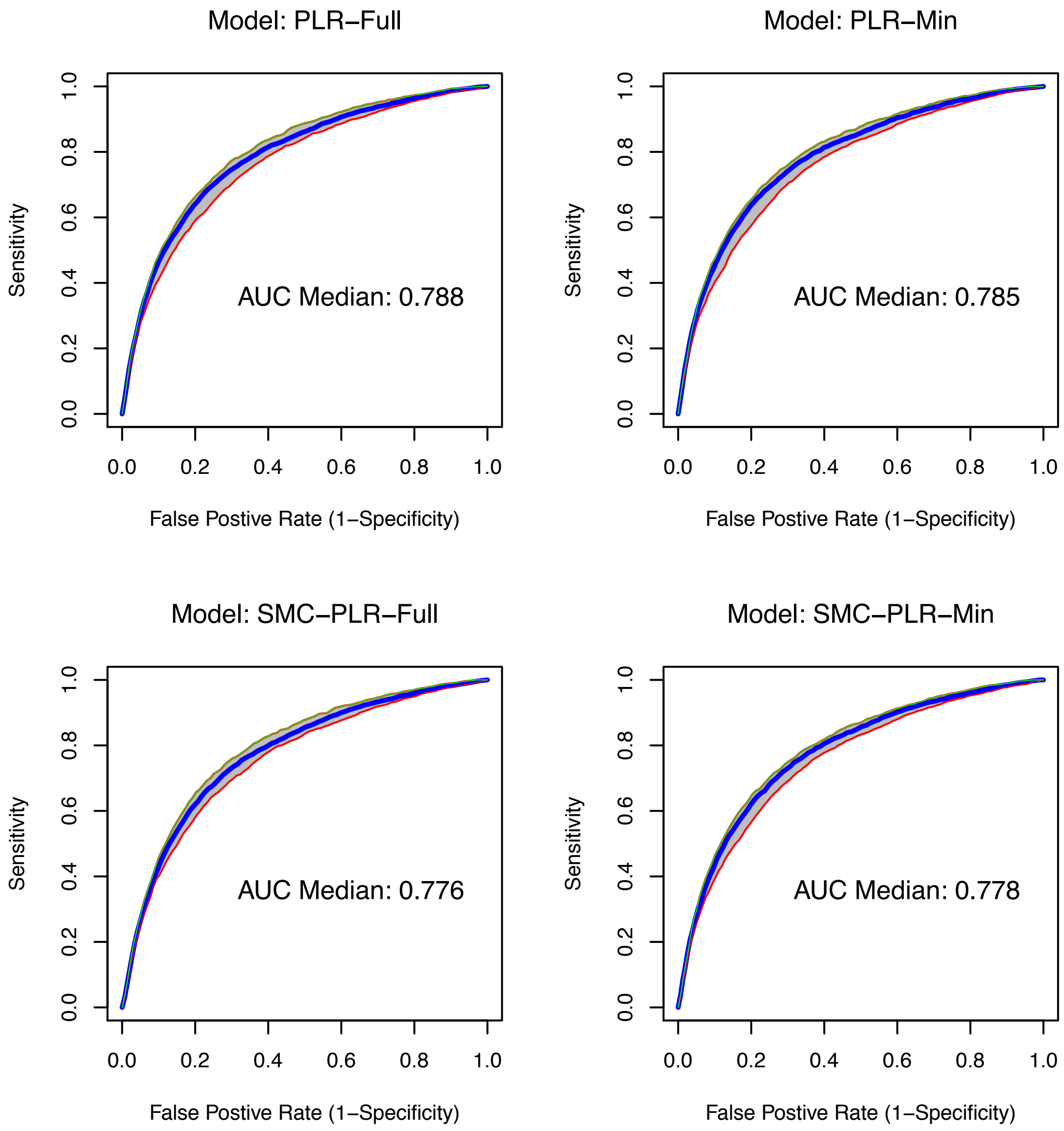


Figure 5.2: Penalised Logistic Regression Models Performance Summary. Models using “all data” i.e. each minute from each physiological signal forms an input to the model. The parameters for the two penalised logistic regression models were obtained by both maximum likelihood and sequential Monte Carlo techniques. The top row of this plot shows the ROC curves obtained from the model whose parameters were obtained by maximum likelihood. The bottom row shows the ROC curves obtained from the model whose parameters were obtained using a sequential Monte Carlo method.

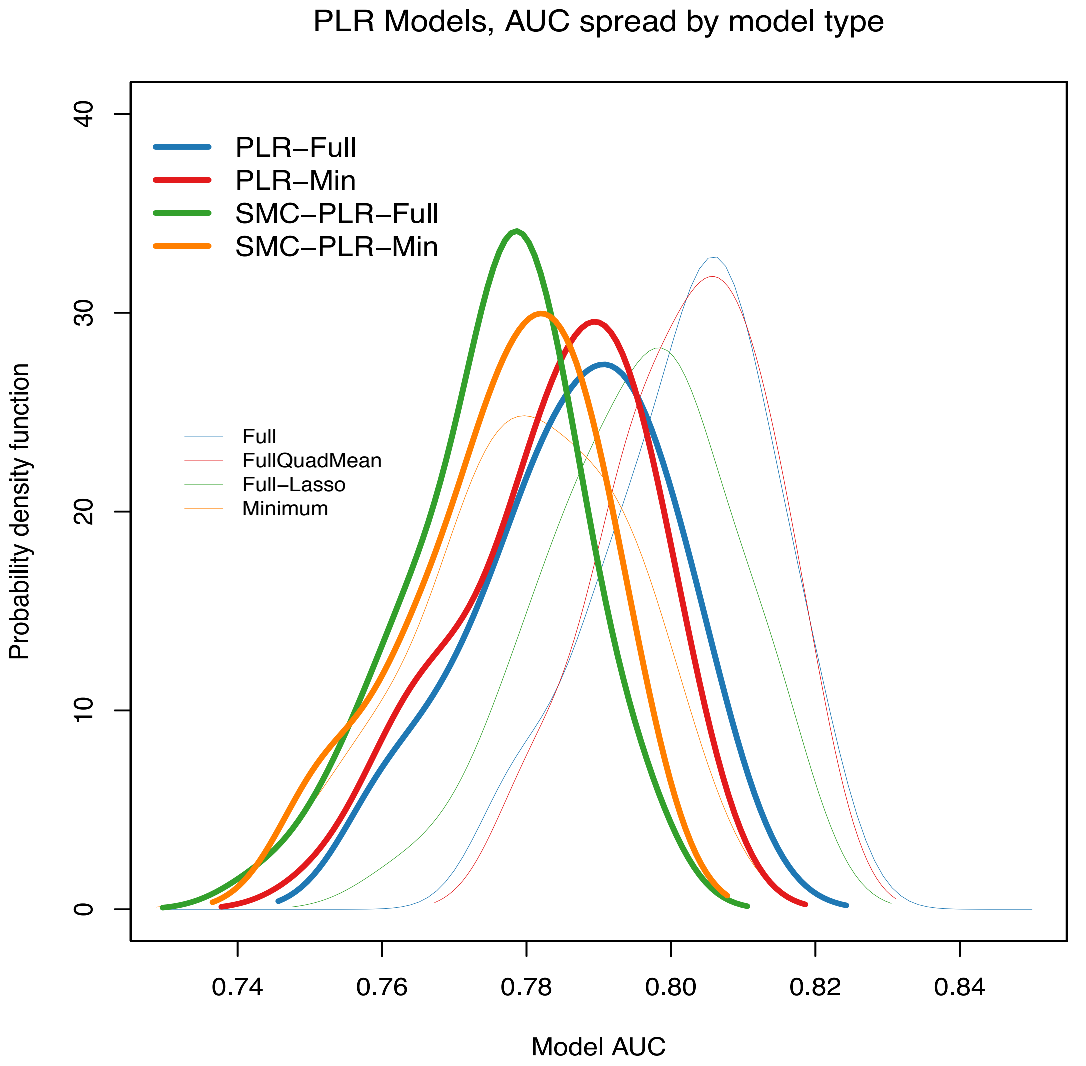


Figure 5.3: Penalised Logistic Regression Models AUC Density Plots. The heavier lines are the density curves, from 50 iterations of each model, showing the spread of AUC values for the two penalised logistic regression models and two fully Bayesian penalised logistic regression models. For comparison density traces are provided from the four statistics based input models. These are shown as the thinner lines on the plot.

5.1.2 Summary of models using statistical measures as input

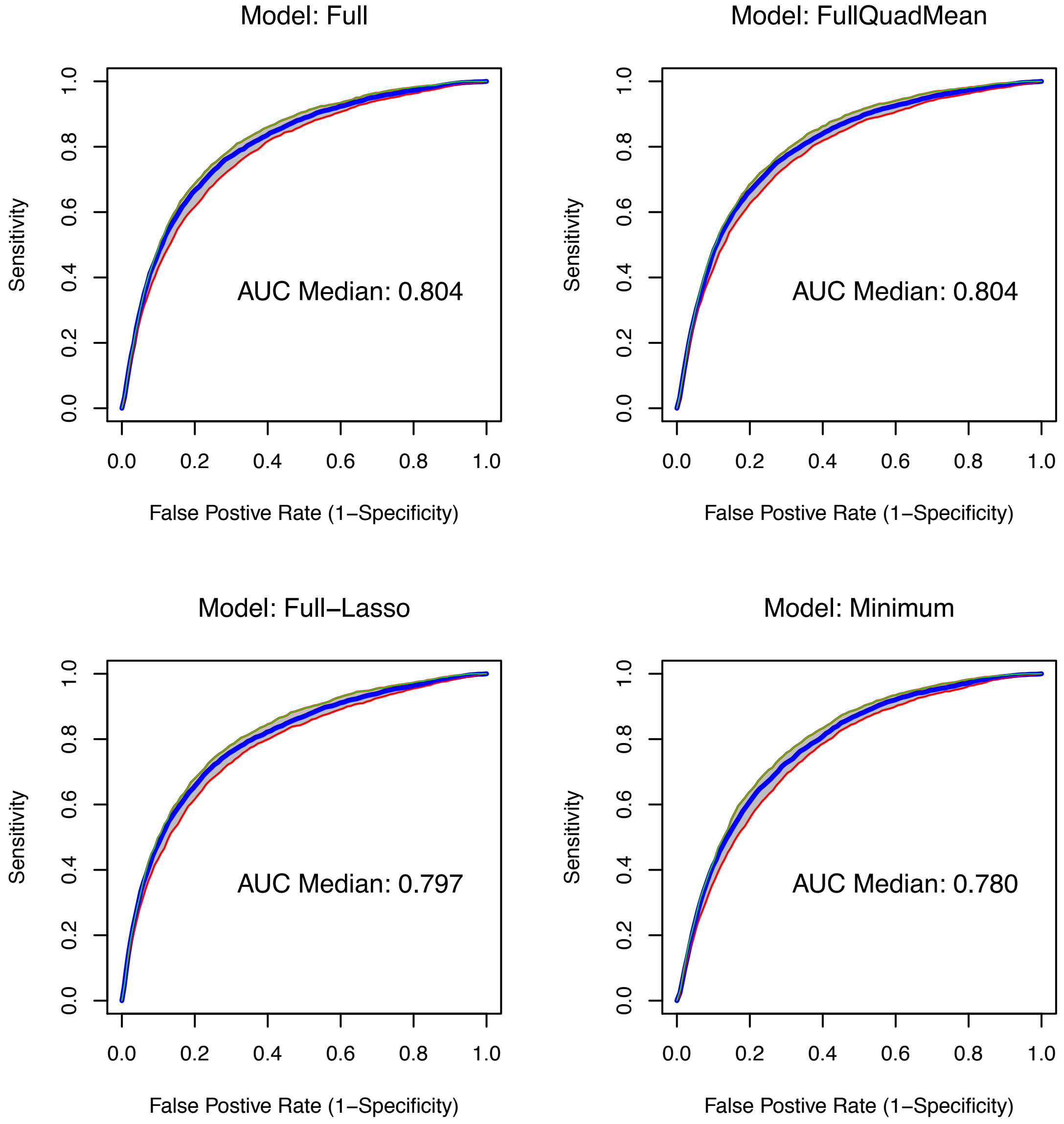


Figure 5.4: Logisitic Regression Model Performance Summary. Models use “statistics based” model inputs. Summary statistics are formed for each physiological signal and it is the values of these statistical measures that form the inputs to the models.

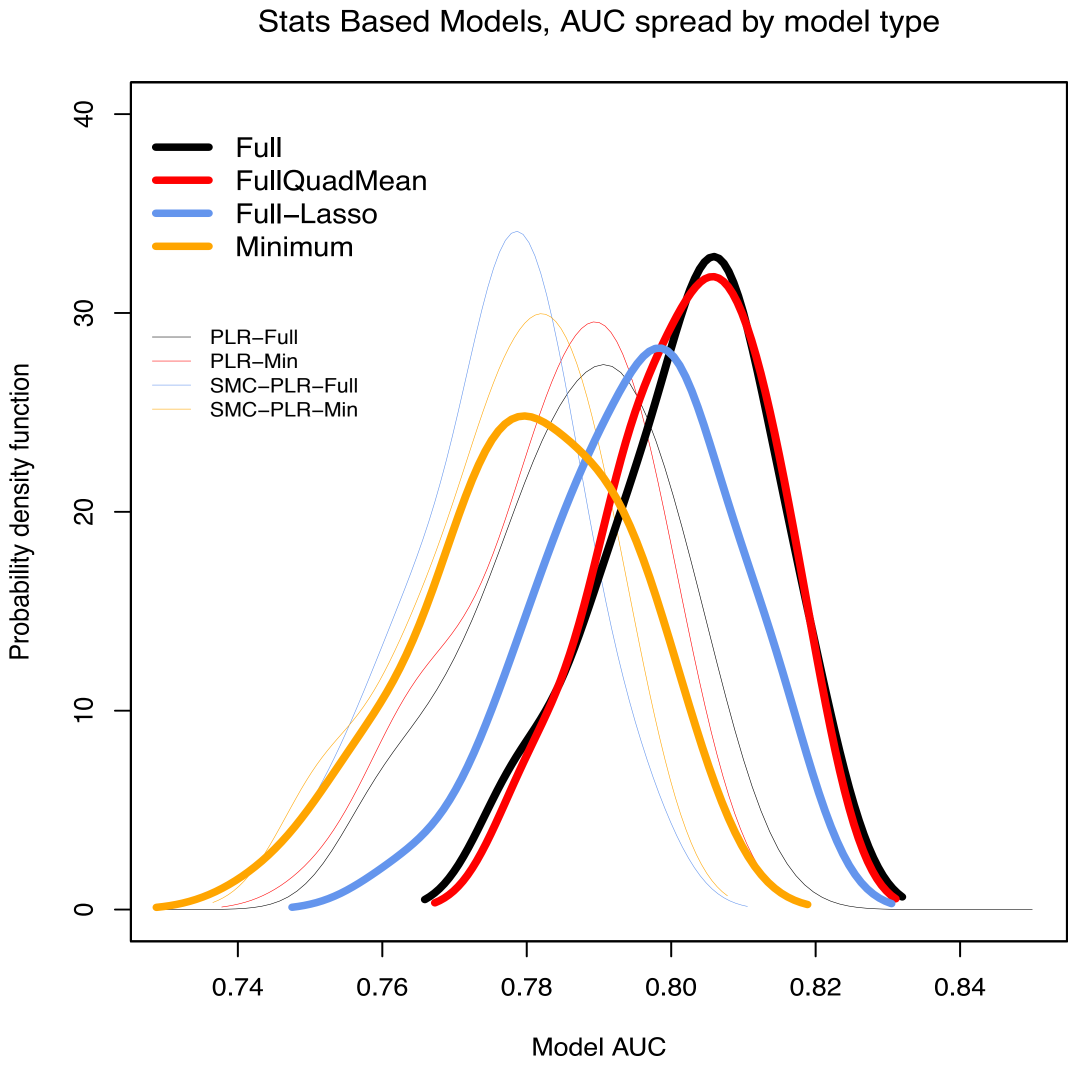


Figure 5.5: Logistic Regression Models AUC Density Plots. The heavier lines are the density curves, from 50 iterations of each model, showing the spread of AUC values for the four logistic regression models that use statistical measures as input.

5.2 Model proposals

There are many different models that could be examined; these range from the very complex, e.g. examining main effects and interactions for all signals, to simpler models containing only clinically significant signals. The aim of the research is to examine whether or not simpler models can be as effective as more complex models. The primary motive for simpler models is that they are easier to explain and hence acceptable to clinical teams. Six proposed models are outlined below.

The first two models use the complete 30 minutes of data from relevant physiological signals, Sections 5.4.1 to 5.4.4 provide the more detailed reasoning behind the model structures. The parameters for the two models are first obtained using a penalised maximum likelihood approach and are also estimated using the Bayesian parameter estimation technique from Section 3.5 in the Methods Review chapter.

The next group of four models, detailed in Sections 5.4.1 to 5.4.4 use a technique of summarising an available physiological signal using four statistical measures. This results in a model with fewer parameters that must be estimated. It is also a straightforward method to explain to clinical colleagues.

5.3 Models using each minute of data

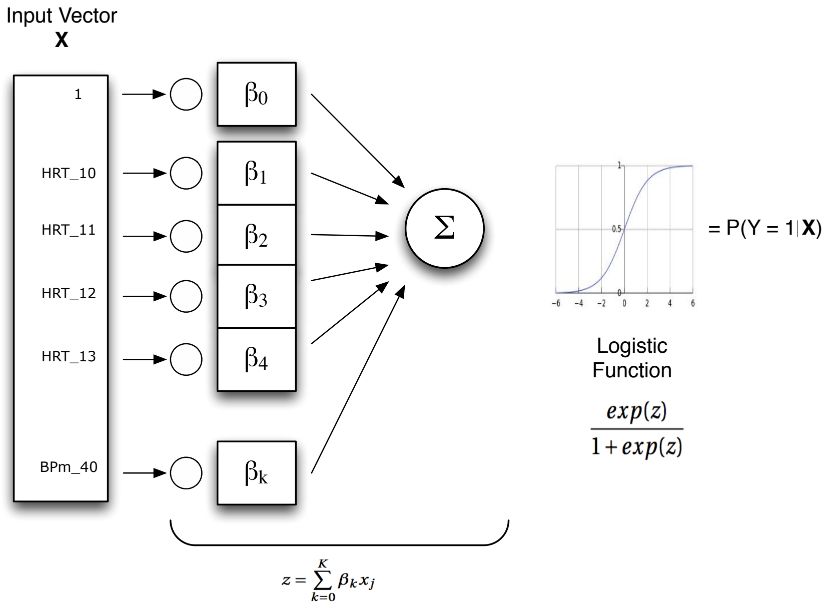


Figure 5.6: Input architecture, "All Data" models. These models use each minute of data as inputs.

- All signals, Model name: PLR-Full — a model using the age and gender of the patient plus all the available signals from Table 2.5. For each of the physiological signals each minute of a 30 minute buffer is used as an input signal.

- Minimum signals, Model name: PLR-Min — a model using the age and gender of the patient plus the signals HRT and BPm which are suggested by clinical colleagues. Again for each of the HRT and BPm signals each minute of a 30 minute buffer is used as an input signal.

In the following sections, the probability of a hypotensive event starting in EH minutes, p_i, is given by,

$$p_i = p(Y_i = 1|\boldsymbol{X}_i) \tag{5.1}$$

and,

$$\operatorname{logit}(Y_i = 1|\boldsymbol{X}_i) = \operatorname{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) \tag{5.2}$$

Also note that all models use normalised data i.e.

$$x_i = x_{i\ input} - \mu_{input} \tag{5.3}$$

with $\mu_{input} = 1/N \sum_{i=1}^{N} x_i$

5.3.1 Determination of optimal penalty setting λ

All the penalised logistic regression models require a value for the penalty parameter λ. The optimal value for λ was determined using a two pass approach along vectors of possible λ values.

The first pass used a coarse grain approach with values for λ from the vector $\{1,10,100,1e3,1e4,1e5,1e6,1e7,1e8,1e9,1e10,1e12,1e14\}$. For each λ value, the β values for the model were estimated and then a predictive loglikelihood value was calculated using a test set of data. This process was repeated for 10 test sets. The resulting vector of loglikelihood values was processed to find the value of λ that gave the maximum loglikelihood.

The second pass used a fine grain approach which takes as its mid point the value of λ that gave the maximum loglikelihood from the first pass. The λ value is converted to a $log(\lambda)$ value and then the range is defined as $log(\lambda_{maxLL}) \pm 2$. This range is then divided into a fine grain vector with a spacing of 0.1. Again a loglikelihood value was calculated using a test set of data. This process was repeated for 50 test sets. The final setting for λ was taken as the value that gave the maximum predictive loglikelihood.

5.3.2 All signals, Model name: PLR-Full

The PLR-Full model is the baseline for the penalised logistic regression group of models. It utilises data from the BrainIT database which reflects the signals that are available from a typical installation of ICU bedside monitors. Full details of the available signals are provided in the medical background, Section 2.4. To briefly recap, this model uses two demographic signals and four measured signals from the ICU monitoring system. The demographic signals are age and gender. The measured signals are: heart rate (HRT); systolic arterial blood pressure (BPs); diastolic arterial blood pressure (BPd); and mean arterial blood pressure (BPm). For each of the four measured signals, this model uses each available minute from a 31 minute buffer as inputs. This gives a total of 127 input parameters which must be estimated. The clinical basis for this model comes from discussions with colleagues and builds on the work of the AvertIT project, (Donald et al., 2012b). Formally the model is defined as:

$$\begin{aligned}\log\left(\frac{p_i}{1-p_i}\right) &= \beta_0 + \beta_1 Age_i + \beta_2 Gender_i \\ &+ \beta_3 HRT\_10_i + \beta_4 HRT\_11_i + \beta_5 HRT\_12_i + \ldots + \beta_{33} HRT\_40_i \\ &+ \beta_{34} BPs\_10_i + \beta_{35} BPs\_11_i + \beta_{36} BPs\_12_i + \ldots + \beta_{64} BPs\_40_i \\ &+ \beta_{65} BPd\_10_i + \beta_{66} BPd\_11_i + \beta_{67} BPd\_12_i + \ldots + \beta_{95} BPd\_40_i \\ &+ \beta_{96} BPm\_10_i + \beta_{97} BPm\_11_i + \beta_{98} BPm\_12_i + \ldots + \beta_{126} BPm\_40_i \qquad (5.4)\end{aligned}$$

with

$$\begin{aligned}\mathscr{L}(\boldsymbol{\beta})_{penalised} &= \mathscr{L}(\boldsymbol{\beta}) - \lambda([(\beta_4 - \beta_3) - (\beta_5 - \beta_4)]^2) \\ &- \vdots \\ &- \lambda([(\beta_{125} - \beta_{124})) - (\beta_{126} - \beta_{125})]^2) \qquad (5.5)\end{aligned}$$

As described in the Methods Review Section 3.4.1.1, estimation of the 127 parameters was carried out using a penalised logistic regression technique. This technique is implemented in the R script **PenalisedLogisticRegression.R**, (see Appendix B.1).

The model performance, as measured using the ROC technique, is presented in Figure 5.7. The profile of the model coefficient values is presented in Figure 5.3.2.2. The full summary of the parameter values is provided in Table 5.1.

5.3.2.1 Model Performance ROC curves (Model: PLR-Full)

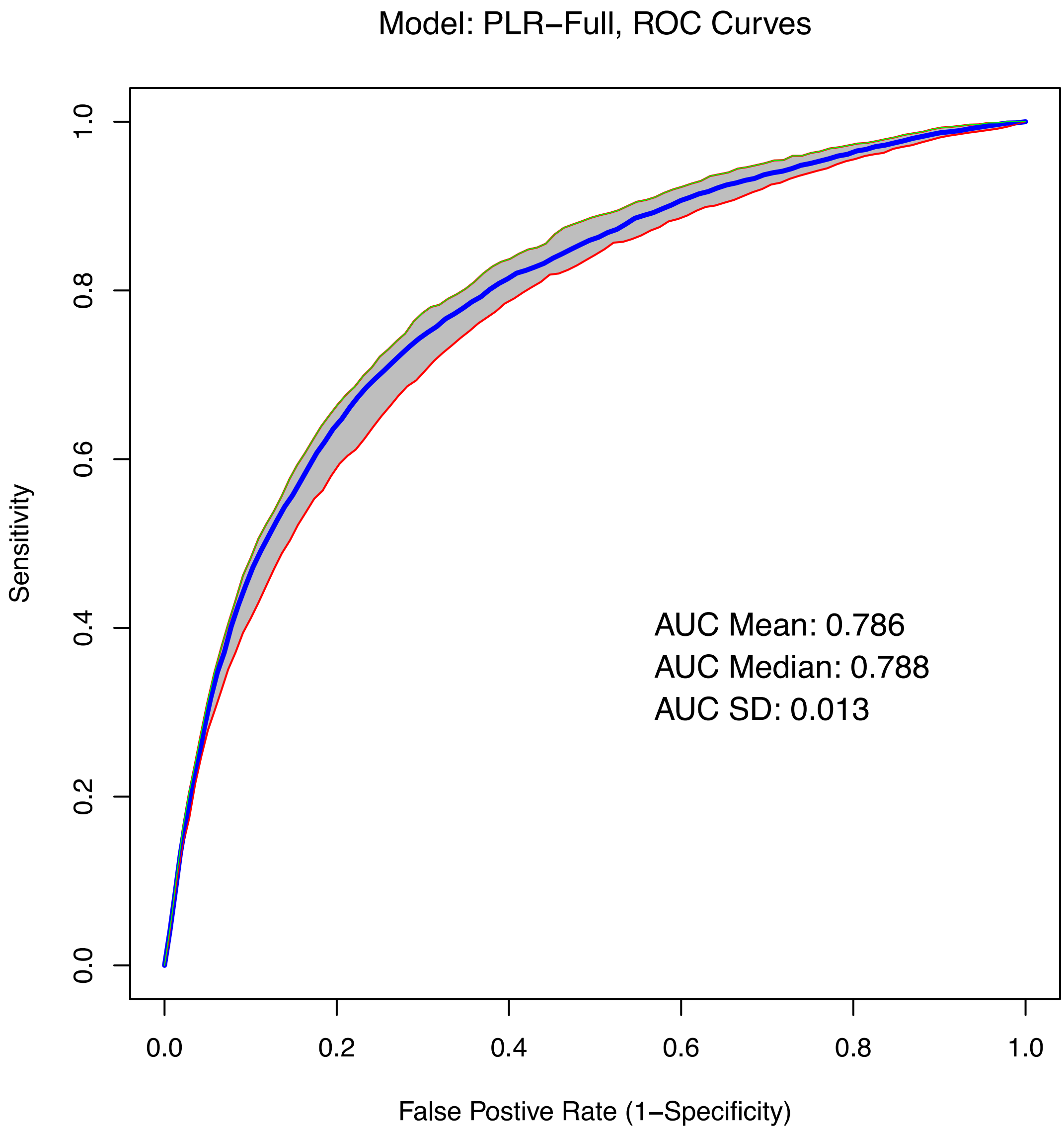


Figure 5.7: Model: PLR-Full, ROC and AUC detail. This plot is generated from 50 runs of the model. Each run consists of a training and test set. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

5.3.2.2 Estimation of lambda (Model: PLR-Full)

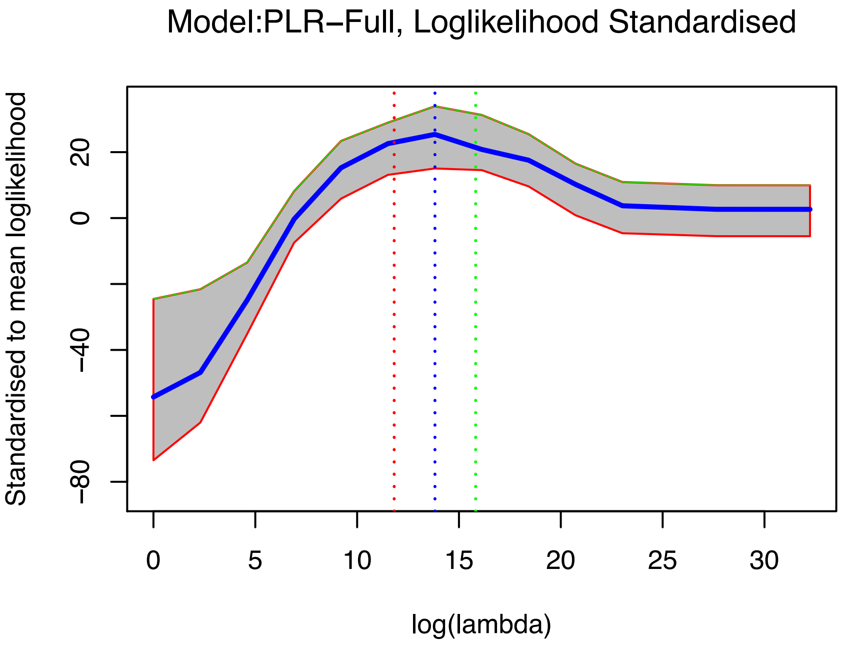


Figure 5.8: PLR-Full: Coarse search for optimal λ. Ten iterations of the coarse search returns a value for $\lambda = 13.82$. The y-axis shows the deviation from the mean loglikelihood at each λ setting across the ten iterations. This accounts for the different absolute values of loglikelihood due to the different data sets.

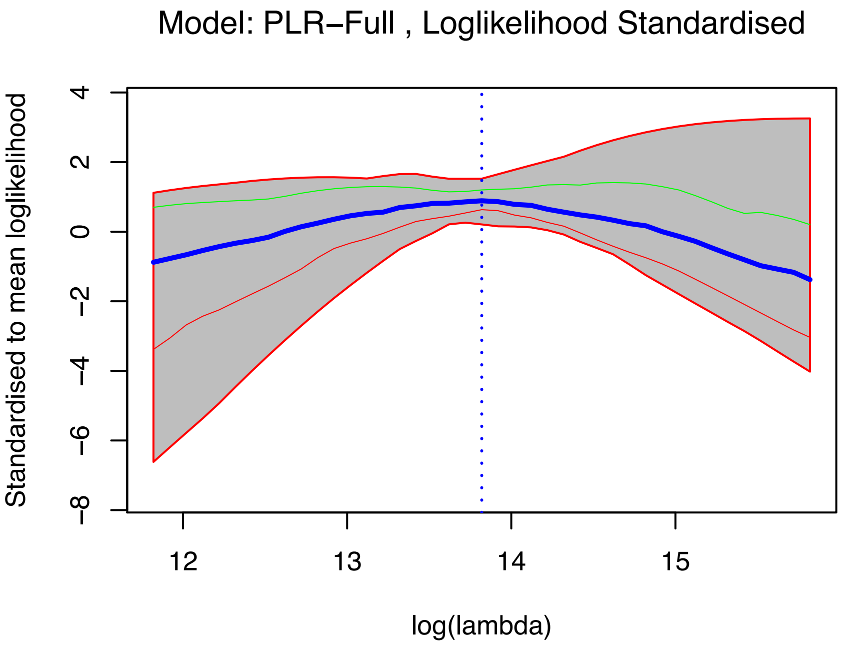


Figure 5.9: PLR-Full: Fine grain search for optimal λ. Thirty iterations of the fine grain search returns the same value of 13.82 for λ. The y-axis shows the deviation from the mean loglikelihood at each λ setting across the 30 iterations.

5.3.2.3 Parameter Profiles (Model: PLR-Full)

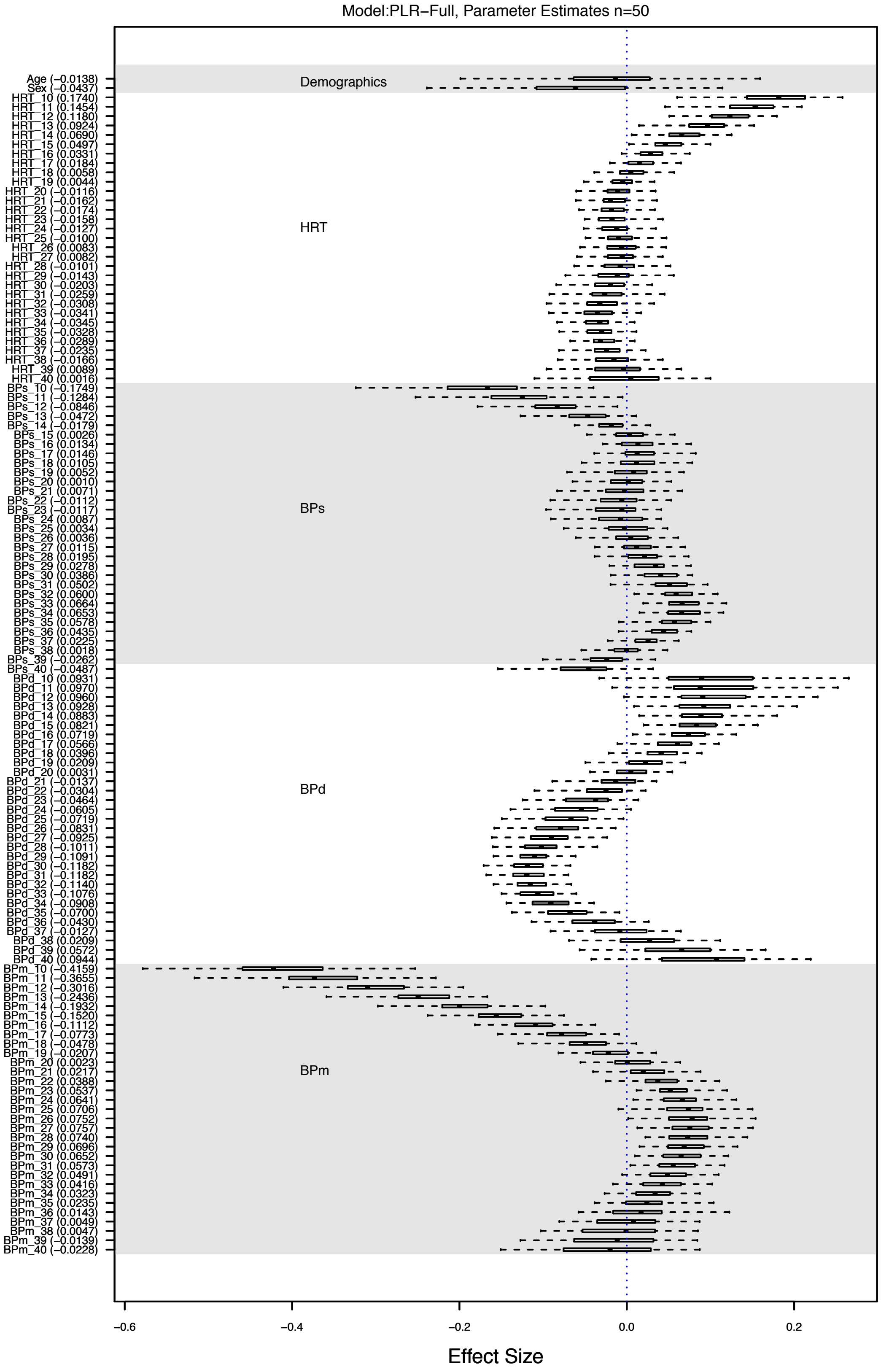


Figure 5.10: Model: PLR-Full, parameter effect size profiles. Effect size is defined as $\beta_j \times$ std dev(x_j)

Table 5.1 details the parameter estimates obtained from 50 runs of the model. Each training run of the model uses a randomly drawn training set from half (approximately 50) of the patients. The model is then tested using the ROC technique on the appropriate test set, containing the other half of the available patients.

Measure	Mean (SE)	Measure	Mean (SE)	Measure	Mean (SE)	Measure	Mean (SE)
HRT_10	0.0088 (0.00038)	BPs_10	-0.0083 (0.00039)	BPd_10	0.0081 (0.00084)	BPm_10	-0.0331 (0.00090)
HRT_11	0.0074 (0.00031)	BPs_11	-0.0061 (0.00032)	BPd_11	0.0082 (0.00074)	BPm_11	-0.0284 (0.00077)
HRT_12	0.0060 (0.00025)	BPs_12	-0.0040 (0.00026)	BPd_12	0.0083 (0.00064)	BPm_12	-0.0238 (0.00065)
HRT_13	0.0047 (0.00021)	BPs_13	-0.0023 (0.00021)	BPd_13	0.0082 (0.00054)	BPm_13	-0.0195 (0.00055)
HRT_14	0.0035 (0.00019)	BPs_14	-0.0009 (0.00018)	BPd_14	0.0078 (0.00046)	BPm_14	-0.0155 (0.00048)
HRT_15	0.0025 (0.00017)	BPs_15	0.0001 (0.00016)	BPd_15	0.0071 (0.00039)	BPm_15	-0.0119 (0.00043)
HRT_16	0.0017 (0.00017)	BPs_16	0.0006 (0.00016)	BPd_16	0.0061 (0.00035)	BPm_16	-0.0087 (0.00040)
HRT_17	0.0009 (0.00017)	BPs_17	0.0007 (0.00017)	BPd_17	0.0048 (0.00033)	BPm_17	-0.0060 (0.00039)
HRT_18	0.0003 (0.00018)	BPs_18	0.0005 (0.00020)	BPd_18	0.0033 (0.00034)	BPm_18	-0.0037 (0.00039)
HRT_19	-0.0002 (0.00019)	BPs_19	0.0002 (0.00021)	BPd_19	0.0018 (0.00036)	BPm_19	-0.0016 (0.00038)
HRT_20	-0.0006 (0.00019)	BPs_20	-0.0000 (0.00022)	BPd_20	0.0003 (0.00037)	BPm_20	0.0002 (0.00038)
HRT_21	-0.0008 (0.00019)	BPs_21	-0.0003 (0.00022)	BPd_21	-0.0012 (0.00039)	BPm_21	0.0017 (0.00038)
HRT_22	-0.0009 (0.00018)	BPs_22	-0.0005 (0.00022)	BPd_22	-0.0026 (0.00040)	BPm_22	0.0031 (0.00038)
HRT_23	-0.0008 (0.00017)	BPs_23	-0.0006 (0.00021)	BPd_23	-0.0039 (0.00040)	BPm_23	0.0042 (0.00038)
HRT_24	-0.0006 (0.00016)	BPs_24	-0.0004 (0.00021)	BPd_24	-0.0052 (0.00040)	BPm_24	0.0050 (0.00037)
HRT_25	-0.0005 (0.00016)	BPs_25	-0.0002 (0.00020)	BPd_25	-0.0063 (0.00039)	BPm_25	0.0056 (0.00037)
HRT_26	-0.0004 (0.00017)	BPs_26	0.0002 (0.00018)	BPd_26	-0.0073 (0.00038)	BPm_26	0.0060 (0.00035)
HRT_27	-0.0004 (0.00018)	BPs_27	0.0005 (0.00017)	BPd_27	-0.0082 (0.00037)	BPm_27	0.0061 (0.00034)
HRT_28	-0.0005 (0.00019)	BPs_28	0.0009 (0.00016)	BPd_28	-0.0090 (0.00035)	BPm_28	0.0059 (0.00034)
HRT_29	-0.0007 (0.00019)	BPs_29	0.0013 (0.00016)	BPd_29	-0.0097 (0.00033)	BPm_29	0.0056 (0.00033)
HRT_30	-0.0010 (0.00019)	BPs_30	0.0018 (0.00017)	BPd_30	-0.0102 (0.00031)	BPm_30	0.0051 (0.00032)
HRT_31	-0.0013 (0.00019)	BPs_31	0.0024 (0.00018)	BPd_31	-0.0103 (0.00029)	BPm_31	0.0046 (0.00032)
HRT_32	-0.0016 (0.00018)	BPs_32	0.0029 (0.00019)	BPd_32	-0.0101 (0.00029)	BPm_32	0.0039 (0.00031)
HRT_33	-0.0017 (0.00017)	BPs_33	0.0032 (0.00019)	BPd_33	-0.0093 (0.00031)	BPm_33	0.0033 (0.00031)
HRT_34	-0.0018 (0.00016)	BPs_34	0.0031 (0.00018)	BPd_34	-0.0080 (0.00035)	BPm_34	0.0026 (0.00033)
HRT_35	-0.0017 (0.00014)	BPs_35	0.0028 (0.00016)	BPd_35	-0.0061 (0.00041)	BPm_35	0.0019 (0.00037)
HRT_36	-0.0015 (0.00014)	BPs_36	0.0021 (0.00014)	BPd_36	-0.0038 (0.00048)	BPm_36	0.0012 (0.00043)
HRT_37	-0.0012 (0.00016)	BPs_37	0.0011 (0.00013)	BPd_37	-0.0011 (0.00057)	BPm_37	0.0004 (0.00052)
HRT_38	-0.0008 (0.00021)	BPs_38	-0.0001 (0.00017)	BPd_38	0.0019 (0.00067)	BPm_38	-0.0004 (0.00061)
HRT_39	-0.0005 (0.00028)	BPs_39	-0.0012 (0.00024)	BPd_39	0.0050 (0.00077)	BPm_39	-0.0011 (0.00072)
HRT_40	-0.0001 (0.00037)	BPs_40	-0.0023 (0.00032)	BPd_40	0.0082 (0.00089)	BPm_40	-0.0018 (0.00083)

Table 5.1: Parameter coefficients for model: PLR-Full

5.3.2.4 Sequential Monte Carlo (SMC) Parameter Estimation

As described in Section 3.5, an alternative method of estimating the parameters of the PLR-Full model was carried out. The alternative method uses a fully Bayesian model, including a Bayesian estimate of the penalty parameter λ. Sampling from the posterior is performed using the Sequential Monte Carlo algorithm. The results, to be compared with the maximum likelihood estimates from Section 5.3.2, are presented in Figures 5.11 and 5.12 and Table 5.2. In addition, diagnostics plots from the SMC process are provided in Figures 5.13 to 5.15.

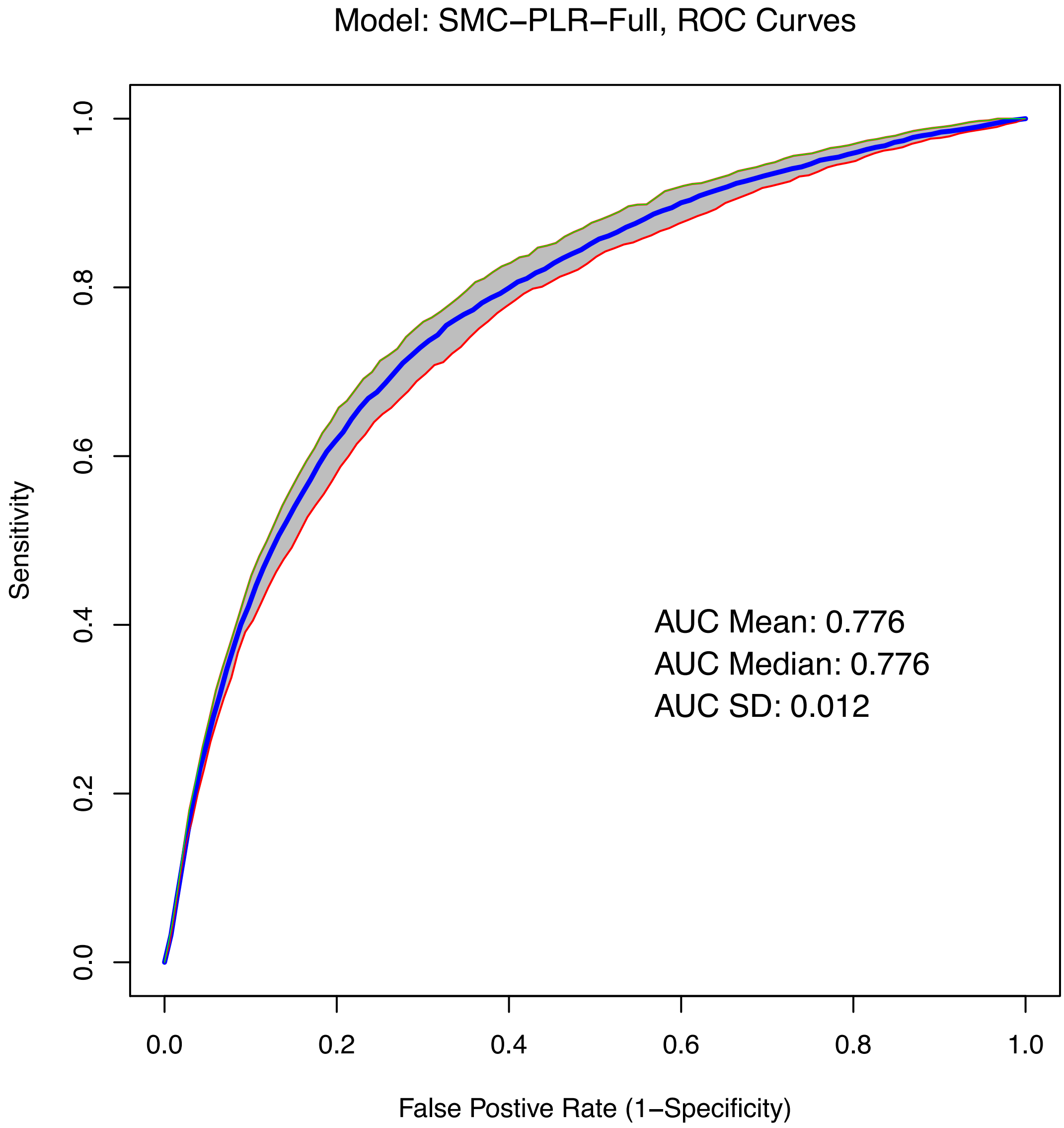


Figure 5.11: Model: PLR-Full, SMC parameter estimation. ROC and AUC detail. This plot is generated from 50 runs of the model. Each run consists of a training and test set. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

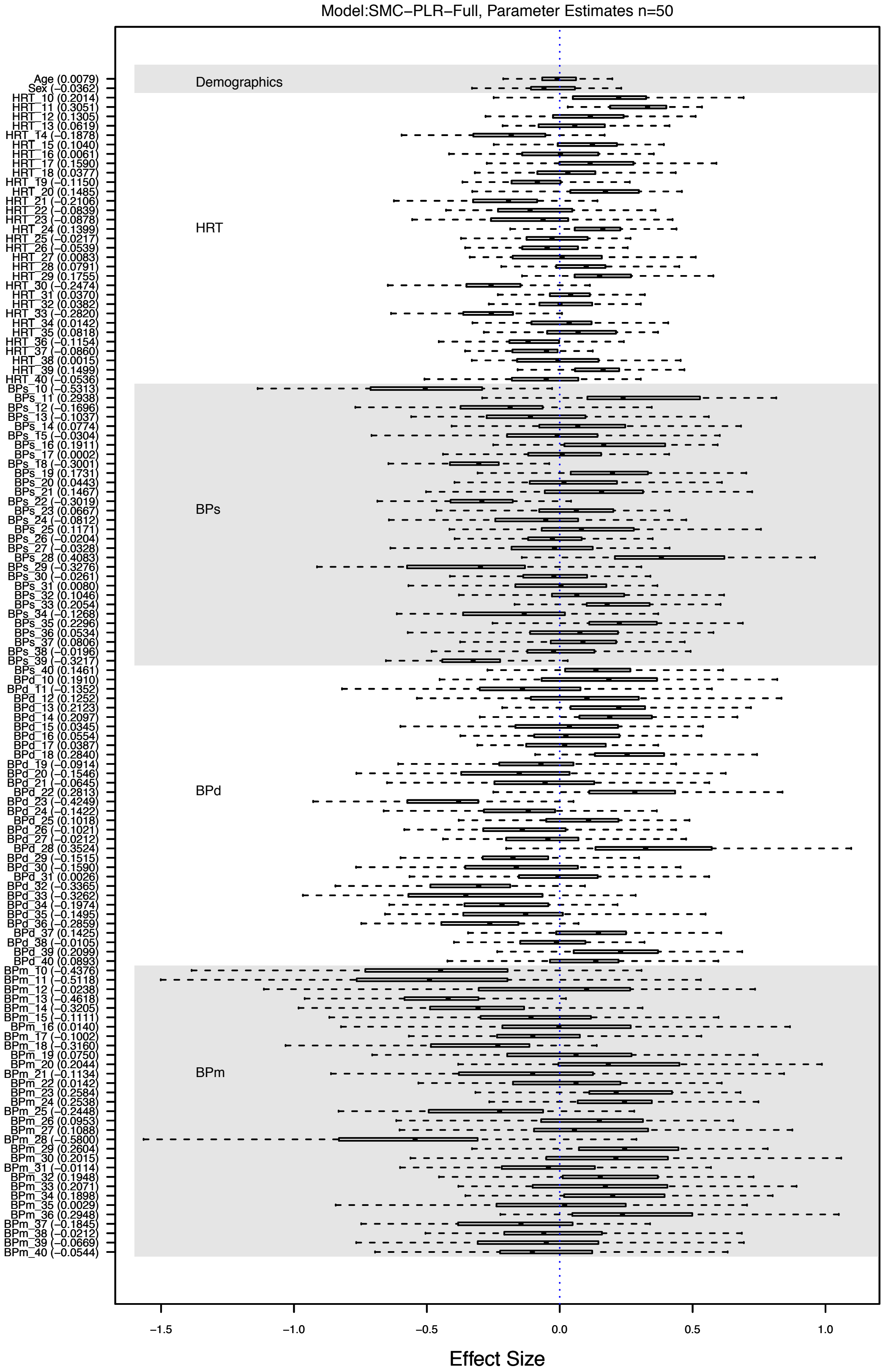


Figure 5.12: Model: PLR-Full, SMC parameter estimation, parameter effect size profiles. Effect size is defined as $\beta_j \times \text{std dev}(x_j)$

Table 5.2 details the parameter estimates obtained from 50 runs of the model. Each training run of the model uses a randomly drawn training set from half (approximately 50) of the patients. The model is tested using the ROC technique on the appropriate test set, containing the other half of the available patients.

Measure	Mean (SE)	Measure	Mean (SE)	Measure	Mean (SE)	Measure	Mean (SE)
HRT_10	0.0102 (0.00145)	BPs_10	-0.0253 (0.00202)	BPd_10	0.0166 (0.00374)	BPm_10	-0.0348 (0.00539)
HRT_11	0.0155 (0.00113)	BPs_11	0.0139 (0.00180)	BPd_11	-0.0115 (0.00404)	BPm_11	-0.0398 (0.00493)
HRT_12	0.0066 (0.00142)	BPs_12	-0.0081 (0.00209)	BPd_12	0.0108 (0.00419)	BPm_12	-0.0019 (0.00521)
HRT_13	0.0031 (0.00129)	BPs_13	-0.0050 (0.00188)	BPd_13	0.0187 (0.00274)	BPm_13	-0.0370 (0.00336)
HRT_14	-0.0096 (0.00141)	BPs_14	0.0037 (0.00206)	BPd_14	0.0185 (0.00311)	BPm_14	-0.0258 (0.00377)
HRT_15	0.0053 (0.00115)	BPs_15	-0.0014 (0.00195)	BPd_15	0.0030 (0.00304)	BPm_15	-0.0087 (0.00413)
HRT_16	0.0003 (0.00140)	BPs_16	0.0091 (0.00164)	BPd_16	0.0047 (0.00321)	BPm_16	0.0011 (0.00397)
HRT_17	0.0081 (0.00156)	BPs_17	-0.0000 (0.00160)	BPd_17	0.0033 (0.00257)	BPm_17	-0.0078 (0.00331)
HRT_18	0.0019 (0.00123)	BPs_18	-0.0142 (0.00128)	BPd_18	0.0240 (0.00273)	BPm_18	-0.0245 (0.00304)
HRT_19	-0.0058 (0.00145)	BPs_19	0.0082 (0.00170)	BPd_19	-0.0079 (0.00350)	BPm_19	0.0059 (0.00435)
HRT_20	0.0075 (0.00132)	BPs_20	0.0021 (0.00182)	BPd_20	-0.0135 (0.00350)	BPm_20	0.0163 (0.00447)
HRT_21	-0.0107 (0.00141)	BPs_21	0.0070 (0.00183)	BPd_21	-0.0057 (0.00372)	BPm_21	-0.0091 (0.00441)
HRT_22	-0.0043 (0.00139)	BPs_22	-0.0144 (0.00115)	BPd_22	0.0241 (0.00312)	BPm_22	0.0011 (0.00350)
HRT_23	-0.0045 (0.00144)	BPs_23	0.0032 (0.00140)	BPd_23	-0.0361 (0.00310)	BPm_23	0.0201 (0.00390)
HRT_24	0.0071 (0.00107)	BPs_24	-0.0038 (0.00154)	BPd_24	-0.0121 (0.00293)	BPm_24	0.0199 (0.00276)
HRT_25	-0.0011 (0.00115)	BPs_25	0.0056 (0.00179)	BPd_25	0.0089 (0.00309)	BPm_25	-0.0195 (0.00327)
HRT_26	-0.0027 (0.00122)	BPs_26	-0.0010 (0.00140)	BPd_26	-0.0089 (0.00323)	BPm_26	0.0076 (0.00358)
HRT_27	0.0004 (0.00141)	BPs_27	-0.0016 (0.00161)	BPd_27	-0.0019 (0.00328)	BPm_27	0.0087 (0.00378)
HRT_28	0.0040 (0.00111)	BPs_28	0.0193 (0.00179)	BPd_28	0.0313 (0.00356)	BPm_28	-0.0465 (0.00443)
HRT_29	0.0089 (0.00121)	BPs_29	-0.0156 (0.00182)	BPd_29	-0.0135 (0.00293)	BPm_29	0.0210 (0.00385)
HRT_30	-0.0126 (0.00114)	BPs_30	-0.0012 (0.00153)	BPd_30	-0.0137 (0.00351)	BPm_30	0.0159 (0.00379)
HRT_31	0.0019 (0.00096)	BPs_31	-0.0004 (0.00152)	BPd_31	-0.0002 (0.00284)	BPm_31	-0.0009 (0.00331)
HRT_32	0.0020 (0.00108)	BPs_32	0.0050 (0.00148)	BPd_32	-0.0297 (0.00267)	BPm_32	0.0157 (0.00328)
HRT_33	-0.0144 (0.00127)	BPs_33	0.0098 (0.00139)	BPd_33	-0.0282 (0.00373)	BPm_33	0.0163 (0.00351)
HRT_34	0.0007 (0.00122)	BPs_34	-0.0061 (0.00171)	BPd_34	-0.0174 (0.00301)	BPm_34	0.0152 (0.00326)
HRT_35	0.0041 (0.00119)	BPs_35	0.0111 (0.00177)	BPd_35	-0.0131 (0.00333)	BPm_35	0.0002 (0.00357)
HRT_36	-0.0059 (0.00135)	BPs_36	0.0026 (0.00174)	BPd_36	-0.0253 (0.00289)	BPm_36	0.0239 (0.00360)
HRT_37	-0.0044 (0.00099)	BPs_37	0.0039 (0.00134)	BPd_37	0.0125 (0.00275)	BPm_37	-0.0148 (0.00336)
HRT_38	0.0001 (0.00136)	BPs_38	-0.0009 (0.00153)	BPd_38	-0.0009 (0.00206)	BPm_38	-0.0017 (0.00302)
HRT_39	0.0077 (0.00118)	BPs_39	-0.0153 (0.00138)	BPd_39	0.0183 (0.00285)	BPm_39	-0.0054 (0.00367)
HRT_40	-0.0027 (0.00131)	BPs_40	0.0070 (0.00131)	BPd_40	0.0078 (0.00292)	BPm_40	-0.0043 (0.00380)

Table 5.2: Parameter coefficients for model: SMC-PLR-Full

5.3.2.5 SMC-PLR-Full, Algorithm diagnostics

The internal operation of the SMC algorithm can be examined to ensure that the procedure is working efficiently. The diagnostic plots in Figures 5.13 through 5.15 show that the SMC technique is performing as expected. This technique works best when the starting point is close to the target distribution and that is ensured by using the coefficients from the maximum likelihood based model of Section 5.3.2. The SMC technique proceeds to move in small stages from the starting distribution and this can be seen in Figure 5.13 where the ESS gradually gets less as a group of particles takes on most of the weight.

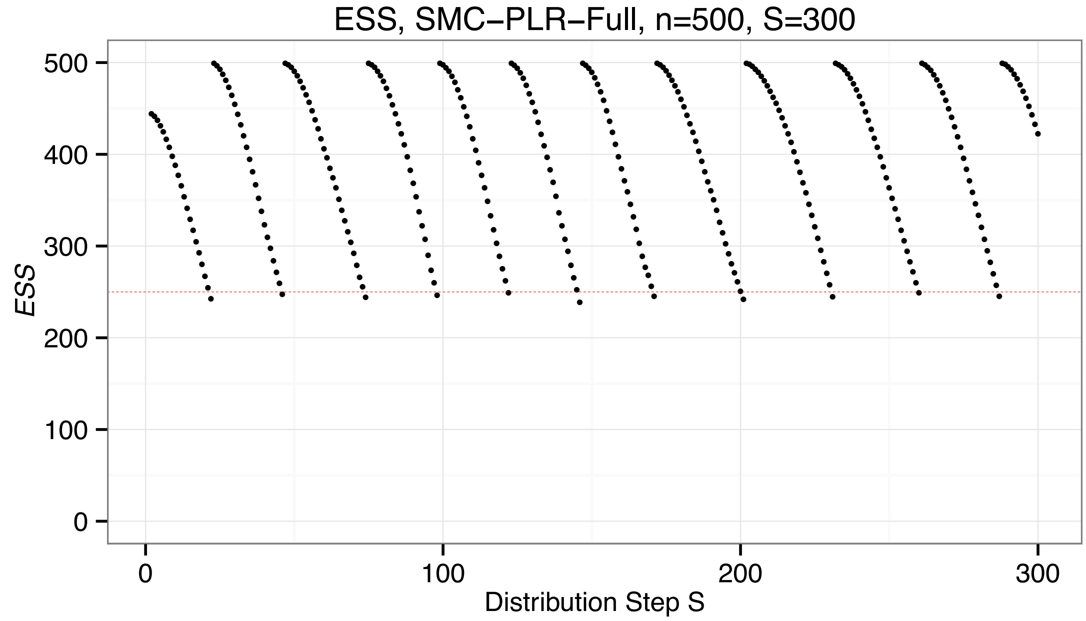


Figure 5.13: Model: PLR-Full, SMC parameter estimation,SMC ESS Diagnostics. S is the number of auxiliary distributions, n is the number of particles. The start distribution is good as the initial ESS value is well above the 50% threshold which triggers a resample procedure.

This deterioration of the particles can also be seen in Figure 5.15, where some of the particles (columns in the plot) begin to move colour from darker blue to lighter colours which indicates that the particle is becoming too dominant within the distribution. Once particles dominate the distribution a resample is performed to smooth out the distribution.

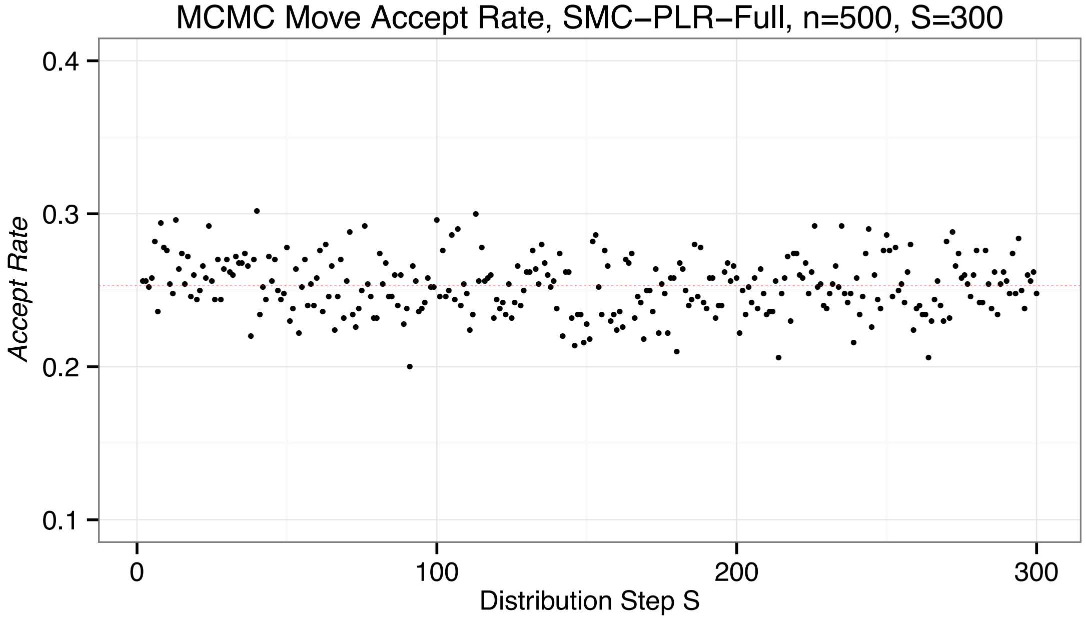


Figure 5.14: Model: PLR-Full, SMC parameter estimation, SMC MCMC Move Accept Rate Diagnostics. This plot shows an evenly distributed accept rate with a mean value of 0.25. The acceptance rate is controlled by τ_j^ν, the Metropolis-Hasting tuning parameter, which for this model is set to 0.2

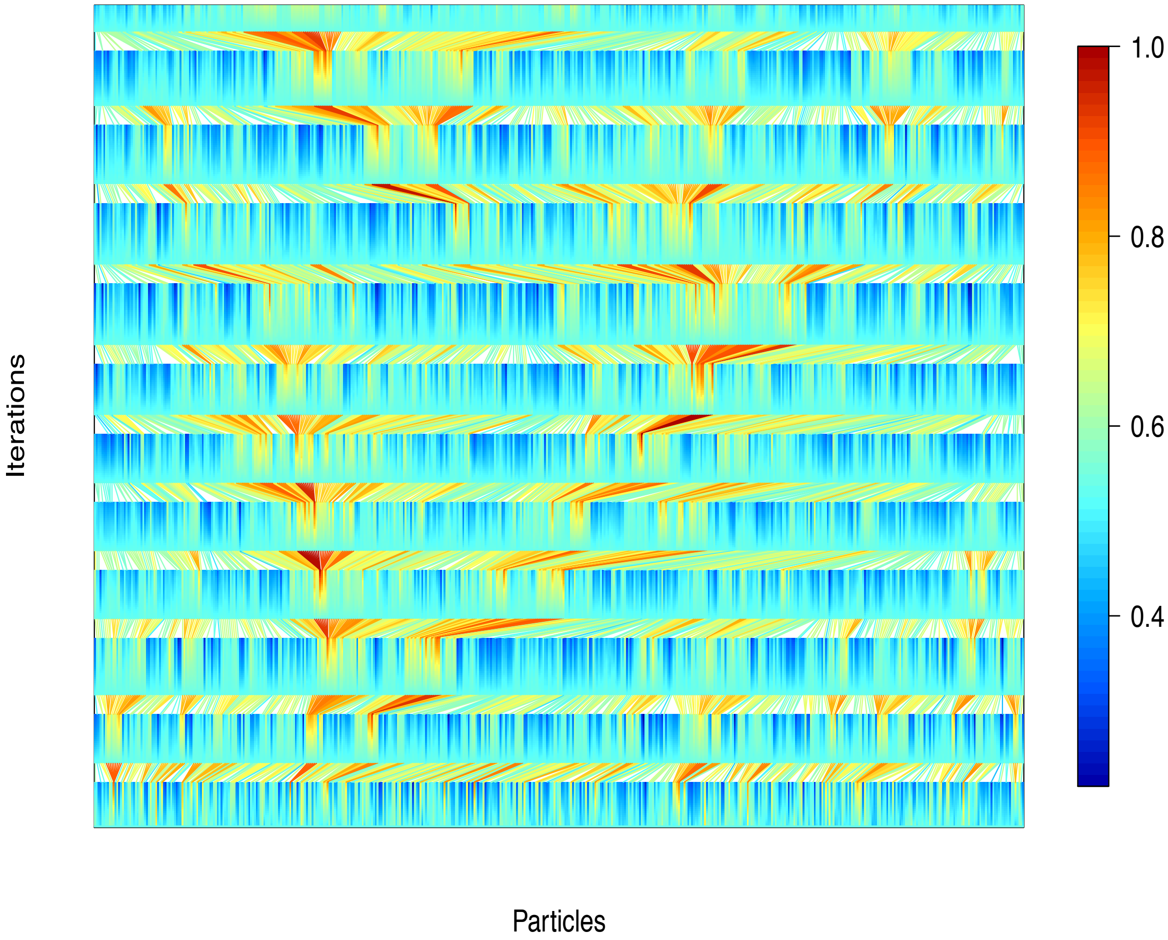


Figure 5.15: Model: PLR-Full, SMC parameter estimation, particle weights across auxiliary distributions. This plot shows the particle weights as the auxiliary distribution moves from the start weights at the bottom of the plot (set using penalised logistic regression) to the final target distribution at the top of the plot. Each band of angled lines is a resample procedure which is triggered as a group of weights begins to dominate (moves from dark blue towards lighter then dark red colours). The colour key is a relative scale with the range being from dark blue for low weights to red for high weights.

5.3.2.6 Model: PLR-Full, Summary

Interpreting the coefficients of the model shows that for the variables HRT, BPs and BPm the initial five to ten minutes of data have the strongest effect. The coefficients for the remaining 20 minutes of data oscillate around zero. The coefficients for BPd show a pattern that starts negative for the first ten minutes in line with the other variables but then swings positive for the following 15 minutes before swinging negative for the final five minutes. The overall effect across the 31 minute data being approximately zero.

The HRT signal has a positive effect over its initial phase i.e. an increase in HRT will tend to increase the probability of a hypotensive episode. The two blood pressure signals BPs and BPm both have negative effects i.e. a decrease in BPs, which would tend to cause a decrease in BPm, will increase the chance of a hypotensive episode starting. The effect is stronger for the BPm signal.

Using the SMC technique, the coefficient profile from Figure 5.12 does not show as distinct a pattern as the PLR-Full model. The penalty effect, which is estimated from the data, is clearly not as strong thereby allowing a larger variation of the coefficient values. The AUC value produced when using the parameters estimated by SMC is slightly down, at 0.776 vs. 0.788, this is most likely due to the reduced penalty effect.

5.3.3 Minimum signals, Model name: PLR-Min

The PLR-Min model is the model suggested by clinical colleagues. It utilises data from the BrainIT database which reflects the signals that are available from a typical installation of ICU bedside monitors. Full details of the available signals are provided in the medical background, Section 2.4. This model uses two demographic signals and the two measured signals. The demographic signals are age and gender. The measured signals are: heart rate (HRT); and mean arterial blood pressure (BPm). For each of the two measured signals, this model uses each available minute from a 31 minute buffer as inputs. This gives a total of 65 input parameters which must be estimated. Again, the clinical basis for this model comes from discussions with colleagues and builds on the work of the AvertIT project, (Donald et al., 2012b). Formally the model is defined as:

$$\begin{aligned}\log\left(\frac{p_i}{1-p_i}\right) &= \beta_0 + \beta_1 Age_i + \beta_2 Gender_i \\ &+ \beta_3 HRT\_10_i + \beta_4 HRT\_11_i + \beta_5 HRT\_12_i + \ldots + \beta_{33} HRT\_40_i \\ &+ \beta_{34} BPm\_10_i + \beta_{35} BPm\_11_i + \beta_{36} BPm\_12_i + \ldots + \beta_{64} BPm\_40_i \end{aligned} \tag{5.6}$$

with

$$\begin{aligned}\mathscr{L}(\boldsymbol{\beta})_{penalised} &= \mathscr{L}(\boldsymbol{\beta}) - \lambda([(\beta_4 - \beta_3) - (\beta_5 - \beta_4)]^2) \\ &- \vdots \\ &- \lambda([(\beta_{63} - \beta_{62})) - (\beta_{64} - \beta_{63})]^2) \end{aligned} \tag{5.7}$$

The model performance, as measured using the ROC technique, is presented in Figure 5.16. The profile of the model coefficient values is presented in Figure 5.3.3.2. The full summary of the parameter values is provided in Table 5.3.

5.3.3.1 Model Performance ROC curves (Model: PLR-Min)

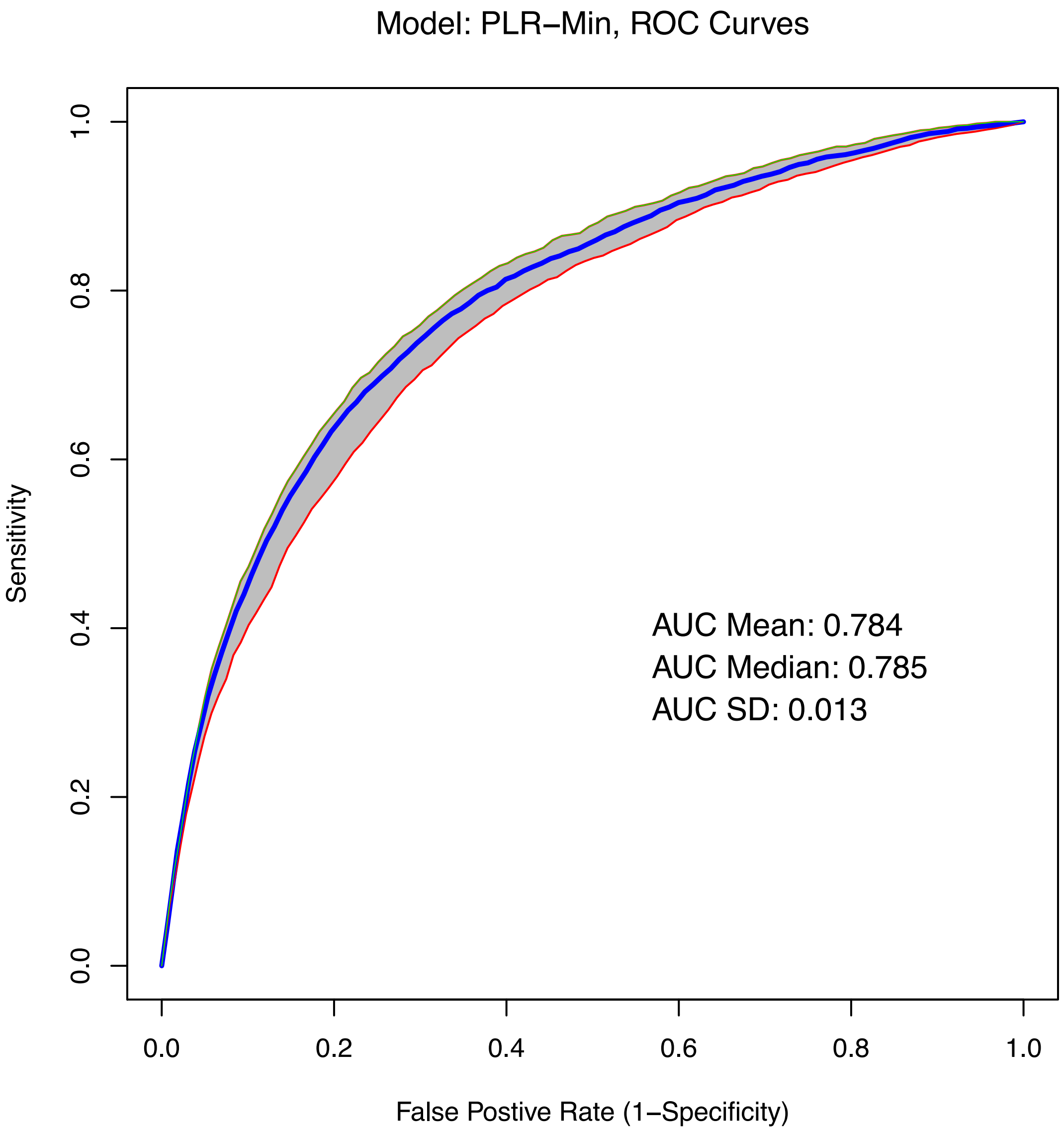


Figure 5.16: Model: PLR-Min, ROC and AUC detail. This plot is generated from 50 runs of the model. Each run consists of a training and test set. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

5.3.3.2 Estimation of lambda (Model: PLR-Min)

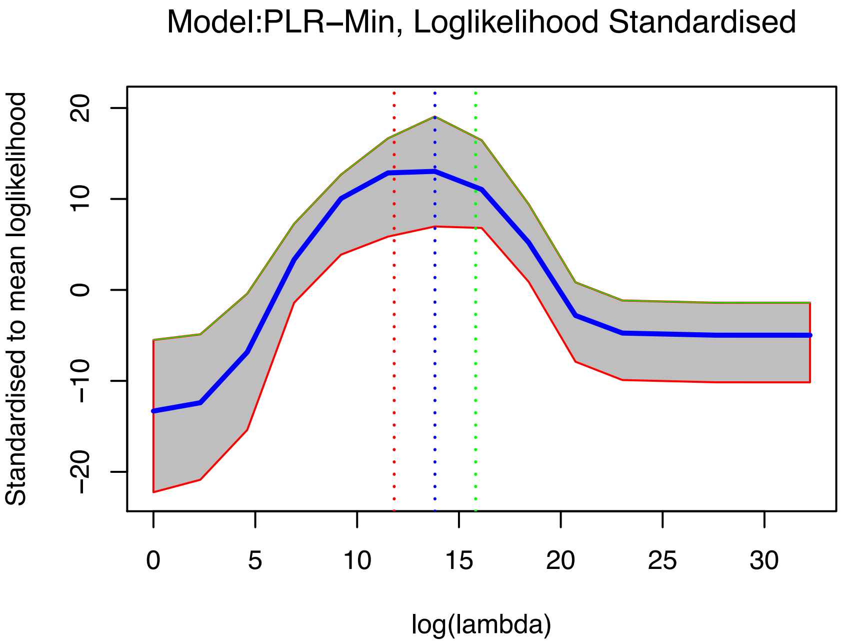


Figure 5.17: PLR-Min: Coarse search for optimal λ. The coarse search using ten iterations returns a value of 13.82 for λ. The y-axis shows the deviation from the mean loglikelihood at each λ setting across the ten iterations. This accounts for the different absolute values of loglikelihood due to the different data sets.

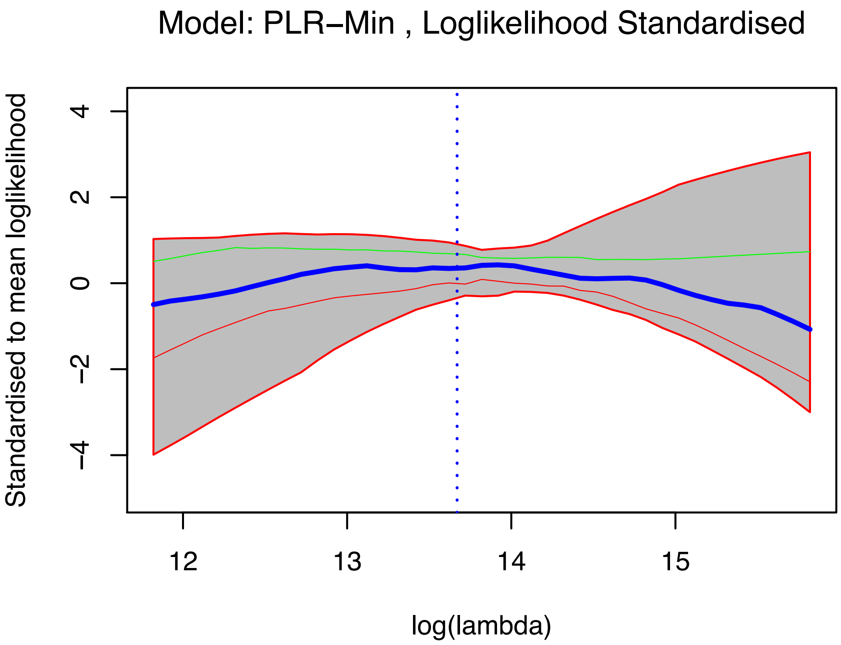


Figure 5.18: PLR-Min: Fine grain search for optimal λ. The fine grain search procedure using 30 iterations returns a value of 13.67 for λ. The y-axis shows the deviation from the mean loglikelihood at each λ setting across the 30 iterations.

5.3.3.3 Parameter Profiles (Model: PLR-Min)

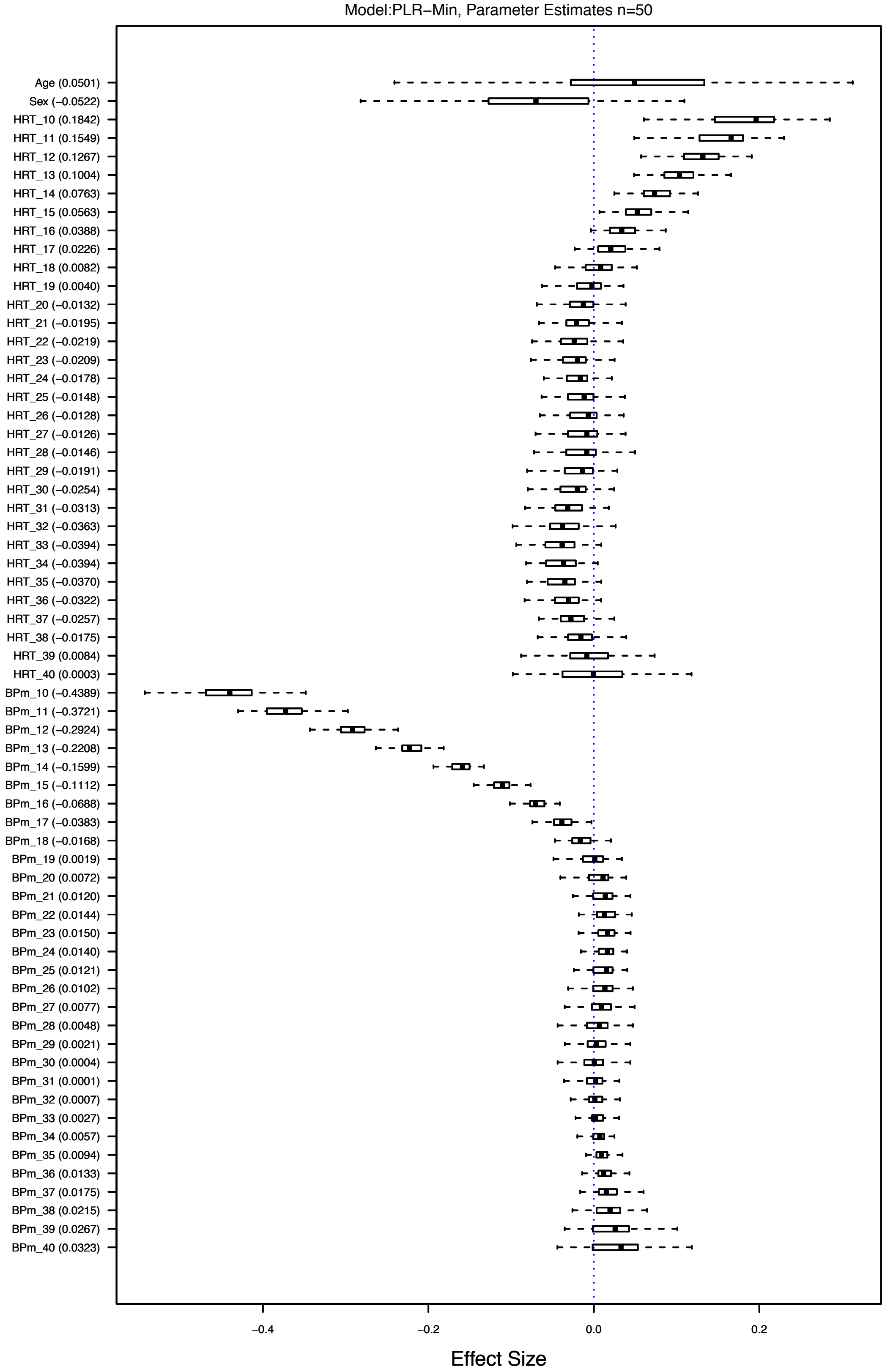


Figure 5.19: Model: PLR-Min, parameter effect size profiles. Effect size is defined as $\beta_j \times$ std dev(x_j)

Table 5.3 details the parameter estimates obtained from 50 runs of the model. Each training run of the model is uses a randomly drawn training set from half (approximately 50) of the patients. The model is then tested using the ROC technique on the appropriate test set, containing the other half of the available patients.

Measure	Mean (SE)	Measure	Mean (SE)
HRT_10	0.0113 (0.00049)	BPm_10	-0.0408 (0.00053)
HRT_11	0.0090 (0.00034)	BPm_11	-0.0319 (0.00039)
HRT_12	0.0066 (0.00027)	BPm_12	-0.0235 (0.00031)
HRT_13	0.0045 (0.00029)	BPm_13	-0.0163 (0.00028)
HRT_14	0.0029 (0.00031)	BPm_14	-0.0103 (0.00026)
HRT_15	0.0020 (0.00030)	BPm_15	-0.0057 (0.00024)
HRT_16	0.0014 (0.00028)	BPm_16	-0.0027 (0.00024)
HRT_17	0.0008 (0.00028)	BPm_17	-0.0013 (0.00026)
HRT_18	0.0001 (0.00029)	BPm_18	-0.0007 (0.00029)
HRT_19	-0.0006 (0.00030)	BPm_19	-0.0003 (0.00030)
HRT_20	-0.0011 (0.00030)	BPm_20	-0.0000 (0.00030)
HRT_21	-0.0015 (0.00028)	BPm_21	0.0001 (0.00027)
HRT_22	-0.0016 (0.00026)	BPm_22	0.0004 (0.00023)
HRT_23	-0.0012 (0.00024)	BPm_23	0.0005 (0.00023)
HRT_24	-0.0006 (0.00023)	BPm_24	0.0007 (0.00025)
HRT_25	-0.0001 (0.00023)	BPm_25	0.0007 (0.00027)
HRT_26	0.0002 (0.00024)	BPm_26	0.0008 (0.00027)
HRT_27	0.0003 (0.00025)	BPm_27	0.0007 (0.00027)
HRT_28	0.0001 (0.00026)	BPm_28	0.0004 (0.00027)
HRT_29	-0.0004 (0.00027)	BPm_29	0.0000 (0.00026)
HRT_30	-0.0012 (0.00027)	BPm_30	-0.0002 (0.00025)
HRT_31	-0.0019 (0.00027)	BPm_31	-0.0002 (0.00022)
HRT_32	-0.0025 (0.00026)	BPm_32	0.0000 (0.00020)
HRT_33	-0.0029 (0.00026)	BPm_33	0.0003 (0.00017)
HRT_34	-0.0028 (0.00026)	BPm_34	0.0007 (0.00016)
HRT_35	-0.0024 (0.00027)	BPm_35	0.0011 (0.00017)
HRT_36	-0.0020 (0.00026)	BPm_36	0.0014 (0.00019)
HRT_37	-0.0014 (0.00023)	BPm_37	0.0015 (0.00022)
HRT_38	-0.0007 (0.00024)	BPm_38	0.0016 (0.00027)
HRT_39	0.0001 (0.00034)	BPm_39	0.0020 (0.00037)
HRT_40	0.0008 (0.00051)	BPm_40	0.0026 (0.00053)

Table 5.3: Parameter coefficients for model: PLR-Min

5.3.3.4 Sequential Monte Carlo (SMC) Parameter Estimation

As described in Section 3.5, an alternative method of estimating the parameters of the PLR-Min model was carried out. This alternative method uses the technique of Sequential Monte Carlo for static problems. The method uses a Bayesian framework and a series of joint distributions of parameters to explore possible parameter combinations for the model. The results, to be compared with the maximum likelihood estimates from Section 5.3.2, are presented in Figures 5.20 and 5.21 and Table 5.4. In addition, diagnostics plots from the SMC process are provided in Figures 5.22 to 5.24.

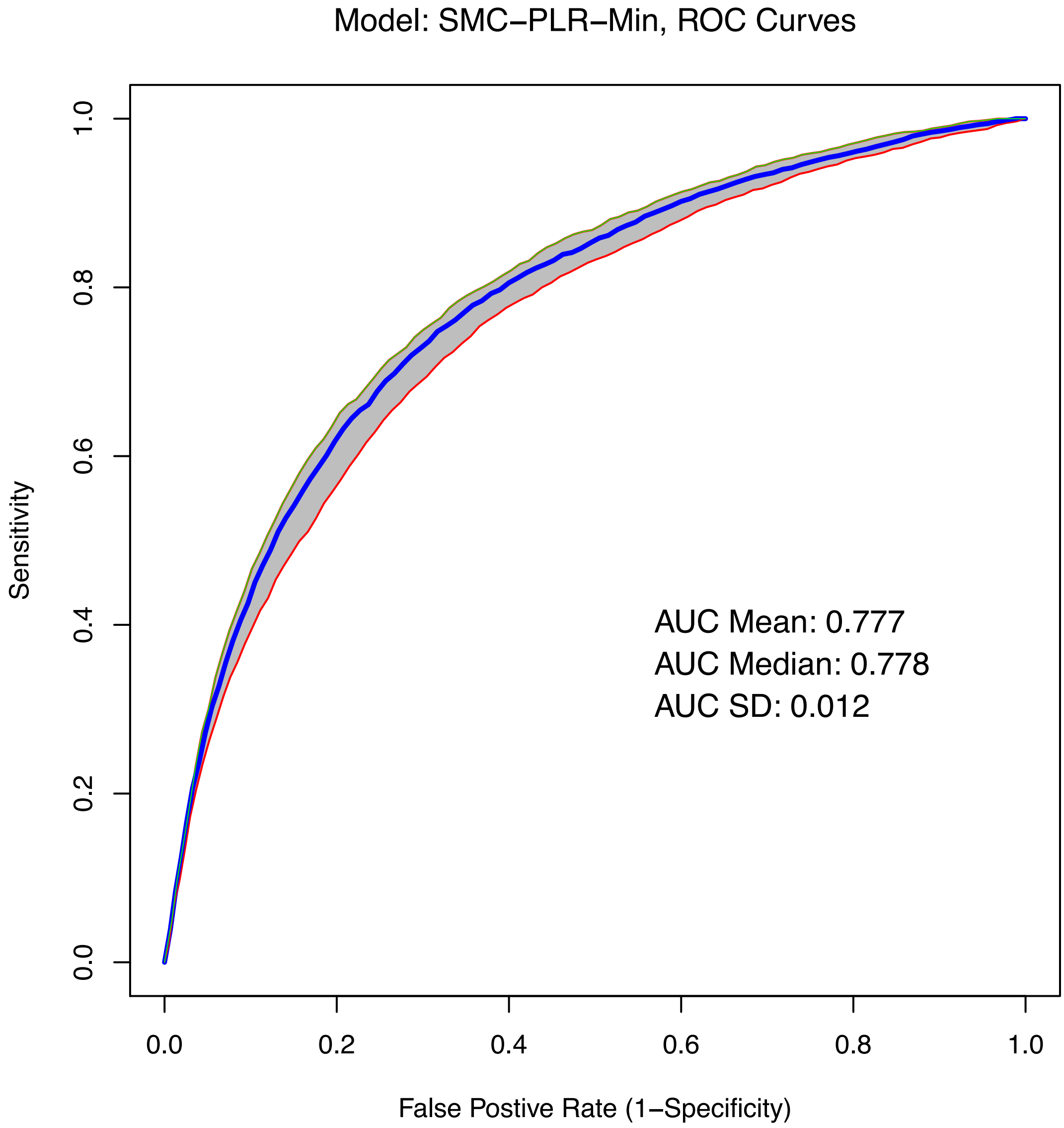


Figure 5.20: Model: PLR-Min, SMC parameter estimation. ROC and AUC detail. This plot is generated from 50 runs of the model. Each run consists of a training and test set. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

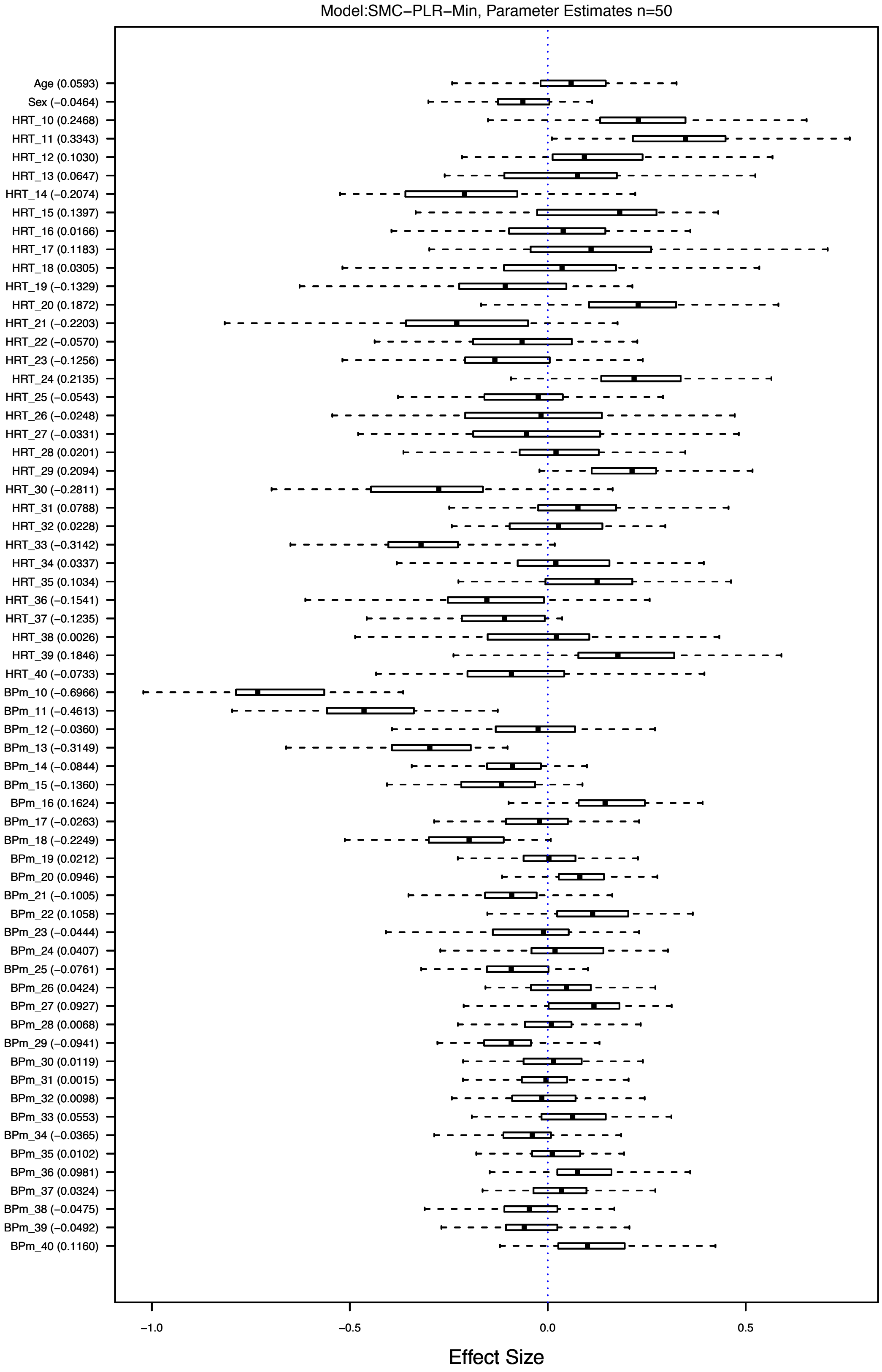


Figure 5.21: Model: PLR-Min, SMC parameter estimation, parameter effect size profiles. Effect size is defined as $\beta_j \times$ std dev(x_j)

Table 5.4 details the parameter estimates obtained from 50 runs of the model. Each training run of the model is uses a randomly drawn training set from half (approximately 50) of the patients. The model is then tested using the ROC technique on the appropriate test set, containing the other half of the available patients.

Measure	Mean (SE)	Measure	Mean (SE)
HRT_10	0.0125 (0.00129)	BPm_10	-0.0554 (0.00170)
HRT_11	0.0170 (0.00122)	BPm_11	-0.0358 (0.00166)
HRT_12	0.0053 (0.00150)	BPm_12	-0.0028 (0.00182)
HRT_13	0.0033 (0.00133)	BPm_13	-0.0252 (0.00166)
HRT_14	-0.0106 (0.00135)	BPm_14	-0.0068 (0.00160)
HRT_15	0.0071 (0.00133)	BPm_15	-0.0107 (0.00171)
HRT_16	0.0008 (0.00133)	BPm_16	0.0128 (0.00146)
HRT_17	0.0060 (0.00145)	BPm_17	-0.0021 (0.00132)
HRT_18	0.0015 (0.00142)	BPm_18	-0.0175 (0.00154)
HRT_19	-0.0068 (0.00172)	BPm_19	0.0017 (0.00159)
HRT_20	0.0095 (0.00139)	BPm_20	0.0075 (0.00111)
HRT_21	-0.0112 (0.00153)	BPm_21	-0.0081 (0.00135)
HRT_22	-0.0029 (0.00136)	BPm_22	0.0084 (0.00148)
HRT_23	-0.0064 (0.00151)	BPm_23	-0.0034 (0.00159)
HRT_24	0.0108 (0.00115)	BPm_24	0.0032 (0.00137)
HRT_25	-0.0028 (0.00127)	BPm_25	-0.0060 (0.00113)
HRT_26	-0.0013 (0.00157)	BPm_26	0.0034 (0.00131)
HRT_27	-0.0017 (0.00158)	BPm_27	0.0074 (0.00145)
HRT_28	0.0010 (0.00117)	BPm_28	-0.0005 (0.00120)
HRT_29	0.0107 (0.00099)	BPm_29	-0.0076 (0.00118)
HRT_30	-0.0143 (0.00142)	BPm_30	0.0009 (0.00118)
HRT_31	0.0040 (0.00118)	BPm_31	0.0001 (0.00117)
HRT_32	0.0012 (0.00103)	BPm_32	-0.0008 (0.00127)
HRT_33	-0.0161 (0.00120)	BPm_33	0.0044 (0.00129)
HRT_34	0.0017 (0.00127)	BPm_34	-0.0029 (0.00129)
HRT_35	0.0052 (0.00122)	BPm_35	0.0008 (0.00114)
HRT_36	-0.0078 (0.00152)	BPm_36	0.0080 (0.00136)
HRT_37	-0.0063 (0.00086)	BPm_37	0.0026 (0.00118)
HRT_38	0.0001 (0.00146)	BPm_38	-0.0038 (0.00113)
HRT_39	0.0094 (0.00131)	BPm_39	-0.0039 (0.00138)
HRT_40	-0.0037 (0.00125)	BPm_40	0.0093 (0.00142)

Table 5.4: Parameter coefficients for model: SMC-PLR-Min

5.3.3.5 SMC-PLR-Min, Algorithm diagnostics

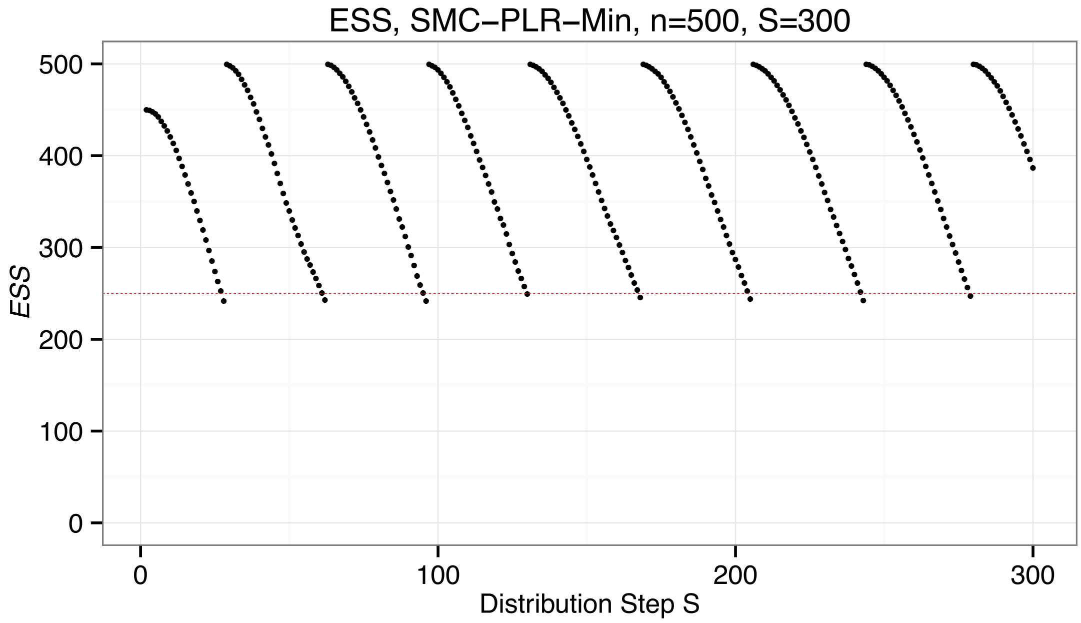


Figure 5.22: Model: PLR-Min, SMC parameter estimation,SMC ESS Diagnostics. S is the number of auxiliary distributions, n is the number of particles. Again, the start distribution is good as the initial ESS value is well above the 50% threshold which triggers a resample procedure.

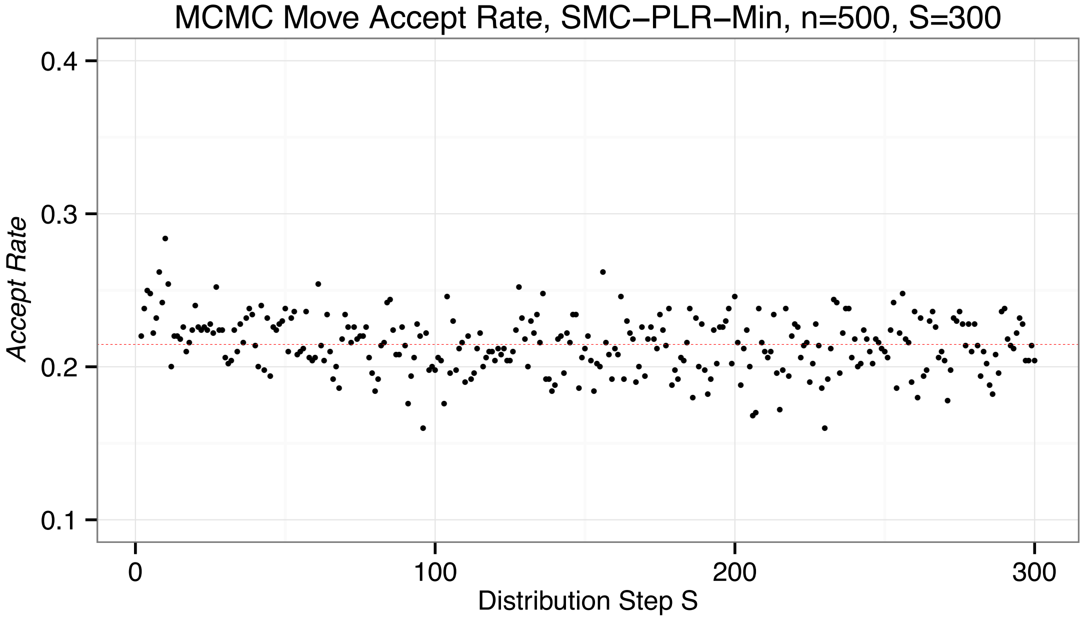


Figure 5.23: Model: PLR-Min, SMC parameter estimation,SMC MCMC Move Accept Rate Diagnostics.This plot shows an evenly distributed accept rate with a mean value of 0.21.

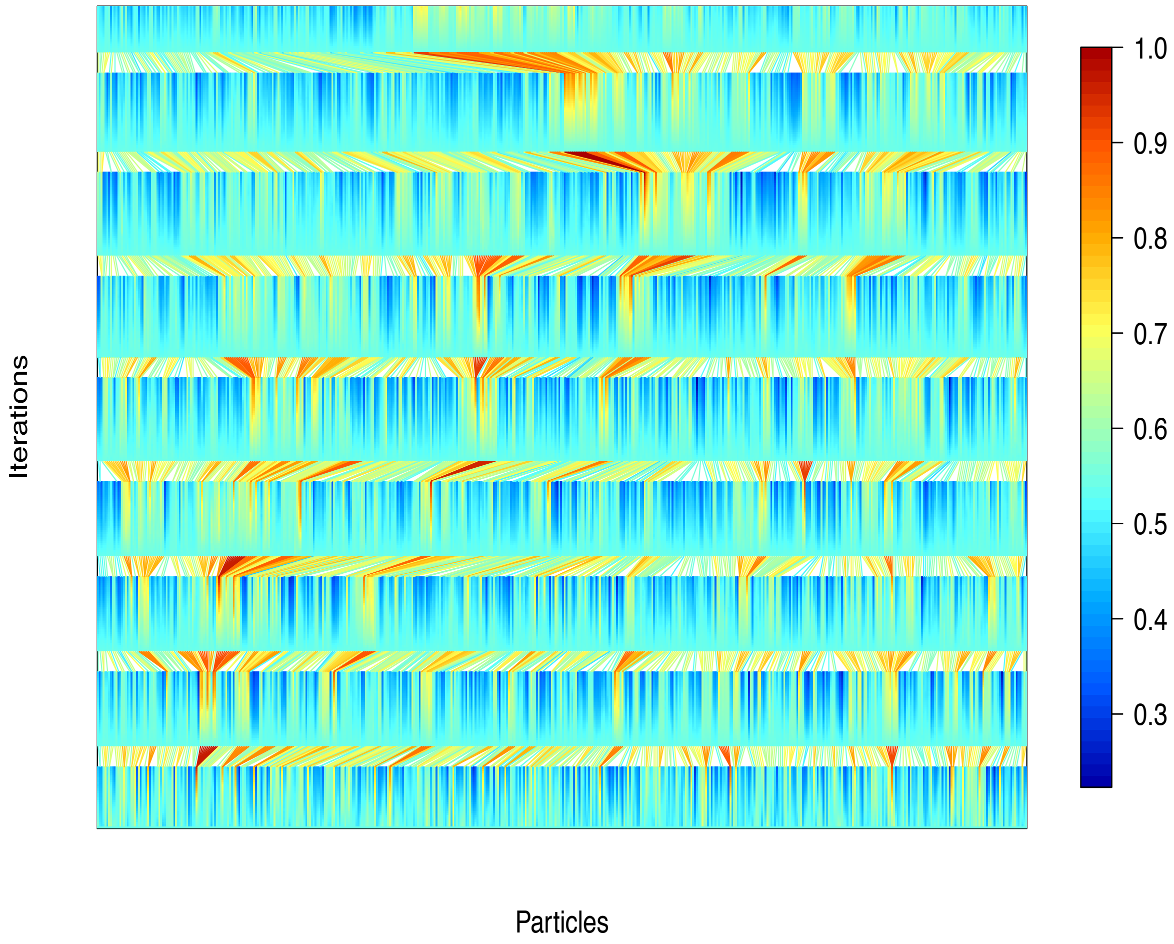


Figure 5.24: Model: PLR-Min, SMC parameter estimation, particle weights across auxiliary distributions. This plot shows the particle weights as the auxiliary distribution moves from the start weights at the bottom of the plot (set using penalised logistic regression) to the final target distribution at the top of the plot. Each band of angled lines is a resample procedure which is triggered as a group of weights begins to dominate (moves from dark blue towards lighter then dark red colours). The colour key is a relative scale with the range being from dark blue for low weights to red for high weights.

5.3.3.6 Model: PLR-Min, Summary

This model performs very closely to the PLR-Full model. The median AUC of 0.785 is only fractions of a percentage point lower than the PLR-Full median AUC of 0.788. Interpreting the coefficients of the model shows that again, for the variables HRT and BPm, the initial five to ten minutes of data have the strongest effect. The coefficients for the remaining 20 minutes of data oscillate around zero.

The same pattern as the PLR-Full model is repeated for the HRT and BPm signals. The HRT has a positive effect, an increase in HRT tends to increase the probability of a hypotensive episode. The BPm variable has a negative effect i.e. a decrease in BPm will increase the chance of a hypotensive episode starting.

For the PLR-Min model, the estimation of the parameters using the SMC technique has caused a slight decrease in AUC performance although not as big a drop (0.7% vs. 1.2%) as in the PLR-Full model. The penalty effect, again estimated from the data, is not as strong as that determined by the two-pass vector approach. This results in a coefficient profile that contains more parameters with non-zero values. The diagnostics show that the SMC algorithm is performing as expected albeit resulting in marginally lower performance.

5.3.4 Models using each minute of data, Summary

Section 5.3 has presented two penalised logistic regression models using two methods to estimate the parameters of the models. The results are very similar with the range of AUC values being 0.788 to 0.776. Both models show that the initial five to ten minutes of data are the main contributors to the probability output of the model. The sequential Monte Carlo technique produced similar values to the more usual maximum likelihood approach albeit using more complex computational techniques.

5.4 Models using statistical measures

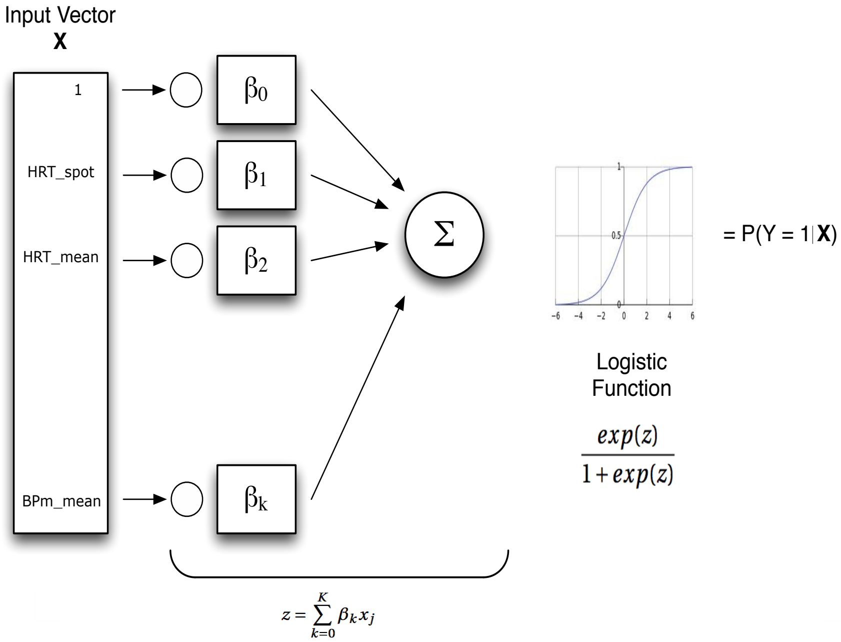


Figure 5.25: Input architecture, "Stats Based" models. Models using statistical measures of the data as inputs

- All signals, Model name: Full — a model examining the main effects using all the available signals (from Table 2.5) and all the available statistical measures.

- All signals + quadratic mean, Model name: FullQuadMean — it is suspected that the relationship between the signals may not be adequately modelled with just linear effects. This model adds a quadratic term of the mean value from each available signal to the full signal model.

- Features identified using lasso regression, Model name: Full-Lasso — this model is automatically built by using the lasso regression technique which uses a penalty factor to reduce the number of non-zero coefficients in a model. The purpose of examining this technique is to see whether or not an algorithmic method could be used to remove the subjective selection of features carried out in the Minimum model.

- Minimum signals, Model name: Minimum — this model is suggested by clinical teams and based on current techniques used in the ICU.

5.4.1 All signals, Model name: Full

The Full model is the baseline for the research. It utilises data from the BrainIT database which reflects the signals that are available from a typical installation of ICU bedside monitors. Full details of the available signals are provided in the medical background, Section 2.4. To briefly recap, this model uses two demographic signals and four measured signals from the ICU monitors. The demographic signals are age and gender. The measured signals are: heart rate (HRT); systolic arterial blood pressure (BPs); diastolic arterial blood pressure (BPd); and mean arterial blood pressure (BPm). The spot value is used along with three statistical measures: mean; standard deviation and slope. The clinical basis for this model comes from discussions with colleagues and builds on the work of the AvertIT project, (Donald et al., 2012b). Formally the model is defined as:

$$
\begin{aligned}
\log\left(\frac{p_i}{1-p_i}\right) = {} & \beta_0 + \beta_1 Age_i + \beta_2 Gender_i \\
& + \beta_3 HRT\_spot_i + \beta_4 HRT\_mean_i + \beta_5 HRT\_sd_i + \beta_6 HRT\_slope_i \\
& + \beta_7 BPs\_spot_i + \beta_8 BPs\_mean_i + \beta_9 BPs\_sd_i + \beta_{10} BPs\_slope_i \\
& + \beta_{11} BPd\_spot_i + \beta_{12} BPd\_mean_i + \beta_{13} BPd\_sd_i + \beta_{14} BPd\_slope_i \\
& + \beta_{15} BPm\_spot_i + \beta_{16} BPm\_mean_i + \beta_{17} BPm\_sd_i + \beta_{18} BPm\_slope_i
\end{aligned}
\tag{5.8}
$$

This model contains 19 parameters which must be estimated. Using the R function **glm()** from the stats package, the parameter values estimated by maximum likelihood are tabulated in Table 5.5. The model performance, as measured using the ROC technique, is presented in Figure 5.26. The profile of the model coefficient values is presented in Figure 5.27.

5.4.1.1 Model: Full, ROC Curves

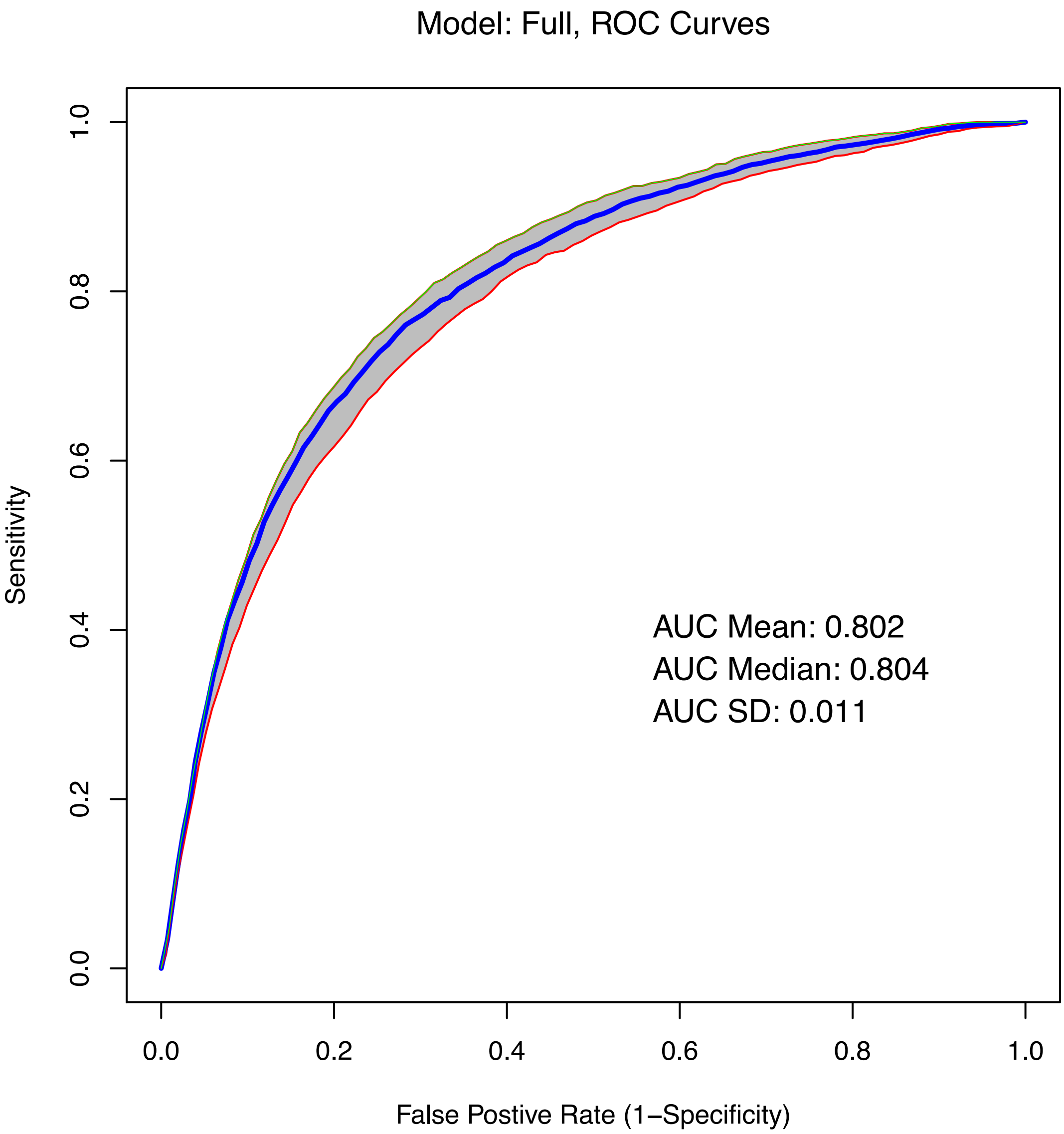


Figure 5.26: Model: Full, ROC and AUC detail. This plot is generated from 50 runs of the model. Each run consists of a training and test set. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

5.4.1.2 Model: Full, Parameter estimates

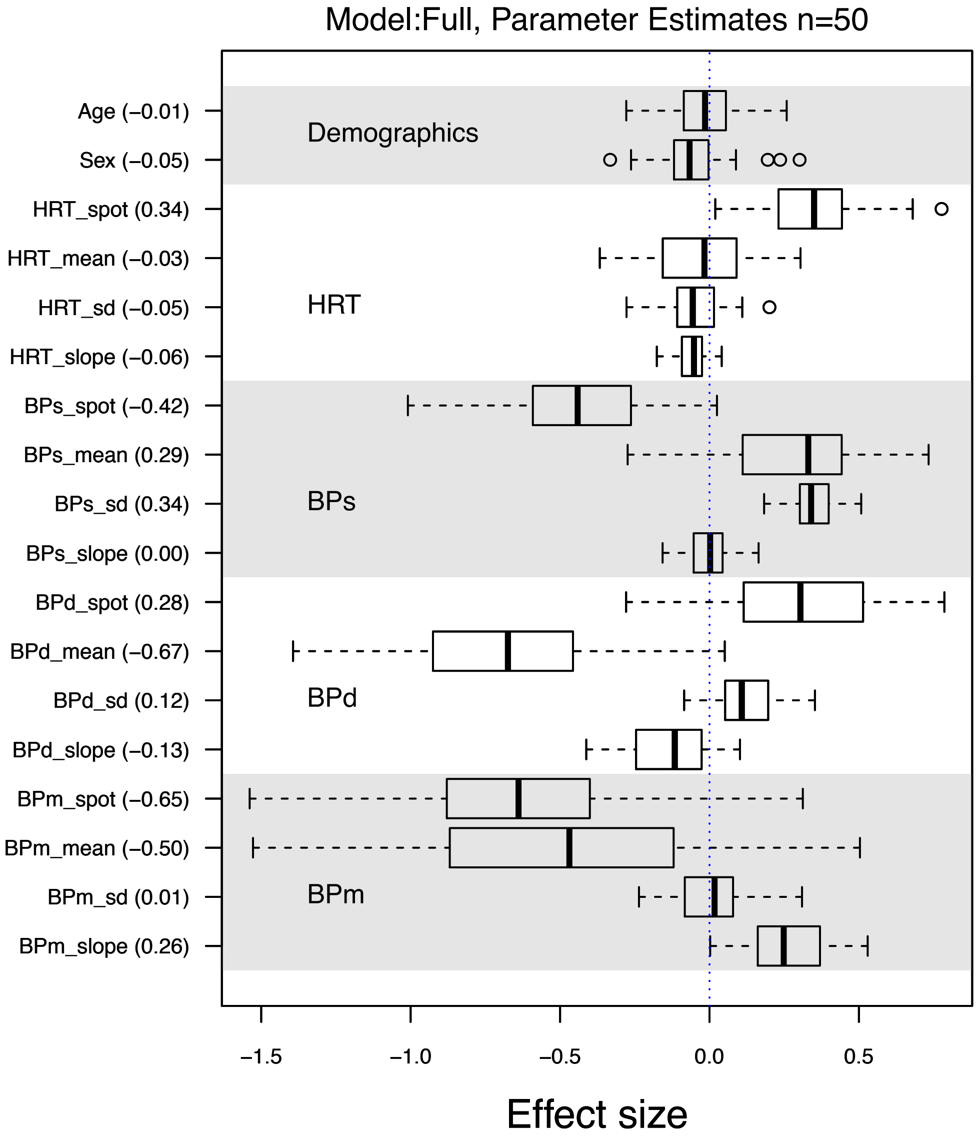


Figure 5.27: Model: Full, Parameter Effect Size. Effect size is defined as $\beta_j \times$ std dev(x_j)

Measure	Mean (SE)
Age	-0.0007 (0.00082)
Sex	-0.1232 (0.03833)
HRT_spot	0.0174 (0.00116)
HRT_mean	-0.0016 (0.00119)
HRT_sd	-0.0146 (0.00372)
HRT_slope	-0.0024 (0.00028)
BPs_spot	-0.0201 (0.00161)
BPs_mean	0.0150 (0.00174)
BPs_sd	0.0715 (0.00237)
BPs_slope	-0.0000 (0.00022)
BPd_spot	0.0245 (0.00336)
BPd_mean	-0.0645 (0.00495)
BPd_sd	0.0363 (0.00399)
BPd_slope	-0.0051 (0.00070)
BPm_spot	-0.0519 (0.00469)
BPm_mean	-0.0439 (0.00652)
BPm_sd	0.0019 (0.00496)
BPm_slope	0.0087 (0.00065)

Table 5.5: Parameter coefficients for model: Full

5.4.1.3 Model: Full, Summary

This is the best performing model from the research. However all the models perform similarly and the performance in terms of AUC value, at 0.804, is only just over 2% better than the much simpler Minimum model.

Interpreting the coefficients, the demographic variable Age has little effect and Sex has a small negative effect which means that males (coded 1) have more chance of having hypotensive episodes.

For the physiological variables the spot value (i.e. the single value 10 mins before an episode start) has a strong effect on the output. For the HRT variable the spot value is the only measurement contributing to the output. For the BPs and BPd variables the spot and mean variables have an effect with opposite signs. For BPs the standard deviation has a positive effect, intuitively this makes sense as it indicates that as the variance of BPs increases the patient is more likely to have an episode start. The slope for BPs has little effect. The same pattern is observed for the BPd measurement, increasing variance indicates an increase in the probability of an episode start. For the BPd signal the slope measurement does appear to have a negative influence on the outcome. For the BPm measurement both the spot and mean values have a negative effect. This indicates that a drop in BPm is associated with an increased risk of a hypotensive event.

5.4.2 All signals + quadratic mean, Model name: FullQuadMean

The FullQuadMean model is included to introduce an additional limiting mechanism on the Full model, mean value signals. By introducing a quadratic term, this model attempts to limit the signals to realistic physiological values. As a signal increases, the quadratic term with a negative coefficient will reduce the signal, thereby providing a damping effect. The opposite effect will occur for signals that are approaching unrealistic low values. The model is defined as:

$$
\begin{aligned}
\log\left(\frac{p_i}{1-p_i}\right) = {} & \beta_0 + \beta_1 Age_i + \beta_2 Gender_i \\
& + \beta_3 HRT\_spot_i + \beta_4 HRT\_mean_i + \beta_5 HRT\_sd_i + \beta_6 HRT\_slope_i + \beta_7 (HRT\_mean_i)^2 \\
& + \beta_8 BPs\_spot_i + \beta_9 BPs\_mean_i + \beta_1 0 BPs\_sd_i + \beta_{11} BPs\_slope_i + \beta_{12} (BPs\_mean_i)^2 \\
& + \beta_{13} BPd\_spot_i + \beta_{14} BPd\_mean_i + \beta_{15} BPd\_sd_i + \beta_{16} BPd\_slope_i + \beta_{17} (BPd\_mean_i)^2 \\
& + \beta_{18} BPm\_spot_i + \beta_{19} BPm\_mean_i + \beta_{20} BPm\_sd_i + \beta_{21} BPm\_slope_i + \beta_{22} (BPm\_mean_i)^2
\end{aligned}
\tag{5.9}
$$

This model contains 23 parameters which must be estimated. Again, using the R function **glm()** from the stats package, the parameter values estimated by maximum likelihood are tabulated in Table 5.6. The model performance, as measured using the ROC technique, is presented in Figure 5.28. The profile of the model coefficient values is presented in Figure 5.29.

5.4.2.1 Model: FullQuadMean, ROC Curves

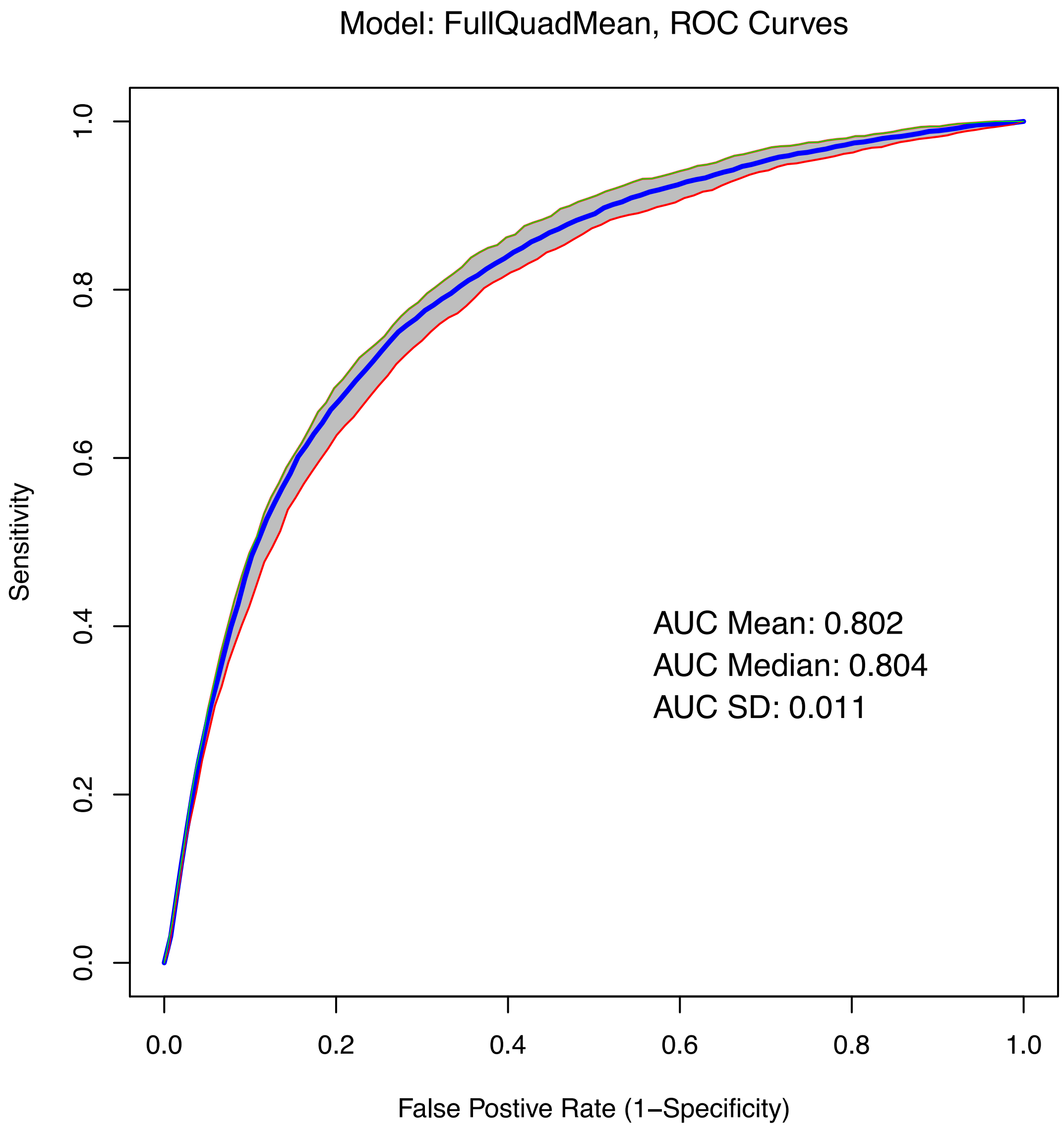


Figure 5.28: Model: FullQuadMean, ROC and AUC detail. This plot is generated from 50 runs of the model. Each run consists of a training and test set. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

5.4.2.2 Model: FullQuadMean, Parameter estimates

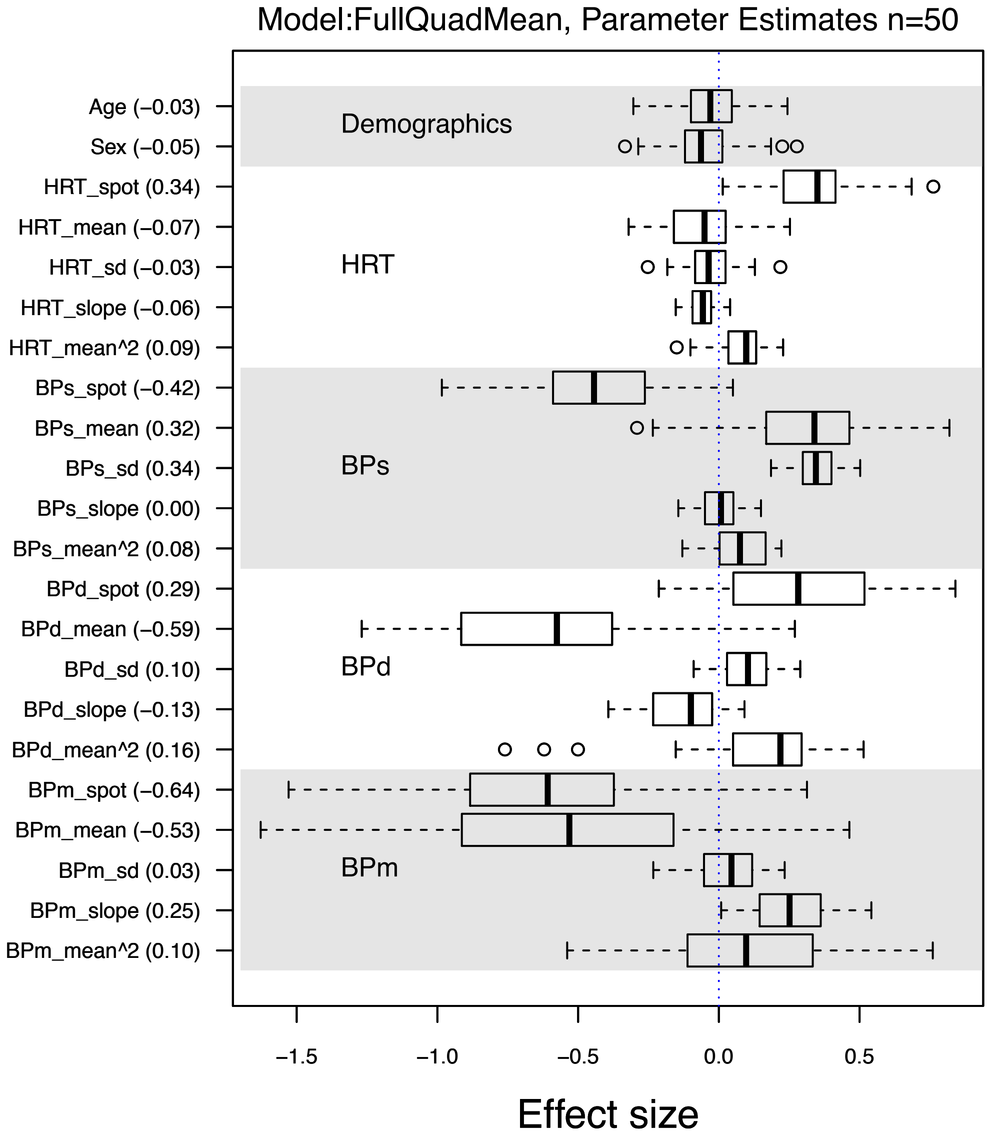


Figure 5.29: Model: FullQuadMean, Parameter Effect Size. Effect size is defined as $\beta_j \times$ std dev(x_j)

Measure	Mean (SE)
Age	-0.0015 (0.00086)
Sex	-0.1122 (0.04069)
HRT_spot	0.0174 (0.00112)
HRT_mean	-0.0034 (0.00105)
HRT_sd	-0.0083 (0.00359)
HRT_slope	-0.0024 (0.00027)
HRT_mean^2	0.0002 (0.00002)
BPs_spot	-0.0200 (0.00158)
BPs_mean	0.0163 (0.00179)
BPs_sd	0.0718 (0.00238)
BPs_slope	0.0000 (0.00022)
BPs_mean^2	0.0001 (0.00002)
BPd_spot	0.0255 (0.00340)
BPd_mean	-0.0571 (0.00534)
BPd_sd	0.0293 (0.00385)
BPd_slope	-0.0051 (0.00072)
BPd_mean^2	0.0009 (0.00020)
BPm_spot	-0.0506 (0.00474)
BPm_mean	-0.0472 (0.00683)
BPm_sd	0.0074 (0.00490)
BPm_slope	0.0086 (0.00066)
BPm_mean^2	0.0005 (0.00019)

Table 5.6: Parameter coefficients for model: FullQuadMean

5.4.2.3 Model: FullQuadMean, Summary

This model produces the same performance figures in terms of AUC values as the Full model. The additional variables formed from the square of the appropriate mean value do have non-zero coefficients but they are not increasing the predictively ability of the model. Taking the usual approach that a more parsimonious model is better, this model would not be used.

5.4.3 Features identified using lasso regression, Model name: Full-Lasso

The Full-Lasso model comes from the field of feature selection, which is still an active area of research in the statistical modelling community. One of the methods that has been used introduces a penalty technique to try to reduce the number of terms that are included from the set of all possible predictors. An example of this technique is "least absolute shrinkage selection operator" (Lasso) regression, (Tibshirani, 1996). In this technique the usual loss function, consisting of the residual sum of squares, is complemented by an L1 norm penalty on coefficients resulting in the absolute values of coefficients being constrained. This constraint has the effect of setting a number of the predictors to zero and these predictors are effectively dropped from the model. This is an algorithmic technique and therefore removes the subjective nature of feature selection. However, the technique still has difficulty with highly correlated signals, and although a *data driven* method, it is not guaranteed to indicate causal signals. The technique is included in this research in order to investigate its use in a clinical environment. The interest lies in whether or not Lasso regression will identify signals which clinicians also believe are driving the underlying physiological processes.

$$
\begin{aligned}
\log\left(\frac{p_i}{1-p_i}\right) = {} & \beta_0 + \beta_1 Age_i + \beta_2 Gender_i \\
& + \beta_3 HRT\_spot_i + \beta_4 HRT\_mean_i + \beta_5 HRT\_sd_i + \beta_6 HRT\_slope_i \\
& + \beta_7 BPs\_spot_i + \beta_8 BPs\_mean_i + \beta_9 BPs\_sd_i + \beta_{10} BPs\_slope_i \\
& + \beta_{11} BPd\_spot_i + \beta_{12} BPd\_mean_i + \beta_{13} BPd\_sd_i + \beta_{14} BPd\_slope_i \\
& + \beta_{15} BPm\_spot_i + \beta_{16} BPm\_mean_i + \beta_{17} BPm\_sd_i + \beta_{18} BPm\_slope_i
\end{aligned}
\tag{5.10}
$$

This model contains 19 parameters which must be estimated. Using the R function **cv.glmnet()** from the glmnet package, the parameter values estimated by maximum likelihood are tabulated in Table 5.7. The model performance, as measured using the ROC technique, is presented in Figure 5.30. The profile of the model coefficient values is presented in Figure 5.31.

5.4.3.1 Model: Full-Lasso, ROC Curves

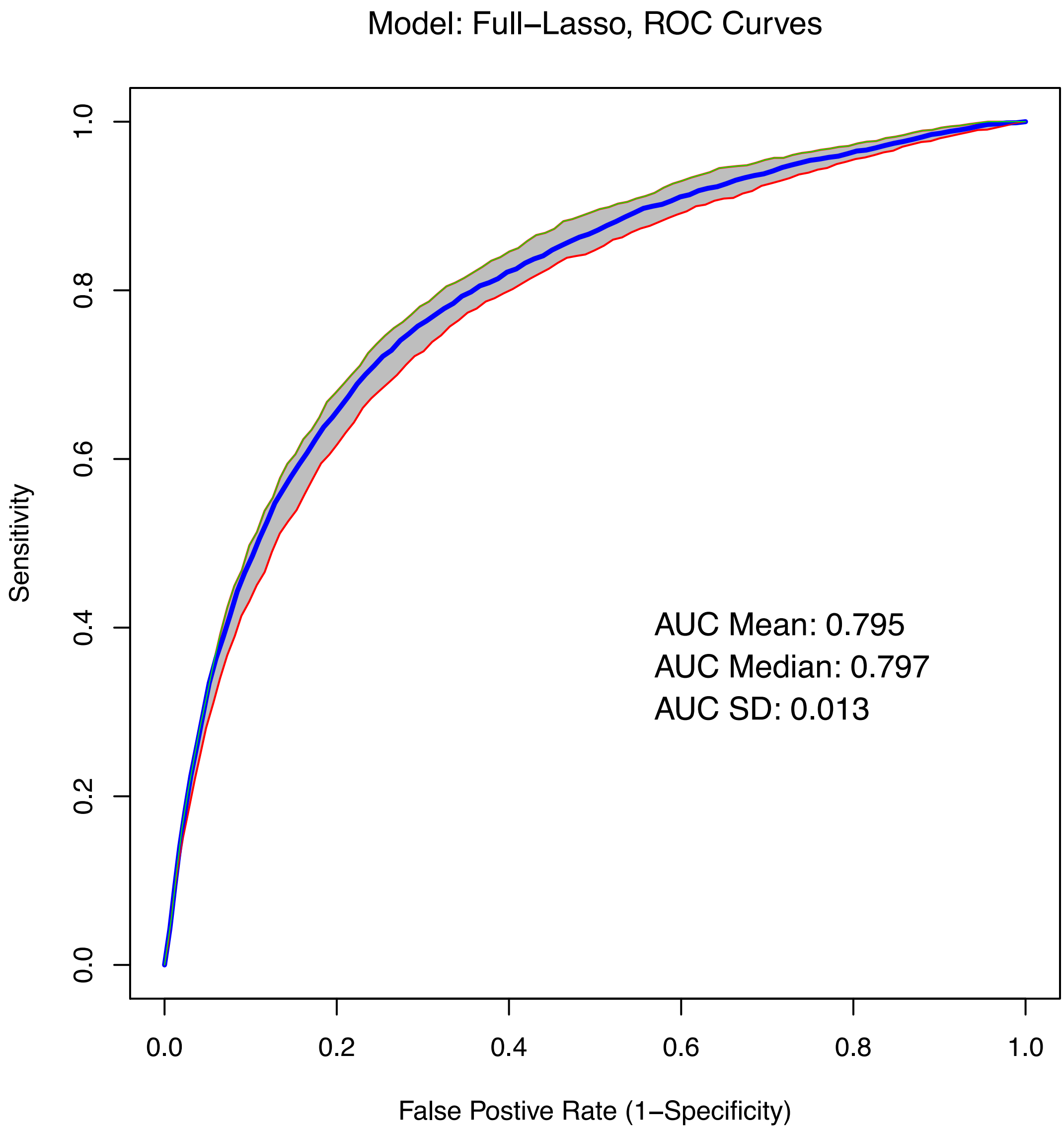


Figure 5.30: Model: Full-Lasso, ROC and AUC detail. This plot is generated from 50 runs of the model. Each run consists of a training and test set. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

5.4.3.2 Model: Full-Lasso, Parameter estimates

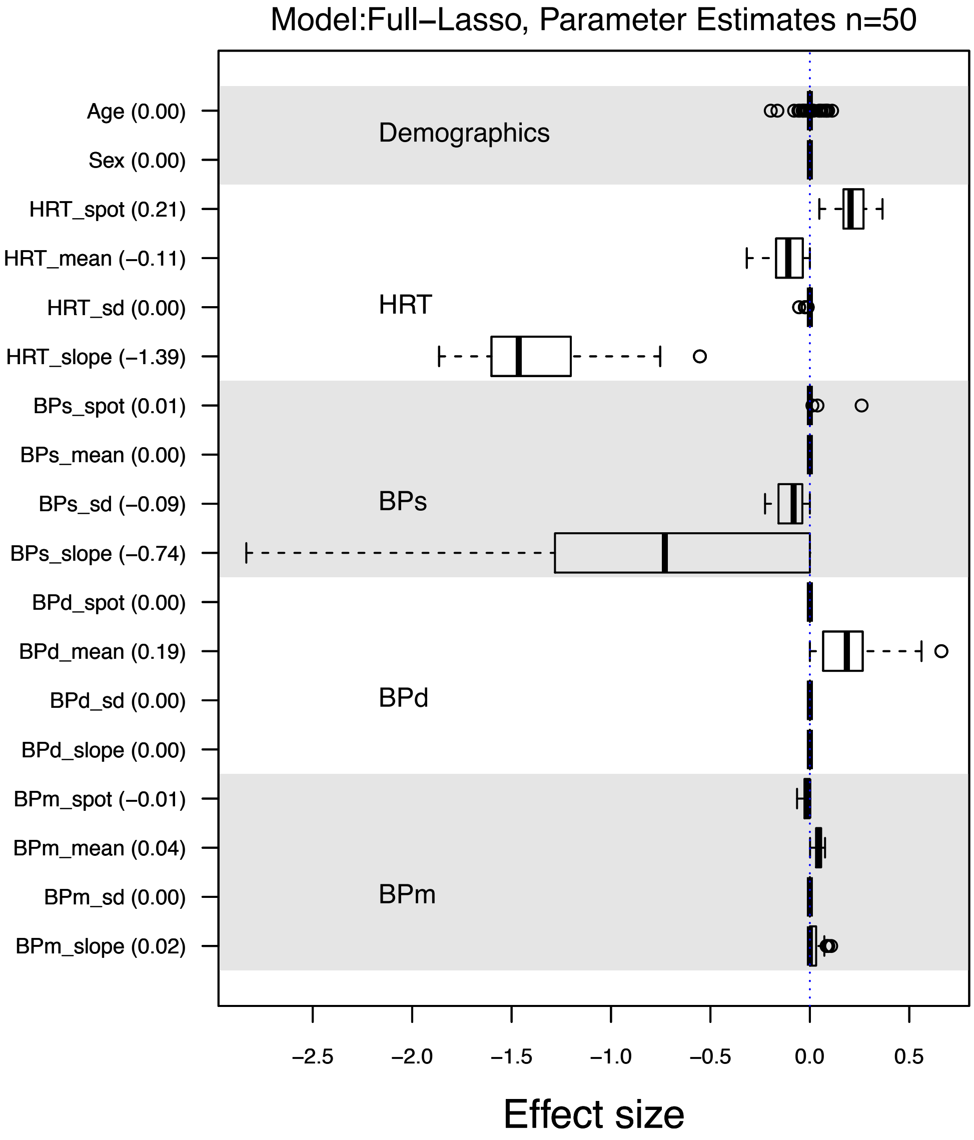


Figure 5.31: Model: Full-Lasso, Parameter Effect Size. Effect size is defined as $\beta_j \times$ std dev(x_j)

Measure	Mean (SE)
Age	-0.0000 (0.00037)
Sex	0.0000 (0.00000)
HRT_spot	0.0107 (0.00053)
BPs_spot	-0.0057 (0.00063)
BPd_spot	-0.0006 (0.00036)
BPm_spot	-0.0558 (0.00168)
HRT_mean	0.0003 (0.00025)
BPs_mean	0.0000 (0.00000)
BPd_mean	-0.0195 (0.00207)
BPm_mean	-0.0161 (0.00217)
HRT_sd	0.0000 (0.00000)
BPs_sd	0.0180 (0.00202)
BPd_sd	0.0000 (0.00000)
BPm_sd	0.0000 (0.00000)
HRT_slope	-0.0012 (0.00019)
BPs_slope	0.0038 (0.00020)
BPd_slope	0.0000 (0.00000)
BPm_slope	0.0006 (0.00014)

Table 5.7: Parameter coefficients for model: Full-Lasso

5.4.3.3 Model: Full-Lasso, Summary

This model is very interesting in that it performs very closely to the Full model being, at AUC = 0.795, only half of a percentage point down on AUC value. The coefficients selected by the procedure are very different from the other models. Age, Gender and BPm are effectively dropped from the model. There are only three variables that have much effect on the model: HRT slope; BPs slope; and BPd mean. This would be a difficult model to explain to a clinical team.

5.4.4 Minimum signals, Model name: Minimum

The Minimum model comes from discussions with clinical teams. This is a very simple model which only uses: a single measure of blood pressure, BPm, as it captures the information from both the systolic and diastolic pressures; the heart rate of the patient, HRT; and the demographic variables of age and gender. The summary statistics used are mean value and standard deviation. When interpreting these signals experienced clinicians intuitively average out the fluctuations which are displayed on the ICU monitors. They also take into consideration the fact that an increase in signal fluctuation often precedes an unstable condition. Translated into statistical modelling terms, their interpretation becomes mean value and standard deviation. Age and gender are included as these variables would be expected to influence the physiological response. The formal model is defined as:

$$\begin{aligned}\operatorname{logit}(Y_i = 1|\boldsymbol{X_i}) = {} & \beta_0 + \beta_1 Age_i + \beta_2 Gender_i \\ & + \beta_3 HRT\_mean_i + \beta_4 BPm\_mean_i \\ & + \beta_5 HRT\_sd_i + \beta_6 BPm\_sd_i \end{aligned} \tag{5.11}$$

This model contains 7 parameters which must be estimated. Using the R function **glm()** from the stats package, the parameter values estimated by maximum likelihood are tabulated in Table 5.8. The model performance, as measured using the ROC technique, is presented in Figure 5.32. The profile of the model coefficient values is presented in Figure 5.33.

5.4.4.1 Model: Minimum, ROC Curves

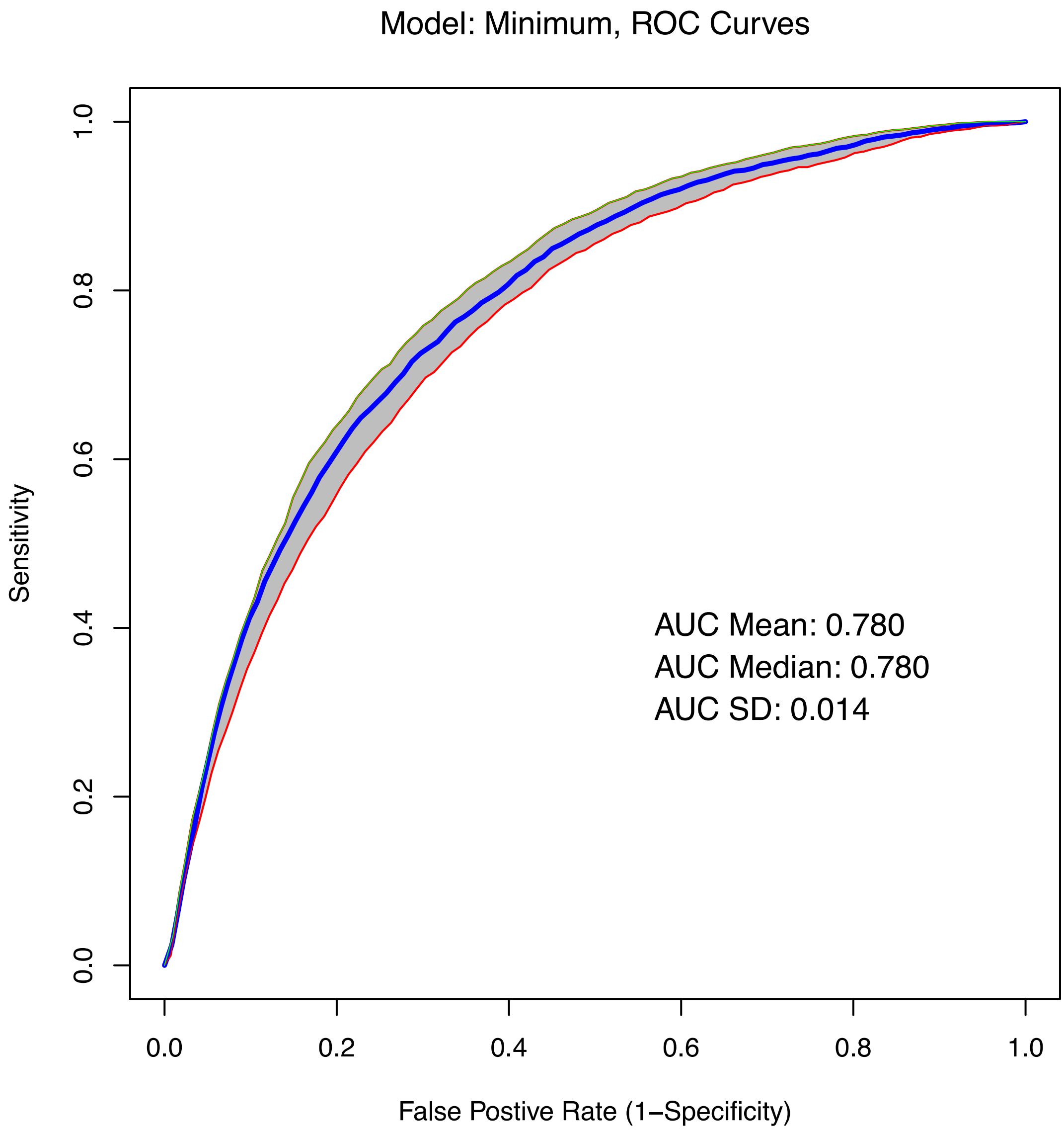


Figure 5.32: Model: Minimum, ROC and AUC detail. This plot is generated from 50 runs of the model. Each run consists of a training and test set. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

5.4.4.2 Model: Minimum, Parameter estimates

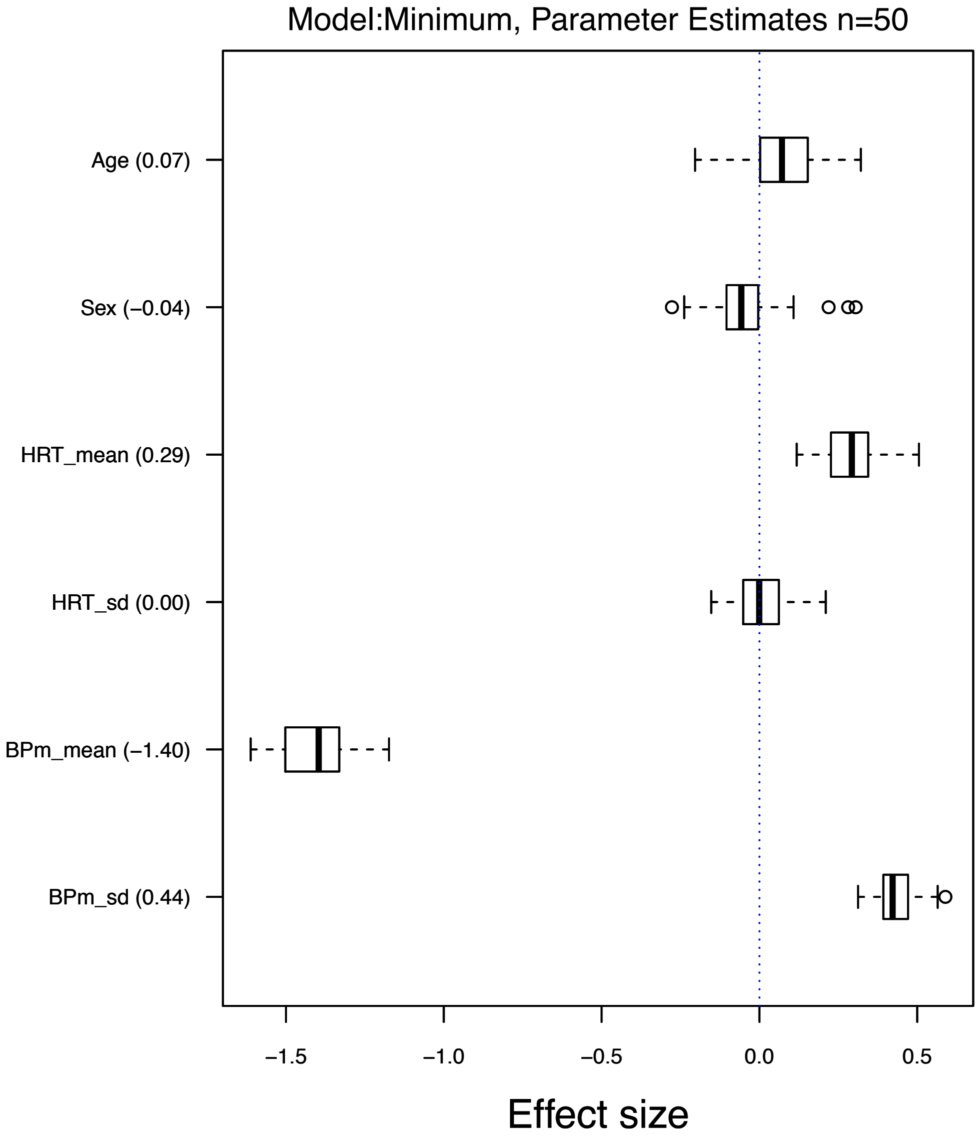


Figure 5.33: Model: Minimum, Parameter Effect Size. Effect size is defined as $\beta_j \times$ std dev(x_j)

Measure	Mean (SE)
Age	0.0037 (0.00084)
Sex	-0.0866 (0.03740)
HRT_mean	0.0152 (0.00062)
HRT_sd	0.0007 (0.00309)
BPm_mean	-0.1240 (0.00137)
BPm_sd	0.1222 (0.00241)

Table 5.8: Parameter coefficients for model: Minimum

5.4.4.3 Model: Minimum, Summary

This simple model performs well and is only 2% lower than the best performing Full model. All the variables apart from the standard deviation of the HRT measurement are contributing to the model output. The largest influence comes from the BPm mean value which has negative effect. This implies that an increase in mean arterial blood pressure will decrease the probability of the onset of a hypotensive episode. This is to be expected as the patient's blood pressure moves away from the EUSIG threshold. Perhaps of more interest are the two coefficients for HRT mean and BPm standard deviation. Both coefficients are positive indicating that a rise in average HRT or the variability of the average BPm indicate an increased risk of a hypotensive episode.

5.4.5 Models using statistical measures, Summary

Section 5.4 has reported the results from four penalised logistic regression models. The results for the three models using all the available measurements are marginally better than the penalised logistic regression models. The simplest model using only two physiological variables and requiring the estimation of only 7 parameters performs very closely to the more complex penalised logistic regression models. The similar performance to the penalised logistic regression models can probably be attributed to the dimension reduction from using the statisitical pre-processing and that fact that most of the information in the data appears to be in the first five to ten minutes of the 31 minute data buffer.

5.5 Varying event horizon and window size

Using the "Stats Based" input architecture described in Section 3.7.2 allows an investigation into the effect of varying both the event horizon and window size used to train and test the models. A group of 30 base data sets was constructed as described in Section 4.2.5. The four "Stats Based" models from Section 5.4 were configured to use these data sets for training and test and a summary of the results is presented in Figure 5.34.

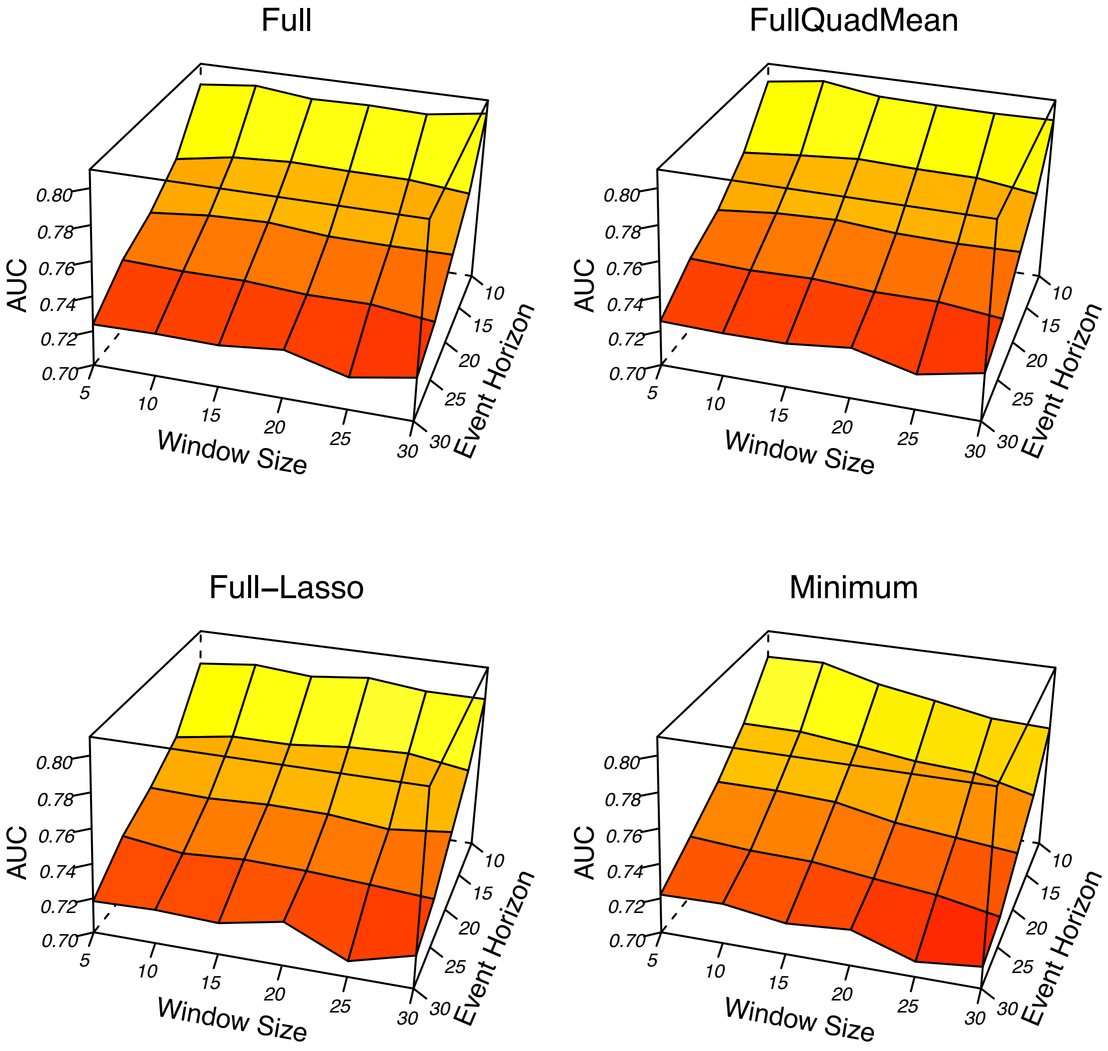


Figure 5.34: Stats based logistic regression model comparison. Each grid point on a 3D plot shows the AUC value from a test set of data with the defined event horizon (EH) and window size (WS). Apart from the Minimum model, window size has little effect on model performance. Increasing the EH used for model training results in a lower predictive ability from the model.

Figure 5.34 presents a 3D plot for each model showing a surface constructed by plotting the AUC value from a test set of data with the defined event horizon (EH) and window size (WS) at each grid point on the surface. Apart from the Minimum model, window size has little effect on model performance. Increasing the EH used for model training results in a lower predictive ability from the model. These results are in line with the results from the previous sections in this chapter. Indeed the top right hand grid point

(EH10, WS30) is the common data set used to assess all models. The models based on using all variables, the “FullQuadMean/-Lasso” models, perform slightly better than the Minimum model. Interestingly the 3D grid for the Minimum model suggests that if a window size of five or ten minutes had been used that its performance would be similar to the other models.

As the surfaces from Figure 5.34 are all similar, the opportunity was taken to investigate the use of the H-score (Section 3.6.3) to see if a more distinct pattern would emerge. Figure 5.35 and Tables 5.9 and 5.10 present the results of this investigation when using the Full model. The conclusion being that using the H-Score gives a similar picture in terms of predictive ability of the model. The procedure was run for all the “Stats Based” models with similar results.

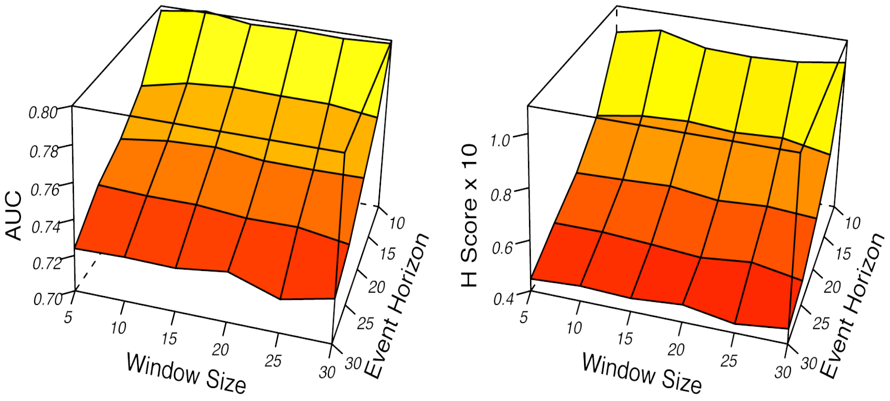


Figure 5.35: Model: Full, varying EH and WS, AUC and H-Score metrics

5.5.1 Model: Full, ROC and H-Score estimates

Event			Window Size			
Horizon	5	10	15	20	25	30
10	0.80 (0.01)	0.80 (0.01)	0.80 (0.01)	0.80 (0.01)	0.80 (0.01)	0.80 (0.01)
15	0.77 (0.01)	0.77 (0.01)	0.77 (0.01)	0.77 (0.01)	0.77 (0.01)	0.77 (0.01)
20	0.75 (0.02)	0.75 (0.01)	0.76 (0.01)	0.75 (0.01)	0.75 (0.01)	0.75 (0.01)
25	0.74 (0.02)	0.74 (0.02)	0.74 (0.02)	0.74 (0.01)	0.74 (0.02)	0.73 (0.02)
30	0.72 (0.01)	0.72 (0.02)	0.72 (0.01)	0.73 (0.02)	0.72 (0.01)	0.72 (0.01)

Table 5.9: ROC assessment results for Model: Full. Each cell represents the average AUC for 50 iterations with (std dev).

Event	Window Size					
Horizon	5	10	15	20	25	30
10	0.99 (0.13)	1.03 (0.17)	0.98 (0.15)	0.98 (0.11)	0.98 (0.13)	1.01 (0.16)
15	0.74 (0.12)	0.77 (0.13)	0.78 (0.13)	0.77 (0.10)	0.78 (0.11)	0.74 (0.11)
20	0.61 (0.11)	0.62 (0.11)	0.63 (0.10)	0.61 (0.10)	0.62 (0.10)	0.61 (0.10)
25	0.54 (0.10)	0.54 (0.12)	0.52 (0.10)	0.51 (0.09)	0.53 (0.09)	0.50 (0.10)
30	0.45 (0.08)	0.46 (0.11)	0.45 (0.08)	0.47 (0.09)	0.44 (0.08)	0.46 (0.07)

Table 5.10: ROC assessment results for Model: Full. Each cell represents the average H Score ×10 for 50 iterations with (std dev).

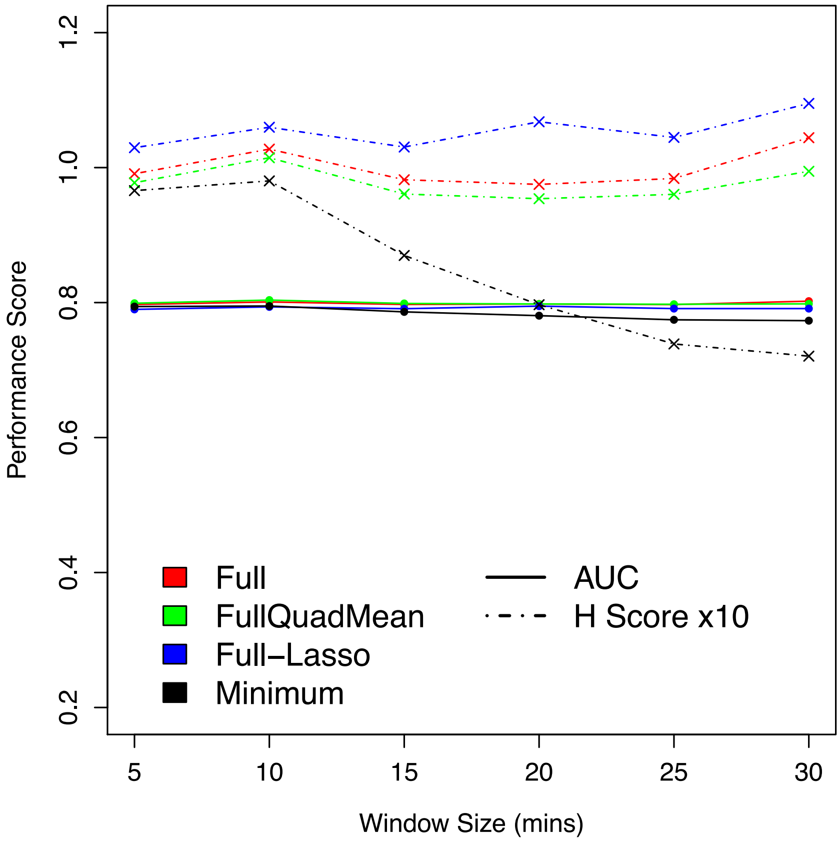


Figure 5.36: EH 10, AUC and H score. This plot shows a “slice” through the 3D surfaces of Figure 5.34 for the different models. The event horizon (EH) is fixed at 10 minutes. The AUC value lines are similar and hard to distinguish between models at this y scale. The H-score shows more separation between models.

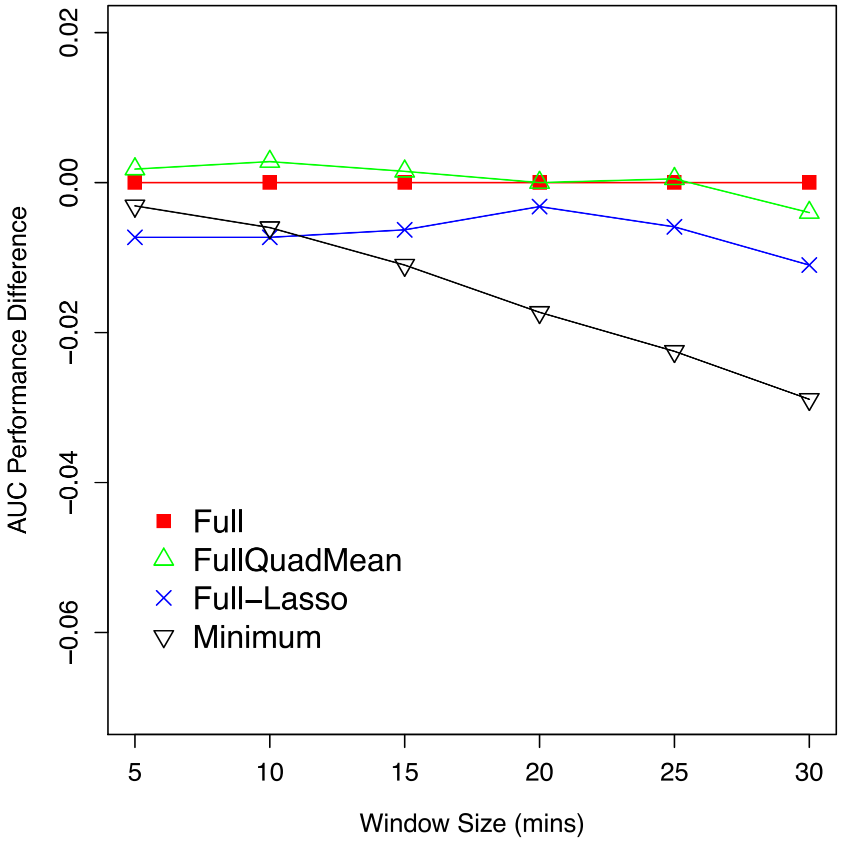


Figure 5.37: EH 10, AUC model comparison. This plots takes as it base line the "Full" model. The plot shows the performance offset for the other models relative to this baseline. Although the "Minimum" model's performance drops as window size is increased, its performance is comparable if a five or ten minute window size is used.

5.6 Logistic regression models, Summary

This chapter has considered eight logistic regression models using two input architectures and two parameter estimation techniques. The results, using a common data set, have been broadly similar and provide evidence that simpler models, admittedly using data pre-processing, can be used to provide early warning of the increased risk of a hypotensive episode.

Chapter 6

Neural Network Models

This chapter describes the tasks required to build a non linear model called a neural network. Two models are presented which are the neural network versions of the full signals and minimum signals models reported in Chapter 5 on logistic regression. This chapter will detail the model topology and the estimation of the parameters of the models using both maximum likelihood and Bayesian techniques.

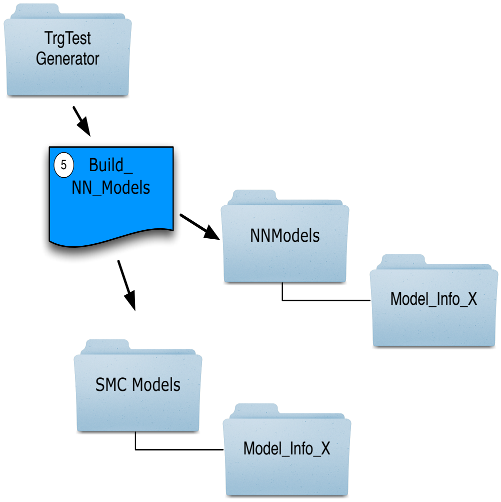


Figure 6.1: Neural networks modelling research tasks

6.1 Neural Network Model Proposals

Two neural network models are investigated. The two models use the same input structure as the "All Data" penalised logistic regression models from Section 5.3.

- SMC-BANN-Full, this model uses the inputs age, gender, HRT, BPs, BPd, BPm. The models uses 12 hidden nodes, the activation functions are logistic sigmoids. The single output node is also a logistic sigmoid unit. Both the hidden layer and output layer have bias inputs for each activation unit, this leads to a total of $(127 \times 12) + (1 + 12) = 1537$ model parameters (aka weights) which must be estimated.

- SMC-BANN-Min, this model uses the minimal set of inputs suggested by clinicians. The inputs are age, gender, HRT and BPm. The models uses 12 hidden nodes, the activation functions are logistic sigmoids. The single output node is also a logistic sigmoid unit. Again, both the hidden layer and output layer have bias inputs for each activation unit, leading to a total of $(65 \times 12) + (1 + 12) = 793$ model parameters to be estimated.

For both models, alternative models structures of four and eight hidden nodes were tried however these configurations gave poorer results. Again note that all models use normalised data i.e.

$$x_i = x_{i\ input} - \mu_{input} \tag{6.1}$$

with $\mu_{input} = 1/N \sum_{i=1}^{N} x_i$

6.1.1 Neural Network Models MLE using R nnet package

It is instructive to investigate what values of AUC could be achieved using the standard R package nnet (Venables and Ripley, 2002). This package uses a maximum likelihood approach to estimate the parameters of the neural network. This provides a comparative target which can be used when assessing the neural networks built using the sequential Monte Carlo method. The script **NNetVariability.R** was written to investigate the "all data" input architecture with "full" and "minimum" signal inputs, (see Appendix B for script details).

An essential part of the the investigation is determining a value for the "weight decay" parameter. This parameter is similar to the penalty term used in the penalised logistic regression models. The suggested weight decay range of 1e-4 to 1e-2 (Venables and Ripley, 2002, §8.10) was extended to 1e-5 to 1e-1 and Figures 6.2 and 6.3 show the AUC results determined from both the training and test sets.

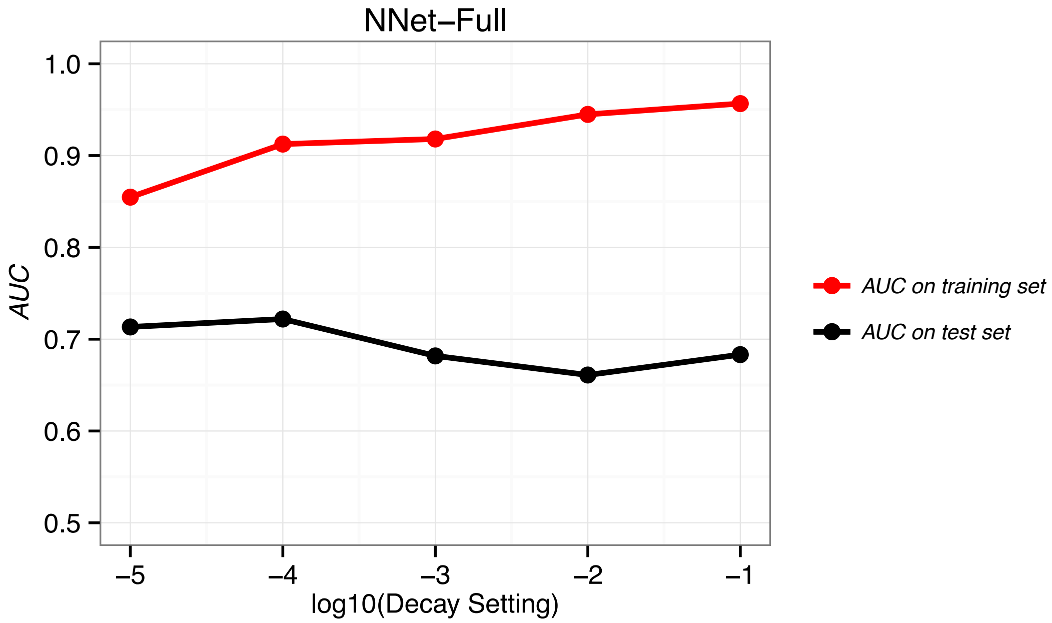


Figure 6.2: NNet-Full, AUC vs weight decay setting

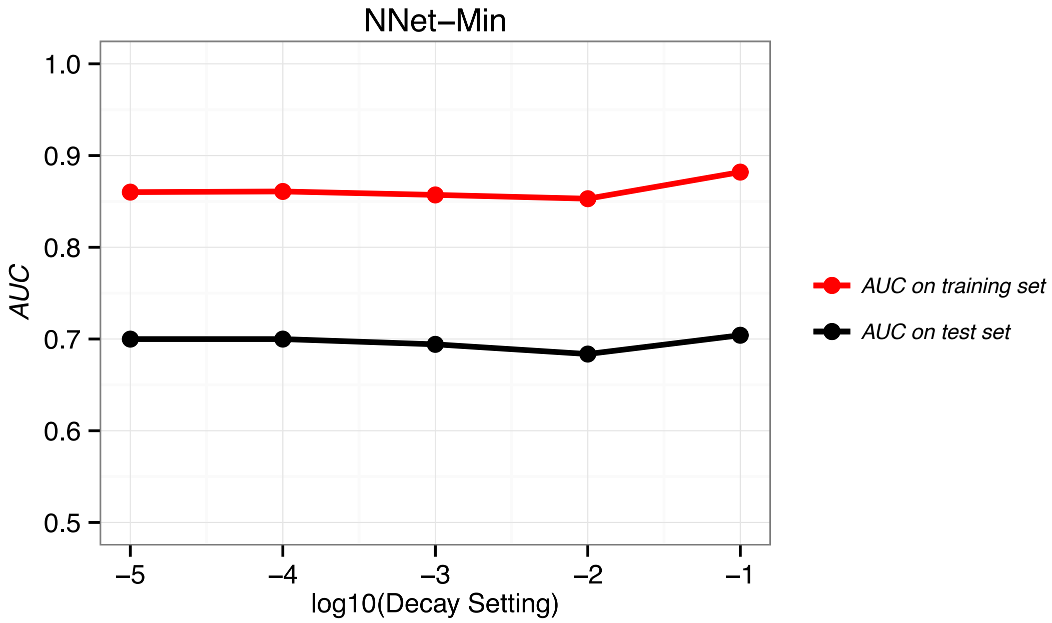


Figure 6.3: NNet-Min, AUC vs weight decay setting

As expected, both plots show that the neural network can achieve very good AUC values on the training set but that the generalisation ability is relatively poor. The best AUC values, on the test sets, of approximately 0.7 are well below the values achieved by the logistic regression models. The variability of the nnet AUC performance is presented in Figure 6.4.

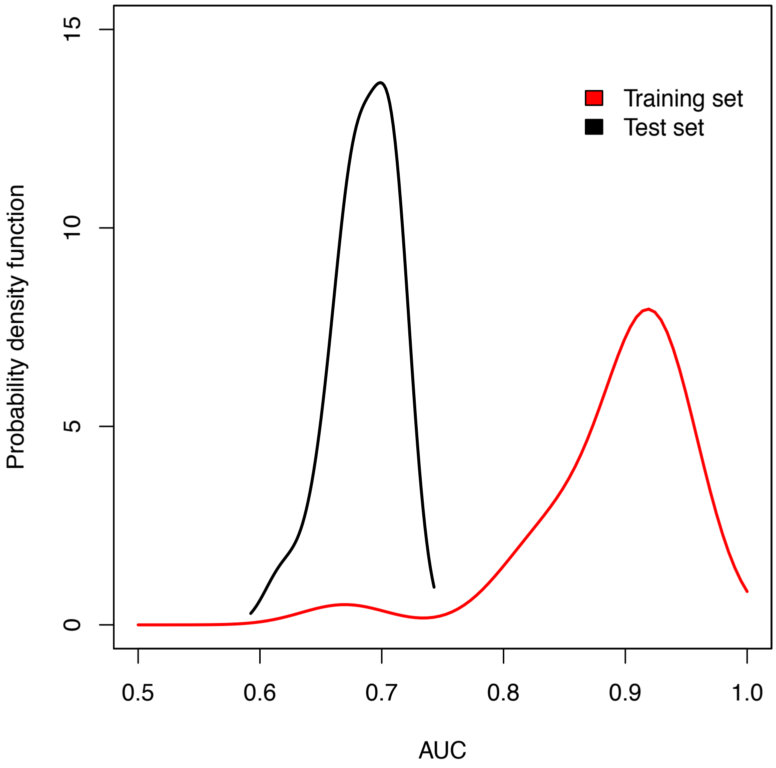


Figure 6.4: NNet-Full, AUC variability.

This section has shown that basic neural networks can be easily constructed using standard R functions. The performance of these models is in line with the logistic regression models of Chapter 5. The AUC value is lower but provides a starting point for the investigation into the use of sequential Monte Carlo techniques for parameter estimation.

6.1.2 NN Models All signals, Model: SMC-BANN-Full

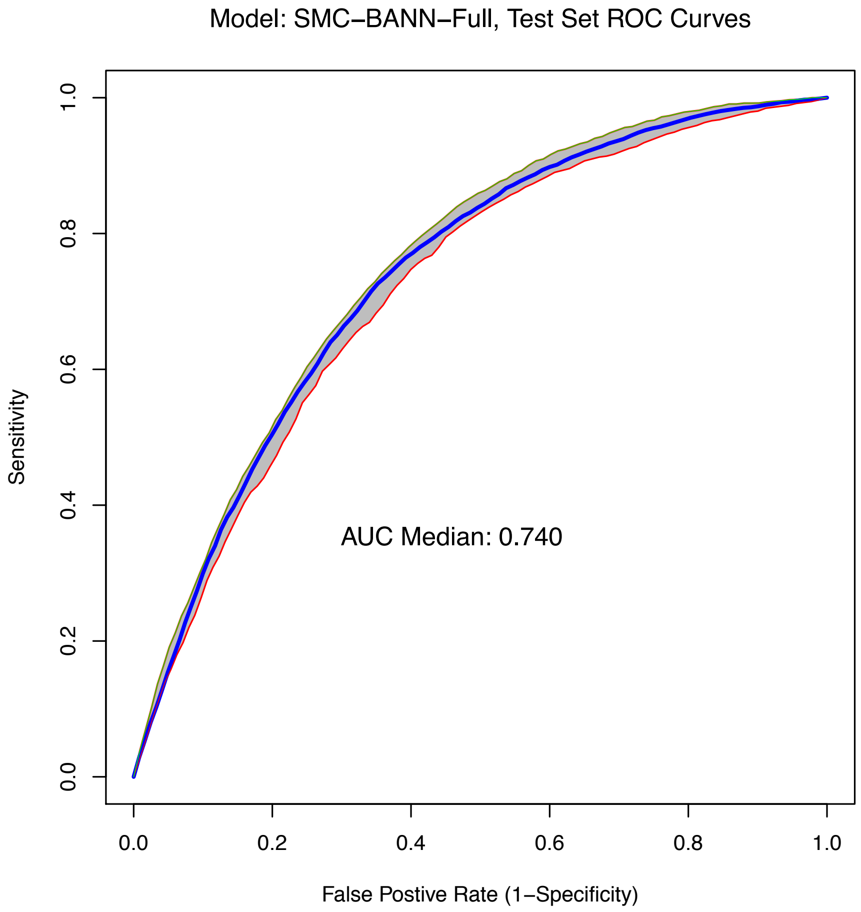


Figure 6.5: SMC-BANN-Full, Test Set ROC curves. This plot is generated from 5 runs of the model. Each run consists of a training and test set and produces 1000 nets each of which contributes an ROC trace to the overall calculations. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

6.1.2.1 SMC-BANN-Full, Algorithm diagnostics

Figures 6.6 to 6.8 provide some insight into the operation of the SMC algorithm. Figure 6.6 plots the ESS value as the distributions step through from the starting distribution π_0 to the final target distribution π_S. This same pattern is displayed in Figure 6.7 which provides a visualisation of the particle weight as the stepping across the distributions occurs. In this figure the distributions steps are displayed on the y-axis. Finally Figure 6.8 provides an expanded view of the initial steps of the distribution movement and shows an area of initialisation that must be assessed when checking the SMC procedure operation.

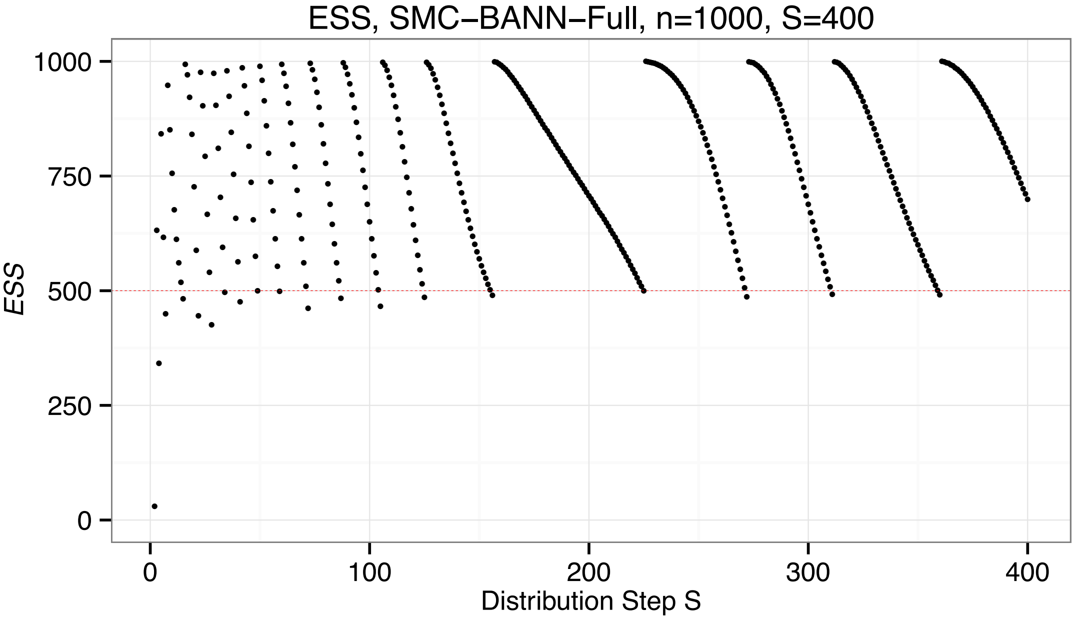


Figure 6.6: SMC-BANN-Full, ESS Diagnostics

Figure 6.6 shows that the SMC procedure has some initial difficulty is starting as the first two points are well below the ESS resampling threshold. The procedure takes approximately 70 steps to settle and then begins to form a more regular pattern. This can also be seen in Figure 6.7 and is discussed with reference to Figure 6.8.

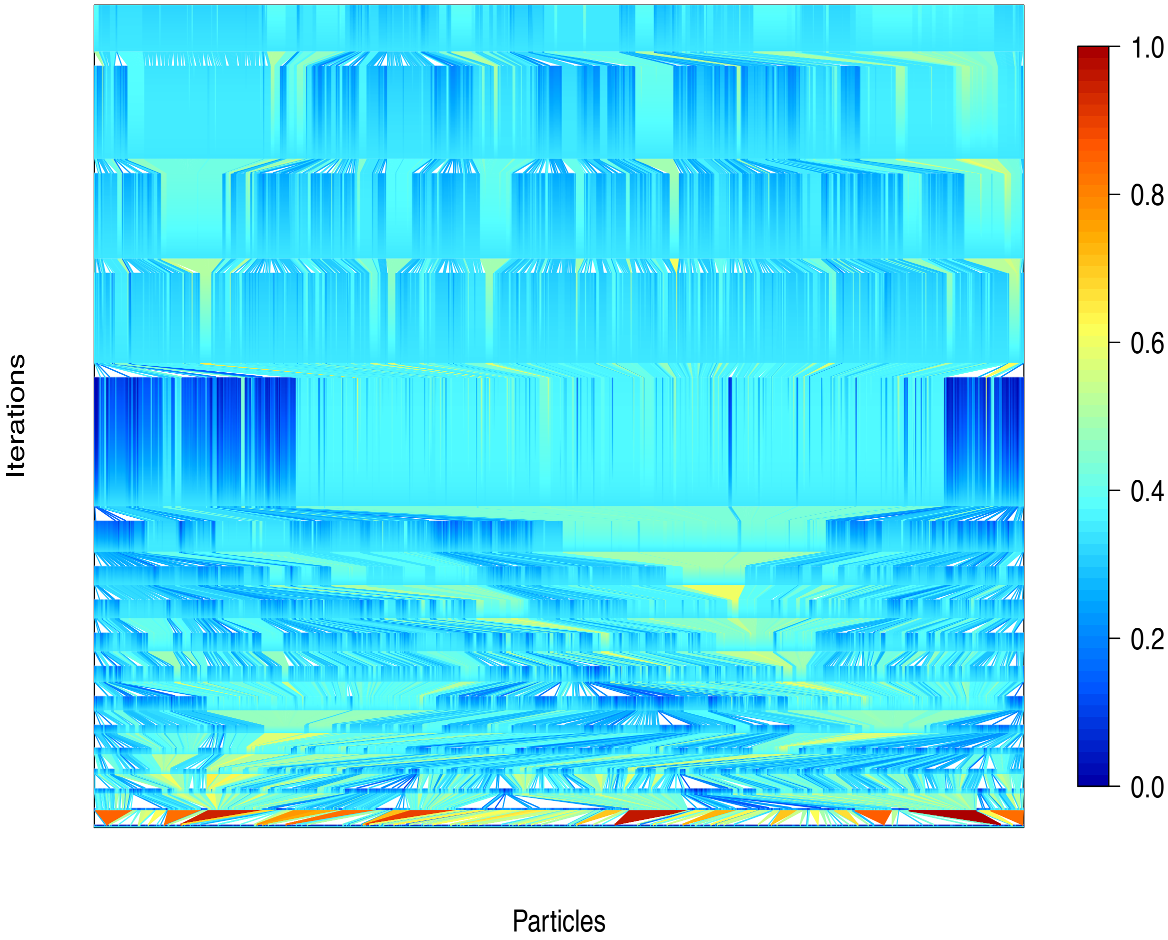


Figure 6.7: SMC-BANN-Full, SMC sampling. This plots shows the 1000 particles as columns of colour rising from the x-axis. The columns are built up as the distribution moves from π_0 at the bottom of the "Iterations" axis to π_S the final target distribution at the top. The colour key is a relative scale with the range being from dark blue for low weights to red for high weights.

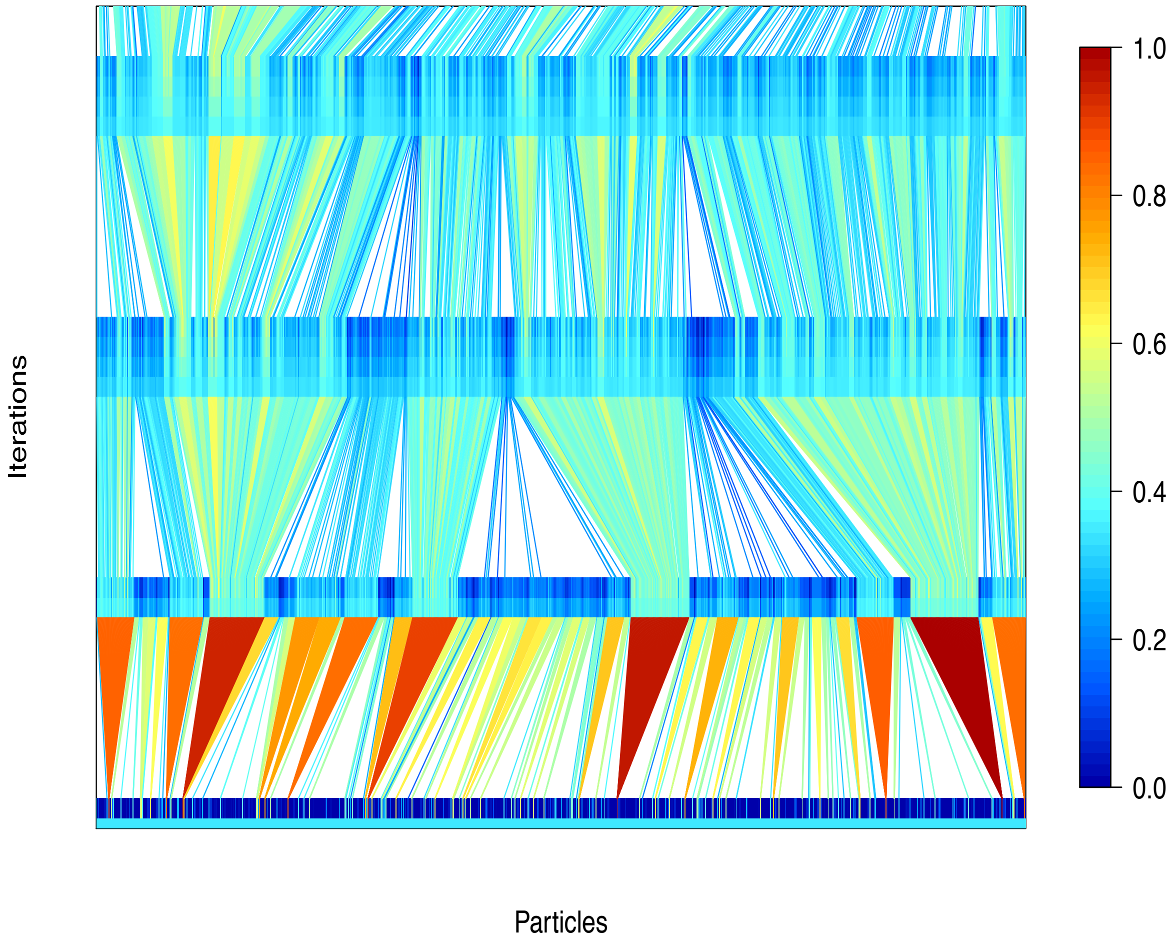


Figure 6.8: SMC-BANN-Full, Initial SMC sampling. The colour key is a relative scale with the range being from dark blue for low weights to red for high weights.

Figure 6.8 illustrates an area that must be assessed when starting the SMC algorithm. The starting distribution can be dominated by a limited number of particles which, although resampled, can limit the amount of exploration of the parameter space. This effect is apparent in Figure 6.8 as the darker red and orange segments that are heavily used during the first resample step. This corresponds to the left hand side of Figure 6.6 as the single point just above the zero line.

6.1.2.2 Model: SMC-BANN-Full, Summary

This section shows that a neural network model can produce AUC values similar to the logistic regression models of Chapter 5. The use of the SMC technique to estimate the model weights has improved the performance of the model compared to using the nnet package with maximum likelihood. However the complexity of the model requiring the estimation of hundreds of parameters has resulted in a poorer performance on this data set when compared to the simpler logistic regression models. The SMC technique has been more difficult to initialise than in the penalised logistic regression models of Sections 5.3.2.4 and 5.3.3.4. The technique recovers well from an initial poor start and provides an even distribution of model weights by the end of the process.

A disadvantage of the neural network technique is the considerable computation complexity required to calculate the parameter estimates. The SMC-BANN-Full model takes approximately eight hours to process a single data set. This is to be compared to a few minutes for a “StatsBased” logistic regression model. A bootstrap procedure was employed during the construction of the logistic regression models to assess the variability of the parameter estimates using 50 randomly drawn training and tests. To replicate this degree of variability using the SMC BANN models would require a prohibitively long calculation run of several weeks (although it must be stressed that the current code base has not been optimised). In order to report some measure of variability the bootstrap runs were reduced to five which still requires in the order of 40 hours of calculations.

6.1.3 NN Models Minimum signals, Model: SMC-BANN-Min

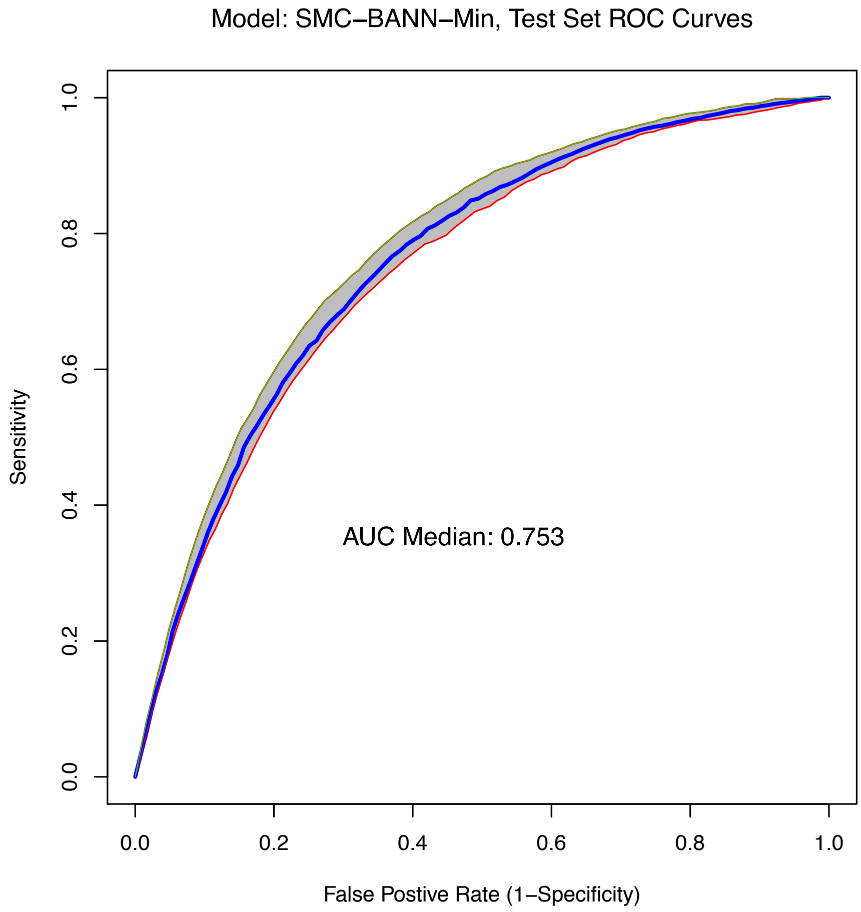


Figure 6.9: SMC-BANN-Min, Test Set ROC curves. This plot is generated from 5 runs of the model. Each run consists of a training and test set and produces 1000 nets each of which contributes an ROC trace to the overall calculations. The blue trace shows the median performance, the green (upper) and red (lower) traces give the 95% and 5% quantiles.

6.1.3.1 SMC-BANN-Min, Algorithm diagnostics

Figures 6.10 to 6.12 provide detail regarding the operation of the SMC algorithm. Figure 6.10 plots the ESS value on the y-axis from the starting distribution π_0 at the left hand side of the x-axis to the final target distribution π_S on the right. Figure 6.7 provides a visualisation of the particle weight as the algorithm steps across the distributionss. This figure displays distribution steps on the y-axis. Figure 6.8 provides an expanded view of the first few steps of the distribution movement.

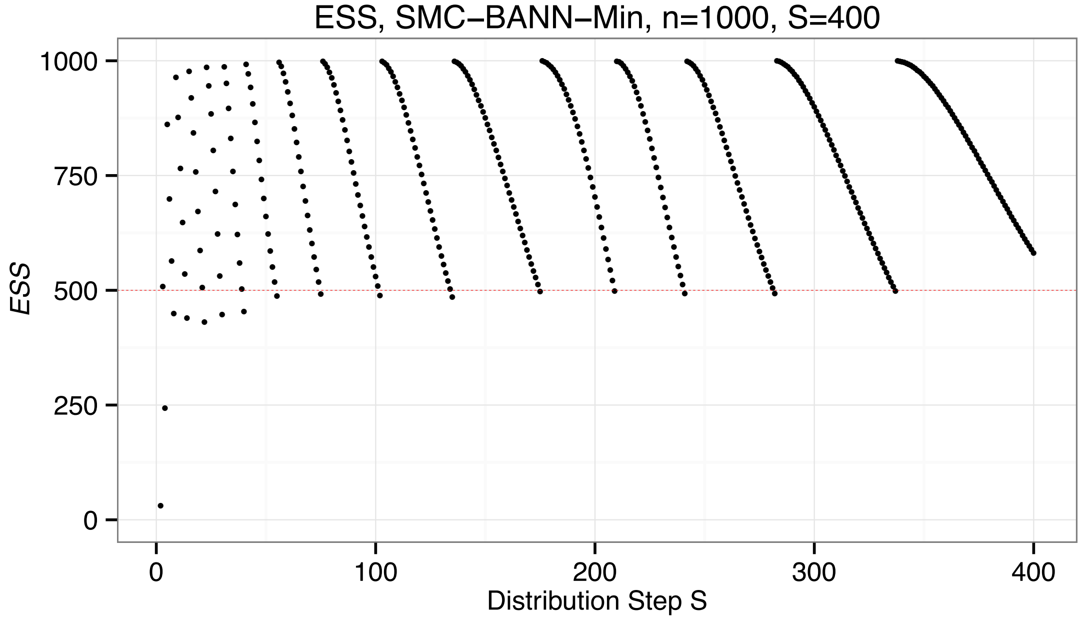


Figure 6.10: SMC-BANN-Min, ESS Diagnostics

Figure 6.10 shows that again the SMC procedure has some initial difficulty is starting although not as much as for the SMC-BANN-Ful model. The initial two points are well below the ESS resampling threshold. For this model the algorithm settles quicker taking approximately 50 steps to settle before forming a more regular pattern. This is shown in Figure 6.11 and is discussed with reference to Figure 6.12.

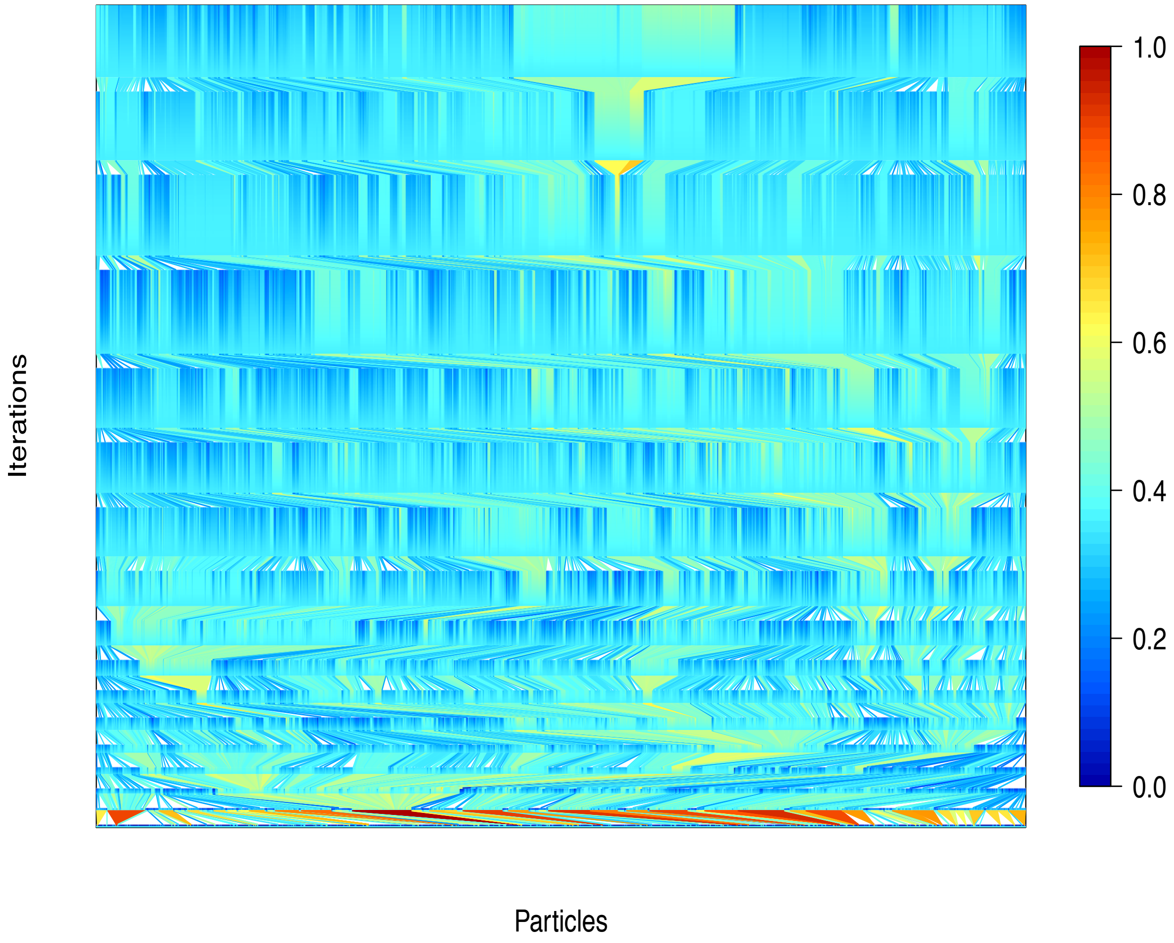


Figure 6.11: SMC-BANN-Min, SMC sampling. This plots shows the 1000 particles as columns of colour rising from the x-axis. The columns are built up as the distribution moves from π_0 at the bottom of the "Iterations" axis to π_S the final target distribution at the top. The colour key is a relative scale with the range being from dark blue for low weights to red for high weights.

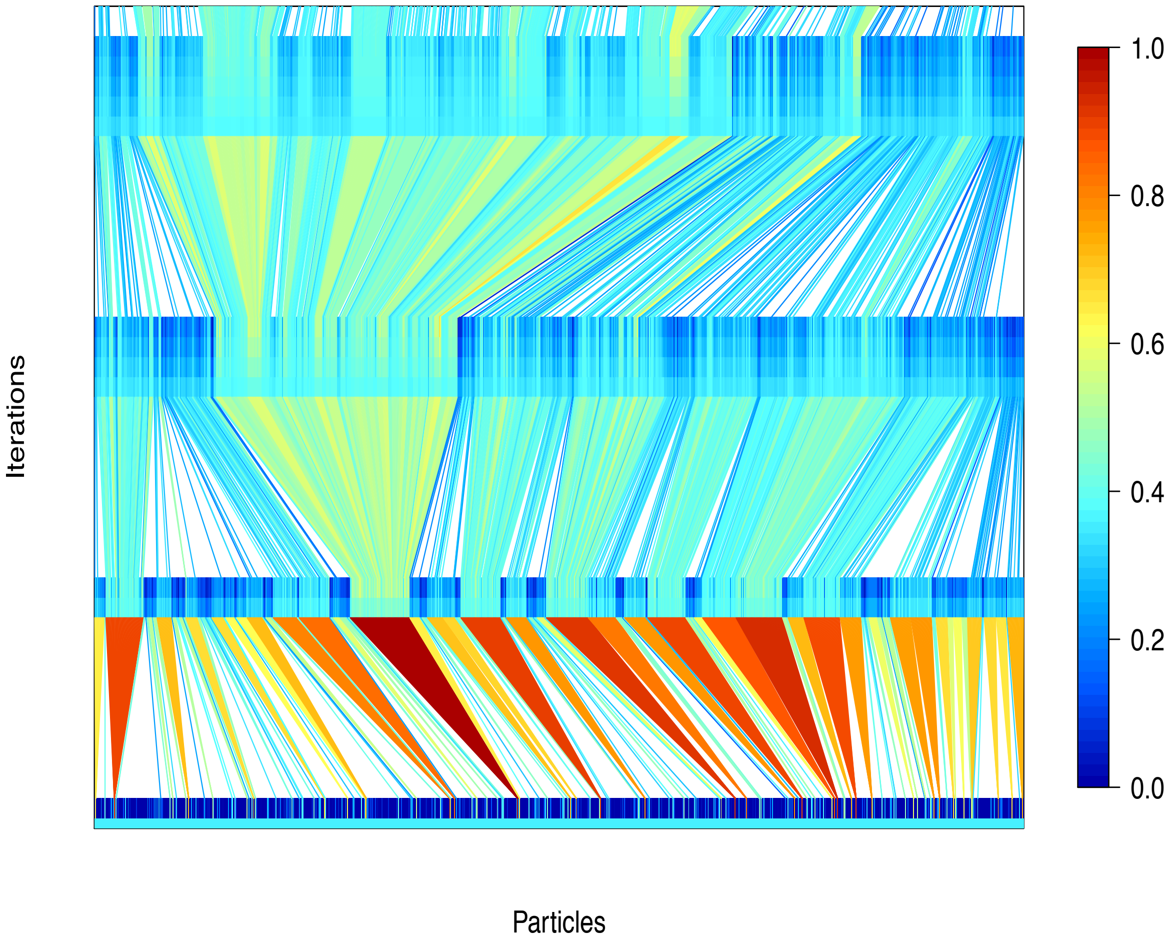


Figure 6.12: SMC-BANN-Min, Initial SMC sampling. The colour key is a relative scale with the range being from dark blue for low weights to red for high weights.

Figure 6.12 shows the starting steps of the SMC algorithm. As in the SMC-BANN-Full model, the starting distribution can be dominated by a limited number of particles limiting the amount of exploration of the parameter space. In Figure 6.12 the darker red and orange segments dominate during the first resample step. This corresponds to the initial points on the left hand side of Figure 6.10.

6.1.3.2 Model: SMC-BANN-Min, Summary

The SMC-BANN-Min model shows a slight improvement with respect to AUC values over the SMC-BANN-Full model. However, it is still well below the best performing logistic regression models of Chapter 5. The complexity of this model still requires the estimation of hundreds of parameters resulting in poorer performance when compared to the simpler logistic regression models. As noted before, the SMC technique is more difficult to initialise than in the penalised logistic regression models. The technique does however recover well from the poor start and provides an even distribution of model weights for the final distribution.

Again, in order to report some measure of variability the bootstrap runs were reduced to five requiring in the order of 30 hours of calculations.

6.1.4 Neural network models, Summary

This chapter has described the performance of two neural networks which have been trained and tested on the same data that was used to build the logistic regression models of Chapter 5. The predictive ability of the models, as measured by the AUC metric (Figure 6.5 for the SMC-BANN-Full model and Figure 6.9 for the SMC-BANN-Min model), is similar to, but lower than, the much simpler logistic regression models. The SMC procedure performed well as a method of estimating the parameters of the model. The algorithm produced an increase in performance when compared to estimation using "off-the-shelf" procedures which shows that careful tuning of neural networks can provide better performance.

However, this flexibility has not produced any increase in performance from the neural network models and suggests that the underlying structure of the data does not warrant the use of these complex models.

Chapter 7

Model Assessment Using Clinical Data

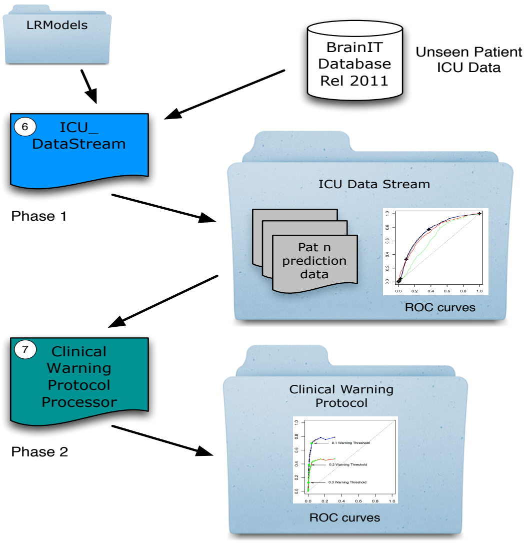


Figure 7.1: Assessing models using unseen patient ICU data

7.1 ICU data stream

The final stage of the model building process is the assessment of the models using unseen clinical data. A cohort of 30 patients from the 2011 release of the BrainIT database was prepared using the Event Analysis Application (EAA) program from Section 4.1.1. This application creates the event and episode list for each patient. Applying the EAA to the model assessment cohort provides the following characteristics.

Total Patients	30			
Females	9	(30.0%)	Median age 44	Range 17 — 64
Males	21	(70.0%)	Median age 33	Range 15 — 68
Total Episodes			Total Events	
864 (ERF=80.1%)			1079	
Type of Trauma				
Fall	Pedestrian	Sport	Traffic Accident	Work
8	2	3	14	3

Table 7.1: Model assessment cohort — demographic summary

Table 7.1 and Figure 7.2 show the demographic characteristics of the data and plots of statistical measures of the hypotensive episodes that occurred within the model assessment cohort. The episode reduction factor (ERF), shows the reduction from EUSIG events to the more relevant *episodes* which cause the clinical teams most concern. These demographic measures are broadly similar to those which were used to construct the system (see Table 4.1 and Figure 4.2).

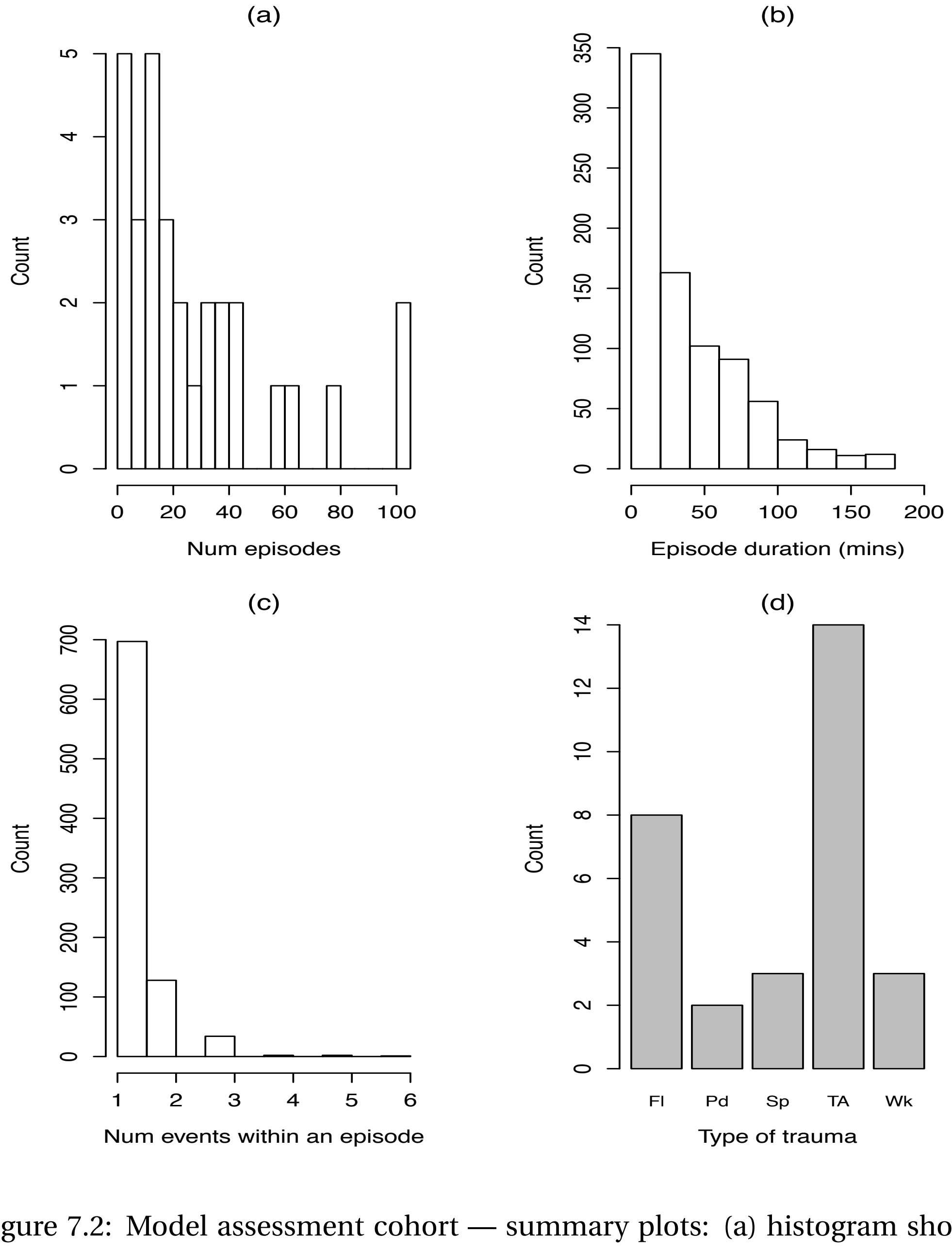


Figure 7.2: Model assessment cohort — summary plots: (a) histogram showing the number of patients (Count) having a given number of episodes during their stay in the ICU; (b) the spread of episode durations across the cohort, limited to durations below three hours; (c) the number of events contained within an episode; (d) types of trauma, Fl = Fall, Pd = Pedestrian, Sp = Sport, TA = Traffic Accident, Wk = Work.

The assessment procedure is to use a given model and calculate an episode probability from each new vector of each new minute of data. This is how a real system would need to operate in an ICU. In order to carry out this assessment, an application, ICU_DataStream, was written. This application takes the minute-by-minute data and carries out the required statistical processes across the model's window size and then presents these pre-calculations to the model equation. The model result is the probability of a hypotensive event starting in the next “event horizon” minutes. For the models being assessed, which have been trained on an event horizon of 10 minutes, this can be read as — "the probability that this patient will have a hypotensive event in the next 10 minutes". The complete assessment process is illustrated in Figure 7.3.

7.2 Testing the model using clinical data

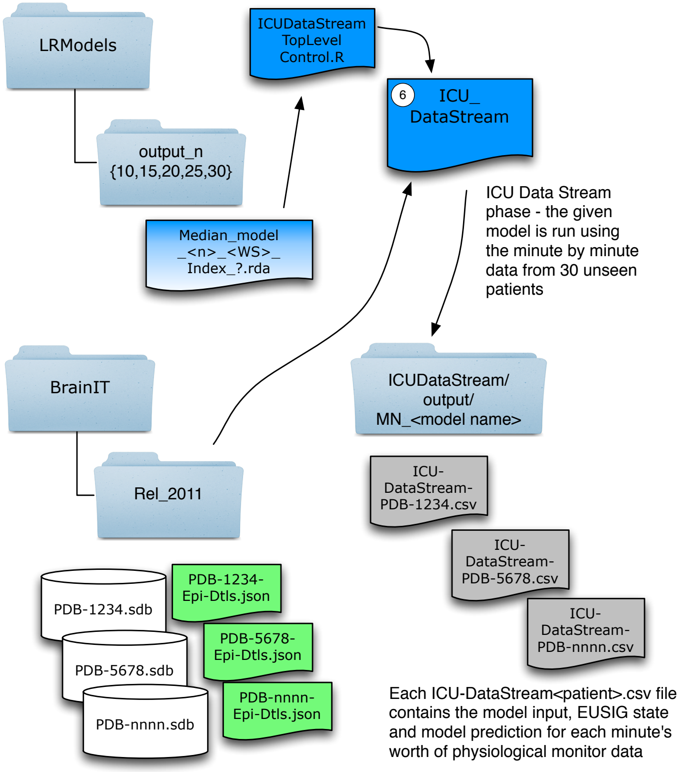

LRModels
output_n {10,15,20,25,30}
Median_model _<n>_<WS>_ Index_?.rda
ICUDataStream TopLevel Control.R
6
ICU_ DataStream
ICU Data Stream phase - the given model is run using the minute by minute data from 30 unseen patients
BrainIT
Rel_2011
ICUDataStream/ output/ MN_<model name>
ICU-DataStream-PDB-1234.csv
ICU-DataStream-PDB-5678.csv
ICU-DataStream-PDB-nnnn.csv
PDB-1234.sdb
PDB-5678.sdb
PDB-nnnn.sdb
PDB-1234-Epi-Dtls.json
PDB-5678-Epi-Dtls.json
PDB-nnnn-Epi-Dtls.json
Each ICU-DataStream<patient>.csv file contains the model input, EUSIG state and model prediction for each minute's worth of physiological monitor data

Figure 7.3: Phase 1 of model assessment on clinical data — ICU DataStream

The results from running the model using a complete minute-by-minute set of ICU data are stored in a .csv file. At the end of the test run, the results from all 30 patients are combined and used to produce the ROC curves shown in Figure 7.4.

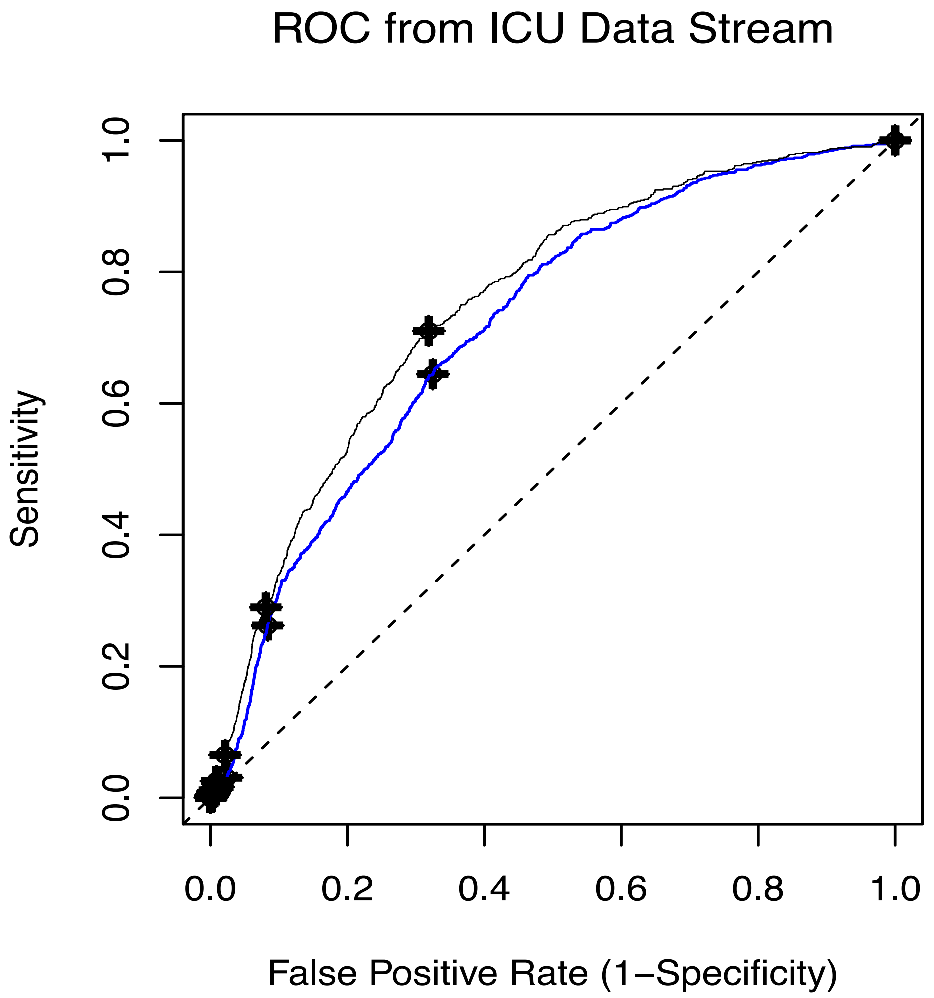


Figure 7.4: ROC for clinical data

Figure 7.4 shows the ROC curves produced by combining the prediction data from 30 unseen patients. For the 10 minute event horizon (EH), this produces a file with 416945 rows of data, made up of 717 positive cases (i.e. timestamps where a hypotensive event did occur EH minutes later) and 416228 negative cases (i.e. the patient was in a stable state). The black line is from the Full 10_5 model; the blue line is from the Minimum 10_5 model.

A criticism of this method of assessing model performance is that it uses too narrow a definition of what constitutes a true positive outcome from the model. If the model delivers a raised probability of an episode 15 minutes before an event, it will be penalised for having produced a false positive even though in fact, it provided the clinical team with *more* warning of a possible hypotensive episode. Using this technique it can be expected that the number of true positives will be biased low while the number of false positives will be biased high. This might result in a valuable model being discarded. In order to provide a more realistic assessment of the model's performance an alternative protocol is proposed in Section 7.3.

7.3 The Clinical Warning Protocol — testing the model in a clinical setting

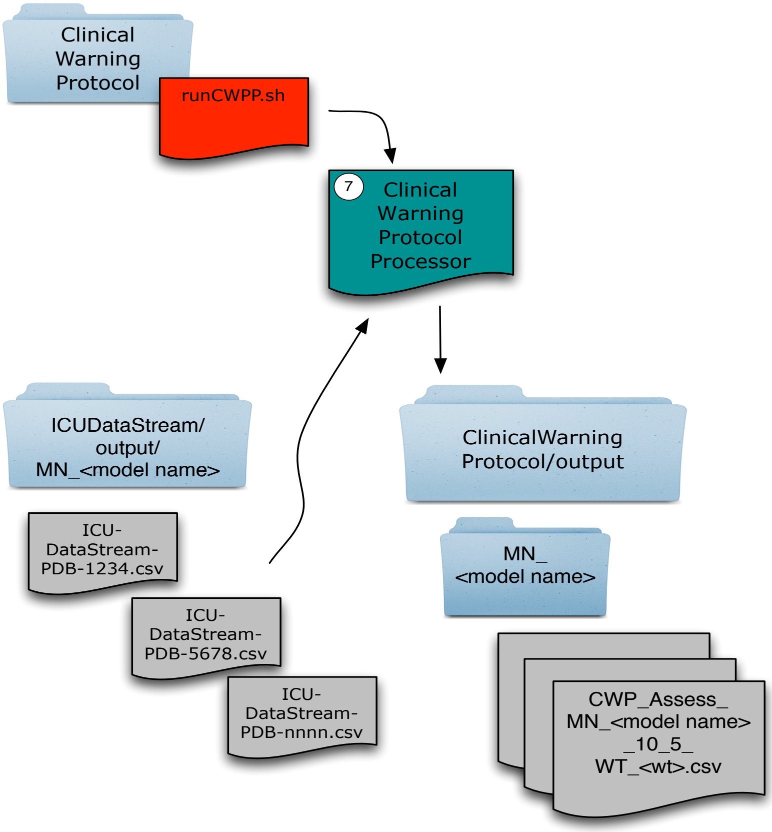

Clinical Warning Protocol
runCWPP.sh
7
Clinical Warning Protocol Processor
ICUDataStream/ output/ MN_<model name>
ICU-DataStream-PDB-1234.csv
ICU-DataStream-PDB-5678.csv
ICU-DataStream-PDB-nnnn.csv
ClinicalWarning Protocol/output
MN_ <model name>
CWP_Assess_ MN_<model name> _10_5_ WT_<wt>.csv

Figure 7.5: Phase 2 of model assessment on clinical data — Clinical Warning Protocol processor (CWPP)

The clinical warning protocol (CWP) is proposed to provide an alternative assessment of model performance. There are two key elements to the protocol. The first is that the model is predicting an episode might start *within* the next 30 minutes rather than at an exact time. The second is the assumption that once a clinical team has been called out, they are prepared to stay in a heightened state of readiness for 30 minutes. If a hypotensive episode does start, it has been a true positive call. If an episode does not occur it has been a *single* false positive call. The effect is that the protocol coalesces individual minute-by-minute predictions into clinical blocks of time. The 30 minute setting is a judgement of the clinical team. It does not mean a senior neurosurgeon stands by the bed, it means the whole team is on heightened alert.

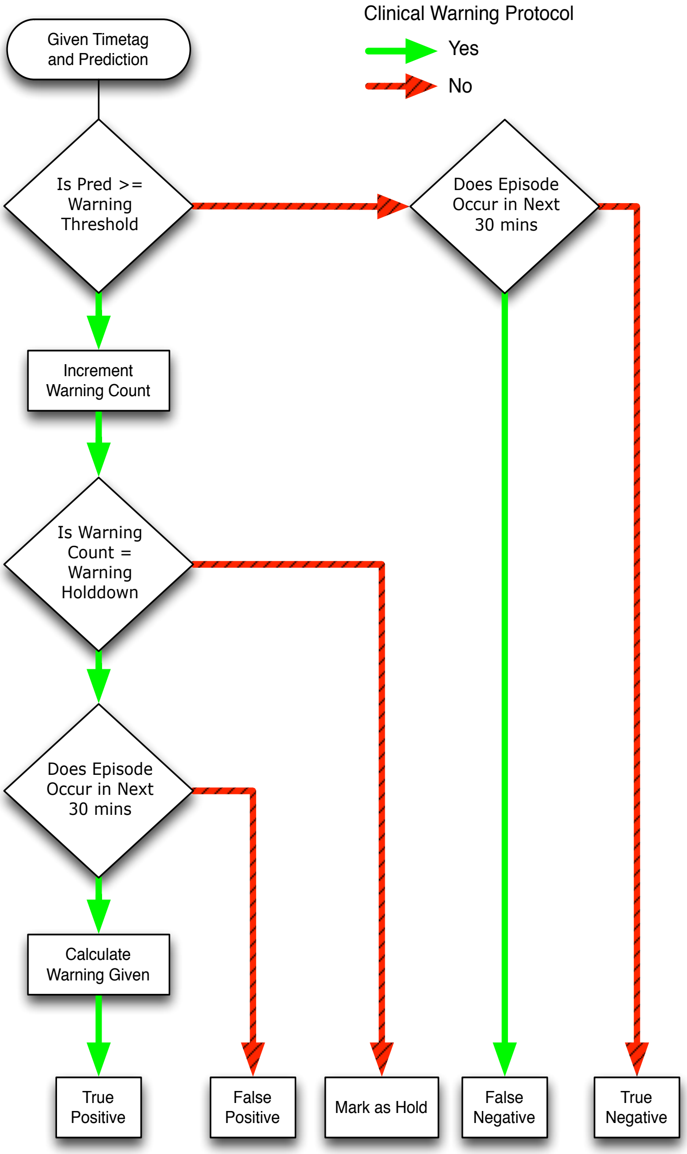

Clinical Warning Protocol
Yes
No
Given Timetag and Prediction
Is Pred >= Warning Threshold
Does Episode Occur in Next 30 mins
Increment Warning Count
Is Warning Count = Warning Holddown
Does Episode Occur in Next 30 mins
Calculate Warning Given
True Positive
False Positive
Mark as Hold
False Negative
True Negative

Figure 7.6: Clinical Warning Protocol

Using the protocol described in Figure 7.6, with warning holddown set to three minutes, the Minimum model produces the results in Table 7.2 which are plotted in the pseudo ROC curves shown in Figure 7.7.

study id	age	gender	trauma type	episodes	sens	sens2	spec	avg wrn
PDB-16138373	45	Male	Fall	4	100.00	100.00	99.34	27.25
PDB-72705050	57	Male	Work	2	0.00	0.00	99.82	0.00
PDB-83705040	56	Female	Fall	19	0.00	0.00	99.98	0.00
PDB-84884871	24	Male	Traffic Accident	33	90.62	87.88	95.03	27.41
PDB-84884946	26	Male	Fall	1	0.00	0.00	98.20	0.00
PDB-84884950	33	Male	Traffic Accident	9	100.00	100.00	96.65	26.22
PDB-84884951	64	Female	Fall	13	0.00	0.00	99.83	0.00
PDB-84884952	36	Female	Sport	103	22.77	22.33	99.60	18.61
PDB-84884953	17	Female	Pedestrian	24	20.83	20.83	99.71	22.60
PDB-84884954	57	Male	Fall	62	66.13	66.13	98.35	23.12
PDB-84884956	59	Male	Fall	27	70.37	70.37	98.97	19.68
PDB-84884958	40	Male	Traffic Accident	40	77.78	70.00	97.07	22.82
PDB-84884959	15	Male	Traffic Accident	57	76.36	73.68	97.40	23.29
PDB-84884960	44	Female	Traffic Accident	25	8.00	8.00	99.80	23.50
PDB-84884961	41	Male	Traffic Accident	12	58.33	58.33	98.74	26.14
PDB-84884962	32	Male	Traffic Accident	7	71.43	71.43	98.85	22.40
PDB-84885004	18	Male	Sport	39	94.74	92.31	95.29	27.03
PDB-84885005	18	Male	Pedestrian	104	90.82	85.58	96.83	23.12
PDB-84885006	51	Female	Traffic Accident	11	0.00	0.00	99.80	0.00
PDB-84885007	25	Female	Traffic Accident	3	33.33	33.33	99.67	17.00
PDB-84885009	62	Male	Fall	15	53.33	53.33	99.47	24.50
PDB-84885011	61	Female	Sport	41	7.32	7.32	99.84	13.33
PDB-84885012	26	Male	Traffic Accident	7	83.33	71.43	99.49	25.80
PDB-84885015	28	Female	Traffic Accident	34	27.27	26.47	99.69	19.56
PDB-84885017	24	Male	Work	79	87.18	86.08	96.47	24.00
PDB-84885060	17	Male	Traffic Accident	16	93.33	87.50	98.07	23.14
PDB-84885062	42	Male	Traffic Accident	17	73.33	64.71	98.67	25.36
PDB-84885067	68	Male	Traffic Accident	11	45.45	45.45	99.53	20.40
PDB-84885068	42	Male	Work	5	60.00	60.00	99.63	18.33
PDB-84885073	19	Male	Fall	44	88.37	86.36	98.03	26.53

Table 7.2: Model assessment summary results, Model: Minimum, 0.15 warning threshold — sens = sensitivity; spec = specificity; avg wrn = average warning given in minutes

Table 7.2 above presents the results for the 30 patient cohort using the 0.15 decision threshold. These results show, for patients detected by the system, the average warning is 23 minutes (sd=4). There is warning given for 25 of the 30 patients (83%).

Two sensitivity results (sens and sens2) are reported. This is to account for the two scenarios caused by complete and incomplete data respectively. Incomplete data results in the model being unable to make a prediction under certain conditions. There are occasions e.g. when the data stream resumes after a break, when an episode will start before the model has sufficient data with which to operate. Consequently the number of episode starts does not match the sum of the true positive calls and the false negative calls i.e. $numEpi-(numTP+numFN)\neq 0$. This mismatch check is carried out by the Clinical Warning Protocol processor (CWPP) software, which is processing the minute-by-minute data, and is reported in the full table of results in Appendix B, B.1 as the column, statsCheck.

As mentioned above, the decision was taken to report both complete and incomplete data scenarios. In Figure 7.7, the blue trace uses the assumption that the missed episode could not be reported due to missing data and it is therefore excluded from the sensitivity calculation. This is reported in the column, sens. The red trace includes the missing episode as a false negative, thereby reducing the sensitivity. This is reported in the column, sens2.

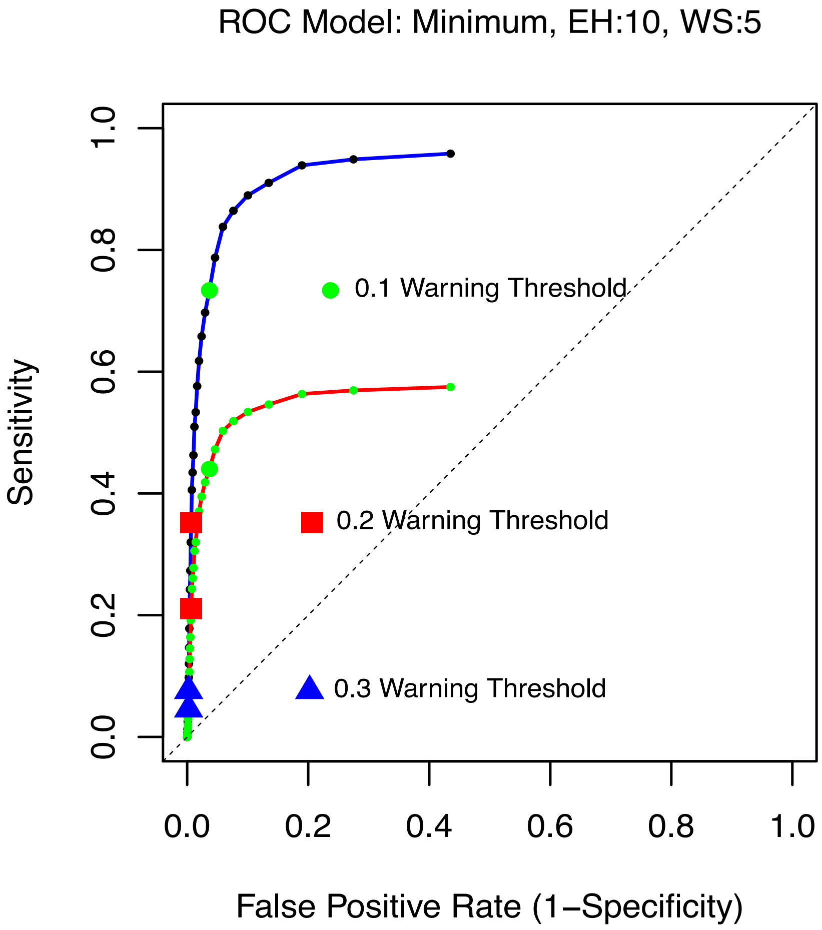


Figure 7.7: Model: Minimum, ROC using Clinical Warning Protocol on ICU data stream

Figure 7.7 shows a pseudo ROC curve constructed from the results produced by the CWPP running the Minimum model. This suggests that, with a decision threshold of between 0.15 and 0.2, the system would detect 30 to 50% of episodes.

7.4 Visual checks — physiological signals and model predictions

The final section in this chapter provides a visual check of the model performance by plotting the time series data as it would appear to a clinical team.

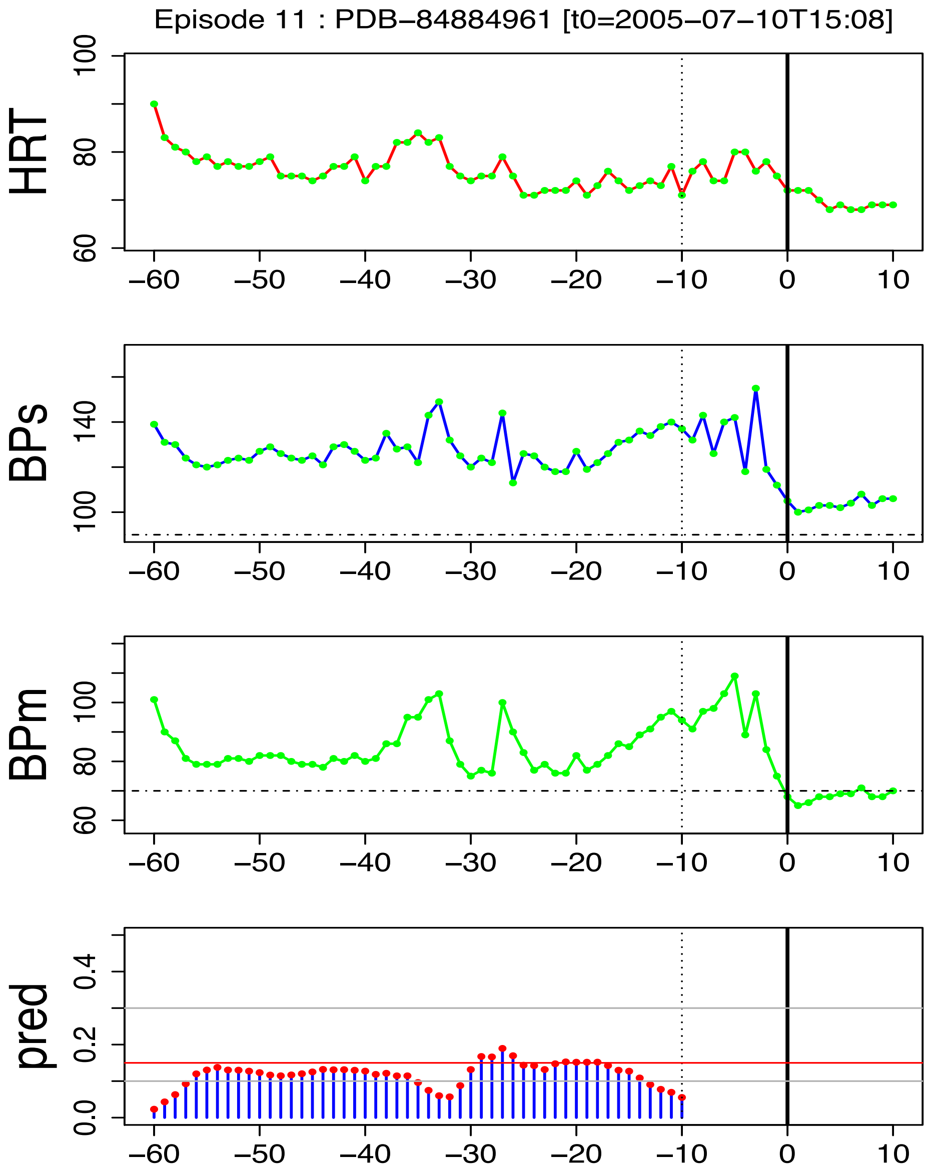


Figure 7.8: Visual checks, Model: Minimum, PDB-84884961 episode 11, warning threshold = 0.15, y-axis pressures in mmHg, HRT in beats/min, x-axis in minutes, pred = output from Minimum model

In Figure 7.8 from patient 84884961, the three physiological signals HRT, BPs and BPm are plotted along with the output of the Minimum model. This shows increased probability approximately 25 to 30 minutes before the episode start.

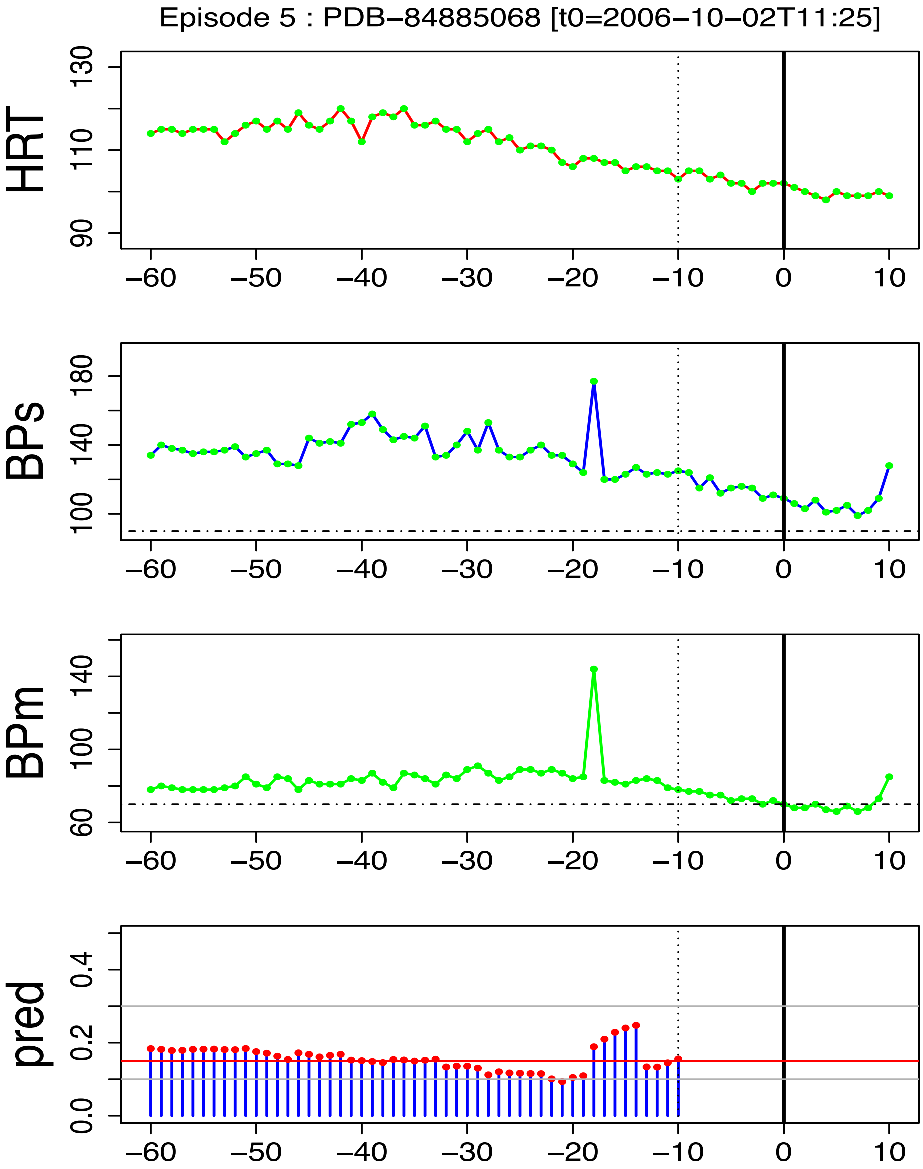


Figure 7.9: Visual checks, Model: Minimum, PDB-84885068 episode 5, 60 mins, warning threshold = 0.15, y-axis pressures in mmHg, HRT in beats/min, x-axis in minutes, pred = output from Minimum model

In contrast, Figure 7.9 shows the signals from patient 84885068 as they are possibly more typical of a clinical situation. It can be seen that warning is given 15 to 20 minutes before the episode start. However, there has also been a false positive call at approximately 60 to 40 minutes before the episode. If viewed across 180 minutes, as shown in Figure 7.10 below, it could be argued that the system has given about 90 minutes warning of an increased risk of a hypotensive episode, as the risk probability is very low before the $t-90$ marker.

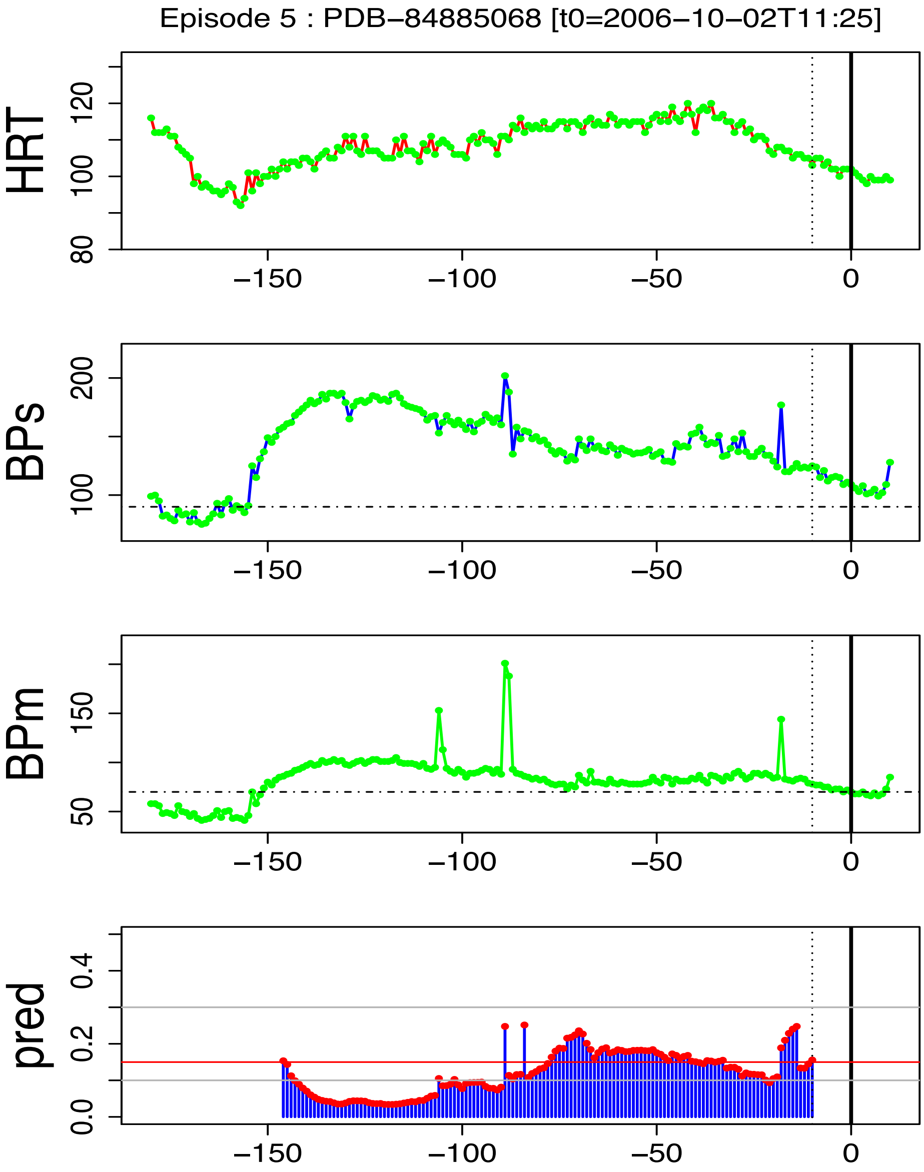


Figure 7.10: Visual checks, Model: Minimum, PDB-84885068 episode 5, 180 mins, warning threshold = 0.15, y-axis pressures in mmHg, HRT in beats/min, x-axis in minutes, pred = output from Minimum model

Chapter 8

Discussion and Conclusions

This chapter provides a discussion of the effectiveness of the models proposed in Chapters 5 and 6 and offers some conclusions regarding the research to date.

8.1 Discussion

The research carried out in this thesis is attempting to answer the question, " ... can simple logistic regression models provide early warning of the start of a hypotensive episode". In order to investigate this question both simple statistical models, using a limited number of inputs and well established statistical techniques, and advanced models, using a large number of possible contributing variables and modern up-to-date statistical methods, have been constructed and tested on real-world clinical data. The results are discussed below and also compared with the current state-of-the-art project that has been carried out in this domain.

Three broad areas of modelling approach have been tested using eight models. These three areas, in order of decreasing complexity, are neural networks, penalised logistic regression and standard logistic regression. Figure 8.1 shows the best performing model from each area along with the simplest model in terms of complexity and inputs.

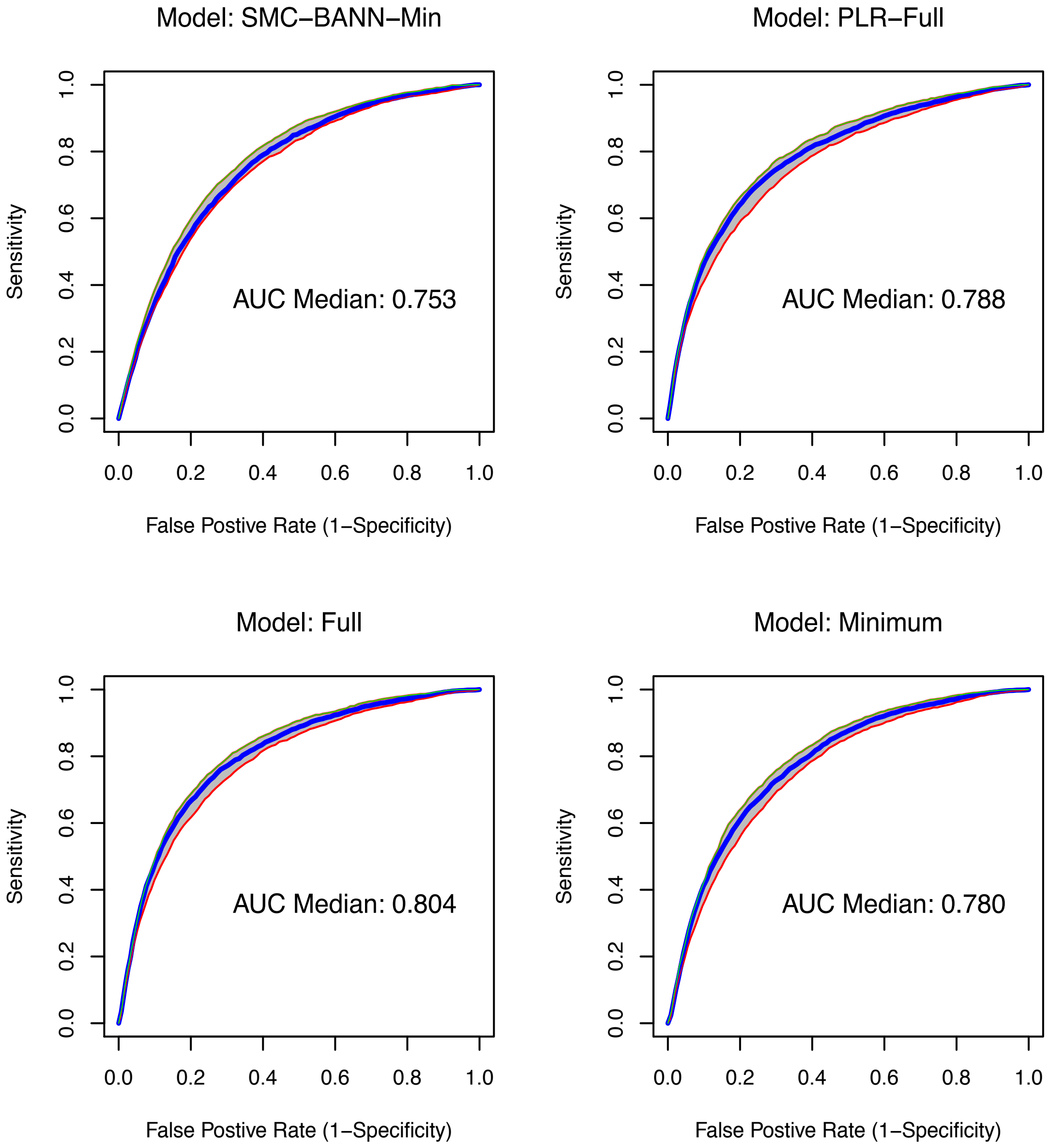


Figure 8.1: Model Assessment Summary Results, the best performing models from each modelling approach along with the simplest model (Minimum). The top row shows models using “All Data”, top left is from the neural network approach, top right used a penalised logistic regression method. The bottom row shows the best performing “Stats Based” model (Full) and the simplest model (Minimum) which is also “Stats Based”. These models were built using the standard R **glm()** function. All models perform broadly the same.

It can be seen that all the models perform broadly similar with the simpler logistic regression model, Full, giving slightly better results. Variants of each technique were built and tested and as expected the more complex logistic regression models can be tuned to increase the AUC metric by a small amount. Table 8.1 reports the median AUC for all the models that were investigated.

Model Name	AUC	Data Source	Approach
SMC-BANN-Full	0.740	All Data	Neural Networks
SMC-BANN-Min	0.753		
PLR-Full	0.788	All Data	Penalised Logistic Regression
SMC-PLR-Full	0.776		
PLR-Min	0.785		
SMC-PLR-Min	0.778		
Full	0.804	Stats Based	Logistic Regression
FullQuadMean	0.804		
Minimum	0.780		
Full-Lasso	0.797	Stats Based	Penalised Logistic Regression

Table 8.1: All Model Approaches AUC Summary, median AUC is reported for all models, grouped by data source and modelling approach.

The results from this research are in line with the results from the AvertIT (2008) project which used the same BrainIT Consortium (2007) clinical database for model training. The AvertIT project used a sophisticated Bayesian Artificial Neural Network (BANN) written in the C programming language. Their system is based on the work of Neal (1996). The project was able to show that the results could be carried forward into a clinical setting (Donald et al., 2012b) and are the only group that have been able to to this within the TBI domain. Their system used a windowing approach to reduce the input dimensionality to the model. As such it is a combination of the two techniques investigated in this thesis i.e. it used a summarising technique for the model's input section coupled with a complex model structure in the form of a neural network.

In Chapter 6, by taking the general architecture from the AvertIT project, but also attempting to simplify their approach, two neural networks were constructed to investigate the effect of using each minute of available data rather than the summary techniques across a window of data. The parameter estimation was carried out using a Bayesian framework and a sequential Monte Carlo (SMC) approach. Although this method would be considered as a fairly complex technique it is arguably less complex than the Bayesian framework with Hamiltonian dynamics based parameter estimation system used by the AvertIT group. The group has, so far, only published early results (Stell et al., 2009, 2010; Donald et al., 2012b) however they have kindly agreed to supply the ROC data from their model construction stage. This is presented in Figure 8.2 along with the best performing neural network model from this research.

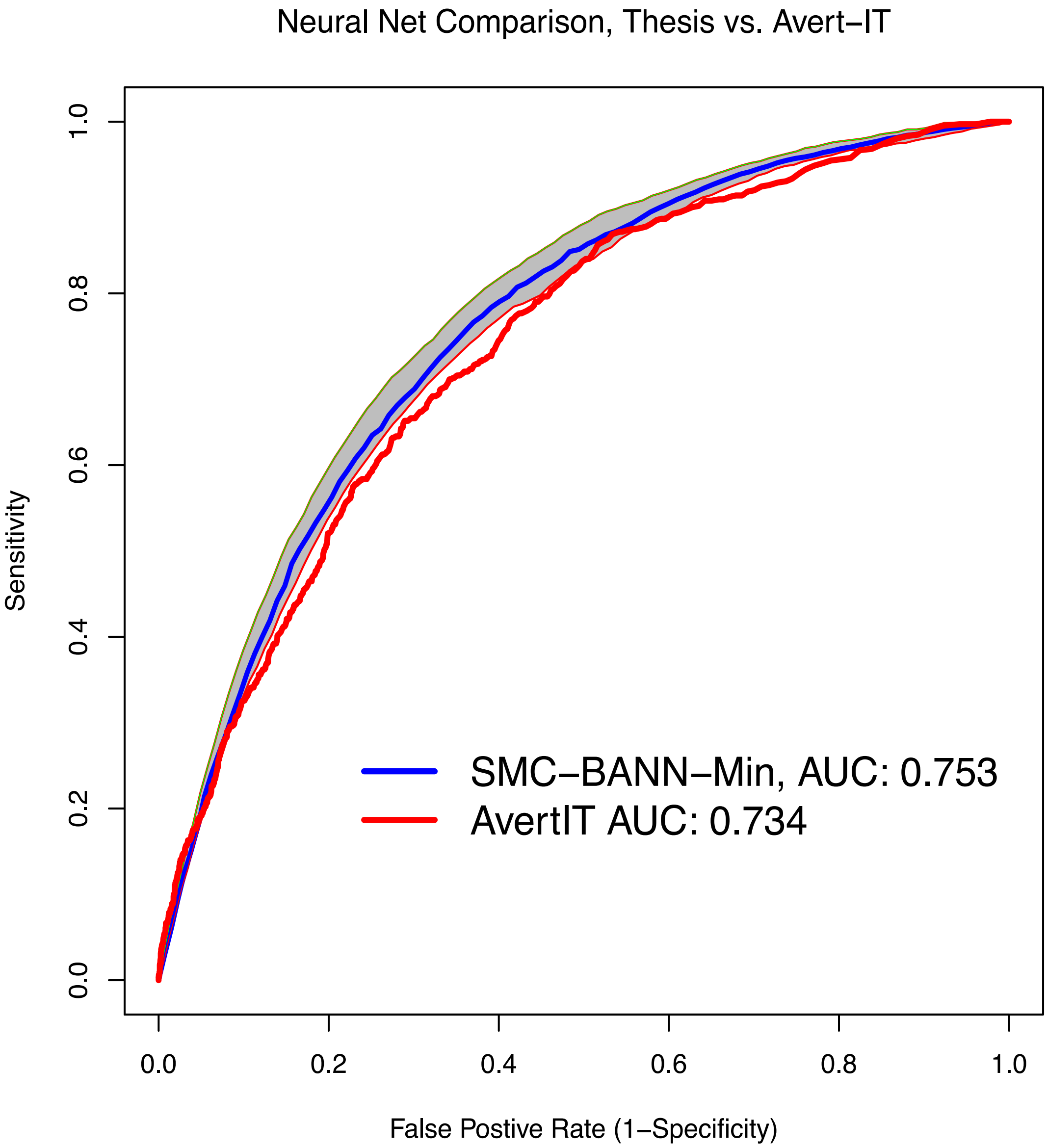


Figure 8.2: AvertIT ROC vs SMC-BANN-Min

Figure 8.2 shows that the neural networks models constructed as part of this research performed slightly better than the Avert-IT BANN. The performance improvement is marginal and it is known that the Avert-IT group were unable to optimise their model due to funding constraints.

Although neural networks *can* be constructed to address the hypotension prediction problem they are complex models to set up as noted by Faraway and Chatfield (1998). They require architectural decisions beyond the choice of inputs e.g. number of hidden nodes, type of node activation units. The SMC parameter estimation technique also requires algorithm configuration choices to be made such as the number of particles and number of auxiliary distributions. Perhaps the most challenging aspect of the neural network model is that they are very difficult to explain to clinicians as the model parameters are not directly related to the clinical inputs of the system.

Turning to simpler, and to clinicians more familiar, models, six logistic regression models, using two different input architectures, were proposed in Chapter 5.

The “All Data” models were constructed using penalised logistic regression as this technique can guard against the expected correlation between the adjacent minutes of the data buffers. These models performed well but looking at the coefficient profiles in Figures 5.10 and 5.19 it is clearly seen that the initial five to ten minutes of the data buffers are contributing most to the probability output of the model. The sequential Monte Carlo (SMC) approach was also used to estimate the parameters of the penalised logistic regression models and it gave very similar AUC performance, 0.776 vs. 0.788 for the PLR-Full penalised logisitic regression model built with maximum likelihood parameter estimates. For both the PLR-Full and PLR-Min models, using SMC for parameter estimation gave slightly lower performance. The main reason for this is that the Bayesian model does not appear to penalise the coefficients enough. This is due to the Bayesian model being more “honest”. The distribution of λ is based on the prior and the training data whereas for the penalised logistic regression models λ was chosen such that it minimised the test set error.

The technique of using summary measures across the data buffers was investigated with three models using standard logistic regression. The models were: the Full model — based on several statistical measures from the heart rate and arterial blood presssure signals (Section 5.4.1); the FullQuadMean model — which added a quadratic term to each of the mean values of the first model to provide a limiting device to an increase or decrease of the base signal (Section 5.4.2); and finally the Minimum model — guided by clinical experience which used a minimum number of signals, in fact only six i.e. age and gender along with the mean values of heart rate and mean arterial blood pressure and the standard deviation of heart rate and the mean arterial blood pressure (Section 5.4.4).

As expected, the correlation between blood pressure terms allows variables to be dropped and the Minimum model, using only BPm from the blood pressure signal, performed well. Because the system is predicting episodes rather than investigating causes, it does not matter which covariates are used as long as the system gives a good prediction. However, it *does* matter when trying to explain/interpret the model. These models also make sense when comparing them to the penalised logistic regression models using the "All Data" input architecture as the penalised logistic regression models show that in general it is the first five to ten minutes which are contributing to the response. The standard logistic regression models use the statistical preprocessing of the input buffer to reduce the complexity of the model and do not appear to lose any information during the process.

The research also investigated the use of the Lasso (Tibshirani, 1996) penalty technique with the Full-Lasso model — which is an attempt at algorithmic feature selection (Section 5.4.3). This model performed well, indeed it gave the second highest score by AUC assessment, but the parameter profile seen in Figure 5.31 would be difficult to explain to a clinical team. For example age and gender are effectively dropped from the model.

In summary, all the models were able to detect hypotensive episodes and all showed similar characteristics. From Section 5.5, the statistical measures based models worked better with smaller event horizons. Apart from the Minimum model, window size appears to have little effect on predictive ability. Intuitively, this is to be expected. It would appear that physiologically, the internal processes which subsequently combine to cause a hypotensive episode, start in the region of 10 to 15 minutes before the episode commences.

When comparing this research to the current state-of-the-art results from the AvertIT project, the observation that most of the information seems to appear in the 15 minutes leading up to the episode may explain why the logistic regression performs as well as the much more complex technique encompassed in the Bayesian Artificial Neural Network (BANN). The AvertIT group used data that was further in front of the event; their event horizons were effectively 15 minutes and 30 minutes. The finding that shorter buffers of information i.e. smaller window sizes, improve the prediction is also of interest. This would suggest that as more information is processed, the averaging effect smooths out the predictive ability. Again, the AvertIT project used a buffer of 15 minutes of information whereas the results from this thesis used 30 minute data buffers.

The results using the standard technique of 50% training data and 50% test data were positive however, it would be of interest to see what level of improvement could be gained by using more training data and a cross validation approach.

Assessing the models on an unseen cohort of patients showed that the Minimum model gave similar predictions to the Full model and has the advantage of being easily explained to clinicians. There is some warning given for 25 of the unseen cohort of 30 patients (83%). Considering the 25 patients where some warning was given, the Minimum model was able to give an average early warning of 23 minutes (sd = 4). It is interesting to note that for five of the patients in the cohort no warning was given at all. A brief investigation of these five patients shows that three of the five were female over 50 years old. In each of these three cases there were more than 10 episodes. The two male patients, age 57 and age 26, had very few episodes, two and one respectively. It would be informative to see if this pattern is repeated using a larger cohort.

From the interpretation of the model coefficients for each model, it would appear that gender has a strong influence, with males having an increased risk of hypotensive episodes. There has been research into this effect by Berry et al. (2009). This group observed that peri (age 45 - 54) and post (age ≥ 55) menopausal females had a lower risk of mortality or further complications after TBI. The group does not offer any explanation for the cause of this effect. However, it may also be the case that gender is acting as a proxy for the type and severity of injury. For example, from the demographics of the BrainIT database, males make up 83% of the road traffic accidents. The median age of this group is 30 (n=52).

It is notable that in both patient cohorts, i.e. the training/test cohort described in 4.1.3 and the clinical validation cohort from Section 7.1, that approximately 20% of episodes contained multiple events. Using the additional tables contained within the BrainIT database, further research could investigate whether these are linked to other clinical events such as blood sampling or physiotherapy.

The introduction of the model into a clinical setting is the ultimate goal. However, assessing how it behaves in a clinical environment could be problematic. Section 7.3 proposed the clinical warning protocol, which crucially assumes that the clinical team will observe the patient for some period of time (in Section 7.3 this was 30 minutes) to see if a hypotensive episode will develop. In practice this would not happen. In a real situation the clinical team, once alerted, will use their experience to decide on a course of treatment. The obvious aim of this course of treatment would be to prevent a hypotenive episode from occurring. If successful, in terms of assessing the model, there will be *no* event within 30 minutes. How then will this be scored? The model has, in a sense, done its job effectively but there is no concrete objective proof that a hypotensive episode would have occurred. Any clinical trial would need to include some mechanism for testing this practical situation as there would be the possibility of unnecessary treatments being carried out and a bias (positive or negative) thus being introduced into the model performance figures. One could argue, of course, that if a model is producing these predictions this has a beneficial effect in terms of patient care even if the actual mathematical performance of the model appears unreliable. These concerns would have to be addressed by the ethics committee scrutinising any proposed clinical trial.

Substantial discussion on clinical usefulness and validation of prediction models can be found in chapters 16 and 17 of the text by Steyerberg (2010).

8.2 Conclusions

This thesis set out to investigate whether or not *simple* logistic regression models could provide early warning of impending hypotensive episodes. The answer is clearly yes. The evidence provided allows the conclusion to be drawn that the simpler models could contribute to further research with an aim of improving the understanding of the factors affecting these predictions.

The preparation of the data is key to the research and Chapter 4 details how this was achieved. Generating the base data sets was by far the most computationally intensive part of the research. The availability of the multi-core server within the University of Glasgow School of Mathematics and Statistics was a significant benefit. However, the emerging ease with which parallel computing systems can be constructed e.g. Hadoop, (The Apache Software Foundation, 2011), and the SNOW package for R, (Tierney et al., 2012), suggests that, with minimal rearrangement of the code base, this stage of the project could be run on larger data sets on standard desktop computers. The code base for the BSG, TTG and CWPP applications has been written in a map reduce style with a top level controlling program processing a routine for a single patient.

Having decided on a logistic regression approach, the model construction and testing is relatively straightforward due to the tools provided by the R statistical programming framework.

The final contribution to the TBI domain is the approach taken to assess the model's performance in a clinical setting. The clinical warning protocol described in chapter 7 and the tool set used to run the protocol would allow rapid assessment of alternative models.

There are several avenues of further research to which this thesis could contribute; the topic of future work is discussed in the next chapter.

Chapter 9

Future Work

A project of this nature often unveils new avenues of possible research. However, in order to complete any thesis, only one approach can be fully explored. From this thesis five key areas have been identified which could form potential future projects for research. These topics are: data quality; episode characteristics; additional covariates; model construction; and clinical acceptance. The following sections provide more detail on each topic.

9.1 Data quality

Starting from Chapter 2, Section 2.2.1, the definition of an event itself specifies the possibility of an “allowable data gap”. This is a consequence of using “real” clinical measurements which frequently contain gaps. The EAA has used a very strict definition of an event and if a break in the data is detected whilst the event is “Active”, the event is discarded from the list of possible events. It would be useful to assess the characteristics and magnitude of this phenomena and understand whether or not a small gap time could be tolerated. It is assumed this would result in more events being added to the list thereby providing additional material for research.

As well as the data frequently containing gaps, there are also patterns which can indicate that the measuring sensor is faulty. Two examples of this type of measurement fault are: when a signal becomes very stable and does not exhibit the normal physiological variation; and when the the difference, $BPm - BPd$ drops below 10 mmHg. Models to identify this type of artifact in the neonatal ICU domain have been studied by a group at the University of Edinburgh, (Williams et al., 2005; Quinn, 2007).

9.2 Episode characteristics

The EAA application produces additional output beyond that required for this thesis. The episode and event list could be studied to look for patterns and correlation between episodes and events, e.g. patterns showing the time from the end of one episode to the start of the next. In addition there are other questions which could be asked regarding episode patterns. Do events have a time pattern i.e. are there more events during the evening/night? Can episodes be attributed to circadian rhythms? Do episodes display a gender pattern?

As previously discussed, it has been noted in both cohorts of patients that approximately 20% of episodes contained multiple events. Using the additional tables contained within the BrainIT database, further research could determine whether or not this multiplicity is linked to other clinical events.

9.3 Additional covariates

The signals that were chosen for this research were based on the availability of data from the BrainIT database and the experience of the clinical teams. The primary signals of interest came from the minute-by-minute physiological measurements. The BrainIT database contains other admission demographic variables that could form additional features that might be included in future models. The inclusion of demographic variables measured at the time of injury and at pre-neurosurgical hospital (PNSH) and neurosurgical hospital (NSH) admission times could also provide stratification information. See the paper by Piper et al. (2010) for a recent description of the BrainIT database.

9.4 Model construction

There are a number of decisions taken during the model construction phase, each of which could become a topic for further research. Perhaps the most beneficial addition to the analysis would be to use a mixed effects model as briefly discussed in Section 3.7. This approach would provide an assessment of the variability of the patient cohort and could ultimately lead to an early warning system that would allow a measure of patient specific treatment. It can be imagined that such a system would initially monitor a newly admitted patient in order to assess to which risk group the patient was likely to belong. These risk groups would be determined from the mixed effects approach. This need to take into account the individuality of each patient's situation is discussed in the work of Nortje and Menon (2004).

Another approach would be to use the techniques of function data analysis. These technique would treat the blood pressure and heart rate signals as functional input into a model.

The use of longer timespans of data, rather than the 30 minute limitation, could be investigated. In particular a measure of the long term trend of the physiological signals over the last 6, 12 or 24 hours could be studied, although artifactual problems could be an issue.

Furthermore, a study could be carried out examining how to combine the output from each one of the useful models into a single prediction. A starting point for this work would be the text by Bishop (2006, Chapter 14).

9.5 Clinical acceptance

The AvertIT project showed that it is possible to obtain early warning of hypotension in approximately 40% of cases, (Donald et al., 2012b; Stell et al., 2012). In the future it is hoped to pursue further research using other techniques to see if this figure can be improved upon, all the time being aware of the need for clinical acceptance. Any research may wish to explore the possibility that a combination of techniques might provide a more acceptable solution. For example, a logistic regression based system, with decision thresholds giving fairly high false positives, working alongside an expert system based on clinical knowledge, may be able to suppress the false positives to an acceptable level. However, while the general notion of combining software and statistics gains clinical acceptance, further research should continue to explore other entirely data driven techniques such as Gaussian processes, HMMs and, of particular interest, combinations of these techniques.

Another major requirement for clinical acceptance is the timely and accurate display of predictions. The introduction of yet another system into a typical ICU environment, which contains many sophisticated pieces of monitoring equipment, often with each one having its own form of alarm enunciation, is problematic. There is considerable literature on the alarm workload of ICU staff, (Gorges et al., 2009; Imhoff and Kuhls, 2006; Chambrin, 2001), which identifies the need for accurate display of information. Because the aim of any future system built on this research will be to provide early warning of a possible hypotensive episode, an acceptable way to raise awareness of this early warning will require careful thought.

Appendix A

Software for Data Preparation and Test

A.1 Background

A large part of the research carried out requires specialised software. This Appendix provides an overview of each application and the commands used to produce the results. More details, and the code available for download, can be found on the project's website `http://www.statsresearch.co.uk`. The following sections assume that the code has been downloaded and installed at a location [INSTALL_DIR].

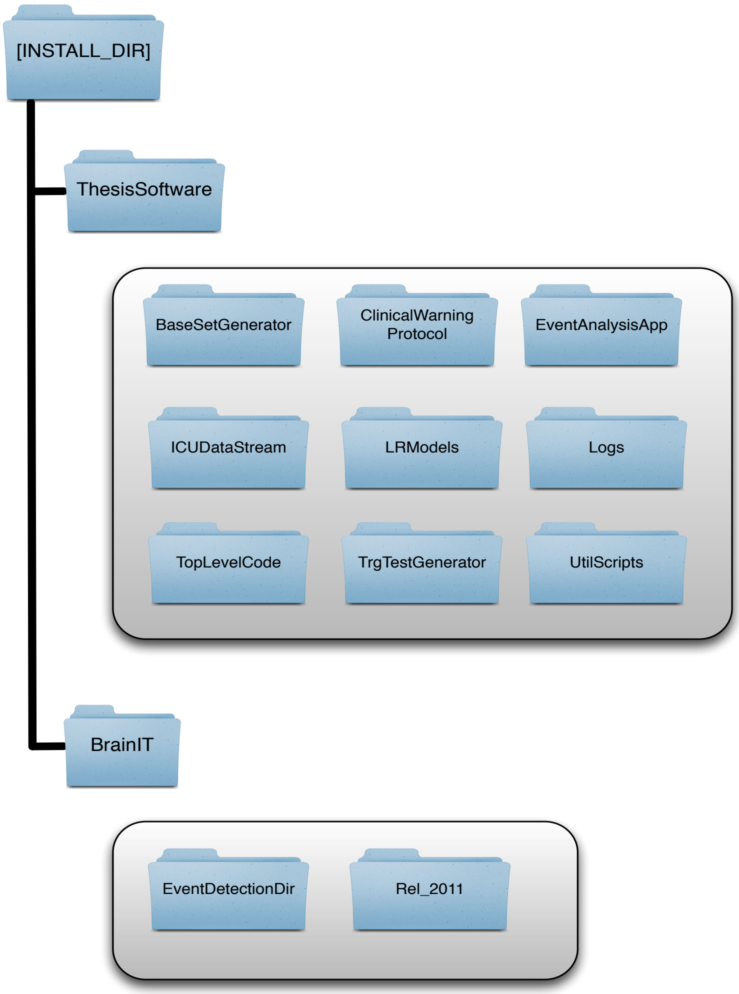


Figure A.1: Project directory structure

A.2 Event Analysis Application

The Event Analysis Application is designed to take a data source which contains time series information and process each time row looking for events which have been predefined in terms of threshold levels and holddown time periods. The application can group events into episodes where an episode is defined as a group of events closely spaced in time. The software also supports the use of entity tables which provide supporting meta data for the time series which is being analysed. The application currently assumes that each patient file is a single SQLite DB, (The SQLite Consortium, 2012).

The application has been designed as a general purpose tool and can be applied to a range of event analysis projects. Some general terminology regarding events and episodes is given in section 2.2.

The application provides support for automatically handling files that fail to process. These are marked as <fileName>.INVALID and can then be processed as required. The commands to run this section are

```
$ cd [INSTALL_DIR]/ThesisSoftware/EventAnalysisApp
$ vi EA.xml (as required)
$ vi EUSIG_EventDef.xml (as required)
$ cd dist
$ java -jar EventAnalysis.jar -cf ../EA.xml
```

On a MacBook Pro, the EventAnalysis app takes about 1 sec/patient. The directory with the patient database files, the output directories for the JSON and log files are all controlled by the EA.xml configuration file.

A.3 Base Set Generator

This application processes the minute by minute data in the physiological table of a BrainIT patient database and the associated event and episode list produced by the Event Analysis Application (see A.2) The operation of the Base Set Generator is described in Section 4.2. This is by far the most computationally intensive part of the research and was run on the University of Glasgow School of Mathematics andd Statistics 24-core server “euclid-18”. This process involved transferring the required BrainIT patient SQLite DB files to a mirror of the predefined structure detailed in Figure A.1. The BSG was then run by the following commands.

```
$ cd [INSTALL_DIR]/ThesisSoftware/TopLevelCode
$ ./CalcAndCompress.sh <event horizon> <window size>
```

Line 2 is repeated for all six window sizes (5,10,15,20,25,30) for a given event horizon, resulting in six processes running in the background. This six process combination then takes just over 24 hours to run. Once the process is complete, all six compressed base data set files are transferred back to the MacBook Pro.

A.4 Training and Test Set Generator

The operation of the Training and Test Set Generator is described in Section 4.3. This application processes the information produced by the Base Data Set Generator application, A.3.

The commands to run this section are

```
1   $ cd [INSTALL_DIR]/ThesisSoftware/TopLevelCode
2   $ Rscript TTG_TopLevelControl.R EH=10
```

Line 2 is repeated for each event horizon. A run for a single event horizon takes about 5.5 hours on the MacBook Pro.

A.5 Logistic Regression Models

The commands to run this section are

```
1   $ cd [INSTALL_DIR]/ThesisSoftware/LRModels
2   $ ./BuildAllModels.sh
3   $ ./RunCompleteLassoModel.sh Full-Lasso-Model-V1-20120424.R Full-Lasso
4   $ ./BuildAllFeatureGrids.sh
5   $ ./BuildFourModelSummaryPlots.sh
6   $ cd [INSTALL_DIR]/RobDonaldThesis
7   $ ./CopyThesisSoftwareFiles.sh
```

Note that shell script BuildAllFeatureGrids.sh, line 4, contains a configuration parameter 'selectionThreshold', value = 0.6, which is used to control the level at which the **Bold Font** display is used within the table.

A.6 ICU Data Stream

This application is used to test models that have been constructed by the logistic regression process.

Testing models on the cohort of 30 unseen patients. The lines below are used to test the Full and Minimum models.

```
1   $ cd [INSTALL_DIR]/ThesisSoftware/TopLevelCode
2   $ Rscript ICUDataStream_TopLevelControl.R EH=10 WS=5 model.name=Full score.type=AUC
3   $ Rscript ICUDataStream_TopLevelControl.R EH=10 WS=5 model.name=Minimum score.type=AUC
```

To combine the files produced by the Phase 1 operations from the above commands (see Figure 7.3) to produce Figure 7.4.

```
1     $ cd [INSTALL_DIR]/ThesisSoftware/ICUDataStream/output/MN_Full_10_5
2     $ ls -1 *.csv > ICU-DSFullList.txt
3     $ cd [INSTALL_DIR]/ThesisSoftware/ICUDataStream
4     $ python CombineICUDataStreamResults.py -e 10 -w 5 -m Full
5     $ Rscript ICUDataStreamROC.R
```

The command on line 4 produces the file that is then processed into an ROC plot by the command on line 5.

A.7 Clinical Warning Protocol Processor

The ClinicalWarningProtocolProcessor.py application processes the ICU Data Stream output and produces an ROC summary for each warning threshold in the top level control file, runCWPP.sh file. The commands below produce the output files which contain the sensitivity and specificity results along with other summary statistics on the model performance. These output files are used by the command processCWPResults.sh which calls two R scripts to produce the plot in Figure B.1 and 7.7.

```
1     $ cd [INSTALL_DIR]/ThesisSoftware/ClinicalWarningProtocol
2     $ ./runCWPP.sh <model name>
3     $ ./processCWPResults.sh <model name>
4     $ python PrepareCWPPThresholdTables.py -t <warning threshold>  -m <model name>
5     $ cd ../../RobDonaldThesis/
6     $ ./CopyThesisSoftwareFiles.sh
```

The command on line 2 will take approximately 45 mins for a given model, whilst the commands on line 3,4 and 6 will each be less than a minute. For this thesis, the command on line 4 is currently using 0.15 for <warning threshold>.

A.8 Visual Checks

The commands to produce the visual check plots Figures 7.8, 7.9 and 7.10 are given below.

```
$ cd [INSTALL_DIR]/ThesisSoftware/LRModels
$ VisualCheckEpisodePrediction.R
$ python PrepareCWPPThresholdTables.py -t <warning threshold>  -m <model name>
$ cd ../../RobDonaldThesis/
$ ./CopyThesisSoftwareFiles.sh
```

A.9 Research machines

Most of the research was carried out on two laptops and one desktop machine. However, the scripts to produce the base data sets (see Sections 4.2 and A.3) were run on the University of Glasgow server euclid-18. This is a 24 core server and each one of the six calculations required to produce the results for a given event horizon was able to be run on a seperate core. The hardware details of the machines are given in Table A.1.

Machine	OS	CPU	Memory	Disk
MacBook Pro	OSX 10.6.8	Core 2 Duo 2.66 GHz	4 GB, DDR 3 1067 MHz	320 GB Hitachi
Dell XPS M1330	Ubuntu 10.04	Core 2 Duo T8300, 2.40GHz	4 GB, DDR 667 MHz	250 GB WDC
Dell Dimension 4700	Ubuntu 10.04	Pentium 4 3.00GHz	4 GB, SDRAM 533 MHz	500 GB WDC
HP Proliant DL360G5	CentOS release 6.2	(24core) X5650 2.67GHz	32 GB	Network Storage

Table A.1: Research machine specifications

A.10 R Support

Version 2.14 of the R statistical framework was used throughout this thesis, (R Development Core Team, 2008).

This section provides details of the R packages that have been used in the experiments that have been carried out to construct the thesis.

A.10.1 A cautionary tale

Checks carried out on a MacBook Pro with 64-bit R and two Ubuntu 10.04 linux machines with 32-bit R showed different output files. The files on the two linux machines were identical but they did not match the file produced on the MacBook Pro.

Debugging traced this to different behaviour of the lm() routine between the two systems when dealing with a zero slope i.e. a group of numbers all the same. In the case of the MacBook Pro, lm() returned an NA for the slope, in the 32-bit R on linux a very small number was returned e.g. 4.123 e-15. The lm() was contained within the method CalcSlope() which was called within a tryCatch block.

A.10.2 R Packages

R Package	Version	Description	Citation
glmnet	1.7.3	Lasso regression support	(Friedman et al., 2010)
lattice	0.20-0	Graphics support	(Sarkar, 2008)
lubridate	0.2.5	Time utilities	(Grolemund and Wickham, 2011)
MASS	7.3-16	Support utilities (write.matrix)	(Venables and Ripley, 2002)
RJSONIO	0.98-1	JSON support	(Lang, 2011)
ROCR	1.0-4	ROC curve support	(Sing et al., 2009)

Table A.2: R packages

A.10.3 Useful R commands

- List installed libraries — >library()
- Install a library — >install.packages(<library name>)
- Install a library (from outside R) — % R CMD INSTALL <library name>

Appendix B

Model Parameter Estimation Software

This appendix provides details of the R code used to obtain the parameter estimates for the six logistic regression models and two neural network models described in the thesis. The code is available for download and can be found by visiting the links

- `http://www.statsresearch.co.uk`
- `http://statsresearch.wordpress.com/about/`

The following sections provide ***very*** brief notes on the routines. The idea is to provide an entry point to the website material. This material will be updated as required.

B.1 Penalised logistic regression

B.1.1 MLE — PenalisedLogisticRegression.R

This script provides functions for building penalised logistic models from first principles. Parameter estimation is carried out by MLE using QR matrix decomposition.

B.1.2 SMC — SMC_LR.R

This script provides functions for building penalised logistic models from first principles. The parameter estimates are obtained using Sequential Monte Carlo techniques.

B.2 GLM logistic regression

B.2.1 Build_LRModels.R

This script is used to train and test three of the "Stats Based" logistic regression models. It uses the **glm** function from the R {stats} package.

B.3 GLMNET LASSO logistic regression

B.3.1 Test_Lasso_LRModels.R

This script is used to train and test the "Stats Based" logistic regression model "Full-Lasso". It uses the **cv.glmnet** function from the R {glmnet} package.

B.4 BANN using Sequential Monte Carlo

B.4.1 SMC_NeuralNet.R

Bayesian Artificial Neural Network (BANN) models are built from first principles using SMC_NeuralNet.R

B.5 Neural Network using NNET

B.5.1 NNetVariability.R

This script was written to build basic neural networks as described in Venables and Ripley (2002, §8.10). It uses the the **nnet** function from the R {nnet} package.

B.6 Model assessment using clinical data

This section provides the full table of results for the Minimum model followed by the summary and full tables for the Full model.

B.6.1 Full results table, Minimum model

study id	age	gender	trauma type	numEpi	sens	sens2	spec	ppv	numTP	numTN	numFP	numFN	statsCheck	wrnMean	wrnSD
PDB-16138373	45	Male	Fall	4	100.00	100.00	99.34	12.90	4	4085	186	0	0	27.25	4.86
PDB-72705050	57	Male	Work	2	0.00	0.00	99.82	0.00	0	6081	27	2	0	0.00	0.00
PDB-83705040	56	Female	Fall	19	0.00	0.00	99.98	0.00	0	5488	1	19	0	0.00	0.00
PDB-84884871	24	Male	Traffic Accident	33	90.62	87.88	95.03	14.87	29	3173	1405	3	1	27.41	4.09
PDB-84884946	26	Male	Fall	1	0.00	0.00	98.20	0.00	0	327	51	0	1	0.00	0.00
PDB-84884950	33	Male	Traffic Accident	9	100.00	100.00	96.65	3.46	9	7232	2199	0	0	26.22	6.14
PDB-84884951	64	Female	Fall	13	0.00	0.00	99.83	0.00	0	1813	5	13	0	0.00	0.00
PDB-84884952	36	Female	Sport	103	22.77	22.33	99.60	19.66	23	23335	172	78	2	18.61	4.61
PDB-84884953	17	Female	Pedestrian	24	20.83	20.83	99.71	18.52	5	7505	35	19	0	22.60	8.73
PDB-84884954	57	Male	Fall	62	66.13	66.13	98.35	17.75	41	11359	956	21	0	23.12	6.77
PDB-84884956	59	Male	Fall	27	70.37	70.37	98.97	9.22	19	17920	843	8	0	19.68	6.52
PDB-84884958	40	Male	Traffic Accident	40	77.78	70.00	97.07	27.72	28	2421	361	8	4	22.82	7.85
PDB-84884959	15	Male	Traffic Accident	57	76.36	73.68	97.40	15.16	42	8811	1448	13	2	23.29	6.13
PDB-84884960	44	Female	Traffic Accident	25	8.00	8.00	99.80	11.76	2	7546	27	23	0	23.50	2.12
PDB-84884961	41	Male	Traffic Accident	12	58.33	58.33	98.74	4.35	7	12111	776	5	0	26.14	2.97
PDB-84884962	32	Male	Traffic Accident	7	71.43	71.43	98.85	11.63	5	3271	210	2	0	22.40	11.72
PDB-84885004	18	Male	Sport	39	94.74	92.31	95.29	12.90	36	4912	1851	2	1	27.03	4.10
PDB-84885005	18	Male	Pedestrian	104	90.82	85.58	96.83	27.64	89	7114	1503	9	6	23.12	6.76
PDB-84885006	51	Female	Traffic Accident	11	0.00	0.00	99.80	0.00	0	26978	95	11	0	0.00	0.00
PDB-84885007	25	Female	Traffic Accident	3	33.33	33.33	99.67	2.78	1	10611	63	2	0	17.00	0.00
PDB-84885009	62	Male	Fall	15	53.33	53.33	99.47	11.27	8	11878	179	7	0	24.50	6.09
PDB-84885011	61	Female	Sport	41	7.32	7.32	99.84	12.00	3	13463	32	38	0	13.33	4.04
PDB-84885012	26	Male	Traffic Accident	7	83.33	71.43	99.49	4.85	5	19227	355	1	1	25.80	5.45
PDB-84885015	28	Female	Traffic Accident	34	27.27	26.47	99.69	14.75	9	16653	85	24	1	19.56	6.04
PDB-84885017	24	Male	Work	79	87.18	86.08	96.47	20.00	68	7436	1971	10	1	24.00	6.66
PDB-84885060	17	Male	Traffic Accident	16	93.33	87.50	98.07	6.39	14	10434	1500	1	1	23.14	7.73
PDB-84885062	42	Male	Traffic Accident	17	73.33	64.71	98.67	10.28	11	7149	632	4	2	25.36	5.84
PDB-84885067	68	Male	Traffic Accident	11	45.45	45.45	99.53	3.14	5	32772	518	6	0	20.40	4.56
PDB-84885068	42	Male	Work	5	60.00	60.00	99.63	3.90	3	19983	289	2	0	18.33	8.74
PDB-84885073	19	Male	Fall	44	88.37	86.36	98.03	12.75	38	12967	1972	5	1	26.53	5.43

Table B.1: Model assessment full results, Model: Minimum, 0.15 warning threshold — sens = sensitivity; spec = specificity; wrn = warning given in minutes. The column statsCheck = $numEpi - (numTP + numFN)$, and should be zero. A positive number indicates the number of episodes missed due to incomplete data.

B.6.2 Summary results table, Full model

study id	age	gender	trauma type	episodes	sens	sens2	spec	avg wrn
PDB-16138373	45	Male	Fall	4	100.00	100.00	99.29	26.25
PDB-72705050	57	Male	Work	2	50.00	50.00	99.20	30.00
PDB-83705040	56	Female	Fall	19	47.06	42.11	99.25	23.75
PDB-84884871	24	Male	Traffic Accident	33	87.50	84.85	96.28	26.54
PDB-84884946	26	Male	Fall	1	0.00	0.00	98.28	0.00
PDB-84884950	33	Male	Traffic Accident	9	88.89	88.89	97.10	24.25
PDB-84884951	64	Female	Fall	13	33.33	30.77	99.05	21.50
PDB-84884952	36	Female	Sport	103	49.48	46.60	99.19	20.29
PDB-84884953	17	Female	Pedestrian	24	37.50	37.50	99.41	20.22
PDB-84884954	57	Male	Fall	62	80.33	79.03	97.22	23.92
PDB-84884956	59	Male	Fall	27	74.07	74.07	99.09	22.20
PDB-84884958	40	Male	Traffic Accident	40	97.22	87.50	92.48	24.06
PDB-84884959	15	Male	Traffic Accident	57	76.36	73.68	97.74	21.86
PDB-84884960	44	Female	Traffic Accident	25	41.67	40.00	99.69	17.60
PDB-84884961	41	Male	Traffic Accident	12	66.67	66.67	98.63	27.00
PDB-84884962	32	Male	Traffic Accident	7	100.00	100.00	97.14	24.71
PDB-84885004	18	Male	Sport	39	86.84	84.62	97.30	25.09
PDB-84885005	18	Male	Pedestrian	104	82.83	78.85	97.43	22.16
PDB-84885006	51	Female	Traffic Accident	11	18.18	18.18	99.82	23.50
PDB-84885007	25	Female	Traffic Accident	3	0.00	0.00	99.86	0.00
PDB-84885009	62	Male	Fall	15	57.14	53.33	98.82	26.38
PDB-84885011	61	Female	Sport	41	50.00	48.78	99.14	20.70
PDB-84885012	26	Male	Traffic Accident	7	83.33	71.43	99.49	26.80
PDB-84885015	28	Female	Traffic Accident	34	53.12	50.00	99.42	19.18
PDB-84885017	24	Male	Work	79	92.31	91.14	96.59	23.94
PDB-84885060	17	Male	Traffic Accident	16	80.00	75.00	98.70	23.58
PDB-84885062	42	Male	Traffic Accident	17	93.33	82.35	98.43	23.93
PDB-84885067	68	Male	Traffic Accident	11	50.00	45.45	99.48	24.60
PDB-84885068	42	Male	Work	5	60.00	60.00	99.71	18.67
PDB-84885073	19	Male	Fall	44	86.05	84.09	98.12	24.49

Table B.2: Model assessment summary results, Model: Full, 0.15 warning threshold — sens = sensitivity; spec = specificity; avg wrn = average warning given in minutes

Table B.2 above shows the results for the 30 patients for the 0.15 decision threshold and this shows that, for patients that are picked up by the system, the average warning is 23 mins (sd = 3). There is some warning given for 28 of the 30 patients (93%). A more detailed set of results for the 0.15 decision threshold is provided in Appendix B.3.

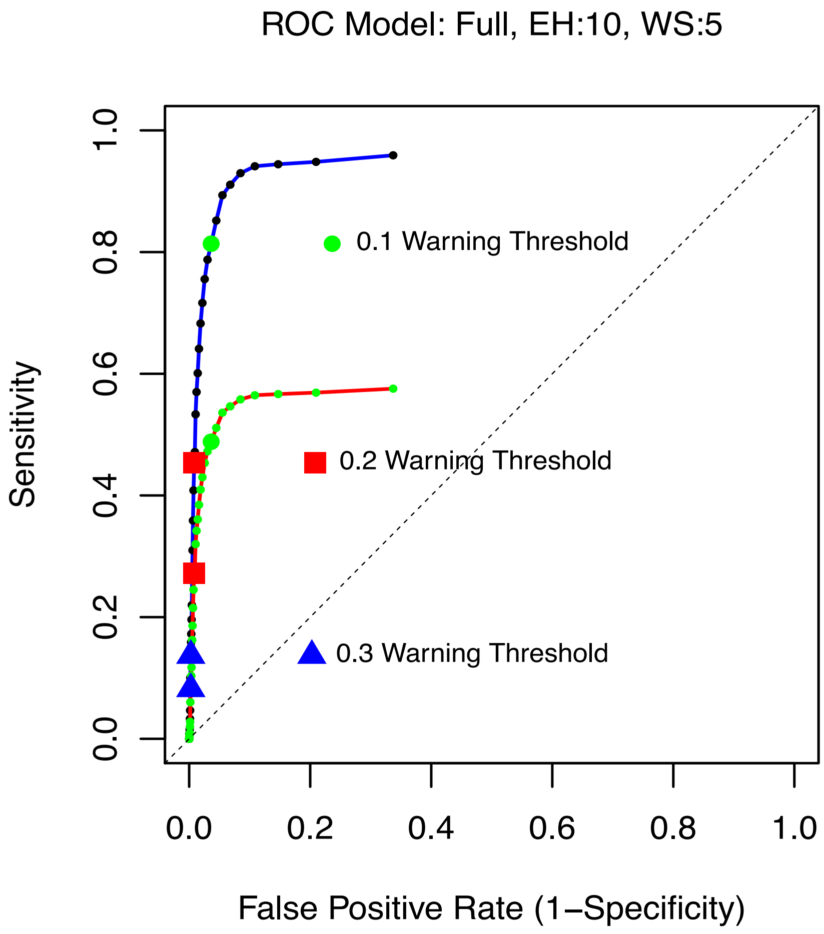


Figure B.1: Model: Full, ROC using Clinical Warning Protocol on ICU data stream

Figure B.1 shows a pseudo ROC curve which is constructed from the results produced by the CWPP running the Full model. This shows that with a decision threshold of between 0.15 and 0.2, the system would detect 30 to 50% of episodes.

B.6.3 Full results table, Full model

study id	age	gender	trauma type	numEpi	sens	sens2	spec	ppv	numTP	numTN	numFP	numFN	statsCheck	wrnMean	wrnSD
PDB-16138373	45	Male	Fall	4	100.00	100.00	99.29	12.12	4	4051	199	0	0	26.25	4.99
PDB-72705050	57	Male	Work	2	50.00	50.00	99.20	2.13	1	5700	145	1	0	30.00	0.00
PDB-83705040	56	Female	Fall	19	47.06	42.11	99.25	17.39	8	4997	123	9	2	23.75	5.70
PDB-84884871	24	Male	Traffic Accident	33	87.50	84.85	96.28	16.57	28	3646	1200	4	1	26.54	5.01
PDB-84884946	26	Male	Fall	1	0.00	0.00	98.28	0.00	0	343	44	0	1	0.00	0.00
PDB-84884950	33	Male	Traffic Accident	9	88.89	88.89	97.10	3.16	8	8203	1882	1	0	24.25	7.69
PDB-84884951	64	Female	Fall	13	33.33	30.77	99.05	21.05	4	1565	54	8	1	21.50	6.61
PDB-84884952	36	Female	Sport	103	49.48	46.60	99.19	21.24	48	21799	422	49	6	20.29	6.40
PDB-84884953	17	Female	Pedestrian	24	37.50	37.50	99.41	17.31	9	7258	79	15	0	20.22	7.81
PDB-84884954	57	Male	Fall	62	80.33	79.03	97.22	15.46	49	9389	1495	12	1	23.92	6.26
PDB-84884956	59	Male	Fall	27	74.07	74.07	99.09	10.81	20	17991	712	7	0	22.20	6.27
PDB-84884958	40	Male	Traffic Accident	40	97.22	87.50	92.48	25.93	35	1230	689	1	4	24.06	7.25
PDB-84884959	15	Male	Traffic Accident	57	76.36	73.68	97.74	15.73	42	9750	1111	13	2	21.86	6.35
PDB-84884960	44	Female	Traffic Accident	25	41.67	40.00	99.69	30.30	10	7376	30	14	1	17.60	5.32
PDB-84884961	41	Male	Traffic Accident	12	66.67	66.67	98.63	4.65	8	11819	816	4	0	27.00	5.04
PDB-84884962	32	Male	Traffic Accident	7	100.00	100.00	97.14	8.43	7	2582	423	0	0	24.71	10.09
PDB-84885004	18	Male	Sport	39	86.84	84.62	97.30	14.54	33	6982	1092	5	1	25.09	6.02
PDB-84885005	18	Male	Pedestrian	104	82.83	78.85	97.43	26.97	82	8412	1076	17	5	22.16	6.75
PDB-84885006	51	Female	Traffic Accident	11	18.18	18.18	99.82	3.92	2	26701	78	9	0	23.50	4.95
PDB-84885007	25	Female	Traffic Accident	3	0.00	0.00	99.86	0.00	0	10624	24	3	0	0.00	0.00
PDB-84885009	62	Male	Fall	15	57.14	53.33	98.82	5.97	8	10589	554	6	1	26.38	3.89
PDB-84885011	61	Female	Sport	41	50.00	48.78	99.14	15.87	20	12267	275	20	1	20.70	6.59
PDB-84885012	26	Male	Traffic Accident	7	83.33	71.43	99.49	4.90	5	18998	346	1	1	26.80	3.11
PDB-84885015	28	Female	Traffic Accident	34	53.12	50.00	99.42	15.32	17	16036	218	15	2	19.18	7.21
PDB-84885017	24	Male	Work	79	92.31	91.14	96.59	20.93	72	7707	1809	6	1	23.94	6.69
PDB-84885060	17	Male	Traffic Accident	16	80.00	75.00	98.70	6.70	12	12726	721	3	1	23.58	6.68
PDB-84885062	42	Male	Traffic Accident	17	93.33	82.35	98.43	11.38	14	6837	751	1	2	23.93	7.11
PDB-84885067	68	Male	Traffic Accident	11	50.00	45.45	99.48	2.86	5	32737	477	5	1	24.60	4.45
PDB-84885068	42	Male	Work	5	60.00	60.00	99.71	4.84	3	20275	162	2	0	18.67	10.26
PDB-84885073	19	Male	Fall	44	86.05	84.09	98.12	12.33	37	13739	1672	6	1	24.49	6.45

Table B.3: Model assessment full results, Model: Full, 0.15 warning threshold — sens = sensitivity; spec = specificity; wrn = warning given in minutes. The column statsCheck = $numEpi - (numTP + numFN)$, and should be zero. A positive number indicates the number of episodes missed due to incomplete data.

Bibliography

American College Of Surgeons. Advanced trauma life support. 2011. URL http://www.facs.org/trauma/atls/program.html. Accessed On: 2012.01.07.

Patrik Andlin-Sobocki, Bengt Jönsson, Hans-Ulrich Wittchen, and Jes Olesen. Costs of disorders of the brain in europe. *EUROPEAN JOURNAL OF NEUROLOGY*, 12(Supplement 1):1–104, 2005.

AvertIT. Avert-it project (FP7-217049-AVERT-IT). 2008. URL http://www.avert-it.org/. Accessed On: 2010.03.02.

Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of markov processe. *Inequalities*, 3:1 – 8, 1972.

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):pp. 164–171, 1970. ISSN 00034851. URL http://www.jstor.org/stable/2239727.

Thomas Bayes and Richard Price. An essay towards solving a problem in the doctrine of chance. by the late rev. mr. bayes, communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions of the Royal Society of London*, 53: 269 – 271, 1763. doi: doi:10.1098/rstl.1763.0044.

Joseph Berkson. Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, 39(227):pp. 357–365, 1944. ISSN 01621459. URL http://www.jstor.org/stable/2280041.

Cherisse Berry, Eric J. Ley, Areti Tillou, Gil Cryer, Daniel R. Margulies, and Ali Salim. The effect of gender on patients with moderate to severe head injuries. *The Journal of Trauma: Injury, Infection, and Critical Care*, 67(5):950–953, 2009. doi: 10.1097/TA.0b013e3181ba3354.

Cherisse Berry, Eric J. Ley, Marko Bukur, Darren Malinoski, Daniel R. Margulies, James Mirocha, and Ali Salim. Redefining hypotension in traumatic brain injury. *Injury*, 1(0):1, 2011. ISSN 0020-1383. doi: 10.1016/j.injury.2011.08.014. URL `http://www.sciencedirect.com/science/article/pii/S0020138311003937`.

Jilles B. Bijker, Wilton A. van Klei, Yvonne Vergouwe, Douglas J. Eleveld, Leo van Wolfswinkel, Karel G. M. Moons, and Cor J. Kalkman. Intraoperative hypotension and 1-year mortality after noncardiac surgery. *Anesthesiology*, 111 (6):1217–1226, 2009. doi: 10.1097/ALN.0b013e3181c14930.

Christoper M. Bishop. *Paterrn Recognition and Machine Learning*. Springer, 2006.

Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

J Martin Bland and Douglas G Altman. The odds ratio. *BMJ*, 320(7247:1468, 2000.

G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):pp. 211–252, 1964. ISSN 00359246. URL `http://www.jstor.org/stable/2984418`.

George E.P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Forecasting and Control (Third Edition)*. Prentice Hall, 1994.

Brain Trauma Foundation. Guidelines for the management of severe traumatic brain injury 3rd edition. *Journal of Neurotrauma*, 1:1 – 117, 2007. URL `https://www.braintrauma.org/pdf/protected/Guidelines_Management_2007w_bookmarks.pdf`.

BrainIT Consortium. BrainIT. 2007. URL `http://www.brainit.org/`. Accessed On: 2007.12.22.

Centers for Disease Control and Prevention. Get the stats on traumatic brain injury in the united states. 2011. URL `http://www.cdc.gov/traumaticbraininjury/pdf/BlueBook_factsheet-a.pdf`. Accessed On: 2012.06.23.

Centers for Disease Control and Prevention. Injury prevention & control: Traumatic brain injury: Severe traumatic brain injury. 2012. URL `http://www.cdc.gov/TraumaticBrainInjury/severe.html`. Accessed On: 2012.07.28.

Marie-Christine Chambrin. Alarms in the intensive care unit: how can the number of false alarms be reduced? *Critical Care August 2001 Vol 5 No 4 Chambrin*, 5 (4): 184–188, 2001.

Chris Chatfield. *The Analysis of Time Series An Introduction (6th Ed)*. Chapman and Hall, 2004.

Randall M. Chesnut, Lawrence F. Marshall, Melville R. Klauber, Barbara A. Blunt, Nevan Baldwin, Howard M. Eisenberg, John A. Jane, Anthony Marmarou, and Mary A. Foulkes. The role of secondary brain injury in determining outcome from severe head injury. *The Journal of Trauma*, 34 (2):216–222, 1993.

Nicolas Chopin. A sequential particle filter method for static models. *Biometrika*, 89 (3):539–552, 2002. doi: 10.1093/biomet/89.3.539.

James C. Christensen, Justin R. Estepp, Glenn F. Wilson, and Christopher A. Russell. The effects of day-to-day variability of physiological data on operator functional state classification. *NeuroImage*, 59(1):57 – 63, 2012. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2011.07.091. URL `http://www.sciencedirect.com/science/article/pii/S105381191100886X`. <ce:title>Neuroergonomics: The human brain in action and at work</ce:title>.

D.J. Christini, F.H. Bennett, K.R. Lutchen, H.M. Ahmed, J.M. Hausdorff, and N. Oriol. Application of linear and nonlinear time series modeling to heart rate dynamics analysis. *Biomedical Engineering, IEEE Transactions on*, 42(4):411 –415, april 1995. ISSN 0018-9294. doi: 10.1109/10.376135.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, 1995. doi: 10.1007/BF00994018.

Huw Talfryn Oakley Davies, Iain Kinloch Crombie, and Manouche Tavakoli. When can odds ratios mislead? *BMJ*, 316(7136):989–991, 3 1998. doi: 10.1136/bmj.316.7136.989.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39 (1):1–38, 1977. URL `http://www.jstor.org/stable/2984875`.

Rob Donald. Event definition analysis, preliminary results. Technical report, Avert-IT Project, 2008.

Rob Donald, Tim Howells, Ian Piper, I. Chambers, G. Citerio, P. Enblad, B. Gregson, K. Kiening, J. Mattern, P. Nilsson, A. Ragauskas, Juan Sahuquillo, R. Sinnott, and

A. Stell. Trigger characteristics of EUSIG-defined hypotensive events. In Martin U. Schuhmann and Marek Czosnyka, editors, *Intracranial Pressure and Brain Monitoring XIV*, volume 114 of *Acta Neurochirurgica Supplementum*, pages 45–49. Springer Vienna, 2012a. ISBN 978-3-7091-0956-4. URL `http://dx.doi.org/10.1007/978-3-7091-0956-4\_9`. 10.1007/978-3-7091-0956-4_9.

Rob Donald, Tim Howells, Ian Piper, I. Chambers, G. Citerio, P. Enblad, B. Gregson, K. Kiening, J. Mattern, P. Nilsson, A. Ragauskas, Juan Sahuquillo, R. Sinnott, and A. Stell. Early warning of EUSIG-defined hypotensive events using a bayesian artificial neural network. In Martin U. Schuhmann and Marek Czosnyka, editors, *Intracranial Pressure and Brain Monitoring XIV*, volume 114 of *Acta Neurochirurgica Supplementum*, pages 39–44. Springer Vienna, 2012b. ISBN 978-3-7091-0956-4. URL `http://dx.doi.org/10.1007/978-3-7091-0956-4\_8`. 10.1007/978-3-7091-0956-4_8.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197 – 208, 2000. URL `http://www.springerlink.com/content/q6452k2x37357l3r/fulltext.pdf`.

Kristin Elf. *Secondary Insults in Neurointensive Care of Patients with Traumatic Brain Injury*. PhD thesis, Uppsala University, 2005.

Y. Fan, D.S. Leslie, and M.P. Wand. Generalised linear mixed model analysis via sequential monte carlo sampling. *Electronic Journal of Statistics*, 2:916 – 938, 2008. doi: 10.1214/07-EJS158.

Julian Faraway and Chris Chatfield. Time series forecasting with neural networks: a comparitive study using the airline data. *Journal of Applied Statistics*, 47(2):231 – 250, 1998.

Tom Fawcett. ROC graphs: Notes and practical considerations for researchers. 2004. URL `http://home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf`. Accessed On: 2009.03.04.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874, 2006. URL `http://www.csee.usf.edu/~candamo/site/papers/ROCintro.pdf`.

Silvia Figini and Mario Maggi. Performance of credit risk prediction models via proper loss functions. Technical report, Universita di Pavia, Department of Economics and Management, 2014.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1): 1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

Stuart Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741, Nov 1984. ISSN 0162-8828. doi: 10.1109/TPAMI.1984.4767596.

Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *Journal of Pattern Recognition and Artificial Intelligence*, 15 (1):1 – 25, 2001.

Matthias Gorges, Boaz A. Markewitz, and Dwayne R. Westenskow. Improving alarm performance in the medical intensive care unit using delays and clinical context. *Anesthesia and Analgesia*, 108 (5):1546 – 1552, 2009.

Garrett Grolemund and Hadley Wickham. Dates and times made easy with lubridate. *Journal of Statistical Software*, 40(3):1–25, 2011. URL `http://www.jstatsoft.org/v40/i03/`.

David Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine Learning*, 77:103–123, 2009. ISSN 0885-6125. URL `http://dx.doi.org/10.1007/s10994-009-5119-5`. 10.1007/s10994-009-5119-5.

David J Hand. Assessing the performance of classification methods. *International Statistical Review*, 80 (3):400 – 414, 2012. doi: 10.1111/j.1751-5823.2012.00183.x.

A.J. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143 (1):29–36, 1982.

James A. Hanley and Barbara J. McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148 (3):839 – 843, 1983.

Frank E. Harrell. *Regression Modeling Strategies, With Applications to Linear Models, Logistic Regression, and Survival Analysis.* Springer, 2001.

Trevor Hastie, Robert Tibshirani, and Jerome Freidman. *The Elements of Statistical Learning, 2nd Edition*. Springer, 2009. URL `http://www-stat.stanford.edu/~tibs/ElemStatLearn/download.html`.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97 – 109, 1970. doi: 10.1093/biomet/57.1.97. URL `http://biomet.oxfordjournals.org/content/57/1/97.abstract`.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):pp. 55–67, 1970. ISSN 00401706. URL `http://www.jstor.org/stable/1267351`.

Jeroen D. Hol, Thomas B. Schön, and Fredrik Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop*, 2006.

Michael Imhoff and Silvia Kuhls. Alarm algorithms in critical care monitoring. *Anesthesia and Analgesia*, 102:1525 – 37, 2006.

B. Jennett and M. Bond. Jennett b, bond m. assessment of outcome after severe brain damage. a practical scale. lancet 1975;l:48u4. *Lancet*, 1:480 – 484, 1975.

Adam M. Johansen and Ludger Evers. Monte carlo methods. 2007. URL `http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic/johansen/teaching/MCM-2007.pdf`. Accessed On: 2009.11.16.

Patricia Jones, Peter Andrews, Susan Midgley, Shirley Anderson, Ian Piper, Janis Tocher, Alma Housley, Jane Corrie, James Slattery, Mark Dearden, and Douglas Miller. Measuring the burden of secondary insults in head-injured patients during intensive care. *Journal of Neurosurgical Anesthesiology*, 6:4–14, 1994.

F Jousset, M Lemay, and JM Vesin. Computers in cardiology / physionet challenge 2009: Predicting acute hypotensive episodes. *Computers in Cardiology*, 36:637–640, 2009.

JSON. Javascript object notation. 2011. URL `http://www.json.org/`. Accessed On: 2011.11.10.

Richard E. Klabunde. Cardiovascular physiology concepts. 2005. URL `http://www.cvphysiology.com/index.html`. Accessed On: 2008.03.05.

Nynke C. Krol. Penalized logistic regression: a quadratic difference penalty. Master's thesis, Universiteit Leiden, Mathematics, Specialization: Statistical Science, 2013. URL `http://www.math.leidenuniv.nl/scripties/MasterKrol.pdf`.

Nan M. Laird and James H. Ware. Random-effects models for longitudinal data. *Biometrics*, 38(4):pp. 963–974, 1982. ISSN 0006341X. URL `http://www.jstor.org/stable/2529876`.

Duncan Temple Lang. *RJSONIO: Serialize R objects to JSON, JavaScript Object Notation*, 2011. URL `http://CRAN.R-project.org/package=RJSONIO`. R package version 0.96-0.

D. Lunn, D. Spiegelhalter, A. Thomas, critique Best, N. (2009) The BUGS project: Evolution, and Statistics in Medicine 28: 3049-3082. future directions (with discussion). The bugs project: Evolution, critique and future directions (with discussion). *Statistics in Medicine*, 28:3049–3082, 2009. URL `http://www.openbugs.net/w/FrontPage`.

D.J. Lunn, A. Thomas, N. Best, and D. Spiegelhalter. Winbugs – a bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing*, 10:325–337, 2000. URL `http://www.mrc-bsu.cam.ac.uk/bugs/`.

IanN. Maciver, LauranceP. Lassman, CharlesW. Thomson, and Ian Mcleod. Treatment of severe head injuries. *The Lancet*, 272(7046):544 – 550, 1958. ISSN 0140-6736. doi: 10.1016/S0140-6736(58)90194-6. URL `http://www.sciencedirect.com/science/article/pii/S0140673658901946`. <ce:title>Originally published as Volume 2, Issue 7046</ce:title>.

W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology,*, 5(4):115 – 133, 1943.

Nicholas Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):pp. 335–341, 1949. ISSN 01621459. URL `http://www.jstor.org/stable/2280232`.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: http:

//dx.doi.org/10.1063/1.1699114. URL `http://scitation.aip.org/content/aip/journal/jcp/21/6/10.1063/1.1699114`.

M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press., 1969.

Patrick Mitchell, Barbara A Gregson, Ian Piper, Giuseppe Citerio, A David Mendelow, and Iain R Chambers. Blood pressure in head injured patients. *Journal of Neurology Neurosurgery and Psychiatry*, 78:399–402, 2007. doi: 10.1136/jnnp.2006.100172. URL `http://jnnp.bmj.com/cgi/reprint/78/4/399`.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *J. R. Statist. Soc. B (2006) 68, Part 3, pp. 411 - 36*, 68 (3):411 – 436, 2006.

Marc Moss, D. Andrew Wellman, and George A. Cotsonis. An appraisal of multivariable logistic models in the pulmonary and critical care literature*. *CHEST Journal*, 123 (3):923–928, 2003. doi: 10.1378/chest.123.3.923. URL `http://dx.doi.org/10.1378/chest.123.3.923`.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer, Lecture Notes in Statistics 118, 1996.

J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):pp. 370–384, 1972. ISSN 00359238. URL `http://www.jstor.org/stable/2344614`.

Jurgens Nortje and David K. Menon. Traumatic brain injury: physiology, mechanisms, and outcome. *Current Opinion in Neurology*, 17:711–718, 2004.

Percival H Pangilinan. Classification and complications of traumatic brain injury. 2012. URL `http://emedicine.medscape.com/article/326643-overview#aw2aab6b3`. Accessed On: 2012.06.23.

Physio Net. Physionet/computers in cardiology challenge 2009: Predicting acute hypotensive episodes. 2009. URL `http://www.physionet.org/challenge/2009/`. Accessed On: 2010.03.07.

Jose C. Pinheiro and Douglas M. Bates. *Mixed-Effects Models in S and S-Plus*. Springer, 2000. ISBN 0-387-98957-0.

Ian Piper, Iain Chambers, Giuseppe Citerio, Per Enblad, Barbara Gregson, Tim Howells, Karl Kiening, Julia Mattern, Pelle Nilsson, Arminas Ragauskas, Juan Sahuquillo, Rob Donald, Richard Sinnott, and Anthony Stell. The brain monitoring with information technology (BrainIT) collaborative network: EC feasibility study results and future direction. *Acta Neurochirurgica*, 152 (11):1859–1871, 2010. ISSN 0001-6268. doi: DOI10.1007/s00701-010-0719-1. URL `http://dx.doi.org/10.1007/s00701-010-0719-1`. 10.1007/s00701-010-0719-1.

Martyn Plummer. Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, 2003. URL `http://mcmc-jags.sourceforge.net/`.

John Quinn. *Bayesian Condition Monitoring in Neonatal Intensive Care*. PhD thesis, Institute for Adaptive and Neural Computation, School of Informatics, University of Edinburgh, 2007. URL `http://omnipresence.org/jq/papers/jqthesis.pdf`.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77 (2):257 – 286, 1989.

L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, January:4 – 16, 1986.

G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk metropolis algorithms. *The Annals of Applied Probability*, 7(1):pp. 110–120, 1997. ISSN 10505164. URL `http://www.jstor.org/stable/2245134`.

J Rose, S Valtonen, and B Jennett. Avoidable factors contributing to death after head injury. *British Medical Journal*, 2:615–618, 1977.

F Rosenblatt. *Principles of Neurodynamics*. Spartan Books, New York., 1962.

Royal Society Edinburgh. James douglas miller - obituary. 1995. URL `http://www.royalsoced.org.uk/cms/files/fellows/obits_alpha/miller_james.pdf`. Accessed On: 2012.02.04.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323 (6088):533 – 536, 1986. doi: 10.1038/323533a0.

Deepayan Sarkar. *Lattice: Multivariate Data Visualization with R*. Springer, New York, 2008. URL `http://lmdvr.r-forge.r-project.org`. ISBN 978-0-387-75968-5.

Robert H Shumway and David S Stoffer. *Time Series Analysis and Its Applications with R Examples (Second Edition)*. Springer, 2006.

Tobias Sing, Oliver Sander, Niko Beerenwinkel, and Thomas Lengauer. *ROCR: Visualizing the performance of scoring classifiers.*, 2009. URL `http://CRAN.R-project.org/package=ROCR`. R package version 1.0-4.

A. Singh, T. Tamminedi, G. Yosiphon, A. Ganguli, and J. Yadegar. Hidden markov models for modeling blood pressure data to predict acute hypotension. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 550 –553, march 2010. doi: 10.1109/ICASSP.2010.5495603.

R. Smiley. Fast fourier transforms as prophecy: predicting hypotension during spinal anesthesia. *Anesthesiology*, 102 (6):1079–1080, 2005.

A. Stell, R.O. Sinnott, R. Donald, I. Chambers, G. Citerio, P. Enblad, B. Gregson, T. Howells, K. Kiening, P. Nilsson, A. Ragauskas, J. Sahuquillo, and I. Piper. A distributed clinical data platform for physiological studies in the brain trauma domain. *Sixth IEEE International Conference on eScience*, 1:65 – 72, 2010. doi: 10.1109/eScience.2010.26. URL `http://dx.doi.org/10.1109/eScience.2010.26`.

Anthony Stell, Richard Sinnott, Jipu Jiang, Rob Donald, Iain Chambers, Giuseppe Citerio, Per Enblad, Barbara Gregson, Tim Howells, Karl Kiening, Pelle Nilsson, Arminas Ragauskas, Juan Sahuquillo, and Ian Piper. Federating distributed clinical data for the prediction of adverse hypotensive events. *Philosophical Transactions of the Royal Society A: Mathematical,Physical and Engineering Sciences*, 367(1898):2679–2690, 2009. doi: 10.1098/rsta.2009.0042. URL `http://rsta.royalsocietypublishing.org/content/367/1898/2679.abstract`.

Anthony Stell, Richard Sinnott, Rob Donald, and Ian Piper. Supporting clinical trials to predict adverse events in the brain trauma domain. In *The 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2012)*, 2012.

Ewout W Steyerberg. *Clinical Prediction Models, A Practical Approach to Development, Validation, and Updating*. Springer, 2010. doi: 10.1007/978-0-387-77244-8_17.

tbi impact.org. Glasgow outcome scale. 2013. URL `http://www.tbi-impact.org/cde/mod_templates/12_F_01_GOSE.pdf`. Accessed On: 2013.08.24.

G.M. Teasdale, L.E. Pettigrew, J.T. Wilson, G. Murray, and B. Jennett. Analyzing outcome of treatment of severe head injury: A review and update on advancing the use of the glasgow outcome scale. journal of neurotrauma. *Journal of Neurotrauma*, 15: 587–597, 1998.

Graham Teasdale and Bryan Jennett. Assessment of coma and imparied consiousness. *The Lancet*, 304:81–84, 1974.

The Apache Software Foundation. Hadoop project. 2011. URL `http://hadoop.apache.org/`. Accessed On: 2012.06.26.

The SQLite Consortium. SQLite. 2012. URL `http://www.sqlite.org/`. Accessed On: 2012.07.06.

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

Luke Tierney, A. J. Rossini, Na Li, and H. Sevcikova. R package snow (simple network of workstations). 2012. URL `http://cran.r-project.org/web/packages/snow/index.html`. Accessed On: 2012.06.26.

Ralf Vandenhouten, Manfred Lambertz, Peter Langhorst, and Reinhard Grebe. Nonstationary time-series analysis applied to investigation of brainstem system dynamics. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 47:729–737, 2000.

Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982. ISBN 0387907335.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL `http://www.stats.ox.ac.uk/pub/MASS4`. ISBN 0-387-95457-0.

Kiri L. Wagstaff. Machine learning that matters. In *Proceedings of the 29 th International Conference on Machine Learning, Edinburgh, Scotland, UK.*, 2012.

P. J. Werbos. *Beyond Regression, New Tools for Prediction and Analysis in the Behavioral Sciences.* PhD thesis, Harvard University, 1974.

B. Widrow and M. E. Hoff. Adaptive switching circuits. In *WESCON*, pages 96 – 104, 1960.

Wikipedia. Wiggers diagram. 2012. URL `http://en.wikipedia.org/wiki/File:Wiggers_Diagram.svg`. Accessed On: 2014.01.14.

Christopher K.I. Williams, John Quinn, and Neil McIntosh. Factorial switching kalman filters for condition monitoring in neonatal intensive care. *NIPS 2005 Online Papers*, 18:1, 2005. URL `http://books.nips.cc/papers/files/nips18/NIPS2005_0596.pdf`.

J.T. Wilson, L.E. Pettigrew, and G.M. Teasdale. Structured interviews for the glasgow outcome scale and the extended glasgow outcome scale: Guide lines for their use. *Journal of Neurotrauma 15(8): 573-85. 1997.*, 15(8):573–85, 1997.

J.T. Wilson, F.J. Slieker, V. Legrand, G. Murray, N. Stocchetti, and A.I. Maas. Observer variation in the assessment of outcome in traumatic brain injury: experience from a multicenter, international randomized clinical trial. *Neurosurgery*, 61(1):123–8, 2007.

Nina Zumel. The simpler derivation of logistic regression. 2011. URL `http://www.win-vector.com/blog/2011/09/the-simpler-derivation-of-logistic-regression/`. Accessed On: 2014.01.18.