Automatic Generalised Seizure Detection in Clinical Electroencephalography Records

David Luke Elliott, M.Sc.

A thesis presented for the degree of

Doctor of Philosophy

Mathematics and Statistics

Lancaster University

United Kingdom

May 2021

Abstract

This thesis examines the application of machine learning pipelines for automatic generalised seizure detection. We begin by introducing the potential pipeline components of a signal classification system (Pre-processing, Feature Engineering, Dimensionality Reduction, and Classification), and review the literature associated with each stage. In the subsequent research chapters, Bayesian optimisation is used to systematically optimise many pipeline/model configurations and hyperparameters to provide practical guidance and inform future pipeline development. There is a focus on ecological validity, therefore we use real-world "raw" patient records collected as part of routine care from multiple healthcare institutions. In chapter 3, using a large feature set and feature reduction techniques, we were able to identify components of EEG records useful for identifying absence epilepsy seizures for pipelines with "classical" classifiers. These pipelines had good overall performance, at the expense of a high false positive rate (FPR); with the best binary classifiers never missing a seizure and accurately marking the full duration of most seizures. As class imbalances were a challenge for effective model training, chapter 4 examined pipelines with balanced ensemble classifiers. Compared to chapter 3, boosted ensembles were faster, with a lower FPR and high precision/specificity. However, typically the full seizure was not marked, meaning they may be more useful for accessing the number of seizures in a record rather than their length. Subsequently, chapter 5 examined the performance of boosted ensembles and deep learning architectures on two different types of generalised seizure. Consistent with human raters, models trained to detect absence seizures, a seizure type with little intra-patient and inter-patient variability, had better performance across all investigated metrics compared to non-specific seizures, which have large intra-patient and inter-patient variabilities. Compared to deep learning models, boosted ensembles provided the best overall performance, were more computationally-efficient, and would be easier to implement into healthcare practice. To our knowledge, this thesis provides the first use of optimal hyperparameters and pipeline components found through Bayesian optimisation methods for absence epilepsy detection. In the future, such pipelines could reduce the current bottleneck of clinical time

required to manually mark EEG records by providing a preliminary marked record to a physiologist.

Acknowledgements

This thesis could never have been realised without the support of so many talented and wonderful people I have been lucky enough to work with. You have all instilled a drive to work collaboratively so that I could achieve a much greater impact from our collective work than I could do alone.

Of course my academic supervisors, Prof. Vincent Reid and Dr. Rebecca Killick, deserve foremost mention for the immeasurable new skills and knowledge I have gained. Their flexibility in helping form this thesis, which included switching to a new PhD degree submission all together, literally is crucial to the very existence of this work... not to mention the thesis would be another 300 pages long without their excellent editing advice! There are of course many other incredible academics and staff at Lancaster University who I have been lucky enough to call my colleges. Without Dr. Abe Karnik or Dr. Judith Lunn giving me the opportunity to write grants with them, I would in no way have had the breath of experience afforded to me. This enabled me to work with other talented students (Kristoffer Geyer and Nathan Rutherford) and spend more time building portable EEG's with Barrie Usherwood and Dr. Peter Tovee. I was often found in the workshop, fidgeting with my designated stress-ball ensuring I didn't touch anything. You both went above and beyond your roles, and responsible for some of the most interesting and exciting moments in my PhD. There are also all the people I got to share an office with over the years, including Ellie Smith who I was lucky enough to work on projects with all the way since my Undergraduate degree, and the rest of the staff in the Psychology department - where an immeasurable number have served as colleagues, mentors, and friends.

This thesis would also not exist without the hard work of another incredible organisation

and the staff who dedicate themselves to the health of others, the NHS. The NHS has so many priorities, their staff constantly busy improving the lives of their communities, so I find it amazing how many people were willing to dedicate their time outside of their already busy schedule to aid my research; despite the difficulties. From early on, this work was championed by Dr. Christian DeGoede, who along with his important work as a Consultant Paediatric Neurologist, is an excellent advocate for health research and was a real joy to work with. There was of course many others at the Royal Preston and Royal Blackburn Hospitals who went above and beyond to help this work become a reality; including Dr. Rosemary Belderbos (Consultant Paediatric Neurologist), Dr. Nicholas Combes (Consultant Neurophysiologist), Gemma Wilkinson (Clinical Physiologist), Andrew Lancaster (Research Nurse), Heather Collier (Research Nurse), and all the other R&D and clinical staff who's efforts did not go unnoticed. I was also fortunate enough to work with staff from Leeds NHS Hospital, with the research team lead by Dr. Munni Ray (Consultant Paediatric Neurologist). You were very generous with your time and guidance, with this thesis benefiting greatly from your involvement. The whole R&D team at Leeds were excellent, providing by far the smoothest data collection process I had during my time doing the PhD.

I also gratefully acknowledge the financial support I have received from a number of sources, without which much of this thesis could not have been achieved:

- North West Doctoral Training Centre (ESRC) studentship,
- EPSRC Vacation Bursary Scheme,
- Faculty of Science and Technology's Research Impact Fund,
- NIHR Clinical Research Network Portfolio Adoption,
- MRC Proximity to Discovery: Industry Engagement Fund,
- Google Cloud Platform Research Credits.

Last, and of course by no means least, I have to acknowledge the wealth of love and support I received from my friends, family, and partner every day of this incredibly tough process. Over the years I was lucky enough to meet some of the most talented, funny, and

interesting people I am fortunate to consider as friends. Although by no means exhaustive, in addition to those already mentioned, this list includes; Anthony Trotter, Becky Stevens, Charles Hunn, David Ramsey, Nina Harrison, Oliver Tate, and Shalmali Joshi. I am so fortunate to have the unwavering dedication from my parents, who's warmth, attention, and unending love has ensured the flourishing of my sister, brother, and I. I am also so grateful for the support from my grandparents, whos dedication to education throughout their lives instilled the importance of education in me from a young age. It is from this foundation that I have been afforded opportunities which would be unthinkable without you all. I have also been so lucky to have love and support from my partner Rebecca. You have been with me everyday of this PhD, through the ups and downs, achievements and disappointments, and I look forward to sharing many more moments with you in our journey together.

Declaration

This thesis is my own work and no portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification at this or any other institute of learning.

David Luke Elliott 19/08/2020

Contents

List of Figures

List of Tables

List of Common Abbreviations

BSS	Blind Source Separation
CNN	Convolutional Neural Network
DT	Decision Tree
DWT	Decimated Wavelet Transform
EEG	Electroencephalography
FFT	Fast Fourier Transform
GBM	Gradient Boosting Machine
ICA	Independent Component Analysis
KNN	K-Nearest Neighbour
LOG	Logistic Regression
MV	Majority Vote
MLP	Multilayer Perceptron
NHS	National Health Service
PCA	Principal Component Analysis
PSD	Power Spectral Density
RBF	Radial Basis Kernel
RF	Random Forest
RNN	Recurrent Neural Network
SD	Standard Deviation
SVM	Support Vector Machine
UDWT	Undecimated Wavelet Transform

Chapter 1

Introduction

Algorithms have been used to assist medical practice for decades. Clinical decision support systems (CDSS), care pathway analysis (eg. Map of Medicine), and forecasting tools are commonly found in clinical practice. The use of algorithms to assist diagnostic imaging has received considerable research interest due to the opportunities to improve accuracy and reduce costs with the advances in large data centres, cloud computing resources, and machine learning. Algorithms to assist diagnostic imaging is by no means a new field; indeed "computer aided" electrocardiograms (ECG) have been used in clinical practice from as early as the 1970's (Thomas Sheffield, 1987). However these algorithms were not "machine learning", instead based on heuristics (static rule based models), and had limited accuracy comparative to modern advancements (Mincholé and Rodriguez, 2019; Mincholé et al., 2019). More recently, many other medical disciplines such as radiology, dermatology, and clinical pathology, have also all had major advancements in the application of machine learning algorithms to aid the diagnostic process (Giger, 2018; Thomsen et al., 2020; Jang and Cho, 2019).

This thesis documents research which began by developing a portable Electroencephalogram (EEG) system for monitoring patients with generalised epilepsy seizures. EEG is widely used in both research and clinical contexts for various applications, including sleep analysis (Fiorillo et al., 2019), seizure detection (Abbasi and Goldenholz, 2019), and surgery (Connor, 2019), due to its high temporal resolution, non-invasiveness, and comparatively low financial cost. At the beginning of the project, hardware, software, and algorithms for

a generalised epilepsy monitor were becoming more available; however few had combined these into a full system. Indeed, subsequently there have been a number of portable EEG headbands developed (e.g. Cognionics Quick-20r), as well as products specific to epilepsy patients (e.g. Epihunter, Epilog) in various early stages of adoption and research. However, during our research with the UK National Health Service (NHS) piloting our developing system, it became apparent that the currently used method of manually marking EEG records collected during routine practice was a bottleneck on services which would likely prevent the implementation of any additional system; even if it promised to improve patient outcomes. Therefore this thesis focuses on developing algorithms which could provide a preliminary marked EEG record, to then be reviewed by a qualified physiologist, to reduce this current bottleneck and facilitate future change.

Although computer aided marking of EEG records has been researched for as long as their application to ECG's (Tzallas et al., 2012), the use of such algorithms is much less common in clinical practice. There are many reasons why automated or semi-automated EEG scoring is not already routinely adopted in the healthcare system, as outlined in Fiorillo et al. (2019):

1. The technical limitations of general EEG classification algorithms mean they typically perform poorly on patients with neurological disorders (Boostani et al., 2017),

2. Difficulties and inconsistencies in EEG scoring rules leads to high inter- and intra-scorer variability (Wilson et al., 2003; Younes et al., 2018),

3. Security and privacy issues for some cloud-based scoring services conflict with data protection policies of healthcare providers (Ali et al., 2018),

4. There is a lack of friendly user interfaces (Marcilly et al., 2016),

5. A general aversion to new technologies in the healthcare sector, which are often perceived as a threat; particularly if they substitute part of the work performed by humans or somehow intervene in the diagnostic process (Fichman et al., 2011).

Although all the barriers above need to be addressed before future implementation, this thesis focuses on the first point (technical limitations) by developing algorithms which specifi-

cally detect seizures with generalised onset. Generalized (or non-focal) seizures are a broad categorisation of seizure which have an onset that manifests quickly across the entire brain (Fisher, 2017). We choose to focus on this particular seizure type as NHS EEG data is difficult and time-consuming to collect and prepare for research purposes, and generalised seizures typically have less intra-patient and inter-patient variability than other types of seizure. We therefore were still able to develop world-class patient-general classification algorithms for this type of seizure despite the number of patient records available. No one has used "raw" NHS patient records collected as part of routine practice, as most published research uses pre-cleaned data. Furthermore, no one has compared seizure classification models using such a systematic approach or focused on the whole classification pipeline.

This thesis begins by examining the application of machine learning for seizure detection with chapter 2, which provides an overview of the main components for a potential seizure detection pipeline. The framework for viewing the layers of such a system (Pre-processing, Feature Engineering, Dimensionality Reduction, and Classification), is then used in the subsequent experimental chapters where Bayesian optimisation is used to assess the performance of different combinations of components within each of these layers. Chapter 3 specifically examines feature selection, reduction, and "classical" machine learning models as components in a classification pipeline used to detect absence epilepsy seizures in routine NHS EEG recordings. Such an approach to pipeline development is important because there is little consensus as to the best features or classifier to use to automatically detect seizures. Chapter 4 then builds upon the findings in chapter 3 by assessing more complex balanced ensemble models on the same and additional patient records. We aim to address a generalisability problem for seizure detection classifiers, where there is often a lack of multi-institution datasets used or compared. Chapter 5 then further expands the scope of the previous two chapters by using a combination of Bayesian and Hyperband optimisation on deep learning model structures and hyperparameters for the detection of a broader range of generalised seizures. This considers the performance of machine learning models on different types of seizures which have distinct intra-patient and inter-patient variabilities. Finally, chapter 6 gives a summary of the key findings and a number of suggestions for future research and impact.

Chapter 2

Methodology Review

2.1 Introduction

This chapter describes and compares the statistical methods for building a system that describes and separates signals into classes of interest; an important component of a complete seizure detection system. We choose to focus primarily on approaches previously used for the automatic detection of seizures in electroencephalography (EEG) data; however, these processing and classification techniques can be applicable to a number of clinical event and prediction detection systems using different recording modalities.

Most complete signal processing and classification systems will generally have the stages of pre-processing, feature extraction, and classification. Pre-processing prepares the raw signal for feature extraction, where aspects of the signal are quantified to best describe attributes of the data, such as biomarkers or artefacts. In order to ensure there are not unnecessary or similar features, there may be a step to reduce the number of features to those that best represent the data or train a model, or combine similar features to make new ones (extraction). The features that are chosen/created can then be classified by applying threshold or model-based criteria after being "trained" if derived from machine learning. These stages all fit into a global strategy, determined by the researcher based on the dataset, to determine which features to calculate, how to combine them, and how to account for contextual information before making a final decision in regards to a classification group (expert system; Varsavsky et al., 2011a). Indeed, selecting the features that provide the

most diversity between classes and similarity within classes is key to achieving the best classification performance (Nasehi and Pourghassem, 2012).

This chapter is predominately structured to follow the pipeline outlined above. However first, a basic introduction to time series is given in subsection 2.1.1. **Time Series**, followed by an introduction to EEG signals in subsection 2.1.2. **EEG Signals**. An overview of decisions that need to be made during the design of a signal classification system are then outlined in section 2.2. **System Design**, to give a holistic view of the process, before covering each part of a typical pipeline in detail. The first step in a typical pipeline is subsequently outlined in section 2.3. **Pre-Processing**, covering specific techniques and filter compositions often used to improve the signal-to-noise ratio of EEG data. Subsequently, section 2.4. **Feature Engineering** outlines some common techniques to describe parts of a signal in terms of their time and frequency components. We then briefly introduce and revisit techniques for reducing the number of features used for model training in section 2.5. **Dimensionality Reduction**, with these sub-categorised into methods that select a smaller subset of features (2.5.1. **Feature Selection**) or by creating new synthetic features through combining features (2.5.2. **Feature Extraction**). Following from this is section 2.6. **Classification** which reviews methods for separating signals into classes, focusing primarily on *supervised* machine learning algorithms. In section 2.7. **Discussion** we then summarise the various approaches to EEG signal classification, including their current limitations, and give some suggestions for future implementation specific to seizure detection.

2.1.1 Time Series

Time series analysis primarily aims to develop models that adequately describe a sample of ordered values. Data sampling usually is restricted by the method of collection, so series often appear as *discrete* time samples spaced equally apart. Sampling intervals of a data source is an important consideration as an insufficient sampling rate can lead to data distortions called *aliasing* (Sun et al., 1993). For the remainder of the thesis, it is assumed that adequate sampling rates have been chosen.

A time series could be considered as a sequence of random variables, $x_1, x_2, x_3, \ldots$ (Shumway and Stoffer, 2017), with a collection of random variables, $\{x_t\}_{t=1,\ldots,N}$, referred

to as a stochastic process. However adjacent points in time may be *correlated*, so that the value x_t depends on the previous values of $x_{t-1}, x_{t-2}, \ldots$. Stationarity is a characteristic of series where the statistical properties do not change over time; with first order stationarity meaning the data has a constant mean, and second order describing data with a constant mean, variance, and covariance which is independent of time. A strictly stationary time series would mean that the probabilistic behaviour of every collection of values is identical to a time shifted set. However, this assumption is often too strong in most applications, so conditions are often placed on the first two *moments* of a series. The conditions for a weakly stationary time series is firstly that the mean value function μ_t is constant and does not depend on time. Secondly, that the autocovariance function:

$$\gamma_x(s,t) = \text{cov}(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)], \tag{2.1}$$

depends only on the difference between s and t so that:

$$\gamma_x(t+h,t) = \text{cov}(x_{t+h}, x_t) = \text{cov}(x_h, x_0) = \gamma(h,0). \tag{2.2}$$

Often EEG signals are not stationary but instead are reduced to epochs (often 20-30 seconds) which are assumed to be stationary.

2.1.2 EEG Signals

The time series gained from an EEG amplifier is a digital sample of analogue voltage recordings generated by the synchronous firing of open field neurons in the brain, observed at several locations across the scalp. The digital EEG therefore approximates the continuous time signal of neural activity $\mathbf{x}^c$ through the discrete sampling of points, $\mathbf{x}_t$, over an interval Δ:

$$\mathbf{x}_t = \mathbf{x}^c(t\Delta), \qquad t = 1, 2, \ldots, N \tag{2.3}$$

At time of acquisition, data needs to be sampled abiding by the *Nyquist criterion*, which states that the sampling rate, F_sHz, can only represent frequencies of half the hertz ($F_s/2$)

of the recorded sampling rate without aliasing. The typical sampling rates for clinical EEG typically lie between 200-500Hz, meaning the spectral components generated by the cortex predominately focused on by neurologists, typically within the 1-30Hz range, can be estimated without aliasing (Kaplan and Shishkin, 2000). Changes in electrical activity can be time-locked and averaged to a stimulus to investigate event related potentials (ERP's), or the oscillatory activity can be investigated through frequency analysis (Luck, 2014b); as discussed in subsection 2.4.2.

EEG is inherently non-stationary due to many factors, such as current cognitive state (e.g. sleeping or wakefulness), or if a patients eyes are open or eyes closed. To reflect these changes, EEG is often windowed by dividing the time axis into sections that may or may not overlap. Most often a window is rectangular so that all signals inside the window range is 1 and outside is zeroed:

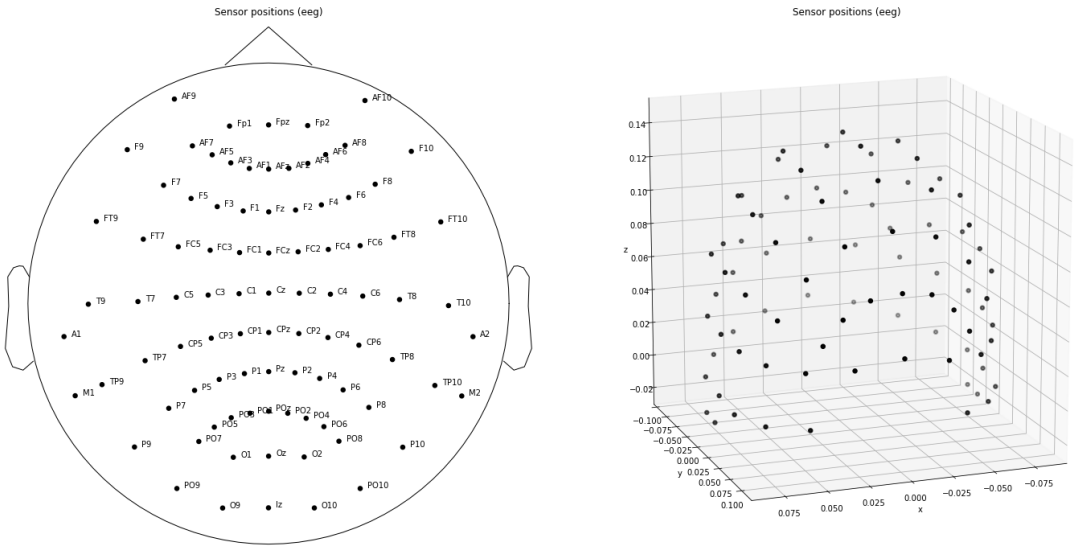
$$H_{n,k} = \begin{cases} 1, & \text{if } k+1 < n < k+N \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

However, rectangular windows have sharp edges which can affect analysis (see subsection 2.3.1), therefore other types of windows are often used; such as the *Hanning* window (Varsavsky et al., 2011b):

$$H_{n,k} = \begin{cases} 0.5\left(1 - \cos\left(\frac{2\pi(n-k+1)}{N-1}\right)\right), & \text{if } k+1 < n < k+N \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

Similar to most signals, recorded EEG data has multiple dimensions of time, frequency, power, phase, and space:

- **Time** is simply how the recorded signal changes amplitude across multiple sequential samples.
- **Frequency** refers to the speed (or the number of cycles per second) of oscillations in the signal, and can be represented in hertz (Hz) or π radians/sample (normalized units where one is half the sampling rate).



(a) Topoplot (b) 3D Locations

Figure 2.1: The international 10-20 electrode placement system.
Note. The precise locations of electrodes will depend on the equipment used.

- **Power** is the amount of energy in a frequency band, as measured by the squared oscillation amplitude.
- **Phase** is the position of an oscillation at a given time point, as measured in radians or degrees. Power and phase are two elements of a single dimension providing independent information on the strength of frequency-band-specific activity and the timing of activity, respectively (Cohen, 2014).
- **Space** refers to the locations of the electrodes on the scalp; a common montage being the 10-20 electrode system (Jasper, 1958), which places electrodes in standardized distances apart to cover the scalp (see figure 2.1).

Rhythmic brain activity contains multiple overlapping frequencies that can be separated through signal-processing techniques. These are typically grouped into bands of delta (2-4Hz), theta (4-8Hz), alpha (8-12Hz), beta (15-30Hz), lower gamma (30-80Hz), and upper gamma (80-150Hz). Other bands include subdelta and omega (up to 600Hz), but these are less commonly represented in the literature due to limitations regarding current scalp EEG's ability to represent such signals (see Gotman, 2013). These groupings of brain oscillations

loosely reflect neurobiological mechanisms of brain oscillations, such as synaptic decay and signal transmission dynamics (Buzsáki, 2009; Steriade, 2006); with faster frequencies (e.g. gamma) thought to generally reflect spatially local processing, and slower frequencies (e.g. delta) reflecting larger scale networks (von Stein and Sarnthein, 2000). Each oscillatory activity is also associated with separate cognitive functions, for example the alpha rhythm is correlated negatively with cortical activation, suggesting it reflects inhibition (Jensen et al., 2012; Klimesch et al., 2007).

Scalp EEG is typically gained by placing 21-256 Ag/AgCl electrodes on the scalp to enable the measurement of the electrical potential between spatially different electrodes. One electrode is dedicated as a *reference* during recording and another as a *ground*. A ground electrode is a common reference for the system voltage that aims to cancel out the common-mode interference that occurs from the body naturally picking up electromagnetic interference. Unless recording takes place in a Faraday cage, this interference often needs to be filtered out during pre-processing (as discussed in section 2.3) if not already conducted at time of recording by the amplifier. The ground electrode can be placed anywhere on the body, although the forehead or the ear are the most common (Light et al., 2010). Conversely, a reference electrode aims to remove unspecific brain activity by representing the electrical potential between an active electrode of interest and a relatively inactive reference. A reference electrode is also still affected by global voltage changes as it is collected against the signal ground. Referencing can be done either by using a physical reference electrode placed on the earlobe, using any electrode during recording and later re-referencing electrodes to the average output of all electrodes, or by measuring the potential between two active electrodes (bipolar recording; Varsavsky et al., 2011a). The combination of an active electrode with a reference and a ground creates a *channel*, and the general configuration of these channels are called a *montage* (Teplan, 2002).

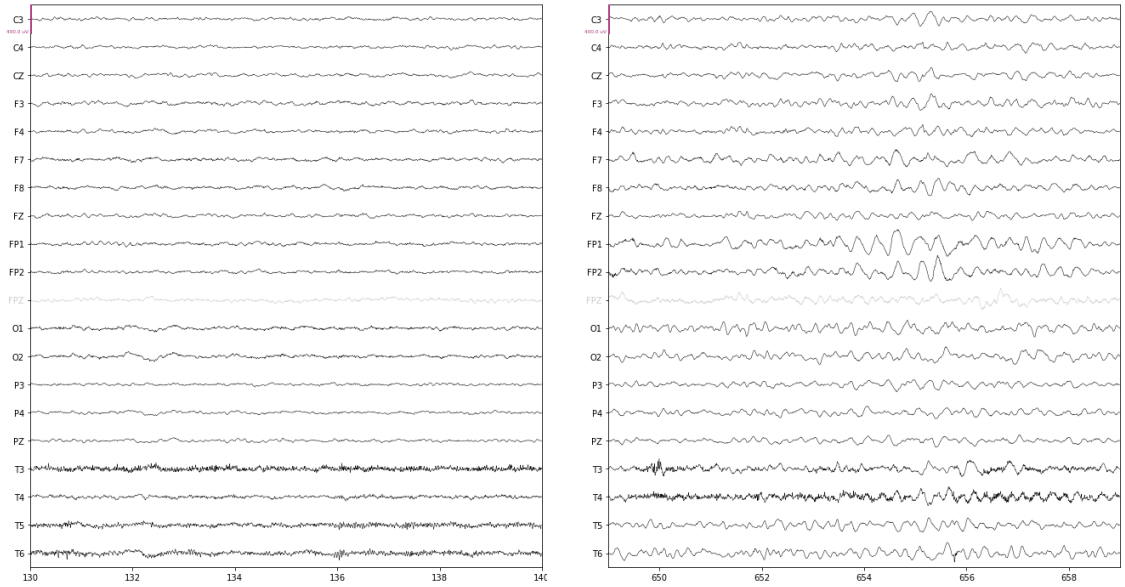
"Normal" EEG

There is a lack of global definition of what normal EEG looks like (Varsavsky et al., 2011a). This is due to EEG changing over the course of a patients life, as well as between levels of cognition (e.g. awake/asleep) or behaviour (e.g. eyes open/closed). For example, the

alpha rhythm (8-13Hz) tends to occur during wakefulness over posterior channels and is best seen when healthy adults have their eyes closed. However, large variability in voltages, spread, and quality have been noted; for example, some otherwise healthy adults have been demonstrated to have no discernible alpha rhythm (Varsavsky et al., 2011a; Niedermeyer and Da Silva, 1999). Furthermore, a common variation in EEG is found between awake and asleep EEG where, in the latter, more global waveforms oscillate at slower frequencies across the head. This global activity is inter-dispersed with fast "spikes", which are commonly found in different sleep stages. Because of such variability, clinical EEG is still assessed by human experts who aim to recognise general patterns either present in the majority of the population, or specific to a diagnostic population, based on training and experience (Varsavsky et al., 2011a). This variability also means EEG alone is rarely sufficient for a clinical diagnosis, with other diagnostic imaging and forms of observation often necessary.

Artefacts

Artefacts reflect electrical phenomena which distort the neural signal (see figure 2.3). Often strategies are employed during data collection to reduce the presence of artefacts in the data



(a) Background EEG

(b) Rhythmic Waveforms

Figure 2.2: Examples of "normal" waking EEG.

(e.g. replacing electrodes), but these require correct identification by EEG technicians at the time of collection. A large number of artefacts in EEG data are caused by improper electrode application or recording preparation, so laboratories and medical facilities often have specific protocols to reduce these (see figure 2.A.1; Spriggs, 2009). Post-collection strategies for reducing the effects of artefacts include manually removing segments of data or channels with excessive artefacts, filtering (see subsection 2.3.1), removal using separation methods (see subsection 2.3.2), or training a system to identify and cope with common artefacts (see section 2.6). These latter methods are preferable due to their ability to preserve more data, but still typically struggle with muscle artefacts which appear in the 15-20Hz range (Gotman et al., 1981; Osorio et al., 1998; Safieddine et al., 2012).

The main artefacts that contaminate EEG are:

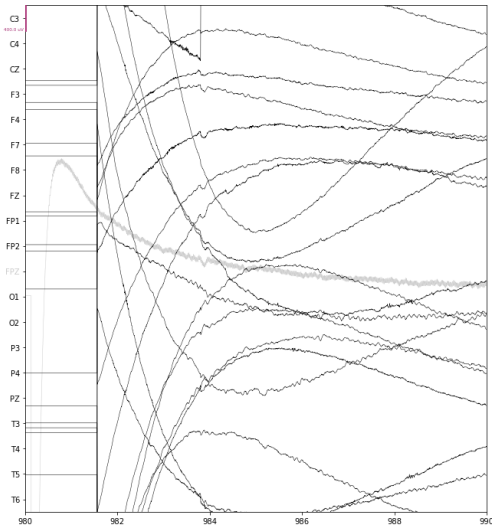
- **Amplifier Saturation** which is caused by a high input signal, such as electrode movements or impedance testing, and causes signal loss.

- **Cardiac Activity** which is often measured using an electrocardiograms (ECG), with this interference in EEG channels typically being of relatively low amplitude (Sörnmo and Laguna, 2005). However, due to its repetitive and regular pattern, it can sometimes be mistaken for epileptiform activity, when ECG is not simultaneously measured (Urigüen and Garcia-Zapirain, 2015).

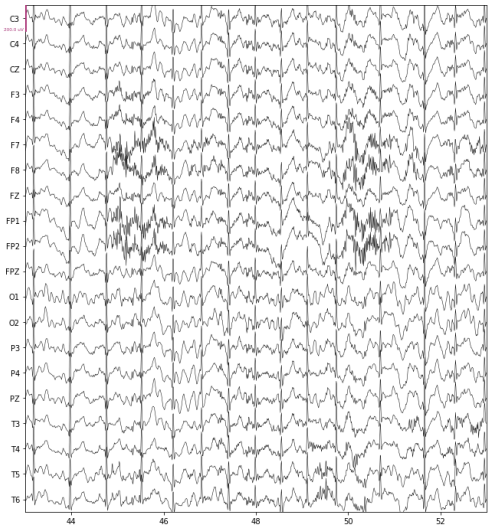
- **Line Noise or High Impedances** which are usually in the frequency range of 50Hz in the United Kingdom (60Hz in the United States), due to this being the frequency used by most electrical devices and outlets (Spriggs, 2009).

- **Myogenic Activity** which is typically measured using an electromyogram (EMG) to capture electrical activity generated by contracting muscles. The shape and amplitude of the interference depends on the muscle contracted and is often difficult to characterise (Goncharova et al., 2003).

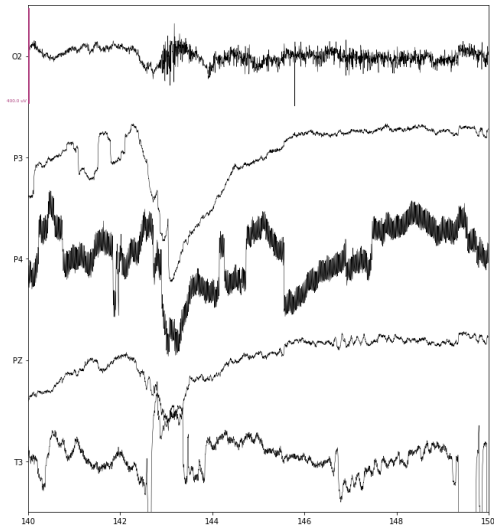
- **Ocular Artefacts** which are often measured using an electrocardiogram (EOG) and contaminate EEG signals primarily in frontal electrodes (Romero et al., 2008). For example, a blink artefact can typically be easily identified as a short large amplitude



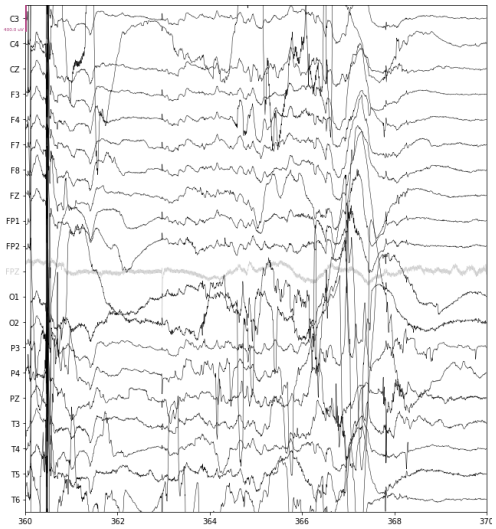
(a) Amplifier saturation found at the start of a recording.



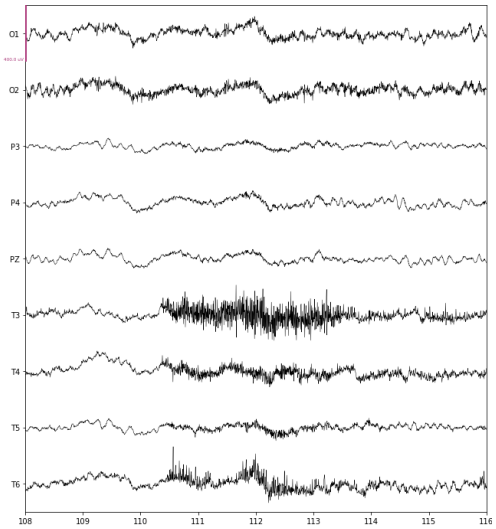
(b) Effect of ECG present in the reference electrode.



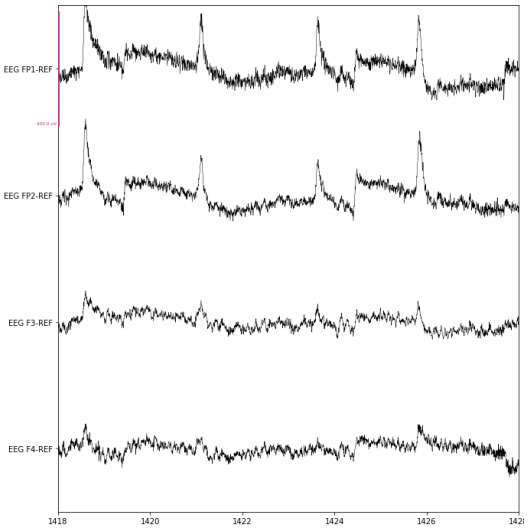
(c) Line Noise in the middle channel (P4) likely due to disconnection from the scalp.



(d) A movement artefact likely caused from full body movement.



(e) An example of an artefact in the temporal electrodes caused by jaw clenching.



(f) Example of a patient blinking in rapid succession

Figure 2.3: Examples of typical artefacts found in EEG.

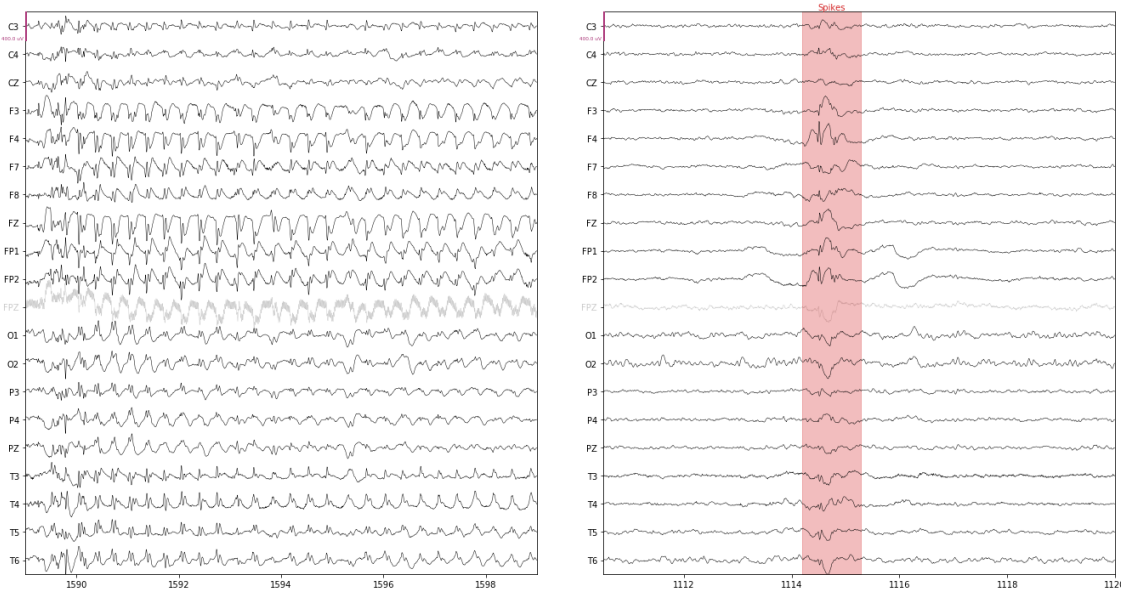
spike that is larger than background activity (Croft and Barry, 2000).

- **Skin Potentials** are less common than other artefacts, but are caused by electrodermal interference from skin potentials and the sweat glands (Urigüen and Garcia-Zapirain, 2015).

Seizures

EEG can be used to identify various clinical markers relevant for diagnosis and treatment; most commonly regarding sleep and epilepsy disorders. There are many different types of epilepsies and causes, but for this review we will adopt the three broad classifications proposed in the International Classification of Seizure Types (Fisher, 2017): focal onset, generalised onset, and unknown onset. Focal (or partial) seizures are caused by an abnormality in a specific part of the brain, with seizures categorised as *secondarily generalized* if they spread to a large proportion of the brain (Varsavsky et al., 2011a). Primarily generalized (or non-focal) epilepsies have seizures where the seizure onset manifests across the entire brain immediately, or at least so fast they seem instantaneous. Unknown onset, encompasses seizures that are yet to be fully classified with confidence into the previous two broad categories. Seizures can also be categorised as continuous (or status epilepticus), which can be either focal or non-focal in nature. Continuous seizures show no observable recovery between seizures and can be life-threatening if they last more than 5 minutes. As this thesis focuses on generalised seizures, the other types are not discussed in any further detail.

Similar to "typical" EEG, there is a lot of variability found between and within seizure types. Nevertheless, comparative to pre-seizure background EEG (often termed inter-ictal EEG), seizures have the common traits of synchronisation across a few or many EEG channels, a large amplitude, and increased oscillatory activity. Onsets and offsets of seizures are typically abrupt, however variations can occur both between patients as well as between seizures. Patients may also have spiked inter-seizure discharges, which are short bursts of high amplitude, syncronized activity around an epileptic focus (see figure 2.4; Varsavsky et al., 2011a).



(a) Generalised Absence Seizure

(b) Spiked Inter-Seizure Discharge

Figure 2.4: Examples of epileptiform activity in EEG.

2.2 Signal Classification System Design

Before looking at each potential step in a signal classification pipeline, which could be used to detect seizures, it is worth taking a broad view of the decisions that need to be made in their design (see figure 2.5). Such considerations are sometimes referred to as part of an *expert system*, where each step in a potential pipeline is tailored to be applicable to a certain problem (Varsavsky et al., 2011a). This includes making decisions regarding...

- ...how to prepare the data for input into the system,
- ...the number and type of features to use,
- ...the order of components in a system,
- ...how the system accounts for known contextual information.

Data can be prepared for a system in many ways, with the process of transforming and mapping data from a "raw" form to a format appropriate for downstream purposes known as *data wrangling* or *data munging*. Although we focus in this chapter on specific pre-processing techniques for signals, this stage of the development will likely also include

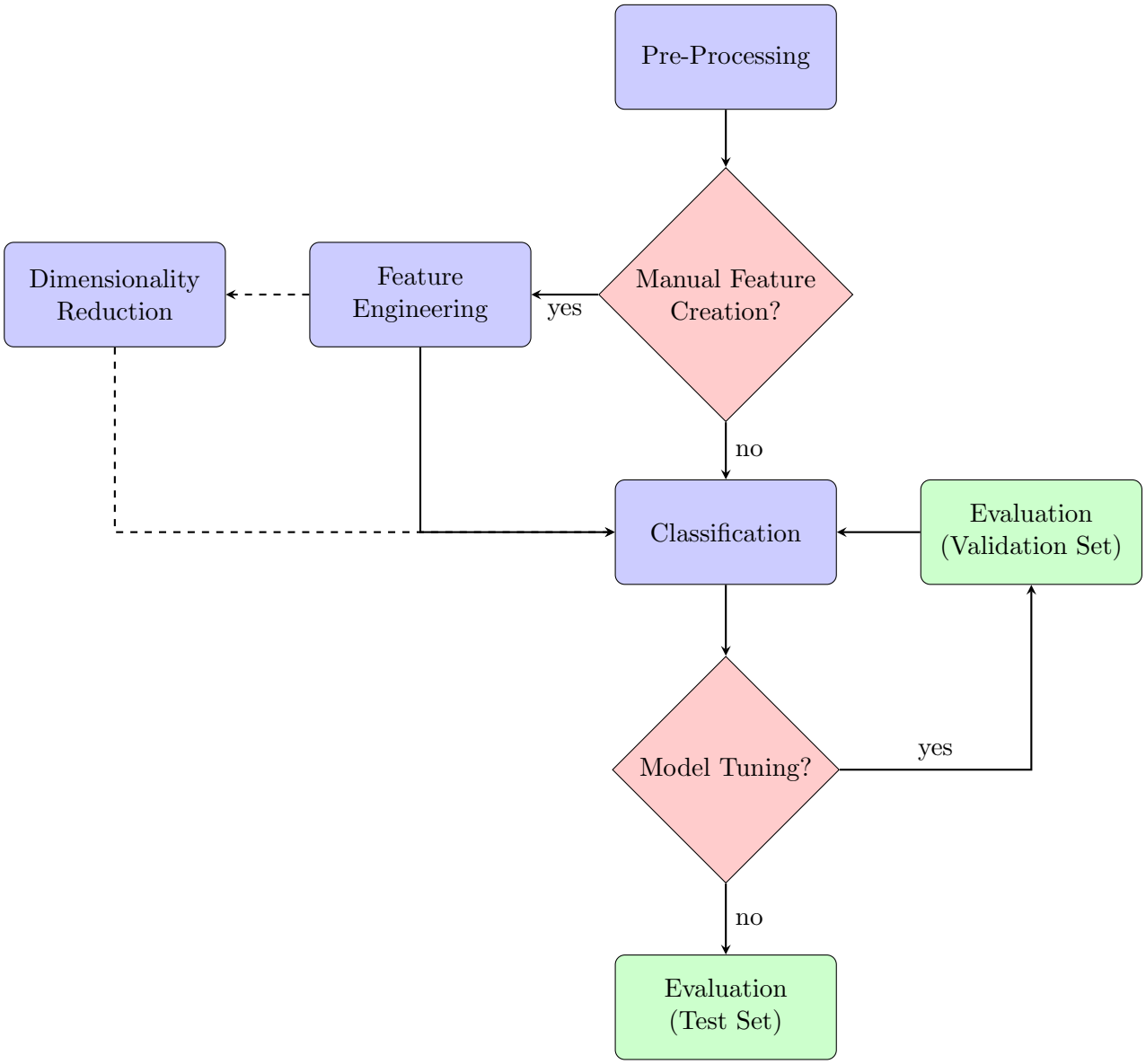


Figure 2.5: A basic signal processing and classification system development pipeline.

other steps of data visualization and data aggregation to aid decision making for all steps of the pipeline. If a classification model is chosen that requires features to be manually created ("hand-crafted"), the number and type of features to use needs to be considered to avoid *over-parameterization*. Over-parameterization can be an issue as it adds redundancy and time to the system, therefore only features that provide new or partially independent information should be added to a model. As well as reviewing appropriate prior literature for features that work best for specific models and applications, there are also data driven techniques to assess which are the best features to aid classification (see section 2.5).

Different authors often take different strategies regarding how they order a system; even when using the same pipeline components. Furthermore, a system need not only have one of the possible components at each stage outlined in this review, and need not strictly follow its ordering. For example, a system may have multiple classification stages, with extracted features first classified by simple threshold rules that feed into a more complex classification system (e.g. Liu et al., 2002), or have multiple classifiers running simultaneously, with the one that models the data best used (e.g. Subasi, 2007b). Specific to an automatic seizure detection system, components specific to detecting unwanted features in the data, such as artefacts and noise, may have their own features and classification system, with its output feeding sequentially into a system used to detect features of interest (e.g. seizures).

Many decisions regarding a pipelines design will likely involve how to account for contextual information. For example, alterations to steps in the pipeline may be made to account for differences in the time of day the EEG was collected (the temporal context); such as by altering the size of labelled windows, the number of predictions in succession needed to output a classification label, or the use of forgetting factors/varied window length to ensure recent events are focused upon more during training (e.g. Wilson et al., 2004). Other known contextual information can also be incorporated into the model design, such as the spatial context. Considering this dimension, EEG channels may be combined into groups, treated separately, or compared at the end of the classification process when making a labelling decision.

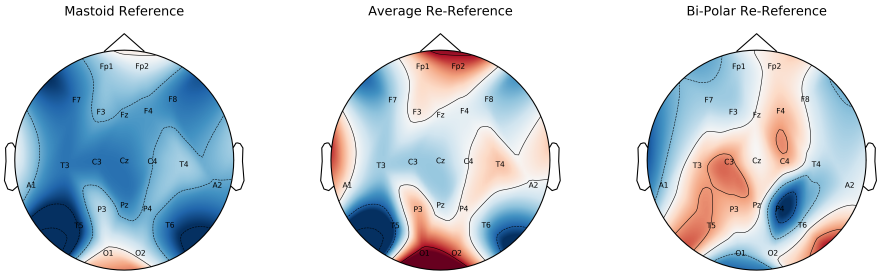
Decisions around pipeline design are informed by the overall purpose of the system. Generally a seizure detector can be classified as a *seizure-event* detector or a *seizure-onset*

detector. Seizure-event detectors aim to identify seizures with the greatest possible accuracy, and tend to focus on being able to be generalised to a whole population. These could enable physicians to better titrate therapy as they could provide a summary of frequency, duration, and time of individual seizures and relate this to the individualised patient therapy plan to maximise their benefit (Kharbouch et al., 2011; Nasehi and Pourghassem, 2012). Seizure-onset detectors aim to detect the onset of a seizure with the shortest possible delay (Nasehi and Pourghassem, 2012); favourable for time sensitive diagnostic and therapeutic interventions, and for life threatening seizures specific to individual patients. Seizure-onset detectors could be used to initiate functional neuroimaging to localise the cerebral origin of a seizure (Nanobashvili et al., 2011), trigger neurostimulators to affect seizure progression (Rothman and Yang, 2003; Theodore and Fisher, 2004), or alert a patient or carer to the patient's condition (Nasehi and Pourghassem, 2012). Therefore, seizure-event detectors tend to favour pipeline components which improve system accuracy, and seizure-onset detectors favour components which improve prediction latency.

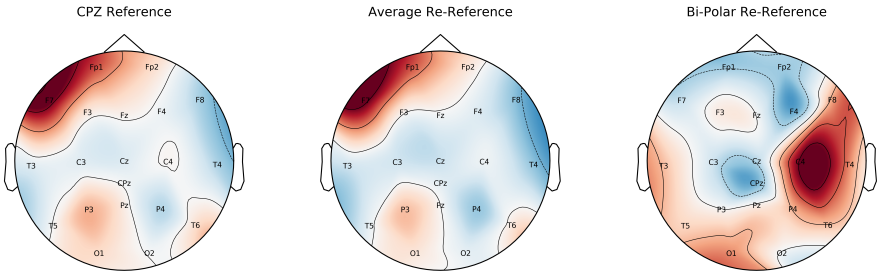
Developing a classification system is often not a linear process, and will tend towards refinement and reiteration. Although some decisions above are best made before developing a system, changes in components and methods in all stages of pre-processing, feature engineering, dimension reduction, and classification will likely occur over time. The following examines each step in a typical pipeline, investigating potential decisions that are possible in each component.

2.3 Pre-processing

Pre-processing refers to transforming a signal through its re-organisation (e.g. extracting epochs), removal of bad/artefactual data (e.g. removing bad electrodes or rejecting epochs with artefacts), or modifying otherwise clean data (e.g. normalising, referencing). Many signal processing systems epoch data into segments either around particular experimental events or in defined window sizes to create quasi-stationary segments of EEG. At this early stage of the process, normalisation is typically applied to convert the signals into a common range so they can be compared. This is especially required if signals are acquired by different



(a) Mastoid Reference



(b) Central Reference

Figure 2.6: Changes in average amplitude across a full EEG record due to re-referencing.

recording equipment or by other researchers/technicians. Signals can be normalised through detrending the signal by removing the mean and/or scaling to a unit variance; as the mean of a single electrode recording is not meaningful as it depends on the setting of the amplifier gain. This is particularly useful for classifiers that use optimization algorithms (e.g. support vector machines), as it makes it easier for the model to learn weights and makes the algorithm less sensitive to outliers (Raschka and Mirjalili, 2019). Re-referencing can also be conducted at this stage to emphasise differences in electrical activity between electrodes. The most common re-reference method for seizure detection uses the linked ears or mastoids, which can be used to show the spike and wave pattern in seizures at a large amplitude (Lopes Da Silva, 1978), but can introduce some bias. Another option is to use bi-polar or average re-referencing, which can be used to reduce the influence of cardiac artefacts if the recording reference electrodes were placed on the mastoid. However, changing a signals reference will inevitably change some of the topological properties of the EEG (see figure 2.6).

Modification of the signal to improve the signal-to-noise ratio is commonly achieved through the use of signal filtering/source separation or removing/interpolation of "bad channels". Identifying bad channels can be done visually, however this approach suffers from a lack of standardisation which is required for generalisability across subjects and paradigms. Instead, a pre-processing pipeline to remove bad channels, such as the PREP pipeline, can be used (Bigdely-Shamlo et al., 2015).

The remainder of this pre-processing section will focus on filtering (subsection 2.3.1) and blind source separation (subsection 2.3.2) methods. Here we focus mostly on time-domain filters purely because they are more common for pre-processing EEG signals, with frequency-based filters (e.g. Fourier and wavelet transforms) more common for feature engineering (section 2.4). Additionally, we focus on *digital filters*, that are used for offline filtering; but the principles discussed also apply to online *analogue filters* that are found in EEG amplifiers. The theory introduced in the following section is applicable to many of the other potential steps in the pipeline, and is specifically used to demonstrate common EEG filters to clean noisy, or "artefactual", segments of EEG data.

2.3.1 Signal Filtering

Temporal or frequency filtering aims to attenuate signal components of a particular frequency/frequency band. Filters are characterised in the time domain by their *impulse response* and in the frequency domain by their *frequency response*. These responses describe the *transfer function* of a filter, which is the effect of a filter on the signal input that results in a filtered output (Widmann et al., 2015). Therefore, lets first look at how to understand these in each respective domain.

Impulse Response Function

A basic filter based in the time domain could average each time point, $\mathbf{x}_i$, with adjacent time points, $\mathbf{x}_{i-1}$, $\mathbf{x}_{i+1}$. This approach can be extended to filter a broader range of high frequencies by averaging a larger number of points. This can be formalised so that a filtered

series, $f(\mathbf{x})$ at time i is computed using:

$$f(\mathbf{x}_i) = \sum_{j=-n}^{n} w_j \mathbf{x}_{i+j} \tag{2.6}$$

where w_j is a weighting value, $w_j = \frac{1}{2n+1}$.We can change the weighting function to account for the temporal proximity of surrounding timepoints, which will increase the temporal precision of the filtered series, by changing $\mathbf{w}$ to a series of weights (e.g. $\mathbf{w} = [0.25, 0.50, 0.25]$). Changing the weighting function will determine the desired properties of the filter. As well as computing the each filtered timepoint, often the weighting function is reversed in time to describe the effect of the current point on the output of the filter. As reversing the weighting function is equivalent to a filtered waveform in response to a voltage spike or impulse, it is known as the *impulse response function* of the filter (Luck, 2014c). Often filters are described by the impulse response function instead of a weighting function, which can be reflected by performing the same filtering operation described in equation 2.6, except reversing the weighting function to create coefficients of the impulse response function. When expressed as an impulse response function, $\mathbb{I}$, filtering can be viewed as a *convolution*:

$$g(\mathbf{x}_i) = \sum_{j=-n}^{n} \mathbb{I}_j \mathbf{x}_{i-j} \tag{2.7}$$

where $\mathbb{I}_j$ is an impulse response function which gives 1 at time j, and 0 elsewhere. The function $g(\mathbf{x})$ simply reverses our weighting vector. Convolution is based on the dot product, which can be interpreted as the sum of elements in one vector weighted by the elements of another, or co-variance between two vectors. Simply a dot product is the multiplication of each element in one vector by the corresponding element in the other:

$$< \mathbf{a}, \mathbf{b} >= \sum_{i=1}^{n} a_i b_i \tag{2.8}$$

Convolution extends the dot product by computing it repeatedly over time. For a signal-processing interpretation, it is easiest to interpret convolution as a time series weighted by another signal that slides along the signal. One vector is the signal and the other is the kernel; which could be a wavelet or sine wave (see subsection 2.4.2). The dot product

between the kernel and corresponding signal is placed in a new vector corresponding to the centre of the kernel, meaning it is often convenient to have an odd number of data points (Cohen, 2014). Furthermore, to ensure the resulting vector after convolution is not shorter than the original signal, the signal is often zero padded.

Frequency Response Function

As previously mentioned, as well as characterised by the impulse response function, properties of the filter also are described by their *frequency response*. The frequency response of a filter can be calculated simply by taking a Fourier transform of the impulse response.

A Fourier transform computes the dot product between a signal and sine waves of different frequencies (*kernels*; Cohen, 2014). Sine waves are oscillations, characterised by their frequency, power, and phase (see subsection 2.1.2). This means the result of a Fourier transform is a three-dimensional representation of the original signal. We denote a sinusoidal signal as $A\sin(2\pi\omega t+\theta)$, where A is the amplitude of the sine wave, ω is the frequency, t is time, and θ is the phase angle.

A *discrete-time* Fourier transform computes the dot product between multiple sine waves, with different frequencies, and the signal. The number of sine waves created, and their frequency, is determined by the number of data points in the time series as the frequencies can range from zero to the Nyquist frequency. Given a periodic sequence $\mathbf{x}_n$, with period N, the *discrete-time Fourier series* representation of $\mathbf{x}_n$ is expressed as (Proakis and Manolakis, 2006):

$$\mathbf{x}_n = \sum_{\omega=0}^{N-1} c_\omega e^{j2\pi\omega n/N} \tag{2.9}$$

Often we wish to obtain the Fourier coefficients c_ω, which provides a description of $\mathbf{x}_n$ in the frequency domain, ω, using:

$$c_\omega = \frac{1}{N}\sum_{n=0}^{N-1} \mathbf{x}_n e^{-j2\pi\omega n/N} \tag{2.10}$$

with c_ω representing amplitude and phase associated with a frequency component. However,

the discrete-time Fourier transform, as outlined above, is rarely used in practice, instead replaced by the fast Fourier transform.

Taking a Fourier transform of a filters impulse response results in two parts: the magnitude response (or amplitude) and the phase response. The magnitude response, typically plotted along the x-axis in linear or logarithmic scale (dB), is effectively multiplied with the spectrum of the signal during filtering (Widmann et al., 2015). Ideally, frequency bands to be attenuated will have values of 0, to remove these spectral components, and those to be passed will have a magnitude value of one, so they are not changed. However, digital filters rarely can meet such a criteria and never completely remove spectral components. The phase response of a filters impulse response reflects the delay of the filters output comparative to its input, with negative values reflecting delays. A filter said to have a *linear phase response* has the same delay for all spectral components. For visualisation it is typically "unwrapped" and plotted along the x-axis in radians or degree (e.g. figure 2.8).

Filtering EEG Signals

Filtering can distort both the temporality and amplitude of an EEG signal (VanRullen, 2011), however if used appropriately and in moderation, it can increase the signal-to-noise (SNR) to reveal more clearly the temporal dynamics of EEG data (Widmann and Schröger, 2012). As hardware filters can only use previous time points, they are more likely to cause a significant phase shift on the data (Luck, 2014b). This is the main reason it is best to do most EEG filtering "offline" rather than "online". However, as filtered values are computed using the surrounding points, if time points do not exist prior or after a given time point, this will cause the filter to produce *edge artefacts* (Luck, 2014b). The signal therefore should be padded at the edges, either by an inverted image of time and amplitude or with a DC constant. This also leads to the recommendation against filtering epoched data, as continuous EEG will have less distortions from padding. Furthermore, filtering across signal discontinuities and DC offset corrections should also be avoided.

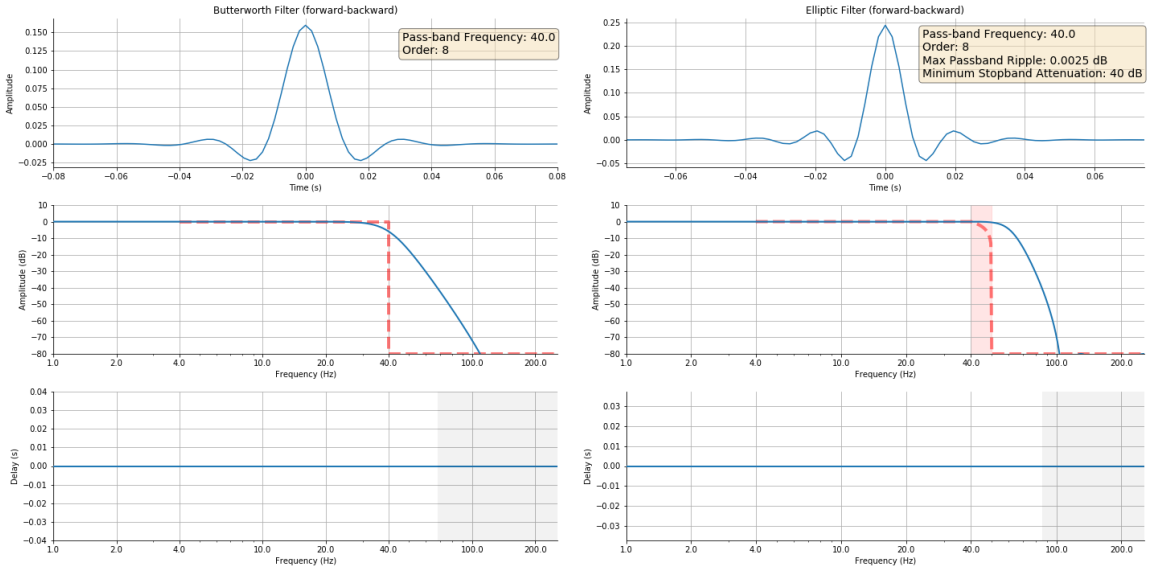
For offline filtering of EEG signals there are various recommendations, but these are intended to be altered depending on their specific application (e.g. Luck, 2014a; Swartz Center for Computational Neuroscience, 2018). For example, some authors argue against high-pass

filtering below 0.1Hz cut-off frequencies, particularly if estimating window mean or peak amplitudes (e.g. Acunzo et al., 2012; Luck, 2014a), or low-pass filtering if estimating onset latencies (e.g. VanRullen, 2011), due to the potential temporal distortions if using filters with a steep rolloff. Finite Impulse Response (FIR) filters (see figure 2.8), are commonly preferred over Infinite Impulse Response (IIR) filters (see figure 2.7); as a causal FIR filter has the same time delay across frequencies (linear-phase), whereas causal IIR filters have non-linear phase, and are generally less numerically stable due to the use of recursive calculations (Parks and Burrus, 1987). This means IIR filters are only recommended in cases where there is a high throughput, due to their computational efficiency, or a sharp cutoff is required, because their reduced ripple (Widmann et al., 2015). For EEG, FIR filters are typically steep for high-pass filters and shallower for low-pass filters (Acunzo et al., 2012; Luck, 2014a). Band-stop filters are almost exclusively used to suppress line (50/60Hz) or cathode ray tube (CRT) noise. However, as sharper filters lead to worse precision in the time domain, band-stop filters are not always recommended (e.g. Widmann et al., 2015). High-pass filters are typically used to force a signal to zero amplitude, to reduce a signals offset caused by direct current (DC), and to attenuate skin potentials and other slow voltage changes. Low-pass filters are used to smooth a signal and reduce the effect of high frequency noise and myogenic activity.

2.3.2 Blind Source Separation

Although simple low-pass, band-pass, and high-pass filters are common methods in EEG research, these are not effective when the spectrum of artefacts and the signal overlap (Sweeney et al., 2012). Common alternative techniques for removing artefacts include blind source separation (BSS), wavelet transform, empirical-mode decomposition, regression, and/or combining these into hybrid methods. We choose in this subsection to focus on BSS techniques; with some of the other methods further discussed in relation to feature engineering (e.g. wavelets) or classification (e.g. regression).

BSS methods cover a variety of unsupervised learning algorithms for separating a set of mixed signals into their component sources, with little information on how they are mixed. Generally, a set of observed signals, X, is assumed to come from a collection of original



(a) Butterworth Filter (b) Elliptic Filter

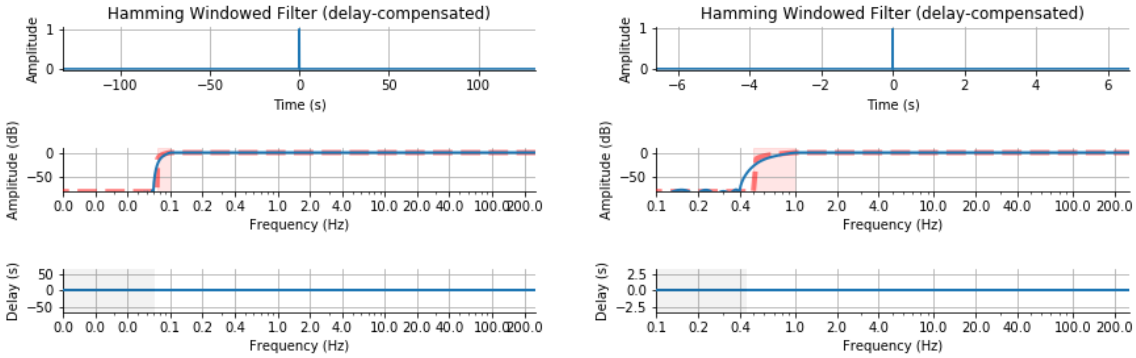
Figure 2.7: Example IIR filters for EEG.

signals mixed with artefacts, U. These signals are linearly mixed via an unknown matrix A, $X = AU$. BSS reverses this algorithm to a reverse mixing of X, $U = WX$, as to estimate the sources by W (Jiang and Bian, 2019). Typically when applied to signal pre-processing, once a signal is split into components, the components of the signal representing artefacts are identified, removed, and the signal is reconstructed. There are many BSS algorithms, such as Canonical Correlation Analysis and EASI, but we will look at the two mostly commonly applied to EEG; Principal Components Analysis and Independent Components Analysis.

Principal Components Analysis (PCA)

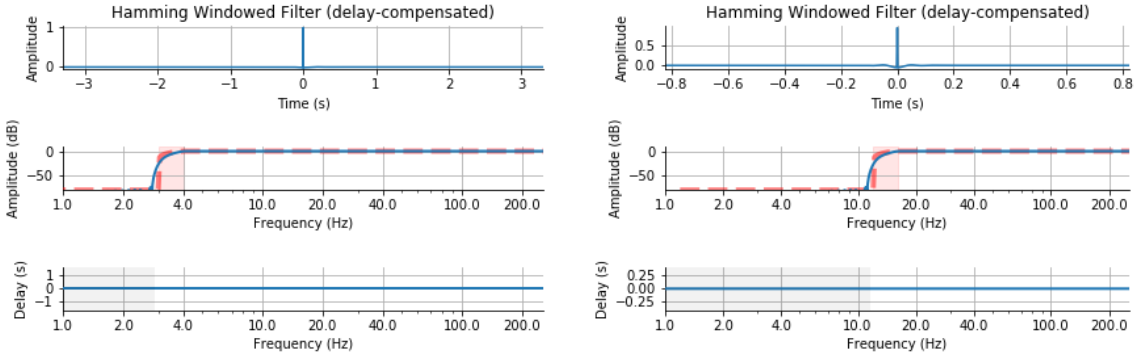
Although briefly covered here, more complete descriptions of PCA can be found in Jolliffe (1986) and Lebart et al. (1984). To separate signal components, PCA aims to find vectors that best explain data variability by transforming data onto an equal or lower dimensional subspace. Principle components are the orthogonal axes of the new subspace giving directions of maximum variance. To estimate the orthogonal space, X is factored as:

$$X = U \times D \times V^T \tag{2.11}$$



(a) Luck (0.1Hz)

(b) SCCN (1Hz)



(c) SCCN (4Hz)

(d) SCCN (16Hz)

Figure 2.8: High-pass FIR filters according to recommendations (Luck, 2014a; Swartz Center for Computational Neuroscience, 2018).

where U and V are left and right singular vector matrices, T being the transpose, and D the diagonal matrix with singular values λ_i (Costa et al., 2014). The PC scores, Z, are then linear combinations of X with the column-vector V (loadings matrix):

$$Z = X \times V = U \times D \tag{2.12}$$

The first principle component will have the largest variance, with subsequent components decreasing in magnitude, as well as each being uncorrelated (mutually orthogonal) to other components. For example, the portion of data variance accounted for by the first p component, as a percentage ratio, is:

$$RV_p = \frac{\sum_{i=1}^{p} \lambda i}{\sum_{i=1}^{n} \lambda i} \times 100\% \tag{2.13}$$

where λi is the eigenvalue associated with the i^{th} PC (Artoni et al., 2018). The number of PCs are typically constrained to a pre-defined threshold so that the "signal" subspace is kept, while the orthogonal "noise" subspace is rejected (Artoni et al., 2018).

Independent Components Analysis (ICA)

ICA is another BSS algorithm, which transforms observed data into latent components that are maximally independent (Hyvarinen and Oja, 2000). For ICA, the different components of a signal are assumed to be statistically independent with non-Gaussian distributions; meaning one variable does not give any information on the values of another. ICA revolves around maximising or minimising contrast functions to provide the optimal separation of latent components (Hyvarinen and Oja, 2000). A commonly used contrast function is *kurtosis*, used to measure non-Gaussianity. Kurtosis describes the peak sharpness of a frequency-distribution curve, with zero representing a Gaussian variable and greater than zero representing most non-Gaussian variables. However, kurtosis is sensitive to outliers (Huber, 1985) as its estimated value may depend on the few observations at the tail end of a distribution. Negentropy is another contrast function that measures non-Gaussianity and is based on *entropy*. Entropy, which quantifies the amount of disorder in a system, can be used as a measure of non-Gaussianity as Gaussian distributions are known to be the least structured distributions (Cover and Thomas, 1991; Papoulis, 1991), and therefore have high entropy values. Negentropy modifies entropy to provide a non-negative value for non-Guassian variable and zero for a Gaussian variable. However this method is computationally difficult, as an estimate of the probability density function of the components is necessary. There are also other contrast functions for ICA estimation available; such as the minimization of mutual information and maximum likelihood estimation. When applying ICA, whitening is often performed by linearly transforming a vector so its components are uncorrelated, with an equal variance of one. This also reduces the complexity of the problem, as it reduces the number of parameters that need to be estimated. If the eigenvalue decomposition of the covariance matrix is used for whitening, you can also reduce the dimension of the data simultaneously by disregarding the eigenvalues of the co-variance matrix that are small, similar to PCA.

Source Separation of EEG Signals

Many research and signal classification applications of EEG use BSS to remove artefacts from the data (e.g. Joyce et al., 2004; Vos et al., 2010). BSS is compatible with a model of EEG that assumes signals are linear mixtures of waveforms from multiple neural and artifactual sources that propagate instantaneously to the scalp (Sarvas, 1987; Safieddine et al., 2012; Urigüen and Garcia-Zapirain, 2015). To use BSS, an appropriate algorithm may be selected per artefact, which can be chosen based on the valid assumptions for each type of artefact (De Vos et al., 2011). For example, PCA has been shown to be more effective for removing ocular artefacts and source localisation than non-BSS methods (Berg and Scherg, 1991; Casarotto et al., 2004); with robust (Shi et al., 2013) and kernelised (Teixeira et al., 2008) variants also previously applied. However, the assumption of orthogonality between neural activity and typical physiological artefacts required for PCA is not often supported (James and Hesse, 2005; Vigário, 1997; Choi et al., 2005). Furthermore, PCA has been shown to be unable to separate some artifactual components from brain signals (Fitzgibbon and Powers, 2007; Urigüen and Garcia-Zapirain, 2015), meaning it is sometimes only used as a whitening step for a subsequent ICA algorithm (Vigário, 1997; Vigário and Oja, 2008).

The basic method of noise removal using ICA is carried out by decomposing the EEG signal into independent components (IC), the component with the most noise is detected and values zeroed, and the newly formed IC matrix is multiplied by the mixing matrix to obtain a cleaned EEG signal (Çınar and Acır, 2017). Such an approach to ICA has been shown to have good performance separating linear mixtures of EOG signals with simulated and experimental data (Vigário, 1997), and when compared to regression methods (Jung et al., 2000). Biomedical signals are commonly processed using SOBI (Belouchrani et al., 1993), InfoMax (Sejnowski et al., 1999), or fastICA (Hyvarinen and Oja, 1997; Hyvarinen, 2008) variations of ICA (Delorme et al., 2012; Albera et al., 2012; Urigüen and Garcia-Zapirain, 2015). Prior knowledge of the signals can be used to semi-automate the selection of components, such as a temporal constraint which uses a reference signal to find components similar to the reference but statistically independent of other sources (James and Hesse, 2005; Lu and Rajapakse, 2006; Romero et al., 2008; Lu and Rajapakse, 2000, 2005; James

and Gibson, 2003), or a spatial constraint by making assumptions of the spatial topography of source projections (James and Hesse, 2005; Hesse and James, 2006; Akhtar et al., 2012). To fully automate this approach, authors have also used measures of kurtosis (Yang, 2015) and modified multiscale sample entropy (mMSE; Mahajan and Morshed, 2015) to determine the component that contains artefacts. However, ICA methods are typically best used in hybrid systems (e.g. Klados et al., 2009; Zhou and Gotman, 2009; Çınar and Acır, 2017).

Nevertheless, similar to PCA, the properties of EEG data does not meet all the assumptions of ICA. Indeed, most signals measured by physical sensors are typically non-Gaussian. Although PCA does not require Gaussianity of the data, it does typically work better with Gaussian data as this can ensure the principle components are independent. PCA is typically used to condense as much variance into the fewest components possible, but should not be expected to separate signals from different generators. Indeed, as artefact components are often correlated with EEG data, PCA is limited when drifts and EEG signals are similar (Jiang and Bian, 2019). However, an priori knowledge of how many components there should be is required when applying ICA, as the the higher-order statistical dependencies that are intended to be reduced by ICA can result in a number of potentially optimal solutions (Lee, 1998); meaning repeated ICA decompositions can give different solutions (Delorme and Makeig, 2004; Duann et al., 2003, 2001; Esposito et al., 2002). Conversely, PCA is not influenced by the number of components selected, as the components generated by PCA are consistent given the same conditions. Also when reducing components of an EEG signal using ICA, additional variance can be induced in reconstructed EEG signals where potential artefactual components are removed, as real signal aspects may also have been removed, or aspects of the artefact may still be present (Pontifex et al., 2017). Nevertheless, both methods are useful for removing artefacts as part of a signal pre-processing pipeline, particularly when addressing ocular and cardiac artefacts.

2.4 Feature Engineering

Key to the performance of any machine learning algorithm is the successful extraction of salient features, which can come from both domain knowledge and computational feature ex-

traction techniques (Raschka and Mirjalili, 2019). We will first look over basic time-domain features that have been previously used with EEG, before revisiting frequency domain representations of signals to cover Fourier and wavelet transforms. These techniques, as well as other methods that fall into time, frequency, and time-frequency domains, are examined throughout in regards to their application to EEG signal analysis.

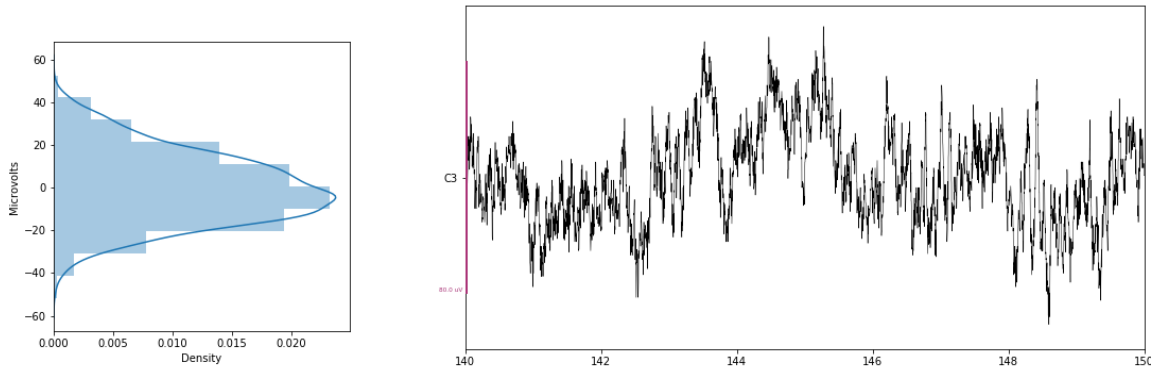
2.4.1 Time Domain Features

For signals, a single sample isolated in time has limited explanatory value. However, observations over time allow for signal dynamics to be better observed. Although raw values from two samples of a signal may be different, they are likely to have a similar distributions; suggesting they are, on average, drawn from the same statistical distribution (see figure 2.9). Averages taken over time, space, or distribution are common to describe a signal. However, the length of the window used to calculate this average is important, as the longer the window, the more likely the estimated statistic reflects the "true" distribution of a signal (Varsavsky et al., 2011a). We have already discussed measurement noise, as the recorded signal differs from the true signal due to the acquisition process and artifacts, but there can also be computational noise. Computational noise results from an inappropriate representation of data and limits the interpretation of a statistic, so caution should be used when choosing statistics to represent signal components.

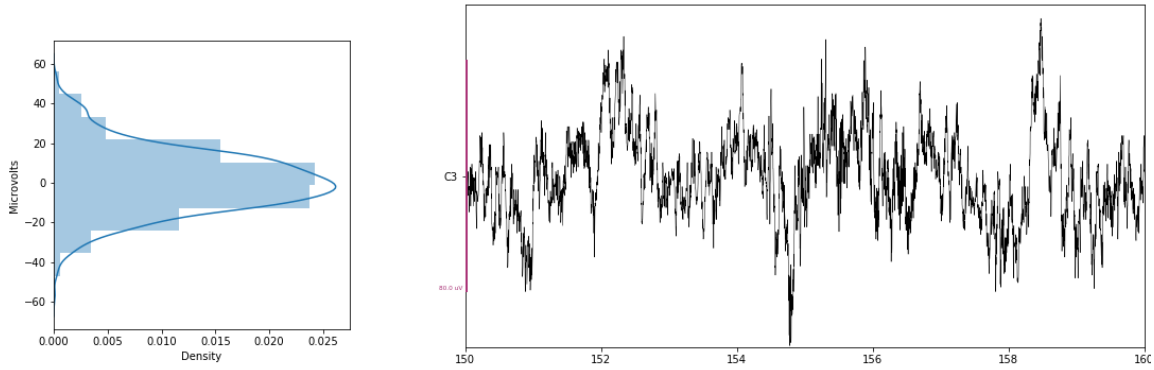
Linear Signal Analysis

Windowing can used to enforce artificial stationarity for the local signal so that linear statistical methods can be performed. Window sizes for EEG are often short, with 20-30 seconds of EEG assumed to be weakly stationary (Elger et al., 2002); but the window size should not be too short as to make the computed statistic invalid (Varsavsky et al., 2011a). Given long window sizes, time domain statistics can be used to give estimates of long-term behaviour at the expense of missing short term patterns, or short windows to better temporal information but less representative signal estimates.

A basic linear time domain feature, that is often used inside a window, is average energy or power. Energy, E of a signal $\mathbf{x}_n$, can be defined as the magnitude, $|.|$ over a finite interval,



(a) Signal 1



(b) Signal 2

Figure 2.9: Probability distribution functions of signals in subsequent 10 second windows.

$-N \leq n \leq N$:

$$E_N = \sum_{n=-N}^{N} |\mathbf{x}_n| \tag{2.14}$$

Instead of $|\mathbf{x}_n|$, it is also common to replace with $|\mathbf{x}_n|^2$. Another alternative statistic is the sample variance, s^2, of a signal. Sample variance computes how a signal deviates from the mean and can be defined as:

$$s^2 = \frac{1}{N-1} \sum_{n=0}^{N-1} (\mathbf{x}_n - \mu_{\mathbf{x}})^2 \tag{2.15}$$

where

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}_n \tag{2.16}$$

The square root of variance (*standard deviation*), is also commonly used (Varsavsky et al., 2011a). As short non-stationarities, such as transient bursts found in EEG, can affect the calculation of $\mu_{\mathbf{x}}$, *variability* is sometimes used instead of variance. Total variability, v_k, of a non-constant signal can be calculated using the number of times a signal changes polarity within a window:

$$v_k = \frac{1}{N-1} \frac{\sum_{n=k+1}^{k+N} |\mathbf{x}_n - \mathbf{x}_{n-1}|}{(\max_{n \in A_k} \mathbf{x} - \min_{n \in A_k} \mathbf{x})} \tag{2.17}$$

where $A_k = \{k+1, k+2, \ldots, k+N\}$. It is also worth noting that DC offsets can affect statistics such as variance and variation, therefore the signal should first be *detrended* or normalised to unit variance before windowing.

Further linear time domain statistics include inferring the synchronicity from the cross-correlation or mean phase coherence, and periodicity from autocorrelation, among others (see table 2.1).

Non-linear Signal Analysis

Most of the statistical approaches discussed so far are linear approaches, which have been applied within a window to categorise non-stationary and non-linear EEG signals. Often non-linear signals are treat as linear because linear signal processing tools are better understood and take less computation time. Furthermore, non-linear methods can be greatly impacted by noise; meaning noise reduction techniques, such as those previously discussed, need to be applied before their application (Diks, 1999). Nevertheless, often the true dynamics of a system are unknown, so these need be reconstructed from experimentally collected data. Takens/Aeyel's theorem (Takens, 1981; Aeyels, 1981) suggests the true dynamics of a system can be reconstructed by taking time delayed versions of the experimentally collected signal. Although this reconstruction does not result in the same appearance as the real data, properties of the signal remain the same (Varsavsky et al., 2011a).

Non-linear approaches generally have two stages, the first is to reconstruct the multidimensional dynamics of a true signal from a single dimensional recording (embedding), and the second is to extract features of a system; such as how complicated it is (dimension), its

predictability (Lyapunov exponents), or its randomness (entropy; Varsavsky et al., 2011a). Dimension is a measure of complexity in the dynamics of a multidimensional system or defined as the number of variables needed to describe a systems behaviour. The dimension of a non-linear system can be a non-integer in that, although the actual dimension is an integer, the coordinates required to specify most of a physical systems state may only be a small volume of the whole phase space. Different fractal dimensions can be used to gain this non-integer; specifically the information, capacity, and correlation dimensions (Parker and Chua, 2012; Varsavsky et al., 2011a). Another non-linear approach is to use Lyapunov exponents to describe the deterministic structure of a system by looking at how the system changes when a small change is introduced; with small Lyapunov exponents indicating predictable behaviour, and large indicating less predictability (Parker and Chua, 2012; Varsavsky et al., 2011a). Entropy can be used as a measure of the randomness, information, and compressibility of a system; with the more "random" or unstructured a variable, the larger the entropy value. Entropy quantifies the amount of disorder in a system, with more disorder indicating more information is transferred in a single measurement and therefore less efficiency in communicating this information. During the calculation of entropy, the signal can be coarsely or finely divided up into larger or smaller bins, with coarse graining having the advantage of being less susceptible to noise than the Lyapunov exponent or dimension measurements. Shannon entropy is commonly used for normalised EEG, which partitions the phase space into a number of bins and calculates the probability by counting the number of data points in each bin (Varsavsky et al., 2011a; Kiranyaz et al., 2014; Iešmantas and Alzbutas, 2020). As entropy measures the complexity or irregularity of biomedical signals, it can be applied to seizure detection as brain activity during a seizure is more predictable than normally, and can therefore be reflected reduction in the entropy value (Yuan et al., 2012; Omerhodzic et al., 2013; Päivinen et al., 2005; Hamdan et al., 2015).

However, there is a number of limitations applying non-linear statistical methods to EEG. Firstly, the window size commonly applied to EEG (20- to 30- seconds) is typically insufficient to reliably compute dimension, Lyapunov exponents, and entropy of the reconstructed EEG signal. Therefore approximations of these statistics are required, such as the effective correlation dimension and maximum Lyapunov exponent. Furthermore, as both di-

Table 2.1: A sample of common time-domain features used for seizure detection.

Feature	Authors
Approximate entropy	Chen et al. (2014) Mitha et al. (2014) Kiranyaz et al. (2014) Awan et al. (2016) Orellana and Cerqueira (2016)
Average Energy	Kiranyaz et al. (2014) Pramod et al. (2014) Zabihi et al. (2013) Shanir et al. (2015) Fergus et al. (2015) Khan and Khan (2017) Yuan et al. (2018b)
Coefficient of Variation	Mitha et al. (2014) Kiranyaz et al. (2014) Harpale and Bairagi (2018)
Interquartile Range	Rafiuddin et al. (2011) Pramod et al. (2014) Paulose and Bedeeuzzaman (2014) Awan et al. (2016) Chandel et al. (2016) Chandel et al. (2017)
Kurtosis	Mitha et al. (2014) Kiranyaz et al. (2014) Fergus et al. (2015) Elmahdy et al. (2015) Hamdan et al. (2015) Awan et al. (2016) Fergus et al. (2016) Tsiouris et al. (2017) Harpale and Bairagi (2018) Ammar et al. (2018)
Mean	Mitha et al. (2014) Kiranyaz et al. (2014) Elmahdy et al. (2015) Hamdan et al. (2015) Awan et al. (2016) Tsiouris et al. (2017) Harpale and Bairagi (2018)
Median Absolute Deviation	Rafiuddin et al. (2011) Pramod et al. (2014) Awan et al. (2016) Chandel et al. (2016) Chandel et al. (2017)
Root Mean Square	Pramod et al. (2014) Mitha et al. (2014) Kiranyaz et al. (2014) Fergus et al. (2015) Fergus et al. (2016) Harpale and Bairagi (2018)
Sample Entropy	Chen et al. (2014) Xiang et al. (2015) Fergus et al. (2015) Hamdan et al. (2015) Awan et al. (2016) Fergus et al. (2016) Zhu et al. (2017)
Shannon Entropy	Kiranyaz et al. (2014) Iešmantas and Alzbutas (2020)
Skewness	Mitha et al. (2014) Kiranyaz et al. (2014) Fergus et al. (2015) Elmahdy et al. (2015) Hamdan et al. (2015) Awan et al. (2016) Fergus et al. (2016) Tsiouris et al. (2017) Ammar et al. (2018)
Standard Deviation	Mitha et al. (2014) Ammar et al. (2018)
Variance	Kiranyaz et al. (2014) Elmahdy et al. (2015) Hamdan et al. (2015) Fergus et al. (2016) Tsiouris et al. (2017) Khan and Khan (2017)
Zero Crossings	Mitha et al. (2014) Kiranyaz et al. (2014) Zabihi et al. (2013) Tsiouris et al. (2017)

mension and Lyapunov exponents are highly sensitive to noise (Kantz and Schreiber, 2004), it is often only used with intra-cranial EEG, with entropy being used more extensively with scalp EEG. Moreover, it is unlikely EEG is truly able capture the dynamics of a highly complex non-linear system such as the brain, due to low spatial resolution. Despite papers that support a non-linear deterministic structure (e.g. Andrzejak et al., 2001; Casdagli et al., 1997; Li et al., 2003), there is also limited evidence that EEG has low dimensionality, especially with scalp recordings. This could be either because the dynamics are very complex, or the skull and scalp cause noise that affects the non-linear characteristics of the signal (Varsavsky et al., 2011a).

2.4.2 Frequency Domain Features

As we have previously discussed in subsection 2.3.1, as well as the time domain, a signal can be represented by its frequency and phase. We will now briefly revisit the Fourier Transform in relation to how it can be used to represent a signal for features in a machine learning model. Beforehand it is worth considering, in its application to EEG, that it is still unclear if "power" increases characterised by Fourier transforms reflect a change in the number of neurons synchronized or the strength of local synchronisation (Cohen, 2014). Although mathematically waveforms can be reconstructed by adding a set of sine waves that vary in amplitude, frequency, and phase (Fourier, 1878), this does not mean physiologically the waveform consists of sine waves oscillating at a particular frequency. The power measured by Fourier-based, or later discussed wavelet-based methods, is not evidence for physiological oscillations per say, as these methods will always give power at frequencies given a signal. Nevertheless, a "true" oscillation by the brain would be represented by these methods, as would a artefact generated by various external generators (Luck, 2014b). Indeed, spectral properties of an EEG not only depend on an individual's state (awake/asleep) and the cognitive tasks being conducted (Cranstoun et al., 2002), but also on factors such as individual differences of brain structure, age, working memory capacity, and brain chemistry (Cohen, 2014), as well as the positioning and referencing of the electrodes.

Frequency Features for EEG

Commonly, frequency features for windowed EEG are based on the *power spectral density* (PSD), which represents the contribution of power in each frequency component of a signal. Typically PSD is calculated using a discrete-time Fourier transform, which for a *windowed* signal $\mathbf{x}_n$ for $n = k+1, k+1 \cdots k+N$ is similar to equation 2.10:

$$c_{\omega,k} = \sum_{n=1}^{N} \mathbf{x}_{n+k} e^{-j2\pi\omega n/N}. \tag{2.18}$$

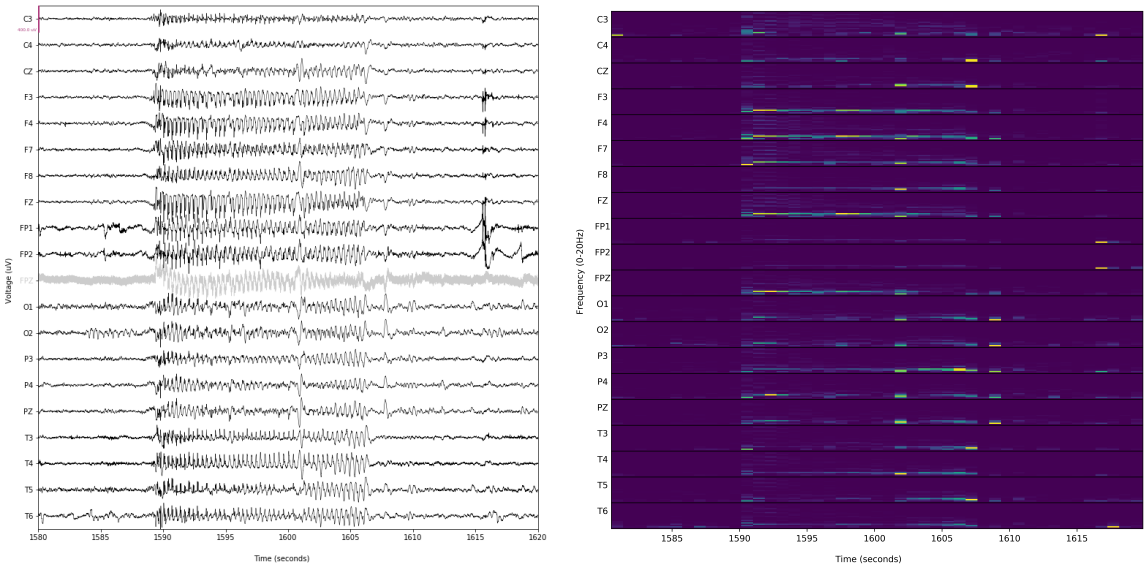
However, an assumption of the Fourier transform is that data is stationary, as violations of stationarity decrease the power in frequencies produced by the Fourier transform. This results in less well defined peaks in a spectrogram and power in other frequencies beside those defined in simulated data. To prevent edge artefacts, as would be present equation 2.18, a taper is typically applied; with popular choices including Hann, Hamming, and Gaussian windows (Cohen, 2014). The short-time FFT also often has overlaps between time segments to improve temporal precision, reduce loss of signal from tapering, and smooth the time-frequency plots. The *Welch* (Welch, 1967) method is one such non-parametric method of power spectrum estimation where periodograms are allowed to overlap (Bartlett and Medhi, 1955). Periodograms are formed for sequential blocks, and averaged over time to gain an estimate of the PSD. To make a more representative PSD, it can be applied across analysis windows and an average taken:

$$P_{\omega,k} = \frac{1}{N} \sum_{n=0}^{N-1} |c_{\omega,k}|^2 \tag{2.19}$$

Increasing N in the above equation has the effect of having better characterisation of the signal but at a worse frequency resolution. As PSD uses averaging, it cannot isolate in time where particular spikes in frequency occur, indeed these are smoothed out, thus frequency resolution is gained at the expense of temporal resolution. A common method to visualise the temporal evolution of frequency content is to use a spectrogram, which calculates sequential PSDs with small analysis windows (see figure 2.10). For EEG, PSDs can be used to reflect the locally stationary properties within a window, as well as time-evolving changes across

windows.

PSD is often used as a basis for calculating various basic statistics (see table 2.2). For example, a measure of peak frequency is common, and merely reflects the frequency of the highest peak in the PSD for a given window. Similarly, median frequency is just the midpoint in the frequency power spectrum (Fergus et al., 2015). However, a limitation of gaining features from a windowed Fourier analysis is that power within the entire window is treat as if it was at the centre of the window. Furthermore, the same window size is used to calculate power in different frequencies despite low and high frequencies yielding greater precision with different window sizes (Luck, 2014c). Indeed, the width of the window can result in poor frequency resolution if too narrow, or poor time localisation that violates the stationarity assumption if too wide (Rosso et al., 2006; Varsavsky et al., 2011a). A multitaper approach for short-time FFT is available, where several tapers with different temporal characteristics are applied (Mitra and Pesaran, 1999; Thomson, 1982); however this is useful mostly for high frequencies, due to the potential to impede frequency isolation from lower frequencies (Cohen, 2014). Additionally, the smooth functions used by windowed Fourier functions have been argued, using the Balian-Low theorem (Benedetto et al., 1994), to not be able to provide the smallest amount of information needed (orthogonal) whilst



(a) Signals

(b) Spectogram

Figure 2.10: The spectrogram of an EEG sequence containing a seizure.

Table 2.2: A sample of common frequency-domain features used for seizure detection.

Feature	Authors
Maximum	Kiranyaz et al. (2014) Zabihi et al. (2013) Ammar et al. (2018)
Median	Kiranyaz et al. (2014) Zabihi et al. (2013) Fergus et al. (2015) Hamdan et al. (2015) Fergus et al. (2016)
Mel Frequency Cepstral Coefficients	Kiranyaz et al. (2014) Zabihi et al. (2013) Golmohammadi et al. (2019) Ramadhani et al. (2019)
Minimum	Kiranyaz et al. (2014) Zabihi et al. (2013) Shanir et al. (2015)
Peak Frequency	Fergus et al. (2015) Hamdan et al. (2015) Fergus et al. (2016)
Spectral Entropy	Mitha et al. (2014) Kiranyaz et al. (2014) Zabihi et al. (2013) Hamdan et al. (2015)

also being localised in time and frequency. This means the physical representation of the energy in an original time series can be lost using this method (Cranstoun et al., 2002).

2.4.3 Time-Frequency Features

Instead of just gaining frequency information, at the expense of temporal resolution, one can use a variety of time-frequency techniques designed to resolve both temporal and frequency content for non-stationary signals. Such techniques include the use of Gabor atoms and Wigner-Ville distributions, but a common approach for EEG is to use a wavelet transformation (WT). WT's have been deemed superior to the Fourier transformation in their application to EEG data analysis as, although they are computationally slower, they give more accurate results with data containing discontinuities and sharp spikes (Kiymik et al., 2005). Wavelets can be used to analyse time series with nonstationary power at different frequency bands (Sakkalis et al., 2006), shown to express discontinuities caused by recording apparatus (Akin and Kiymik, 2000), and are useful for identifying and removing artefacts (e.g. eye and muscle movements; Khatun et al., 2016; Mammone and Morabito, 2014; Olund et al., 2014). They have also been used to investigate a number of pathologies such as Autism (Bhat et al., 2014), Alzheimer disease (Sankari et al., 2012) and obsessive compulsive disorder (OCD; Hazarika et al., 1997).

Wavelet Transform

A wavelet, $\psi(x)$, is a function that (Rao and Bopardikar, 1998):

1. Integrates to zero: $\sum_{n=-\infty}^{\infty} \psi(x) = 0$,

2. Has finite power: $\sum_{n=-\infty}^{\infty} |\psi(x)|^2 < \infty$.

A basic wavelet has the properties of a, by which it is scaled, and b, by which it is shifted (or translated) across samples:

$$\psi_{a,b}(x) = 2^{-a/2}\psi(2^{-a}x - b), \qquad a, b \in \mathbb{Z}. \tag{2.20}$$

If $a = 1$ and $b = 0$ then $\psi_{ab}(x)$ is known as the *mother wavelet* (Varsavsky et al., 2011a). The parameter a represents different *scales* for which temporal and frequency content will be extracted at different *resolutions*. Small values of a give more detailed temporal information (or *temporal scaling*), and occupy higher frequencies, than large values. This illustrates the *uncertainty principle* that resolution can be high for either time or frequency. Furthermore, the range of frequency details the wavelet covers becomes smaller for larger values of a (*frequency scaling*; Varsavsky et al., 2011a).

A time-frequency representation of a signal can be gained from the convolution of wavelets of different frequencies with a signal (Cohen, 2014). A wavelet representation of a function, $f(x)$, can be expressed as:

$$f(x) = \sum_{a=-\infty}^{\infty} \sum_{b \in \mathbb{Z}} d_{a,b}\psi_{a,b}(x), \tag{2.21}$$

where $d_{a,b}$ are the wavelet *detail* coefficients. These detail coefficients give us the contributions at each scale-location pair. The most common way of calculating d is through a *discrete wavelet transform* (see figure 2.A.2). This is an orthogonal transformation, as there is no overlap of frequency content at different scales, and can be achieved by restricting the dilations and translations of the mother wavelet (*dyadic sampling*). An example of this is the Haar wavelet family (see figure 2.11). $d_{a,b}$ can be iteratively computed using the coefficients from a higher scale, that can also be thought of as filters, that describe the wavelet family (Varsavsky et al., 2011a).

Other wavelet transforms exist, of note is the *stationary* wavelet transform (or *undecimated* wavelet transform, UDWT; Holschneider et al., 1989) and wavelet packet decom-

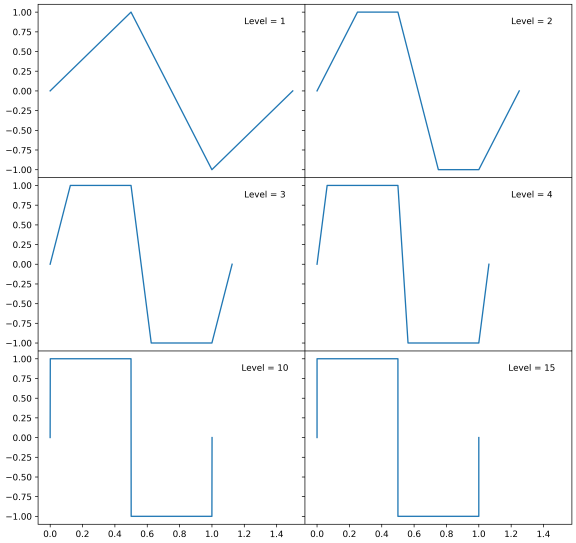


Figure 2.11: Piecewise linear approximations of the Haar wavelet family at various scales. *Note.* Obtained using the cascade algorithm (see, The MathWorks Inc., 2020).

position (WPD; Wickerhauser, 1996). UDWT aims to address the problem of translation-invariance in DWT, which means the transform coefficients and data are not shifted along by the same integer amount due to the decimated step. For DWT, an odd or even decimation can be made (e.g. $a = 2^A$), whereas UDWT use both odd and even transformations at each scale (see figure 2.A.3). UDWT is a more computationally intensive method than DWT, but can result in better discrimination between noise and activity, as well as more precise frequency localization. However, UDWT's are not orthogonal (Gyaourova et al., 2002), and frequencies close to each other may provide similar, if not identical, results due to frequency smoothing; meaning more frequency bins may unnecessarily increase computation time without increasing information (Cohen, 2014). WPD differs from the previous methods as, rather than only passing wavelet approximation coefficients through low- and high-pass filters at each level, both the detail and approximation coefficients are passed to create a binary tree (see figure 2.A.4). Due to the over-complete signal representation provided by this method, a number of algorithms have been proposed to prune the tree to a more sparse representation based on cost functions; such as entropy (Coifman and Wickerhauser, 1992) or thresholding based on the energy values before further decomposition at each node (Mojsilovic et al., 1997). WPD has been suggested to have better frequency resolution than DWT (Alakus and Turkoglu, 2018), particularly in higher frequencies, due

to decomposing detail coefficients to gain information that would otherwise be lost (Yang et al., 2006).

Other Time-Frequency Methods

There are a number of other time-frequency methods that appear in the seizure detection literature that we will briefly mention. For example, empirical mode decomposition (EMD; Huang et al., 1996) is a nonlinear time–frequency technique which breaks up time series signals into independent groups of functions or components called intrinsic mode functions (IMFs Jaber et al., 2014). EMD differs from Fourier and wavelet domains as it is adaptive, rather than having a prior fixed basis (Yash, 2018); meaning it is not assumed that the components of a signal are fixed in frequency over time and therefore can be generated from a dynamic signal generator. EMD provides empirically derived frequencies, useful for identifying changes in instantaneous frequency in non-stationary data. As such, this method has been applied, among other things, to detecting epileptic spikes (Oweis and Abdulhay, 2011). However, interpreting EMD components can be difficult, with oscillations of interest in multiple IMFs (Cole, 2016), and changes in hyperparameter values can result in significantly different results (Cole and Voytek, 2019). Furthermore, IMFs require the number of extrema and the number of zero-crossings to be equal or differ at most by one across the whole dataset (Cole, 2016), which is unlikely to occur in EEG data.

Other time-frequency decomposition methods of note include autoregressive modelling, matching pursuit, and the p-episode. However, these are not further discussed as autoregressive modelling has largely been replaced by wavelet convolution (Cohen, 2014), matching pursuit has a large convergence time (Pati et al., 1993) and, due to different sets of atoms used in the decomposition of each signal, phase across frequencies in simultaneously recorded signals is difficult to compare (Subhash Chandran et al., 2016), and the p-episode is influenced by a manually chosen threshold and is not appropriate for noisy data (Caplan et al., 2001; Montez et al., 2009; van Vugt et al., 2007; Cohen, 2014).

Time-Frequency Features for EEG

Among several families of mother wavelets, cubic spline functions have previously been recommended as mother wavelet for natural signals, due to their symmetry, smoothing, and numerical properties (Unser and Aldroubi, 1996). However, the Daubechies wavelet, typically of order 4 (db4; see figure 2.12), is the most commonly used wavelet for EEG. Db4 smooths the frequency filtering enough to characterise the EEG well, but is also computationally efficient (Kjaer et al., 2017; Subasi, 2007b). Nevertheless, other wavelet families such as the Coiflet (Gandhi et al., 2011; Uyulan and Erguzel, 2016) and Symlets families (Al-Qazzaz et al., 2015; Akkar and Ali Jasim, 2017) have been demonstrated to be optimal for classification and denoising EEG in certain applications.

DWT is currently the most commonly applied wavelet transformation method for extracting wavelet coefficients for EEG; likely due to its computational efficiency and simplicity of application compared to other WT methods. DWT has been used in a broad range of EEG research from emotion recognition (e.g. Jenke et al., 2014), to Alzheimer's Disease (e.g. Ghorbanian et al., 2013), and seizure detection (e.g. Ocak, 2009). However, as previously noted, UDWTs have potential applications for noise elimination. Mamun et al. (2013), for example, found different UDWT's were best for noise elimination in healthy and epileptic subjects. Specifically the db8 wavelet function was more useful for healthy subjects and the orthogonal Meyer wavelet function for epileptic subjects. This can be particularly useful in assessing epileptic seizures which have high amplitude muscle and physiological artefacts (e.g. tonic-clonic), which can make them difficult to analyse visually (Rosso et al., 2004). Wavelet packets have also been applied to a number of clinical diagnosis domains (e.g. Zhang et al., 2015; Bhat et al., 2018), including seizure detection (Alakus and Turkoglu, 2018; Raghu et al., 2017). Alickovic et al. (2018) compared EMD, DWT, and WPD for automated epileptic seizure detection and prediction using multiple classification models. They found models using features from EMD had the poorest performance, with the authors suggesting WPD as a feature extractor for random forest or support vector machine classifiers. However, most authors have only applied this technique on small intracranial epilepsy datasets (e.g. the EEG database from the University of Bonn; Andrzejak et al.,

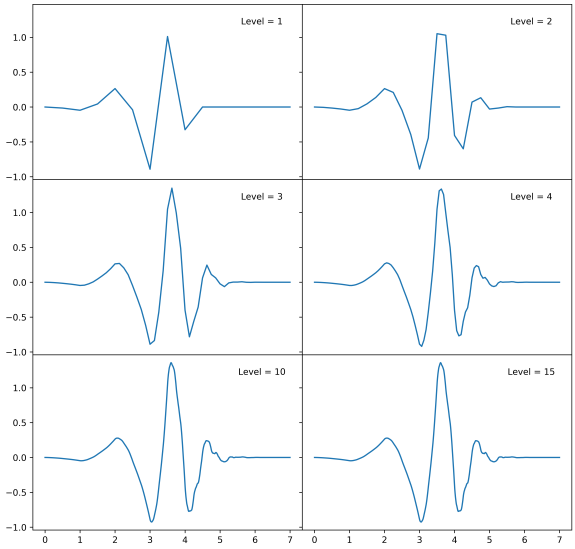


Figure 2.12: Piecewise linear approximations of the Daubechies 4 wavelet family at various scales.
Note. Obtained using the cascade algorithm (see, The MathWorks Inc., 2020).

2001). Furthermore, often all the wavelet scales from a WPD are used as features, but this is inefficient without the use of approaches to gain a selection of the subband trees (Huang and Aviyente, 2006; Raja and Gangatharan, 2015). Therefore, more work is required to observe its application to larger, extracranial EEG datasets, as well as applying dimensionality reduction techniques.

Similar to frequency features, it is uncommon for the wavelet coefficients to be used directly as features. Wavelet filters isolate spectral information in different frequency ranges similar to the previously described PSD. As such, the features computed from the wavelet coefficients are often similar to those computed on frequency methods (see table 2.3).

There are several limitations to time-frequency power analyses in general, with some also applying to other neuroimaging methods such as ERP's and fMRI. The model parameters selected will affect the outcome of the analyses, such as the decisions around temporal and frequency precision, shape of the wavelet or band-pass filter, and baseline time period and normalization. Exploratory data analyses are often required due to problems with time-frequency analyses, with multiple comparisons requiring conservative statistical corrections that cannot identity subtle effects. Finally, EEG time-frequency features may not represent neural oscillations, however could reflect a manifest variable rather than the latent variable

Table 2.3: A sample of common time-frequency domain features used for seizure detection.

Feature	Authors
Coefficient of Variation	Rafiuddin et al. (2011) Khan et al. (2012) Kiranyaz et al. (2014) Zabihi et al. (2013) Hussain (2018)
Energy	Bugeja et al. (2016) Awan et al. (2016) Mitha et al. (2014) Kiranyaz et al. (2014) Rafiuddin et al. (2011) Pramod et al. (2014) Chen et al. (2014) Kiranyaz et al. (2014) Mitha et al. (2014) Zabihi et al. (2013) Orosco et al. (2016) Tsiouris et al. (2017) Chandel et al. (2016) Chandel et al. (2017) Kaleem et al. (2018) Hussain (2018)
Entropy	Pramod et al. (2014) Chandel et al. (2016)
Maximum	Pramod et al. (2014) Kiranyaz et al. (2014) Zabihi et al. (2013) Orosco et al. (2016)
Mean	Pramod et al. (2014) Kiranyaz et al. (2014) Zabihi et al. (2013) Javaid et al. (2015) Orosco et al. (2016) Chandel et al. (2016) Chandel et al. (2017) Alickovic et al. (2018) Harpale and Bairagi (2018) Selvathi and Meera (2018)
Minimum	Pramod et al. (2014) Kiranyaz et al. (2014) Zabihi et al. (2013)
Relative Scale Energy	Kiranyaz et al. (2014) Zabihi et al. (2013)
Shannon Entropy	Kiranyaz et al. (2014) Zabihi et al. (2013) Ibrahim and Majzoub (2017) Sopic et al. (2018) Perera et al. (2017)
Standard Deviation	Pramod et al. (2014) Zabihi et al. (2013) Javaid et al. (2015) Ibrahim and Majzoub (2017) Harpale and Bairagi (2018)
Variance	Kiranyaz et al. (2014) Das et al. (2016) Perera et al. (2017) Selvathi and Meera (2018) Chandel et al. (2019)
Power	Javaid et al. (2015) Perera et al. (2017)
Kurtosis	Alickovic et al. (2018) Chandel et al. (2019)

of a neural oscillation. However, this latter criticism does not mean that time-frequency analyses is not useful. Indeed, Cohen (2014) suggests this uncertainty means that findings should be described as "band-limited" or "frequency-band-specific" to be more conservative when describing a finding, and reserving the use of "neural oscillation" as an interpretation or speculation around the results.

2.5 Dimensionality Reduction

Once features have been created, in order to reduce a models complexity, run time, and potential for over-fitting to the training data, dimension reduction techniques are often applied. Broadly they can be grouped into methods that create a subset of the original set of features (Feature Selection) or methods that create new synthetic features through combining the original features and discarding less important ones (Feature Extraction).

2.5.1 Feature Selection

A simple method for feature selection could be to impose a sparsity constraint when training a classifier to ensure a model favours fewer features. However, one could also use model stacking, where the input to one model is the output of another. This allows for non-linearities to be captured in the first more complex model, and the subsequent use of an efficient linear model as the last layer. Deep learning is an example of model stacking, as often neural networks are layered on top of one another to optimise both the features and the classifier simultaneously (Zheng and Casari, 2018). Another example of model stacking is to use the output of a decision tree–type classifier as input to a linear classifier. As decision trees (see subsection 2.6.1) rank the importance for each feature on the model, you can use these importance values to reduce the features down to features that contribute most to assigning a class membership (see figures 4.A.4 & 4.A.5). However, if features are highly correlated, which is often the case in EEG (e.g. figures 3.A.7 & 3.A.8), one feature may be ranked highly while the information of the others not fully captured (Raschka and Mirjalili, 2019). Nevertheless, this method has been used by Birjandtalab et al. (2017) for finding the best few EEG channels for seizure detection using the same spectral feature set in each channel. They suggest identifying the best channels for seizure detection may enable limited-channel EEG, which has a faster run time, lower power consumption, and increased accuracy by avoiding non-focal/unnecessary channels.

2.5.2 Feature Extraction

As some features may be highly correlated to others, PCA (introduced in subsection 2.3.2) can be used to compresses them into an lower dimensional subspace. As PCA is sensitive to data scales, components need to be standardized before applying PCA, because equal importance should be given to all features despite being measured on different scales. However, when used for dimensionality reduction, the number principal components must be set; which can be based on a trade-off between computational efficiency and classifier performance (Raschka and Mirjalili, 2019), or the use of a threshold that accounts for a desired proportion of total variance (Zheng and Casari, 2018).

Another method for reducing non-linear data with high-dimensionality down to a lower-dimensional subspace is t-Distributed Stochastic Neighbor Embedding (t-SNE), commonly used for data visualisation. It has been applied to seizure detection by Birjandtalab et al. (2017) as a feature extraction technique, however t-SNE is not intended primarily as a pre-processor for models as it fits clusters onto the training data which are difficult to apply to a separate test set. A better alternative is the use of Uniform Manifold Approximation and Projection (UMAP; McInnes et al., 2018), a supervised and unsupervised dimension reduction technique that can be used for non-linear dimension reduction (McInnes and Healy, 2018). UMAP is intended as a replacement for t-SNE as it has better runtime performance and often preserves more global structure. As well as unsupervised dimension reduction (e.g. PCA or t-SNE), class labels can also be used; as well as a combined approach using labelled data embedding with new unlabelled points added after. However when used for clustering data, UMAP and t-SNE do not completely preserve density and can result in a finer clustering than is present in the data, and are not as interpretable as PCA (McInnes et al., 2018). Still, it is an interesting feature extraction method which to date that has not been applied to EEG seizure detection.

2.6 Classification

Once approaches have been taken to transform a signal into relevant descriptors, classifiers can be run on the data to detect signal differences and separate signals into different classes.

Classifiers can separate pre-determined classes under the assumption that the presented data belongs to one of the classes. A classifier can simply impose a threshold on features, or employ more complex methods, such as using machine learning algorithms, which require training and subsequent testing on unseen data (Varsavsky et al., 2011a). However, it is worth noting that machines are not literally "learning", as one would traditionally use the word, as they are in fact finding a mathematical formula that produces desired outputs based on inputs. These formulas can then be used on new data, provided they have a similar statistical distribution to the data the model was trained on (Burkov, 2019).

Machine learning algorithms can be broadly categorised by the level of *supervision* required for learning (see figure 2.13). Supervised learning aims to determine how to map labels to data using training examples. This type of learning is a classification task, with "supervised" referring to where the desired output labels are already known (Raschka and Mirjalili, 2019). The aim of supervised learning is to use a dataset, $(\mathbf{x}_i, y_i)_{i=1}^{N}$, where $\mathbf{x}_i$ is the D-dimensional real-valued feature vector of example $i = 1, \cdots, N$, and y_i is a real-valued label (or target). Its worth noting that the output, y_i, can be more complex, such as a vector, matrix, tree, or graph, but is most commonly a categorical label, a number that can be used to deduce a label, or a real valued continuous label (regression; Burkov, 2019). A simple algorithm for categorical labelling may be trained to distinguish between two cases; such as the case of spam email detection, where mail is sorted into spam or not spam. Most machine learning algorithms are able to do such *binary classification* and, if not naturally multi-class, can be extended to multi-class classification (e.g. gmail categories: Social, Promotions, Updates) using techniques such as the One-versus-Rest or One-versus-One methods (Raschka and Mirjalili, 2019). In these approaches, after all models in the *ensemble* have been trained, every model outputs prediction probabilities based on their inputs and the labels from the most certain model is chosen (Burkov, 2019).

Supervised classification models can be further sub-categorised by whether they are generative or discriminative algorithms. Generative algorithms, such as Naive Bayes', hidden Markov, and Gaussian mixture models, learn the joint probability of data instances and their labels. Conversely, discriminative algorithms (e.g. Logistic Regression and Support Vector Machines) model the boundaries separating labels (Mohr et al., 2017). Additionally

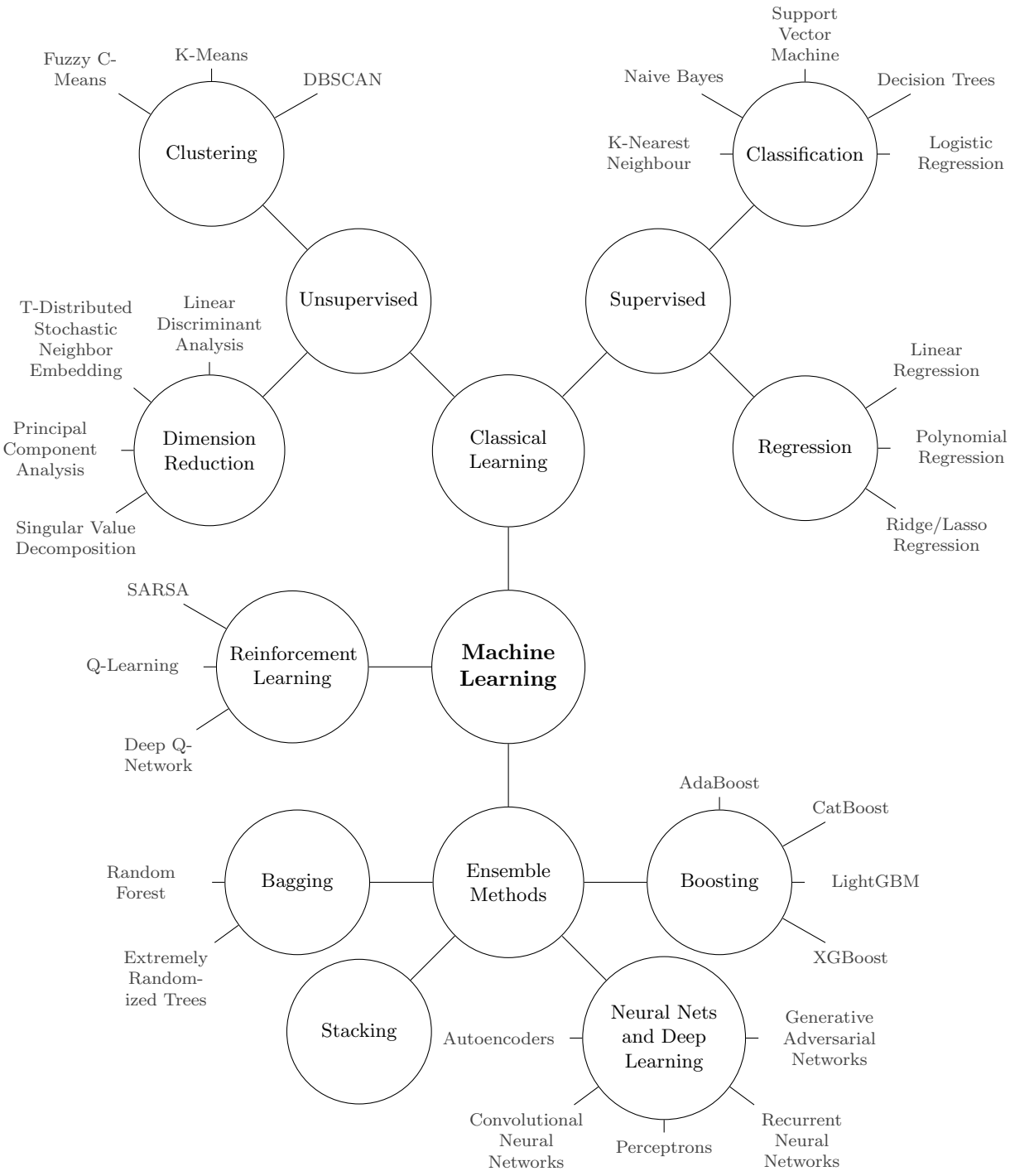


Figure 2.13: Categorisation for a sample of machine learning methods.

instead of learning internal parameters from training data which can be later discarded, labels for new data can be assigned by directly comparing them to data points in the training data (e.g. k-Nearest Neighbours; Burkov, 2019).

Other types of learning include unsupervised and semi-supervised learning. Unsupervised methods aim to find hidden structures within data as labels are not provided (Mohr et al., 2017). Unsupervised methods can be grouped into clustering algorithms (e.g. K-means and hierarchical clustering), which cluster similar data together, anomaly detection (e.g. one-class support vector machines), which identify instances different from the majority of the data, and the previously discussed dimensionality reduction algorithms (e.g. principle component analysis), which remove multicollinearity and retain important information by creating new synthetic features. Semi-supervised methods use both labelled and unlabelled data as training samples, and are practical for large scale data where there is a higher ratio of unlabelled to labelled data (Mohr et al., 2017). A type of semi-supervised learning is active learning, which ask users to provide a new label when data is generated that has not previously been classified. This allows users to update models by adding in additional labels, which could be used to make a model more personalized, even if it does incur a labelling burden (Mohr et al., 2017; Settles, 2010). In their general application, although it may seem detrimental to add uncertainly to the model, unlabelled data does allow for better information of the probability distribution the data is drawn from, which can be leveraged by models (Burkov, 2019).

2.6.1 Classical Methods

Classical methods are categorised typically in relation to ensemble or neural network/deep learning models. They have a background in statistics, rather than computing, and were used for solving mathematical problems such as finding similarities in data points and searching for patterns. We have already discussed some unsupervised classical learning methods, specifically regarding dimension reduction (see section 2.5), so the rest of this subsection will focus on supervised classical methods; Linear Regression, Logistic Regression, Support Vector Machines, k-Nearest Neighbours, and Decision Trees. These are by no means obsolete comparative to modern ensemble methods, and for simple classification problems where

explainable outcomes are required, these are often preferable.

Linear Regression

Discriminative *classification* algorithms are typically used for pre-processing or to group EEG data based on its content (e.g. inter-ictal, ictal). However, basic understanding of linear *regression*, which provides continuous outcomes rather than categorical labels, can be useful for supporting understanding of the subsequently discussed discriminative models.

In linear regression, the distance between explanatory variables ($\mathbf{x}_i$), of which there can be multiple, ($x_{i,j}$), and a model/prediction is minimised to gain a real-valued target/response (y_i). The model is parametrized using a linear combination of features (Raschka and Mirjalili, 2019):

$$y_i = w_0 x_{i,0} + w_1 x_{i,1} + \cdots + w_D x_{i,D} + \epsilon_i = \sum_{j=1}^{D} w_j x_{i,j} + \epsilon_i = \mathbf{w}^T \mathbf{x}_i + \epsilon_i. \tag{2.22}$$

Here, $\mathbf{w}$ is a D-dimensional vector of parameters, with w_0 as the y-axis intercept (sometimes denoted by b) and $x_{i,0}$ is a bias unit equal to 1. This model can be used to predict an unknown label $\hat{y}_i$ for a new $\mathbf{x}_i$ once the optimal values, $\mathbf{w}^*$, have been found. To find optimal values of $\mathbf{w}$, an objective function is maximised or minimised, such as the mean squared error:

$$\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{2.23}$$

where $\hat{y}_i = \mathbf{w}^* \mathbf{x}_i$.

As the optimisation criteria above is convex, meaning it has only one global minimum, optimisation algorithms can be used to find this global minimum. Generally, this is achieved by computing a prediction error for each instance, multiplying this by the feature values, and then taking the average over all training instances (Géron, 2019). However, for large or high dimensional datasets, this approach is not always feasible due to the large computational cost. Therefore, often approximate optimizations are used by training the algorithm in batches of data (e.g. Batch Gradient Descent, Mini-batch Gradient Descent), or an instance at a time (Stochastic Gradient Descent; Géron, 2019). Gradient descent finds a local

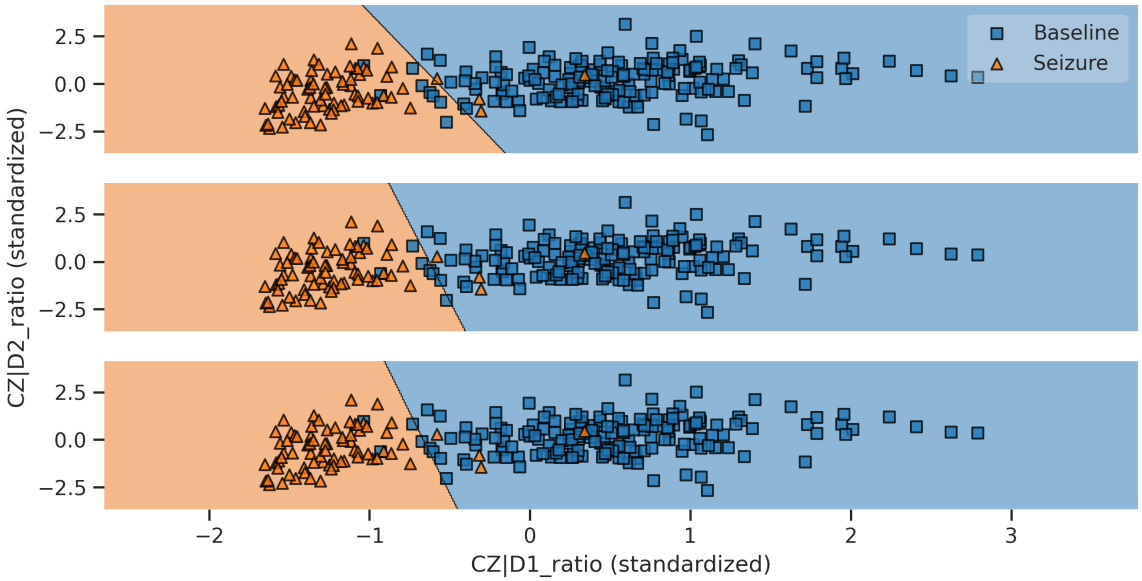


Figure 2.14: Changes to decision boundaries according to regulation.
Data Source: The Epileptologie Database (Andrzejak et al., 2001)

minimum of a function by starting at a random point and taking steps down in proportion to the slope (negative) of the gradient at its current point (Burkov, 2019). Each parameter has a partial derivative calculated, proceeding in epochs (entire runs of the provided training data), in order to update the parameters. At each epoch, $\mathbf{w}$ and σ are updated using the partial derivatives in respect to the learning rate, η, subtracting the derivatives from the parameter values until convergence is reached (change is minimal):

$$\mathbf{w} \leftarrow \mathbf{w} - \eta(y_i - \sigma(\mathbf{w}^T \mathbf{x}_i))\mathbf{x}_i. \tag{2.24}$$

Improvements on gradient descent means it is more common to use algorithms such as RMSprop and Adam, which are variants of stochastic gradient descent (Burkov, 2019) and are commonly used with modern deep learning methods (see subsubsection 2.6.2).

When optimising the model, it is commonly regularised to build a less complex model in an attempt to prevent overfitting. Overfitting means that the model can predict the training data well, but poorly predicts unseen data. This is also refereed to as "high variance", to reflect the lack of consistency (or variability) in model predictions which would change if the model was retrained on different subsets of the training dataset (Raschka and Mirjalili,

2019). This can be due to either the model being too complex, or the model training with too many features on a small set of data (Burkov, 2019). L1 (Equation 2.25) and L2 (Equation 2.26) regularisation are the most common methods for dealing with overfitting:

$$L1 : \lambda||\mathbf{w}|| = \lambda \sum_{j=1}^{D} |w_j|, \tag{2.25}$$

$$L2 : \lambda||\mathbf{w}||^2 = \lambda \sum_{j=1}^{D} w_j^2. \tag{2.26}$$

Both methods add a penalising term to the objective function, producing higher values when the models are more complex. The regularization parameter λ is used to control the regularisation, with the higher the value of λ, the stronger the regularization. In practice L1 mostly performs feature selection and L2 maximises performance and is better to use in combination with optimisation algorithms.

Logistic Regression

Logistic regression is a common discriminative algorithm used for the classification of a dependent variable which has a limited number of possible values, so is more applicable to seizure classification. Logistic regression is similar to linear regression, in that it computes a weighted sum of input features with a bias term, but instead outputs the logistic (S shaped sigmoid function) of the result to model the probability of class membership (Raschka and Mirjalili, 2019):

$$\phi(z_i) = \frac{1}{1 + e^{-z_i}}, \tag{2.27}$$

where z is the linear combination of weights and inputs, similar to equation 2.22. The output of the sigmoid can be interpreted as the probability of an example belonging to class 1, $\phi(z_i) = P(y = 1|\mathbf{x}_i; \mathbf{w})$. A binary outcome is then typically gained by using a threshold:

$$\hat{y}_i = \begin{cases} 1, & \text{if } \phi(z_i) \geq 0.5 \\ 0, & \text{otherwise.} \end{cases} \tag{2.28}$$

In order to learn the weights, $\mathbf{w}$, we can use an optimisation algorithm (such as gradient descent) to minimise a log-likelihood function (as a cost function J):

$$J(\mathbf{w}) = \sum_{i=1}^{N} \left[-y_i \log\left(\phi(z_i)\right) - (1 - y_i) \log\left(1 - \phi(z_i)\right) \right]. \tag{2.29}$$

Similar to linear regression, for large datasets this optimisation can take place using gradient descent or other optimisation algorithms. As logistic regression produces a decision boundary to separate data, this can be adapted by changing the strength of the regularisation on the model (Géron, 2019). As well as using one-versus-all and one-versus-one strategies, logistic regression can also be extended to separate multiple classes using softmax regression or multinomial logistic regression. For example, in softmax regression each instance has a score computed for each class, with the probability estimated by applying the softmax function; which computes the normalised exponential of the scores:

$$\zeta(\mathbf{z}) \stackrel{\text{def}}{=} \left[\zeta^1, ..., \zeta^N\right], \text{ where } \zeta^i \stackrel{\text{def}}{=} \frac{\exp(z_i)}{\sum_{s=1}^{N} \exp(z_s)}. \tag{2.30}$$

This allows for class membership to be predicted based on the class with the highest estimated probability. Training such a model means minimising the cross entropy cost, as this penalises low target probabilities, so measures how well estimated class probabilities match a target class (Géron, 2019).

Support Vector Machine (SVM)

A SVM is another common discriminative algorithm which distinguishes classes of objects by finding a hyperplane that provides the maximum margin of separation for data points belonging to different classes. Each feature vector is represented as a point in high-dimensional space, the size of which is the number of features in the dataset. For example, a dataset with 2 features could be plotted in 2 dimensional space (using an x- and y-axis), with the dimensions increasing as the number of features increase (24 features as 24-dimensional space). Imaginary hyperplanes (lines) are drawn to separate the classes 1 dimension less than the space (e.g. 1D line in 2D space), these being parallel to the decision boundary

which separates the classes (Burkov, 2019). During training, the model attempts to find the optimal values $\mathbf{w}^*$ to separate the classes so that negative-class examples are on one side of the negative hyperplane, and all positive-class examples fall behind the positive hyperplane (Raschka and Mirjalili, 2019):

$$\mathbf{w}^T\mathbf{x}_i \geq 1 \quad \text{if} \quad y_i = 1, \tag{2.31}$$

$$\mathbf{w}^T\mathbf{x}_i \leq -1 \quad \text{if} \quad y_i = -1. \tag{2.32}$$

Also, the distance between the positive and negative hyperplanes (*margin*) needs to be maximised:

$$\frac{\mathbf{w}^T(\mathbf{x}_{\text{pos}} - \mathbf{x}_{\text{neg}})}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}; \tag{2.33}$$

although in practice, often the reciprocal term $\frac{1}{2}||\mathbf{w}||^2$ is minimised (Raschka and Mirjalili, 2019).

When optimising the model, a subset of training data, known as support vectors, are selected to compute the optimal separation hyperplane. If data can be linearly separated, then a *hard margin* of separation can be used; whereby a point on the edge of a class is used as the support vector for the decision boundary. These two points, which are the closest examples of the two separate classes, are used to provide the hyperplane that draws the largest separation (or margin) between them. This is so that when new data is provided from a similar distribution, the model has the highest chance of correctly identifying its class membership (generalisation). However, this method is sensitive to outliers, so a more flexible method may be preferable. Instead, a *soft margin* can be used to compute a hyperplane that still provides a maximum margin of separation, whilst allowing for some errors. This introduces a slack variable, ξ, to the linear constraints:

$$\mathbf{w}^T\mathbf{x}_i \geq 1 - \xi_i \quad \text{if} \quad y_i = 1, \tag{2.34}$$

$$\mathbf{w}^T\mathbf{x}_i \leq -1 + \xi_i \quad \text{if} \quad y_i = -1, \tag{2.35}$$

and objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C\left(\sum_i \xi_i\right). \tag{2.36}$$

The hyperparameter C determines the trade-off between increasing the size of the decision boundary and ensuring that $\mathbf{x}_{i,j}$ is on the correct side of the decision boundary; resulting in a trade-off between optimal separation of the training data and classification of future examples (Burkov, 2019). Furthermore, if classes cannot be linearly separated, the input feature space can be projected to higher dimensions using the kernel trick (e.g. radial basis kernel; Cover, 1965; Varsavsky et al., 2011a), where the data may be separable linearly (see figure 2.15). The kernel trick avoids actually doing the transformation of the vectors into higher dimensions and computing their dot product, by using the kernel function as a modified dot product that only works with the original lower dimensional space; relying on the fact that each coordinate of a transformed vector $\Phi(\mathbf{x})$ is a function of the coordinates of the lower dimensional vector $\mathbf{x}$ anyway. After "transformation", the data can then be mapped back into the original feature space to create a nonlinear separation boundary (Duun-Henriksen et al., 2012b). SVMs can therefore be used to model non-linear decision boundaries and are generally robust to overfitting in high-dimensional space. Comparative to other subsequently discussed models (e.g. Deep Learning), SVMs are faster to implement on small- to medium-sized datasets (Varsavsky et al., 2011a), and are not as effected by an

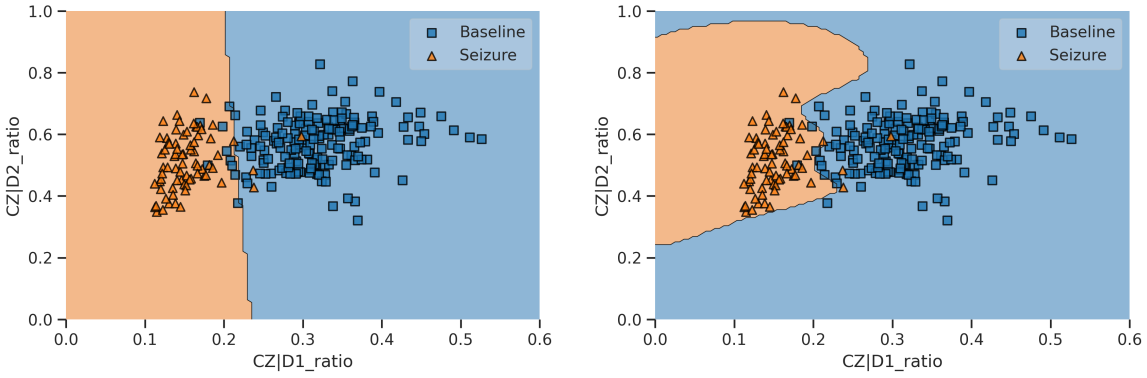


Figure 2.15: Linear and non-linear SVM decision boundaries on two features.
Data Source: The Epileptologie Database (Andrzejak et al., 2001)

over-representation of data in the training phase due to over-parameterization or over-fitting (Gonzalez-Vellon et al., 2003; Shoeb et al., 2004). Furthermore, SVMs always converge on the same answer given identical data. However, even though kernels do not actually transform data onto higher dimensions, they are still generally computationally expensive and do not scale well to larger datasets; linear kernels having a running time of $O(mn)$, and non-linear kernels running between $O(m^2n)$ and $O(m^3n)$.

Decision Tree (DT)

Decision trees, in their most basic form, effectively ask a series of questions in order to partition datapoints into nodes (bins). An algorithm starts at a tree root and then splits the data based on the features that gives the largest *information gain*. This splitting procedure occurs until all the samples within a given node all belong to the same class. A limit on nodes, or tree depth, is often set to avoid over-fitting due to a deep tree. To split using information gain relies on calculating the difference between an impurity measure of a parent node and the sum of the impurities of its child nodes; information gain being high when impurity of the child nodes is low. An optimisation function to maximise information gain, IG, at each split can be defined as (Raschka and Mirjalili, 2019):

$$\mathrm{IG}(V_p, f) = I(V_p) - \sum_{c=1}^{C} \frac{N_c}{N_p} I(V_c). \tag{2.37}$$

Here f is the feature used for the split, V_p and V_c are the parent and child dataset nodes, I is the impurity measure (see equation 2.39), N_p is the total number of training examples in the parent node, and N_c is the number of examples in the cth child node (Raschka and Mirjalili, 2019). Different algorithms can be used to define how trees are produced, such as Iterative Dichotomiser 3 (ID3; Quinlan, 1986), C4.5 (Quinlan, 2014), and Classification And Regression Tree (CART; Breiman et al., 1984). We choose here to focus on the CART algorithm as an optimised version is implemented into the popular Python package `Scikit-learn`, used in this thesis. `Scikit-learn`, similar to other libraries, reduces the search space by

implementing binary trees, meaning parent nodes always have two children:

$$\mathrm{IG}(V_p, f) = I(V_p) - \frac{N_{\text{left}}}{N_p} I(V_{\text{left}}) - \frac{N_{\text{right}}}{N_p} I(V_{\text{right}}). \tag{2.38}$$

The training set is split into two subsets using a threshold on a single feature by searching for the pair that produces the "purest" subset, based on size and minimisation of a cost function. Once split, it uses the same logic recursively until the maximum depth is reached or a split cannot be found that reduces impurity.

Three impurity measures that are commonly used in binary decision trees are Gini impurity, entropy, and classification error (Raschka and Mirjalili, 2019). For example, entropy, I_H, can be used to gain the proportion of the examples that belong to class l, $p(i|V_t)$, for a particular node, V_t:

$$I_h(V_t) = -\sum_{i=1}^{l} p(i|V_t) \log_2 p(i|V_t). \tag{2.39}$$

where $p(i|V_t) \neq 0$. This results in entropy being 0 if all examples at a node belong to the same class, and maximal if there is a uniform class distribution (Raschka and Mirjalili, 2019).

Due to their hierarchical structure, decision trees can easily model non-linear decision boundaries, and are generally robust to outliers and scalable to large datasets. However without regulating the tree depth, they are prone to overfitting to the data (see figure 2.16). Furthermore, decision trees tend to model their decision boundaries as orthogonal straight lines, meaning they are sensitive to the rotation of the dataset. This can be helped by the use of PCA to rotate the data before fitting the model (Géron, 2019). Nevertheless, due to the fact decision trees randomly (stochastically) choose features and are sensitive to small variations in the data, they can create very different models if data is removed or different random states are assigned (Géron, 2019).

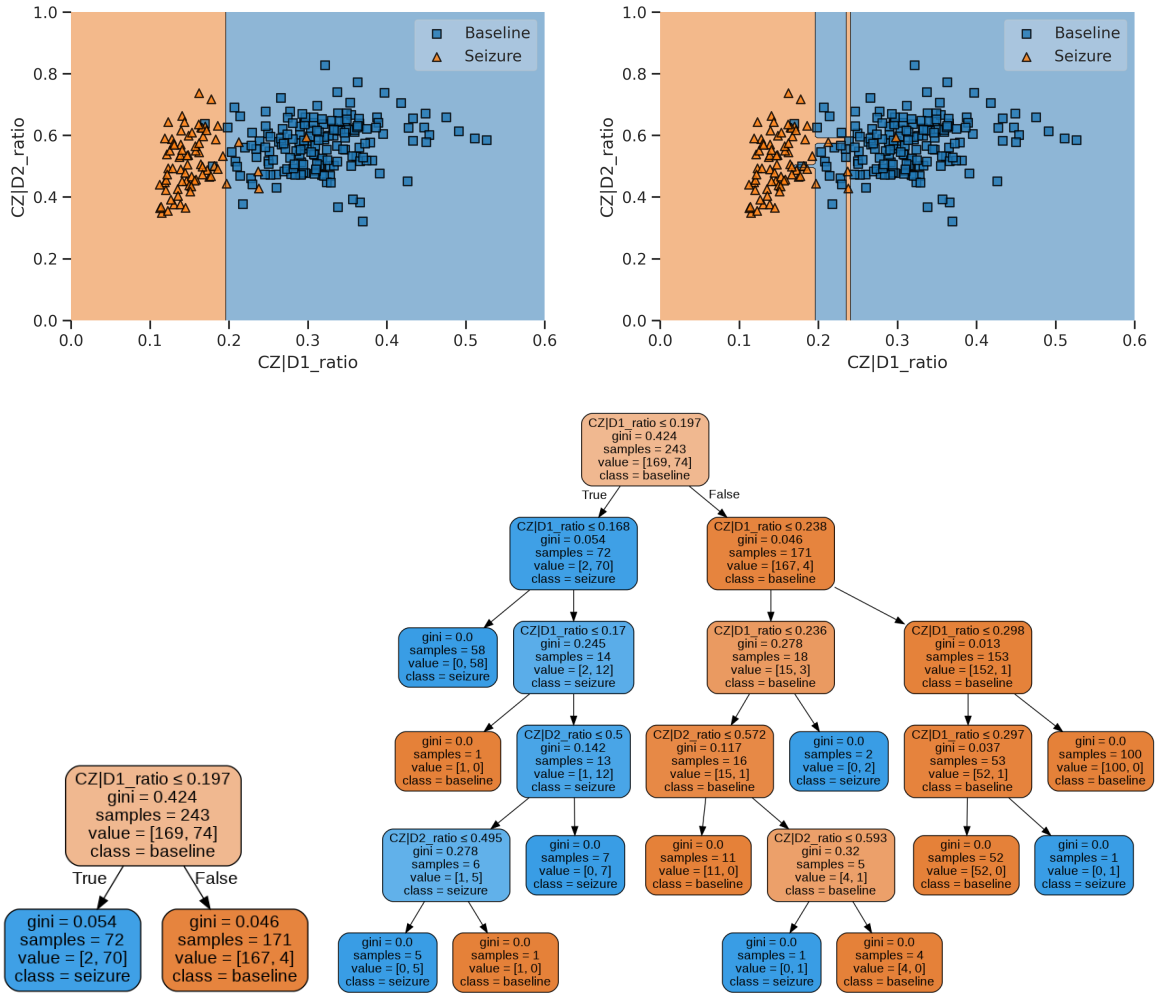


Figure 2.16: The effects on decision boundaries of decision tree splits when the maximum depth is set to 2 or there is no maximum.
Data Source: The Epileptologie Database (Andrzejak et al., 2001)

k-Nearest Neighbour (KNN)

KNN is different from the previously discussed algorithms as it does not learn a discriminative function from the training data, instead memorizing the training data directly (a lazy learner; Raschka and Mirjalili, 2019). KNN finds the k number of samples that are the most similar to a data point, $x_{i,j}$, to be classified (nearest neighbours), based on a given distance metric, and uses them to assign a class label using a majority vote (Raschka and Mirjalili, 2019). Multiple distance metrics are available, such as the Manhattan distance ($p = 1$) and Euclidean distance ($p = 2$):

$$d(\mathbf{x}_i, \mathbf{x}_k) = \sqrt[p]{\sum_{j=1}^{D} |x_{i,j} - x_{k,j}|^p}, \tag{2.40}$$

or cosine similarity:

$$s(\mathbf{x}_i, \mathbf{x}_k) \stackrel{\text{def}}{=} \cos(\angle(\mathbf{x}_i, \mathbf{x}_k)) = \frac{\sum_{j=1}^{D} x_{i,j} x_{k,j}}{\sqrt{\sum_{j=1}^{D} x_{i,j}^2} \sqrt{\sum_{j=1}^{D} x_{k,j}^2}}. \tag{2.41}$$

The classifier can be easily adapted as new data becomes available, however KNN is susceptible to overfitting due to the curse of dimensionality; where the feature space becomes more sparse as the number of dimensions of the feature space increases (Raschka and Mirjalili, 2019). Furthermore, classification complexity linearly increases with the number of data in the training set unless data structures such as k-d trees, a combination of decision trees and KNN, are used (Raschka and Mirjalili, 2019; Friedman et al., 1977).

Classical Methods for EEG Classification

Regression-based models are commonly used as a baseline model to compare algorithms to when classifying EEG data for its content (Jiang and Bian, 2019). For example, comparative to other machine learning models, logistic regression provides an efficient and interpretable model, as it outputs probabilities for each class which are easy to regularise. Logistic regression has been found to have comparable performance to other models for patient-independent seizure detection (Fergus et al., 2015; Samiee et al., 2017), and good sensitivity for patient-specific seizure detection but a high false positive rate (Supratak et al., 2014). However, it has a number of limitations in its application; including a tendency to overfit when data is not independent, as it can over-weight the significance of particular observations. This therefore means there is a burden of feature engineering to ensure features are not too correlated. Furthermore, logistic regression can only provide a linear decision boundary, limiting its use to linear problems, or when feature engineering has accounted for non-linearities. Often other, more complex models, have preferable performance comparative to logistic regression, but at greater computational cost; such as multilayer perceptrons (Alkan et al., 2005) and support vector machines (Zhang et al., 2018).

The most commonly applied classifier for seizure detection is currently a support vector machine (SVM), typically including features from a Daubechies 4 mother wavelet (e.g. Duun-Henriksen et al., 2012b; Adeli et al., 2003; Henriksen et al., 2010). Commonly a radial basis function kernel is used (e.g. Gu et al., 2018; Perera et al., 2017), providing greater performance than a linear hyperplane (Paulose and Bedeeuzzaman, 2014). However, a linear hyperplane may be preferable in particular situations where computational efficiency is important (Elmahdy et al., 2015; Selvathi and Meera, 2018); such as online limited channel classification (Petersen et al., 2011), or at scale in parallel cluster environments (Sendi et al., 2018). A feature selection/reduction step before a SVM classifier is common in the more recent literature; such as PCA (e.g. Yuan et al., 2019a; Selvakumari et al., 2019) or ICA (e.g. Wang et al., 2017; Ramadhani et al., 2019). However, although commonly used, some authors have found other classifiers perform better than SVMs when compared on the same dataset, features, and evaluation paradigm, although not exclusively (e.g. Javaid et al., 2015; Alickovic et al., 2018; Zhang et al., 2018; Kaleem et al., 2018). These classifiers include K-nearest neighbour (KNN; Fergus et al., 2015; Hamdan et al., 2015), Gaussian Mixture Model (GMM; Awan et al., 2016), AdaBoost (Amin and Kamboh, 2016), Logistic Regression (LR; Samiee et al., 2017), Random Forest (RF; Samiee et al., 2017; Wang et al., 2017), Extreme Learning Machine (ELM; Chen et al., 2014), Convolutional Neural Networks (CNN; Cao et al., 2017; Iešmantas and Alzbutas, 2020), and other deep learning variants (Yuan et al., 2019b).

KNN, applied to EEG commonly in the brain–computer interface (BCI) literature, has been successfully applied to seizure detection by a number of authors (e.g. Shanir et al., 2018; Chandel et al., 2019). The hyperparameter k is typically set to 3, with Polat and Ozerdem (2016) finding this though optimisation, and the Euclidean distance is often used to determine the nearest neighbours (although distance measures are not always reported). Rather than supervised learning, it has also been implemented in an adaptive learning method, so users can provide feedback if there is a false detection and this updates the model, such as by adding more features (Ibrahim and Majzoub, 2017). KNN has been found to have better performance than other classifiers, such as LDA (Chandel et al., 2019) and ensemble methods (Roy et al., 2019a); however others have found it outperformed by

SVM (Kaleem et al., 2018) and RF (Bhattacharyya and Pachori, 2017) models.

Decision trees have also been previously applied for seizure detection with mixed performance. Authors such as Polat and Güneş (2007) and Mohammadpoory et al. (2017) find decision trees to have favourable performance compared to other model designs, including neural networks and other classical models, but on a small inter-cranial dataset (Bonn Epileptologie Database). However, on other datasets they have been found to have average (e.g. Fergus et al., 2016; Tzallas et al., 2009) or worse performance comparative to other classical models (e.g. Zeng et al., 2016). Feature selection is not commonly used before a tree-based model as the models produce feature importances based on their criterion function to measure the quality of a split. Although not always reported, the C4.5 algorithm (Quinlan, 2014) has been used generate decision trees (Mohammadpoory et al., 2017; Tzallas et al., 2009), although CART is another popular choice to generate trees. Decision trees have also been used to identify seizures using video rather than just EEG (e.g. Pediaditis et al., 2012). However in more modern literature, decision tree models are more commonly used in ensembles (e.g. random forests/boosting).

2.6.2 Ensemble Methods

Ensemble methods aim to improve generalisability of an algorithm by combining the predictions of several estimators (Raschka and Mirjalili, 2019). To achieve this there are two general methods: *averaging* and *boosting*.

Averaging Classifiers

Averaging methods build several separate estimators and then average their predictions, reducing variance and chance of overfitting an estimator. A *bagging* method can be used to average, where an ensemble of base classifiers are each fit on random subsets of a dataset. Specifically, bagging is when sampling is produced with replacement (Breiman, 1996) and without replacement being called *pasting* (Breiman et al., 1999). Therefore both bagging and pasting allow training to be sampled several times across multiple predictors (Géron, 2019). A random forest is a version of bagging where multiple decision trees are averaged together to build a robust model. The random forest algorithm draws a random bootstrap

sample of data and grows a decision tree on this sample. At each node, a number of features are randomly selected without replacement and the node is split using the feature that provides the best split according to a given function. This process is repeated and the prediction is aggregated to assign the class label by majority vote or probabilistic prediction (Raschka and Mirjalili, 2019; Breiman, 2001). The reason for taking a random subset of features is to prevent individual trees in the forest becoming too correlated and it reduces the model's variance, which in turn reduces the chance of overfitting (Burkov, 2019).

A group of classifiers are not always all decision trees, as multiple different classification pipelines can be combined. This aggregation can be done by simply selecting the class label that has been predicted by the majority of the classifiers (more than 50% of votes) for *hard voting*. Certain classifiers return the probability of a predicted class label and this can be used for *soft voting* instead of class labels (Raschka and Mirjalili, 2019). Soft voting often achieves a higher performance than hard voting because highly confident votes are given more weight (Géron, 2019). Ensemble methods work best when the predictors are as independent as possible, so one way of achieving this is to get diverse classifiers. This increases the chance they each make different types of errors, which in combination will improve the overall accuracy (Raschka and Mirjalili, 2019).

Boosted Classifiers

Unlike bagging methods, which tend to work best with complex models (Scikit-learn, 2019), boosting methods typically use weak estimators that are built sequentially, with each estimator attempting to reduce the bias of the predecessor (Géron, 2019). Weak learners initially often only have a slight performance advantage over random guessing, but by focusing on training samples that are hard to classify, the overall performance of the ensemble is improved (Raschka and Mirjalili, 2019). Compared to bagging models, boosting can lead to a decrease in bias, but boosting algorithms such as AdaBoost are also known for over-fitting to the training data (high variance; Raschka and Mirjalili, 2019). AdaBoost (Freund and Schapire, 1997) works by first training a base classifier to make predictions on the training set and then increases the weights for misclassified training instances. A second classifier then is trained with these new weights and a prediction of the classes are made. This is

then repeated until all predictors are trained, wherein the ensemble makes predictions like bagging, except with weights depending on overall accuracy on the weighted training set (Géron, 2019).

Gradient boosting works similar to AdaBoost, in that it sequentially adds weighted predictors to correct predecessors in an ensemble; however, instead of changing weights, it fits a new predictor to the residual errors (Géron, 2019). Gradient boosting does not calculate a gradient, as was previously outlined with linear regression, instead the residuals are used as a proxy of the gradient to show how the model should be adjusted. Gradient boosting enables the handling of large datasets with lots of examples and features, and typically outperforms random forests; even if some implementations of the method are slower to train due to their sequential nature (Burkov, 2019). However, two very effective, efficient, and parallelizable algorithms, popular at time of writing, are *XGBoost* (Chen and Guestrin, 2016) and *lightGBM* (Ke et al., 2017). Both algorithms improve upon basic gradient boosted decision tree (GBDT) algorithms in a number of ways. Both algorithms can grow trees leaf-wise, so that each split is in the leaf that reduces the most loss, rather than a level-wise strategy, which maintains a more balanced tree with splits generally increasing as the levels increase. Although leaf-wise training is more prone to overfitting, it is more flexible and applicable to large datasets. Both algorithms allow for methods to find the best split of features for each leaf, such as histogram-based bin sampling and ignoring sparse inputs, with some specific to lightGBM; such as data subsampling and exclusive feature bundling. Histogram-based methods subsample the number of splits evaluated by a model by grouping features into a set of bins before building each tree. Gradient-based one-side sampling, available in lightGBM, concentrates on data points with larger gradients rather than data points that contribute less to training. In order to ensure ignoring small gradients does not lead to biased sampling, data with small gradients are randomly sampled and these samples are given increased weight when assessing their contribution to the change in loss. Another package of note is *CatBoost* (Prokhorenkova et al., 2018), which makes improvements on the handling of categorical features and has been shown to be quicker on some datasets than XGBoost and LightGBM. However, although in both industrial and academic applications gradient boosted trees are becoming known as consistently strong performing classifiers, the

interpretability of final model is still worse than classical models.

Deep Learning

Another popular ensemble method is *deep learning*, where layers of artificial neurons or other formulas are stacked on top of each other. *Artificial neural networks* (ANN's) can be supervised or unsupervised (e.g. self-organizing maps) and are often compared to networks of neurons in the brain; however, although originally inspired by biological neurons (McCulloch and Pitts, 1943), modern implementations are usually far from how the brain operates (Géron, 2019). Nodes in the network are interconnected and typically arranged into input, middle (hidden), and output layers to complete a given task. The *Perceptron* is one of the simplest ANN architectures based on artificial neurons called threshold logic units (TLUs) or linear threshold units (LTUs). The inputs and outputs are numerical, with each input associated with a weight. The TLU computes a weighted sum of the inputs, applies a step function (e.g. Heaviside step function):

$$\text{heaviside}(z_i) = \begin{cases} 0, & \text{if } z_i < 0 \\ 1, & \text{if } z_i \geq 0 \end{cases} \tag{2.42}$$

and outputs the result. A single TLU can be used for simple linear binary classification by computing a linear combination of inputs, and if these exceed a threshold, outputs a positive or negative class (Géron, 2019). A perceptron is a single layer of TLUs with each neuron connected to all the inputs (see figure 2.17). Special pass through neurons called input neurons, which output whatever input is fed, tend to be first in the network; along with an extra bias neuron that outputs 1 consistently (Géron, 2019). For training the network, the connection weight between two neurons is increased when they have the same output as the training label, taking into account the network error, and connections that lead to the wrong output are not reinforced. One training instance is fed at a time, with each making a prediction. For each output neuron that produced a wrong prediction, it reinforces the connection weights that would have contributed to the correct prediction (Géron, 2019). Perceptrons have a few limitations in their application. They are similar

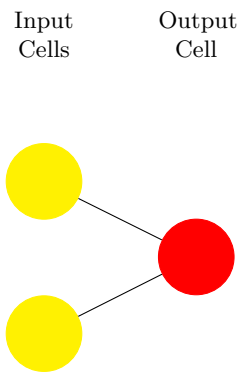


Figure 2.17: A perceptron as a node map.
Note. Reproduced from van Veen and Leijnen (2019)

to logistic regression classifiers, but cannot output a class probability, as they only make predictions based on a hard threshold. Also, as was shown by Minsky and Papert (1969), they are incapable of solving some trivial problems. However some of these limitations can be addressed by organising them into multiple layers (Géron, 2019).

Multi-layer perceptrons are comprised of three general layers; a pass-through input layer, one or more layers of TLUs (hidden layers), and a final layer of TLUs (output layer; see figure 2.18). Every layer has a bias neuron and is fully connected to the next layer. All unit inputs are joined to form an input vector, with this vector having a linear transformation applied to it similar to linear regression. The unit then applies an activation function (g_l) to produce a real valued number, which is then used as the input to the next layer. A single input layer for a 2D input, $\mathbf{x}$, would be:

$$y_{1,1} \leftarrow g_1(\mathbf{w}_{1,1}\mathbf{x}). \tag{2.43}$$

Each unit in a layer is just indexed as a row ($\mathbf{w}_{l,u}$) with l being the layer and u being the unit. The output of above then feeds into a subsequent layer:

$$y_{2,1} \leftarrow g_2(\mathbf{w}_{2,1}\mathbf{y}_1). \tag{2.44}$$

The last layer in a feed-forward network usually just has one unit which either has a linear or logistic activation function depending if it is to be used as a regression or classification model. Furthermore, when the output classes are exclusive, the output layer is modified by

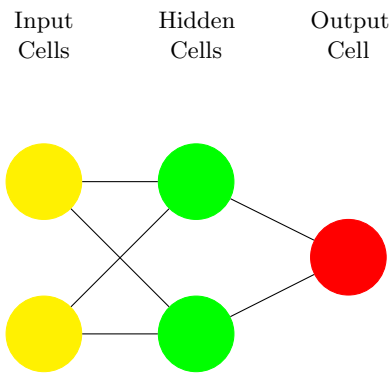


Figure 2.18: A multi-layer perceptron as a node map
Note. Reproduced from van Veen and Leijnen (2019)

a shared softmax function so the output of each neuron corresponds to the estimated probability of the corresponding class (Géron, 2019). There are a number of activation functions available, with them enabling a model to approximate non-linear functions (Burkov, 2019; Chollet, 2017a). As well as the previously mentioned logistic function (output range 0 to 1), common functions include hyperbolic tangent function (tanh), which outputs ranges from -1 to 1 helping make each layers output more normalised, and the ReLU function, which is continuous and has an abrupt change in slope (Géron, 2019). In the modern literature variants on the ReLU are often used; such as SELU, ELU, and Leaky ReLU depending on the model architecture (see figure 2.19; Chollet, 2017a).

When an ANN has two or more hidden layers it is called a deep neural network (DNN). This was previously intractable due to problems of exploding or vanishing gradients. As previously mentioned, a network is trained by back-propagation (or gradient descent using reverse mode autodiff; Rumelhart et al., 1986) by the process of:

1. Feeding training instances to the network.
2. The output of each neuron in each layer is measured against the output error.
3. How much each neuron in the last hidden layer contributed to each output neurons error is computed.
4. This is repeated for each previous hidden later until the algorithm reaches the input layer.

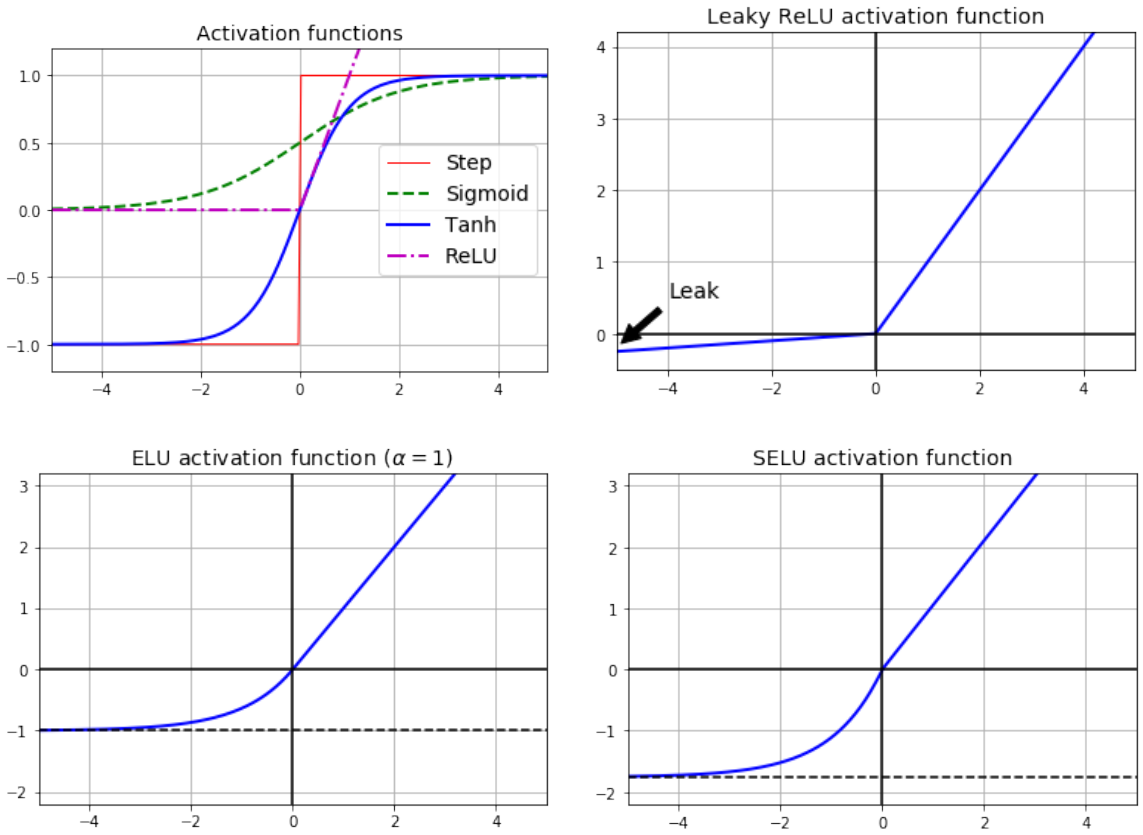


Figure 2.19: Visualisation of various neural network activation functions.
Note. Reproduced from Géron (2019)

5. The connection weights are then tweaked to reduce error.

For this process to work, the step function has to be changed by an activation function to allow gradient descent to make progress each step. However, the previous problem was that the gradient became vanishingly small, preventing parameters from changing, before the use of modern activation functions and other techniques such as skip connections, gradient clipping, batch normalisation, early stopping, and dropout. *Dropout* means each time a training example is run through the network, some units are randomly excluded, with the more neurons excluded the higher the regularisation. This regulates the model as it prevents neurons becoming solely reliant on a small number of their inputs, or as sensitive to slight input changes, thus making the model as a whole more robust (Géron, 2019). *Early stopping* refers to the process of saving the model at each full training pass of the data and using the model that performs best on the validation data; as training tends to eventually lead

Table 2.4: A selection of common neural network layers and their associated applications.

Name	Description	Use
Fully Connected (Dense) Layers	A set of linear functions of all of the input features used for every independent output.	Global pattern detection
Convolutional Layers	Use a subset of inputs for each output commonly by moving filters across the inputs in strides.	Local pattern detection Image recognition Voice recognition Natural language processing
Response Normalization Layers	Divide neurons output by a function of the collective total response.	
Pooling Layers	Combine multipule inputs into a single output using averaging, summing, or taking the maximum value.	
Recurrent Layers	Have forward and backward connections so that activations can flow back and forth.	Sequence data

to overfitting to the training data, to the detriment of performance on the validation data. *Batch normalisation* regulates the model as it standardises the outputs of each layer before the subsequent layer use them as input (Burkov, 2019).

Networks can be loosely grouped into their number of layers (single or multi-layered), whether each layer projects only to later layers (feed-forward) or if they also project to earlier layers (recurrent/feedback), and the number of connections between the layers (fully interconnected or partially interconnected). Many different layers (see table 2.4) and network architectures exist; with multi-layer perceptrons consisting of all fully connected layers, convolutional neural networks using fully connected, convolutional, and pooling layers, and recurrent neural networks using recurrent, and fully connected layers.

Convolutional Neural Network (CNN) models typically have 2D or 3D images as their input, on which a square moving window is applied to train multiple smaller regression models on each patch of data. Each regression model learns a parameter matrix which is convolved with the input matrix to output higher values if they are similar. In deep learning, convolution refers to element-wise multiplication and addition using a weighting matrix which is learned during training. A single layer can therefore be viewed as a collection of multiple convolution filters (each with a bias parameter) which convolve across an image from left to right, top to bottom, computing a convolution at each iteration with a non-linear activation function applied to the sum of the convolution (Burkov, 2019). Each subsequent convolutional layer then treats the preceding layer as a collection of image matrices called

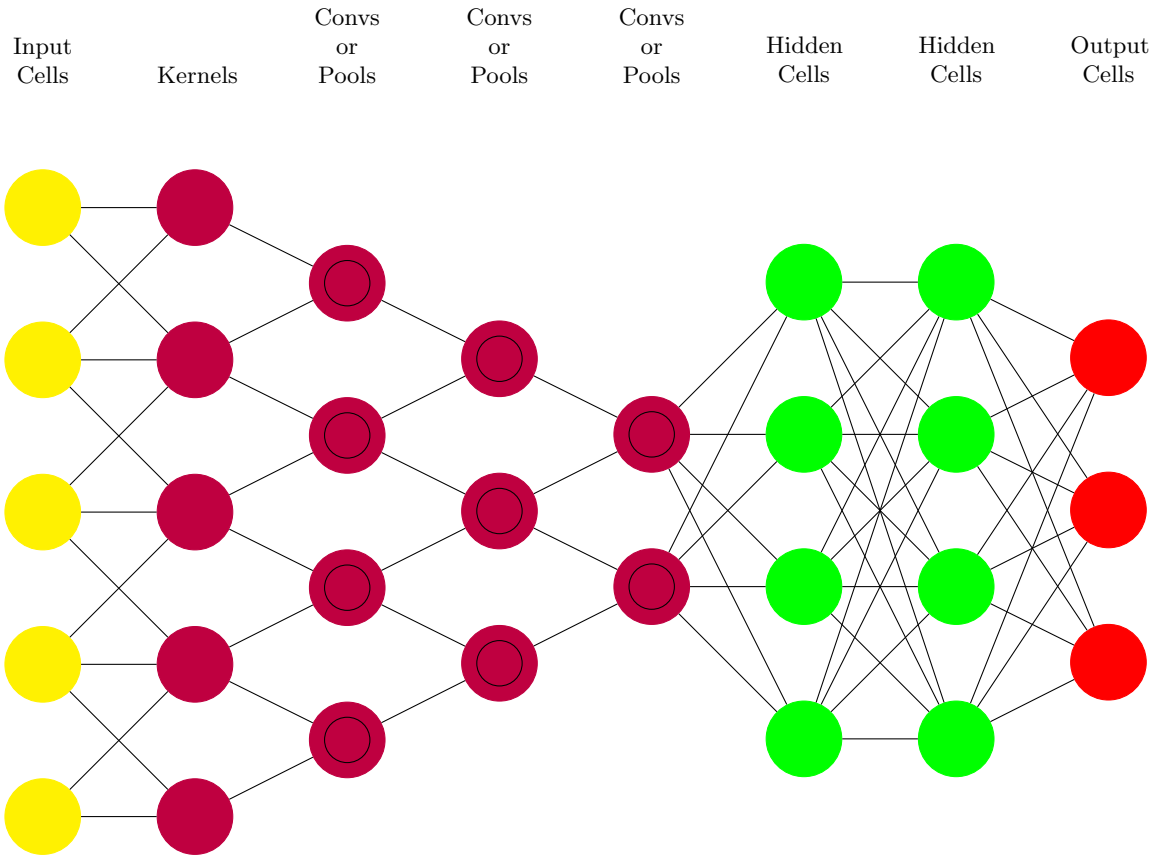


Figure 2.20: A deep convolutional network as a node map.
Note. Reproduced from van Veen and Leijnen (2019).

volumes, which is the sum of convolutions for the patches of each matrices. Convolutions are influenced by the stride step of each moving window, meaning how many cells a window moves across a matrix, as bigger strides create smaller output matrices. Padding is often used to add additional cells around the image/volume before it is convolved with the filter (usually with 0); useful when larger filters are used as it aids "scanning" of the image boundaries. Usually after a convolutional layer there is a pooling layer, which applies a filter with a fixed operator moving window, usually the max or average, rather than a trainable filter. This typically improves a model as it reduces the parameters in the network. This differs from an architecture such as MLP, where each layer added adds an additional $(size_{l-1} + 1) \cdot size_l$ parameters; meaning 1 million parameters are added for each 1000-unit layer (Burkov, 2019).

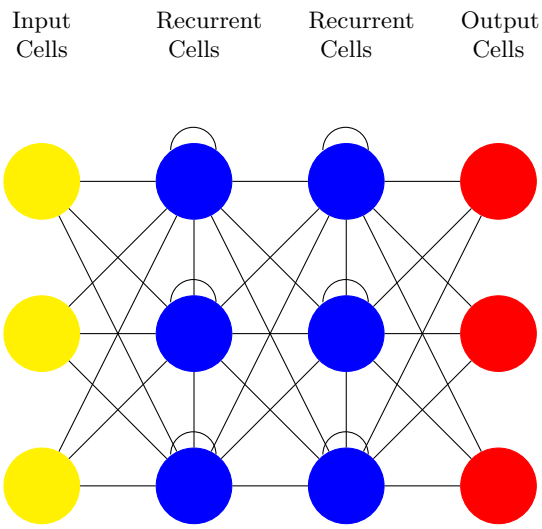


Figure 2.21: A recurrent neural network as a node map
Note. Reproduced from van Veen and Leijnen (2019)

Recurrent Neural Network (RNN) models can be used to label, classify, or generate a sequence matrix where each row is a feature vector and the row order matters. For these models, labelling refers to predicting a class for each feature vector, classification is a class prediction for an entire sequence, and generation outputs a different sequence relevant to the input sequence (Burkov, 2019). Instead of feeding states forward, units in recurrent layers have loops so that they have a real-valued state analogous to a memory of prior timesteps. Feature vectors are input sequentially by order of timestep, so that the state of each unit is updated by calculating the linear combination of the input feature vector and the previous timestep in the same layer. The output of a RNN model is typically a vector unless an MLP is used at the end of the network. RNN models suffer from the vanishing gradient problem, particularly for long input sequences, and in handling long-term dependencies; meaning feature vectors at the start of a sequence can be forgotten by the end of a sequence (Burkov, 2019). This means often gated RNN and long short-term memory (LSTM) networks are used to allow networks to store information in units for future use. In these models, activation functions control the reading, writing, and erasing of stored information in the unit to interpret later timesteps, with what information to control in this way learned from the data (Burkov, 2019).

Ensemble Methods for EEG Classification

Ensembles have an increasing popularity across a number of domains, due to typically performing better than classical methods. For example, the winning and third place submissions in an online seizure detection Kaggle competition both used random forests as classifiers with predominately frequency derived features (Baldassano et al., 2017). In the literature, "hardware friendly" random forest implementations are often tested (Wang et al., 2017; Sopic et al., 2018). Since the class distribution for seizure detection is highly imbalanced, some authors also use methods such as data under-sampling (e.g. Roy et al., 2019a), over-sampling with interpolation (e.g. de la Cal et al., 2018), or boosting. Classification models which use a hybrid method of sampling and boosting, such as the RF variant RUSBoost (Seiffert et al., 2008), Adaboost, and XGBoost (Roy et al., 2019a), have also been applied to seizure detection as these methods have low computational cost and high performance comparative to common models such as SVM (Solaija et al., 2018; Amin and Kamboh, 2016). However, despite their common use in other applications and online competitions, bagging and boosted ensembles are less commonly used in the current seizure detection literature comparative to the previously discussed "classical" and deep learning models.

In a systematic review of the literature, Roy et al. (2019b) found that most studies applying deep learning to EEG focus on its application to sleep staging (e.g. Sors et al., 2018) and abnormalities (e.g. Ruffini et al., 2019), seizure detection and prediction (as reviewed below), brain-computer interfaces (e.g. Yoon et al., 2018), and cognitive and affective monitoring (e.g. Almogbel et al., 2019). CNN models are currently the most popular deep learning model architecture for EEG, despite RNN architectures designed to explicitly take temporal dependencies into consideration. Indeed, this is consistent with a recent systematic evaluation of architectures which concluded that currently convolutional networks tend to perform better than recurrent networks on a number of sequence modelling tasks (Bai et al., 2018). Indeed, specific to seizure detection, Tjepkema-Cloostermans et al. (2018) found 2D CNNs and 2D CNN-LSTMs were the best models from a number of different deep learning architectures. Although raw EEG data, with the matrix consisting of segmented batches of channels and time, has been used (Alhussein et al., 2018; Truong et al., 2018; Yuvaraj et al.,

2018; Zou et al., 2018), EEG recordings are commonly converted to a topomap (Manoranjan and Parvez, 2015; Thodoroff et al., 2016) or STFT spectrogram (e.g. Cao et al., 2017; Yuan et al., 2019a; Alkanhal et al., 2018) for input into a CNN model. Furthermore, due to a large class imbalance in seizure data, training of deep learning models is often done on randomly under-sampled data (Yuvaraj et al., 2018; Yao et al., 2019; Yuan et al., 2019a).

CNN architectures are sometimes combined with other methods to improve performance. For example, combining a CNN with an Autoencoder (Alhussein et al., 2018), or using CNN as a feature extraction method and SVM as a classifier (Muhammad et al., 2018), has been demonstrated to perform better than just using CNN alone. In other deep learning applications, there are also architectures which combine aspects of RNN and CNN architectures, such as replacing fully-connected layers in an LSTM with convolutional layers (Shi et al., 2015), mixing convolutional and recurrent layers (Bradbury et al., 2017), and adding dilation to the recurrent architecture (Chang et al., 2017; Bai et al., 2018); however comparisons of these mixed approaches to other methods are limited for seizure detection (Choi et al., 2019; Thodoroff et al., 2016). There are also a number of CNN models that now can take into account the history of sequence data for prediction, temporal convolutional networks (Bai et al., 2018), as well as improvements to RNN models, such as independently recurrent neural networks (Yao et al., 2019).

Deep learning networks are useful for EEG classification as they are adaptive, so are suitable for non-stationary signals, and can be trained to detect artefacts that may trigger false seizure classifications (Varsavsky et al., 2011a). Rather than depending on chosen features, which largely affect the performance of other machine learning methods (Bengio et al., 2013), deep learning adopts a data-driven approach which reduces the demands of signal pre-processing and feature engineering. Although downsampling, re-referencing, and STFT transformed data are still common pre-processing steps used with these methods, explicit handling of EEG artefacts only occurs in around 50% of papers, suggesting they are not always required to achieve meaningful results (Roy et al., 2019b). Although they may overfit to data with small sample sizes, deep learning generally achieves a better performance as the sample size increases as they can identify intricate data characteristics missed by traditional machine-learning methods (Mohr et al., 2017). A further interesting property

of deep learning models in general is transfer learning, where model parameters can be transferred from one model to another. This enables models to be trained to identify an individuals seizure expression without starting from scratch as a model can be adapted from a general patient model and tailored to suit an individuals seizure expression (e.g. Page et al., 2016).

Although, comparative to classical machine learning models, deep learning models generally have been shown to improve accuracy in their application to EEG by around 5.4%, improvements specific to seizure detection are generally lower than in other areas (e.g. sleep scoring; Roy et al., 2019b). Furthermore, ensemble and neural network methods are also generally harder to interpret (Géron, 2019), with deep learning in particular referenced as a "black box" approach. However, despite this common criticism, there are a number of model inspection techniques that are being developed for deep learning methods. For example, methods applied to models trained on EEG include a class activation map (Ghosh et al., 2018), Deeplift (Lawhern et al., 2018), saliency maps (Volker et al., 2018), input-feature unit-output correlation maps (Schirrmeister et al., 2017), and retrieval of closest examples (Deiss et al., 2018). These are important in clinical settings as understanding a models choices will aid clinical decision making and could lead to future discoveries in brain functioning (Roy et al., 2019b). Despite work in this area, there are still a number of potential limitations in applying deep learning models to EEG data. A practical limitation is that there are generally fewer labelled examples available for training than in other common applications (e.g. computer vision and natural language processing). This is largely due to time and financial costs, as well as ethical issues around privacy, associated with clinical data collection and labelling. Other approaches beyond supervised learning therefore may be required to account for this, such as active learning, semi-supervised learning, and the aforementioned transfer learning. Furthermore, EEG has a low signal-to-noise ratio, differing it from other successful uses of deep learning in image, text, and speech recognition (Roy et al., 2019b). Additionally, deep learning models from the current literature tend to be difficult or impossible to reproduce due to data or code unavailability (Roy et al., 2019b), with papers generally having poor reporting practices. One aspect particularly lacking is the description of hyperparameter optimisation, with Roy et al. (2019b) finding 80% of

reviewed papers not declaring their strategies. Of papers that did report their strategies, most use manual trial and error (e.g. Acharya et al., 2018; Dong et al., 2018) or Gridsearch (e.g. Liao et al., 2018; Aznan et al., 2018), with only a few using more optimal strategies such as bayesian methods (e.g. Stober et al., 2015; Schwabedal et al., 2018).

2.7 Discussion

There are numerous algorithms for the automatic detection of seizure events, however there are generally a lack of independent comprehensive reviews of seizure detection algorithms (e.g. Pauri et al., 1992; Wilson et al., 2004) or evaluations of algorithms on the same dataset (e.g. Varsavsky et al., 2011a; Baldassano et al., 2017). It is difficult to compare algorithms between authors and several factors should be considered when assessing a detectors validity. Firstly, the general application of the methods, be this for intra- or inter-subject classification, will impact performance. Intra-subject models, which are trained and tested on each subject data independently, tend to lead to better performance due to reduced variability of the data (Roy et al., 2019b). The choice of validation procedure will also impact the reported performance of the model. Methods such as Leave-N-Subjects-Out tend to lead to lower performance but, as it uses different subjects for training and testing, it is more applicable to real-life scenarios where a model has not trained on the data it is presented with (Roy et al., 2019b). Conversely, K-fold cross-validation, where data is combined from all subjects in the training and test sets, typically will lead to higher performance (see Deiss et al., 2018). Even with the same applications and validation procedure, there is a lack of norms regarding the calculation and reporting of performance metrics. Although common metrics, such as accuracy, precision, sensitivity, and F1-score, are often reported, these measures do not inherently account for the timescale to which the scoring relates, and can therefore be scored in different ways. For example, the Any-Overlap Method (OVLP; see Ziyabari et al., 2017) is popular across neuroengineering research domains (e.g. Gotman et al., 1997). OVLP considers an event correctly detected if a prediction is made in close proximity to the reference event and groups multiple predicted events in the reference event as a single correct prediction; regardless if a there are gaps between these correct predictions

or if they only cover a short proportion of the actual event. OVLP has been supported in its application to seizure detection on the grounds that seizure onset/offset times are typically ambiguous (Wilson et al., 2003), so this reduces the chance detections of epileptic activities leading up to a seizure are counted as false positives (Varsavsky et al., 2011a). This is further supported by the fact that fundamentally algorithms are always compared to experts visual assessment of the EEG data, which includes bias (Varsavsky et al., 2011a), and therefore reflects relative, rather than absolute, measures of performance. Nevertheless, the adoption of performance metrics using OVLP sampling may lead to a misrepresentation of the actual performance of an algorithm, as it could result in artificially high sensitivities. Another approach is to use Epoch-Based Sampling (EPOCH), where signals and their associated labels are sampled at a fixed epoch duration (e.g. 1 or 30 second windows). Indeed EPOCH based sampling is often the method of choice to manually label the data; for example, sleep data is commonly visually labelled in 30 second windows. However, although this mitigates some of the problems in the OVLP approach, because the annotations are given fixed time windows, long seizures inherently have a higher weighting in performance evaluation; a problem for seizure detection where seizures often vary in length. Furthermore, models that predict labels across windows of the same duration as it was marked could also see improved performance to those that choose different window sizes. A number of other metrics and methods of evaluation have also been proposed that could be applied to seizure detection and sequential pattern recognition applications in general (Ziyabari et al., 2017; Wilson et al., 2003; Kelly et al., 2010; Baldassano et al., 2016), which may in the future help to standardise reporting of detection algorithms.

However, although evaluation methods commonly differ between authors (see table 2.A.1), the use of shared standardised datasets is becoming more common (Wagenaar et al., 2015). Indeed in a recent systematic literature review of general applications of deep learning to EEG (Roy et al., 2019b), 54% of the 156 papers reviewed used a publicly available dataset; however only 19% released their source code, so when taken in combination with the dataset, only 7% of studies could be fully reproduced. Its worth noting that the definitions of reproducibility and replicability are often inconsistent or contradictory across different institutions and scientific disciplines. Here we use definitions from National Academies of

Sciences Engineering and Medicine (2019), where *reproducibility* (or "computational reproducibility") is the act of "a second researcher recomputing the original results, and it can be satisfied with the availability of data, code, and methods that makes that recomputation possible. This definition of reproducibility refers to the transparency and reproducibility of computations" (p.45). *Replicability* in this case therefore refers to:

> When a new study is conducted and new data are collected, aimed at the same or a similar scientific question as a previous one, we define it as a replication. A replication attempt might be conducted by the same investigators in the same lab in order to verify the original result, or it might be conducted by new investigators in a new lab or context, using the same or different methods and conditions of analysis. If this second study, aimed at the same scientific question but collecting new data, finds consistent results or can draw consistent conclusions, the research is replicable. (p.45)

Furthermore, we also define *generalisability* where "a second study explores a similar scientific question but in other contexts or populations that differ from the original one and finds consistent results" (p.45-46).

Currently, the most common dataset in the literature appears to be the Bonn Epileptologie Database (http://epileptologie-bonn.de/cms; Andrzejak et al., 2001), which contains 100 single-channel intracranial EEG segments of 23.6 seconds. Although limited in both size and ecological validity comparative to other more recent datasets, it is still commonly used by authors to demonstrate classification pipelines in a simple and comparable way. The European Epilepsy Database (Schulze-Bonhage et al., 2010; Ihle et al., 2012; Klatt et al., 2012) is another common dataset (e.g. Manzouri et al., 2018), due to its large collection of scalp and intracranial EEG data; although it is not open-source as it has a limited access licence. On-the-other-hand, open-source data sharing has been facilitated by the increase of dedicated data sharing platforms such as Kaggle (https://www.kaggle.com), and the Open Science Framework (https://osf.io), as well as more specific sites such as iEEG (https://www.ieeg.org; Wagenaar et al., 2013) and PhysioNet (https://physionet.org; Goldberger et al., 2000). Kaggle, owned by Google LLC, is an online community of data scientists

where datasets and classification competitions are hosted. One such competition was the UPenn and Mayo Clinic's Seizure Detection Challenge (https://www.kaggle.com/c/seizure-detection; Baldassano et al., 2017), where 200 teams competed to develop seizure detection algorithms for 1 second ictal and interictal intracranial EEG segments from 4 canines (Coles et al., 2013) and 8 patients (Stead et al., 2010; Brinkmann et al., 2009) with epilepsy. iEEG (Wagenaar et al., 2013) is a specific data sharing platform, aimed at providing access to over 1,200 datasets of continuous scalp and intracranial EEG hosted on commercial cloud (Amazon Web Services, using Amazon's S3 and RDS) and on a local intranet (using Tomcat, NFS and MySQL) resources (Wagenaar et al., 2015). However iEEG can be difficult to navigate without knowing specific dataset ID's. Conversely, PhysioNet's PhysioBank databases are easy to access and come with software (PhysioToolkit) to aid signal processing and analysis. PhysioNet hosts the CHB-MIT Scalp EEG Database (https://www.physionet.org/pn6/chbmit/), which is commonly used due to ease of access and provision of continuous extracranial recordings of 22 patients (198 seizures total). However the largest open-source corpus of clinical EEG data, containing 15,757 hours of EEG recordings from 13,539 patients, is the TUH EEG corpus (Obeid and Picone, 2016). The corpus contains archival records from Temple University Hospital (TUH) paired with corresponding clinician reports. The TUH EEG Seizure Corpus (1.5.0; Shah et al., 2018) is a subset of the TUH corpus which has been manually annotated to categorise a total of 3055 seizures from 642 patients, split into training and validation sets. As both the CHB-MIT and TUH EEG Seizure Corpus are among the most common and largest data-sets available in the literature, research using these datasets are often referenced in the subsequent chapters. Furthermore, for the reasons outlined above regarding the difficulty comparing performance metrics, results of many papers using these datasets are presented in tables 2.A.1 and 2.A.3, rather than written here, to allow for a more holistic context for comparison.

Ultimately, the seizure detection algorithms that have been discussed in this chapter aim to be integrated into a seizure detection system used in clinical practice (e.g. Mirarchi et al., 2017; Alhussein et al., 2018; Muhammad et al., 2018). Indeed there have already been a number of detection systems that have been sold commercially to varying success (see table 2.A.2). In a review of some of the commercially available algorithms by Varsavsky

et al. (2011a), CNet was found to have the highest true positive rate (TPR), but as Reveal had a lower false positive rate (FPR), this was the most reliable detector. The authors determined in this study that an FPR of 6 per hour was reasonable, due to this meaning only 10% of an EEG record needed to be subsequently visually reviewed, and a sensitivity of 80% reasonable, due to the previous literature suggesting experts only agree 80% of the time on seizure labelling (Wilson et al., 2003). However, the previously discussed academic advancements in algorithm performance and design promise greater accuracy and performance in a healthcare environment. Nevertheless, it is worth noting that generally most modern detection algorithms provide good results for routine intra- and extra-cranial EEG recording, but are poorer in ambulatory settings due to increased artefacts (Chavakula et al., 2013), and generally should still be considered complimentary in clinical settings to traditional EEG visual analysis (Rosso et al., 2006).

2.A Appendix A

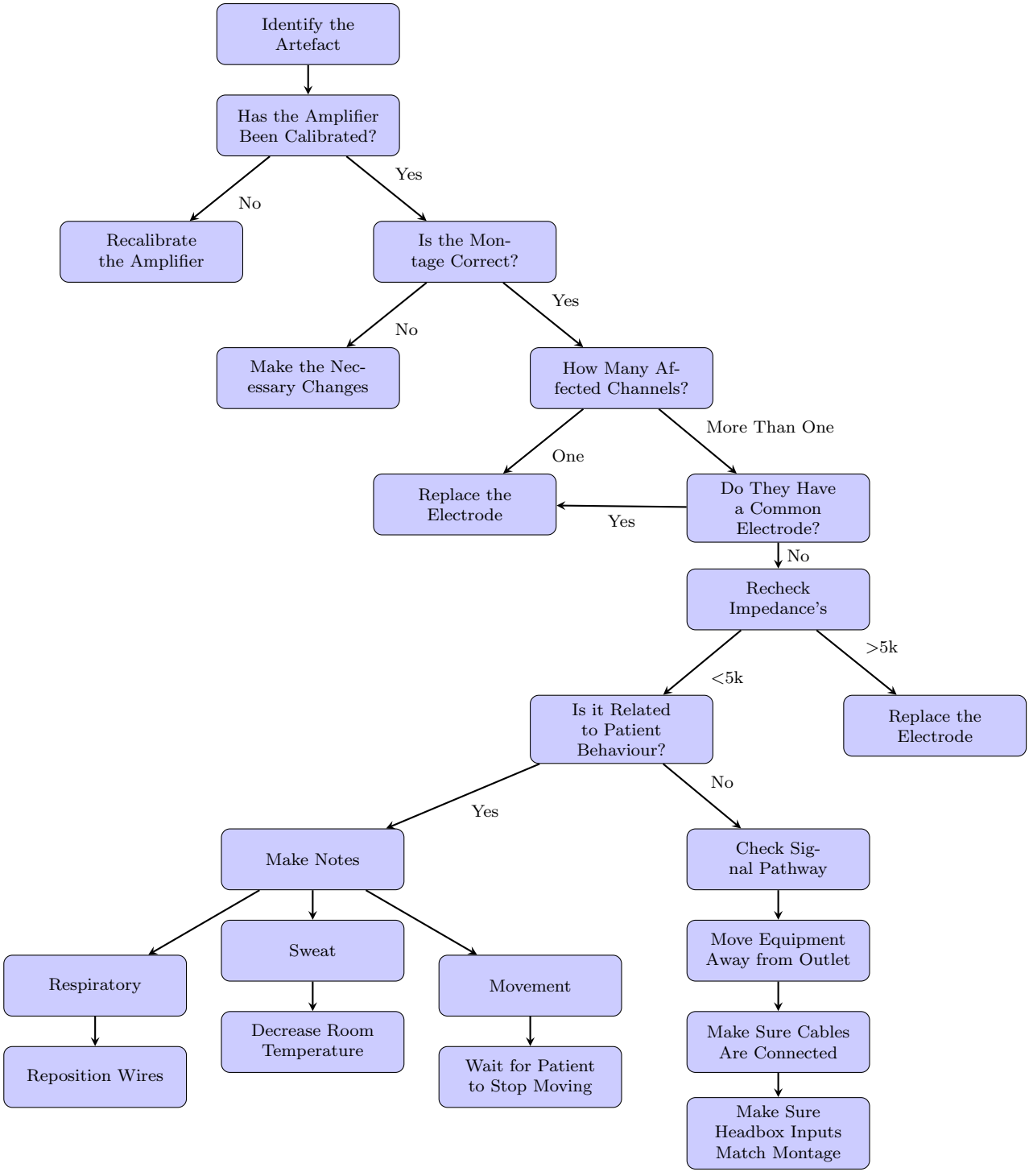


Figure 2.A.1: Methods for manual correction of artefacts during data collection
Note. Reproduced from Spriggs (2009)

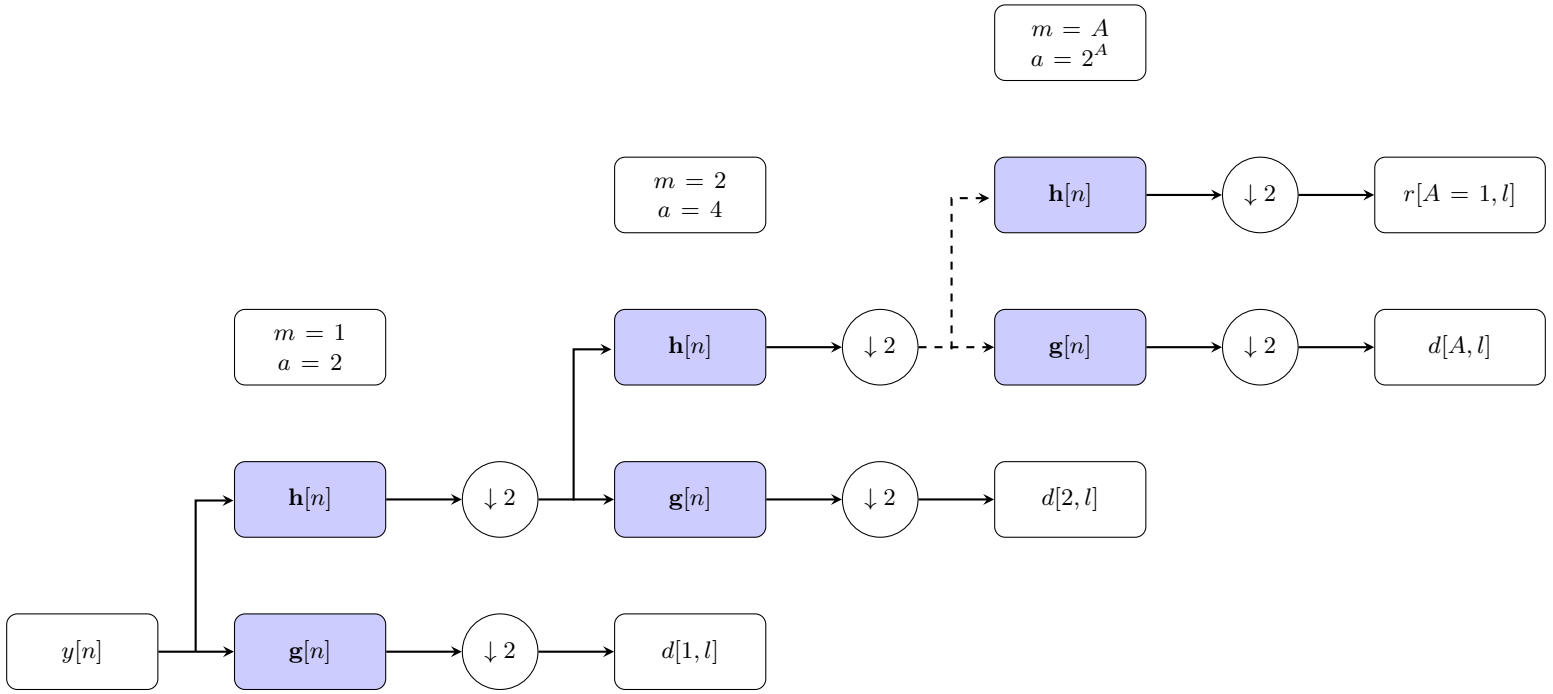


Figure 2.A.2: Decimation procedure for a discrete wavelet decomposition into components.
Note. Adapted from Varsavsky et al. (2011b)

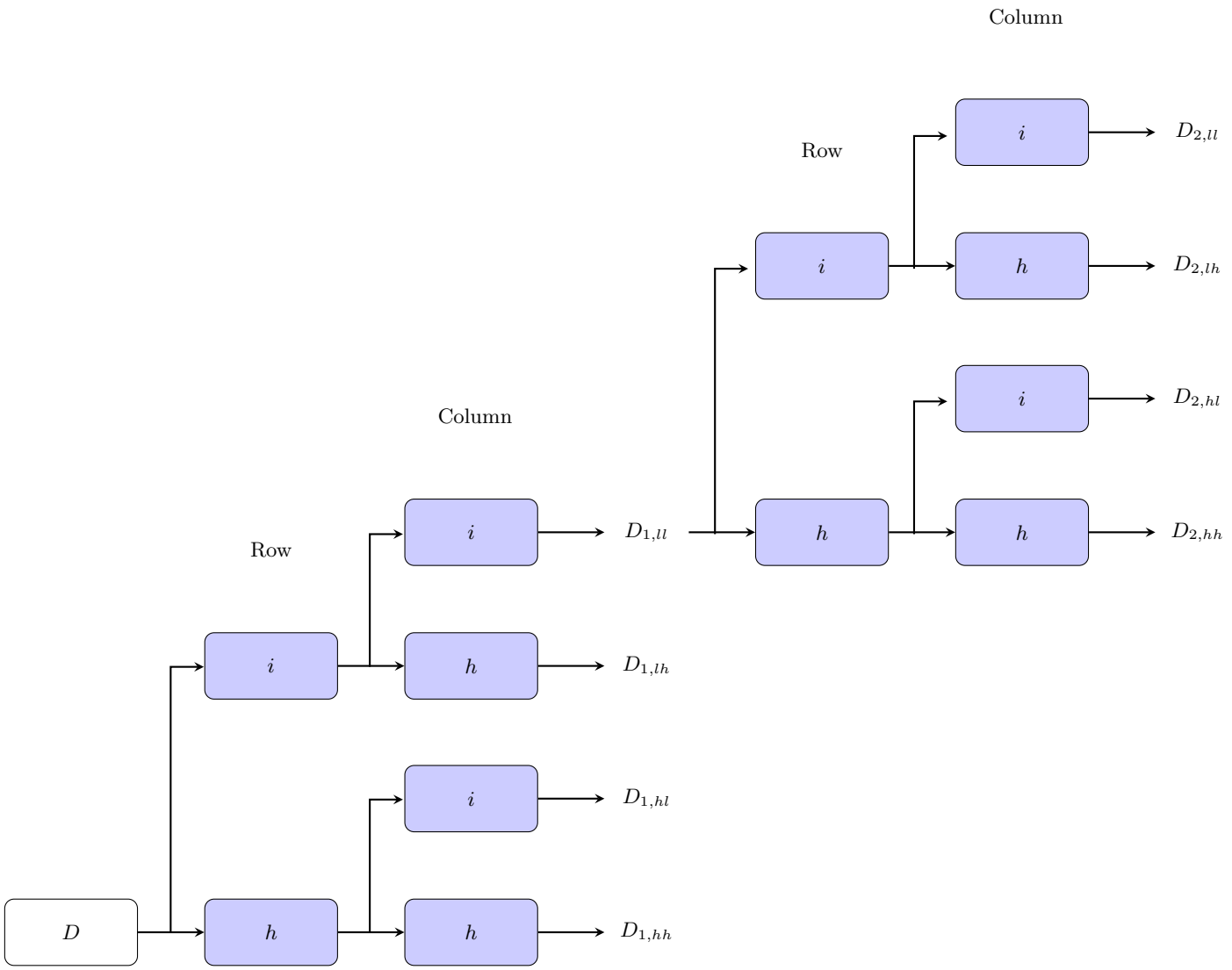


Figure 2.A.3: Decimation procedure for a stationary wavelet decomposition.

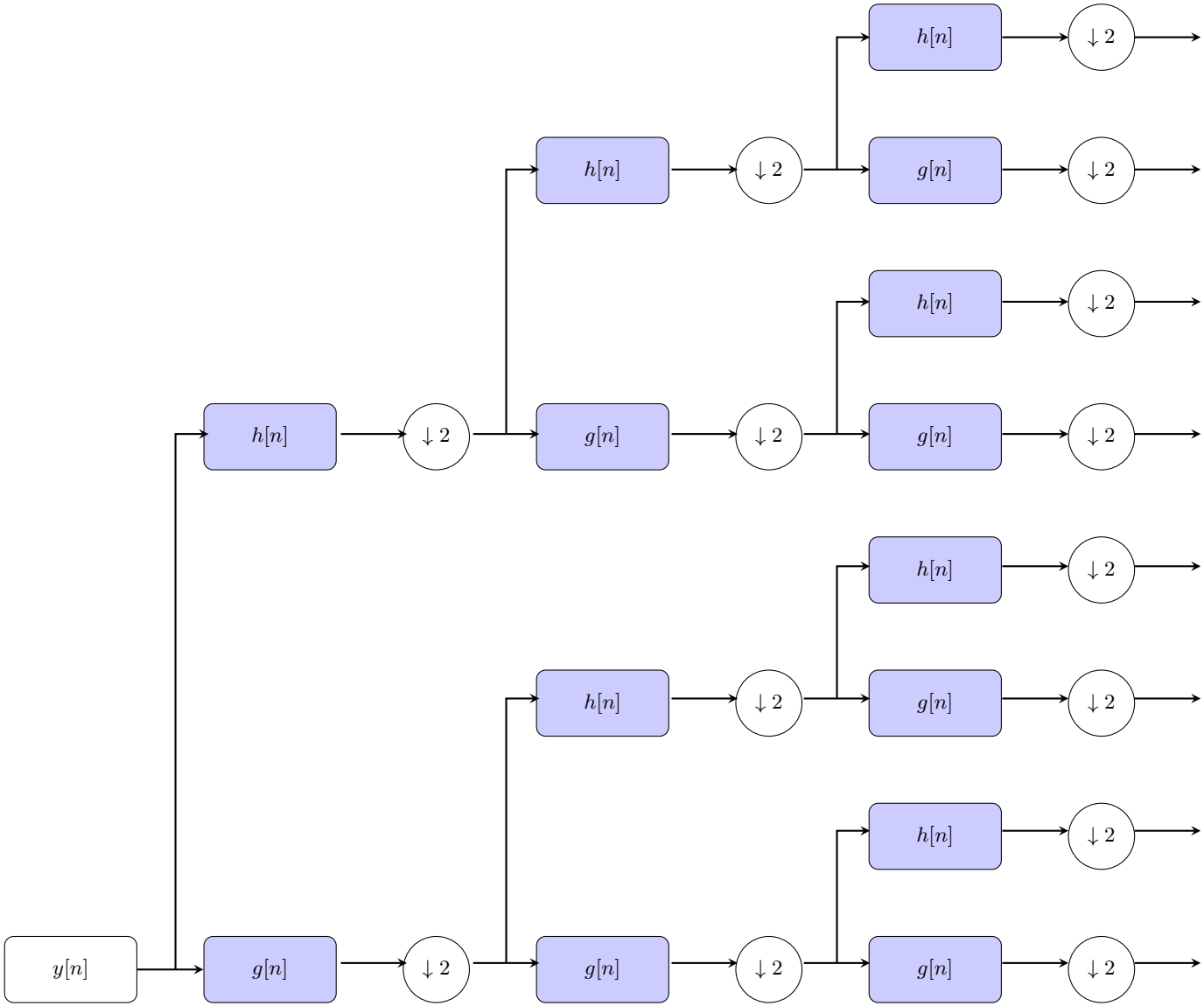


Figure 2.A.4: Decimation procedure for a wavelet packet decomposition with 3 levels.

Table 2.A.1: Seizure onset and detection research papers using CHB-MIT dataset.

Reference	Patients	Feature(s)	Feature Reduction	Classifier(s)	Evaluation Method	ACC	SEN	SPEC	PREC	F1	AUC	FPR	Latency (s)
Shoeb and Guttag (2010)	23	1 frequency	-	SVM	Leave-one-record-out cross-validation	-	97.00	-	-	-	-	0.08/h	4.60
Rafiuddin et al. (2011)	23	2 time 2 time-frequency	-	LDA	20% hold-out	80.16	-	-	-	-	-	-	-
Khan et al. (2012)	5	2 time-frequency	-	LDA	20% hold-out	91.80	83.60	100.00	-	-	-	-	-
Nasehi and Pourghassem (2013)	23	1 time-frequency/frequency	-	MLP	10% hold-out	-	98.00	-	-	-	-	0.13/h	0-10
Ahammad et al. (2014)	23	2 time 6 time-frequency	-	LC	40% hold-out	-	98.50	-	-	-	-	-	1.76
Chen et al. (2014)	12	2 time 1 time-frequency 1 phase	-	SVM ELM	Leave-one-patient-out cross-validation	- -	88.20 92.60	- -	- -	- -	- -	0.06/h 0.08/h	- -
Kiranyaz et al. (2014)	23	27 time 5 frequency 12 time-frequency	-	CNBC	75% hold-out	-	89.01	94.71	-	-	-	-	-
Mitha et al. (2014)	?s segments 1000 ictal 1000 inter-ictal	16 time 2 frequency 1 time-frequency	FS	SVM	??	92.80	88.66	87.00	-	-	-	-	-
Paulose and Bedeeuzzaman (2014)	23	1 time	-	LOG SVM	40% hold-out	71.62 96.57	- -	- -	- -	- -	- -	- -	- -
Pramod et al. (2014)	16	10 time 1 frequency 1 time-frequency	-	MLP	Leave-one-patient-out cross-validation	-	98.06	99.29	73.29	-	-	0.50%	-
Supratak et al. (2014)	6	5 raw 3 raw 1 raw	-	LC & SA	Leave-one-record-out cross-validation	- - -	87.18 100.00 100.00	- - -	- - -	- - -	- - -	1.86/h 7.90/h 83.13/h	11.18 6.87 3.36
Zabihi et al. (2013)	4	31 time 7 frequency 12 time-frequency 1 phase	CMIM	SVM	50% hold-out	98.94	93.78	97.65	-	-	-	-	-
Alotaiby et al. (2015)	9	1 time	-	SVM	Leave-one-record-out	93.16	87.04	98.28	-	-	-	-	-
Behnam and Pourghassem (2015)	100hrs	4 frequency	-	MLP & BPNN	30% hold-out (20% test, 10% validation)	94.40	-	-	-	-	-	-	-
Elmahdy et al. (2015)	23	4 time 2 frequency	-	SVM	??% hold-out	94.82	91.64	98.01	-	-	-	2.16%	-
Fergus et al. (2015)	23	6 time 2 frequency	FR PCA	LDA QDC UDC POLY LC KNN DT PARZEN SVM	80% SMOTE hold-out	- - - - - - - - -	82.00 87.00 52.00 82.00 88.00 93.00 90.00 96.00 90.00	90.00 92.00 91.00 90.00 87.00 94.00 90.00 98.00 89.00	- - - - - - - - -	- - - - - - - - -	56.00 63.00 70.00 92.00 94.00 98.00 94.00 82.00 93.00	- - - - - - - - -	- - - - - - - - -
Hamdan et al. (2015)	60s segments	5 time	SS	LDC	20% hold-out	-	78.00	88.00	-	-	55.00	-	-
Javaid et al. (2015)	32,290 epochs	3 time-frequency	PCA	SVM QDA ANN	10-fold cross-validation 50% hold-out	96.30 94.00 92.88	93.50 90.00 75.75	97.40 96.00 98.66	- - -	- - -	- - -	- - -	- - -
Shanir et al. (2015)	3	2 frequency	-	LC	40% hold-out	99.81	100.00	99.81	-	-	-	-	-
Xun et al. (2015) Xun et al. (2016)	4	-	SAE	SVM	20% hold-out	77.07	-	-	-	-	88.80	-	-
Amin and Kamboh (2016)	23	1 time-frequency	-	ADABOOST SVM	50% hold-out	- -	97.87 96.00	- -	- -	- -	- -	0.08/h 0.15/h	2.7 4.62
Awan et al. (2016)	12	12 time 3 time-frequency	WSR AB	KNN SVM GMM MV	??	80.90 83.60 84.70 88.10	80.10 84.30 86.10 87.90	81.40 83.70 83.60 89.20	- - - -	- - - -	- - - -	- - - -	- - - -

Reference	Patients	Feature(s)	Feature Reduction	Classifier(s)	Evaluation Method	ACC	SEN	SPEC	PREC	F1	AUC	FPR	Latency (s)
Bugeja et al. (2016)	Reduced epochs around seizures	1 time 1 frequency	-	SVM ELM	3-fold cross-validation	- -	97.98 98.99	83.73 81.39	- -	- -	- -	- -	2.95 1.26
Chandel et al. (2016)	23	3 time-frequency 2 time	-	LC	40% ictal hold-out	-	100.00	-	-	-	-	-	1.90
Das et al. (2016)	5	1 time-frequency	-	SVM	20% hold-out	98.33 (T7-P7)	97.25 (T7-P7)	98.34 (T7-P7)	-	-	-	-	-
Fergus et al. (2016)	171 Ictal 171 interictal	2 frequency	PCA LDA LDAi LDAf LDAb GS - -	QDC UDC POLY LOGL KNN DT PARZEN SVM		- - - - - - - -	84.00 51.00 78.00 82.00 88.00 82.00 81.00 85.00	86.00 91.00 88.00 84.00 88.00 81.00 93.00 86.00	- - - - - - - -	- - - - - - - -	60.00 70.00 89.00 90.00 93.00 89.00 61.00 90.00	- - - - - - - -	- - - - - - - -
Orellana and Cerqueira (2016)	23	2 time 1 frequency 1 phase	PCA	RF	10-fold cross-validation	92.46	89.73	94.77	-	-	-	6.87/h	-
Orosco et al. (2016)	18	3 time-frequency	LW	LDA PRNN	30% hold-out	- -	92.60 79.90	99.90 99.70	- -	- -	- -	0.30/h 3.90/h	0.20 18.40
Thodoroff et al. (2016)	23	1 frequency	-	CNN & RNN	Leave-one-patient-out cross-validation	-	85.00	-	-	-	-	0.80/h	-
Van Esbroeck et al. (2016)	23	1 frequency	-	SVM	Leave-one-record-out cross-validation	-	-	-	-	-	96.20	41.73/h	10.12
Vidyaratne et al. (2016)	5	-	-	RNN	Leave-one-patient-out cross-validation	-	100.00	-	-	-	-	0.08/h	7.00
Zabihi et al. (2016)	23	6 phase	PCA	LDA & NB	50% hold-out	94.69	89.10	94.80	-	-	-	-	-
Ammar and Senouci (2017)	3	5 time 2 frequency	-	ELM	??	94.85	-	-	-	-	-	-	-
Bhattacharyya and Pachori (2017)	23	1 time-frequency	-	RF C4.5 FT Bayes-net NB KNN	10-fold cross-validation	99.41 98.64 98.90 97.98 95.16 98.35	97.91 95.44 95.58 91.93 88.46 94.02	99.57 99.09 99.30 98.70 95.74 98.82	- - - - - -	- - - - - -	99.90 98.80 97.90 99.30 98.10 96.40	- - - - - -	- - - - - -
Bolagh and Clifford (2017)	23	1 frequency	-	SVM	Leave-one-patient-out cross-validation	89.84	85.77	-	-	-	-	0.77/h	5.24
Cao et al. (2017)	12	1 Frequency	-	SVM 7-layer CNN	Leave-one-ictal-record-out	80.53 90.13	76.06 96.50	- -	- -	- -	- -	- -	- -
Chandel et al. (2017)	14	2 time 2 time-frequency	-	LC	25-40% hold-out	98.60	96.43	98.64	-	-	-	-	1.70/0.90
Ibrahim and Majzoub (2017)	10	2 time-frequency	-	KNN	Leave-one-record-out cross-validation	-	94.50	-	-	-	-	1.14/h	8.60
Khan and Khan (2017)	23	4 time	-	QC	4-fold cross-validation	86.58	83.47	90.27	86.87	-	-	-	3.43
Samiee et al. (2017)	23	1 time-frequency	-	LOG RF SVM	75% hold-out	- - -	70.39 66.35 60.42	99.09 99.29 99.49	- - -	98.85 98.62 98.89	85.41 82.79 83.00	- - -	- - -
Vidyaratne and Iftekharuddin (2017)	23	1 time 1 frequency	-	RVM	Leave-one-ictal-record-out cross-validation	-	96.00	-	-	-	-	0.10/h	1.89
Wang et al. (2017)	20	1 frequency	ICA RF SVM-RFE	SVM	Leave-one-record-out	-	74.20	-	-	-	-	0.36/h	6.00
Yuan et al. (2017c)	9	-	SAE	Wave2Vec	20% hold-out	92.42	-	-	-	95.06	96.77	-	-
Yuan et al. (2017b)	9	1 time-frequency	-	SA	20% hold-out	95.71	98.65	-	96.08	97.71	-	-	-
Zhu et al. (2017)	23	2 time	-	QDA	Leave-one-ictal-record-out cross-validation	-	99.00	100.00	-	-	-	-	-

Reference	Patients	Feature(s)	Feature Reduction	Classifier(s)	Evaluation Method	ACC	SEN	SPEC	PREC	F1	AUC	FPR	Latency (s)
Alhussein et al. (2018)	23	-	-	7-layer CNN & Autoencoder	hold-out?	99.20	93.50	-	-	-	-	-	-
Alickovic et al. (2018)	8s segments 1000 inter-ictal 1000 ictal 1000 pre-ictal	5 time-frequency 1 empirical mode decomposition	Manual	RF SVM MLP KNN	10-fold cross-validation	100.00 100.00 100.00 100.00	100.00 100.00 100.00 100.00	100.00 100.00 100.00 100.00	- - - -	- - - -	- - - -	- - - -	- - - -
Ammar et al. (2018)	1	3 time 2 frequency	PSO	SVM	??% hold-out	-	91.00	95.00	-	-	-	-	-
Deng et al. (2018)	22	3 time-frequency	-	TTL-FSs	1-4 different patients trained for each tested	94.04	91.29	93.16	-	-	-	-	-
Harpale and Bairagi (2018)	5	4 time 1 frequency 4 time-frequency	ANOVA	FC	50% hold-out	96.48	96.52	95.34	-	-	-	3.52%	-
Hussain (2018)	6	4 time-frequency	-	MLP	50% ictal hold-out	92.27	90.54	93.07	-	-	-	-	-
Kaleem et al. (2018)	23	3 time-frequency	T-Test	LDA SVM NB KNN CT	5-fold cross-validation	95.70 99.60 96.20 99.50 99.30	97.60 99.90 98.00 100.00 99.50	93.80 99.40 94.30 99.10 99.10	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -
Ke et al. (2018)	18	1 time	-	9-layer VGGNet (CNN & MLP)	64%, 16%, 20% hold-out	98.13	98.85	97.47	-	-	-	-	-
Khanmohammadi and Chou (2018)	23	1 time	PCA-CSP	CP	Leave-one-record-out cross-validation	-	96.00	-	-	-	-	0.12/h	4.21
Muhammad et al. (2018)	23	1 frequency	8-layer CNN/ Autoencoder	SVM	Leave-one-patient-out cross-validation	99.02	92.35	93.64	-	-	-	-	-
Park et al. (2018)	23 (92 Segments)	-	-	11-layer CNN	I ictal event hold-out per patient	85.60	91.70	80.60	-	-	-	-	-
Perera et al. (2017)	7	3 time-frequency	ANOVA	SVM	25-40% hold-out	-	90.30	99.00	-	-	-	2.60	3.41
Ramakrishnan and Muthanantha Murugavel (2018)	23	3 time-frequency	FRB	LDAG & SVM	50% hold-out	98.00	99.00	96.00	-	-	-	-	-
Selvathi and Meera (2018)	?	2 time-frequency	-	SVM	50% hold-out	95.60	-	-	-	-	-	-	-
Sendi et al. (2018)	23	1 time	LDA	SVM	10-fold cross-validation	99.32	99.41	95.25	99.86	99.64	98.17	-	-
Shanir et al. (2018)	22	3 time	-	KNN	Patient specific 3-fold cross-validation	-	99.20	99.80	-	-	-	0.47	-
Solaija et al. (2018)	23	-	DMD	RUSBoost	50% hold-out	-	87.32	98.93	-	-	-	0.64/h	-
Sopic et al. (2018)	10	5 time-frequency 2 frequency	-	RF	Leave-one-patient-out cross-validation	-	96.95	95.77	-	-	-	-	-
Truong et al. (2018)	23	1 frequency	-	5-layer CNN	Leave-one-patient-out cross-validation	-	-	-	-	-	96.10	-	-
Wang and Ke (2018)	18	1 time	-	MLP	5-fold cross-validation	97.93	99.42	96.76	-	-	-	-	-
Yao et al. (2018)	23	-	-	BiLSTM	70%, 15%, 15% hold-out	83.89	83.72	84.06	-	83.63	-	-	-
Yuan et al. (2018b)	9	1 time	PCA	MLP	??% hold-out	96.61	-	-	-	97.85	98.47	-	-
Yuan et al. (2018a)	9	1 time-frequency	-	DAE	20% hold-out	95.71	98.65	-	96.08	97.25	98.74	-	-
Zhou et al. (2018)	21	1 frequency	-	3-layer CNN	6-fold cross-validation	97.50	96.90	98.10	-	-	-	-	-
Alkanhal et al. (2018)	23	1 frequency	-	9-layer CNN	Leave-one-patient-out cross-validation	-	87.95	86.50	-	-	-	0.75/h	-
Chandel et al. (2019)	18	7 time-frequency	-	LDA KNN	40% ictal & all-1hr iter-ictal 5-fold cross-validation	98.00 99.45	100.00 98.36	98.05 99.62	- -	- -	- -	4.02/h 0.80/h	1.41 6.32
Choi et al. (2019)	23	1 frequency	-	CNN & RNN	1 patient hold-out	99.40	89.00	99.50	-	-	-	0.50/h	-
Nandy et al. (2019)	23?	4 time 7 frequency 1 time-frequency	MOEA	LDA QLDA SVM	???	76.41 80.79 97.05	76.09 73.4 98.99	76.73 88.2 95.12	76.61 86.17 95.29	76.35 79.27 97.11	83.05 89.98 99.54	23.27% 11.80% 4.88%	- - -
Selvakumari et al. (2019)	23	7 phase	PCA	SVM/NB	50% hold-out	96.28	97.50	94.50	-	-	-	-	-
Yao et al. (2019)	23	-	-	LSTM CNN RNN	10-fold reduced cross-validation	84.35 82.90 86.55	84.40 84.80 84.20	84.25 81.00 88.90	84.70 82.56 88.51	84.25 83.05 86.22	- - -	- - -	- - -
Yuan et al. (2019b)	23	-	PCA SSAEs Skip-gram Med2Vec -	SVM SVM SVM SVM Wave2Vec	20% balanced hold-out	79.76 87.04 86.84 68.99 93.92	- - - - -	- - - - -	- - - - -	86.38 91.44 91.45 77.12 96.05	85.52 91.71 90.39 76.29 98.33	- - - - -	- - - - -

Reference	Patients	Feature(s)	Feature Reduction	Classifier(s)	Evaluation Method	ACC	SEN	SPEC	PREC	F1	AUC	FPR	Latency (s)
Yuan et al. (2019a)	23	-	PCA	SVM	5-fold subject independent cross-validation	79.46	-	-	-	0.77	52.00	-	-
			PCA	NN		72.68	-	-	-	29.32	57.89	-	-
			-	WT-CtxFusionEEG		90.25	-	-	-	72.02	92.87	-	-
			-	CNN & MLP		94.37	-	-	-	85.34	95.72	-	-
Yuvaraj et al. (2018)	23	1 frequency	-	11-layer CNN	4-fold subject independent reduced cross-validation	-	86.29	-	-	-	-	0.74/h	2.10
Zabihi et al. (2019)	23	3 phase	-	LDA/NN	50% hold-out per patient	95.11	91.15	95.16	-	-	93.16	-	-
Zhang et al. (2019)	12	1 frequency	W-FPE-F	ELM	50% reduced hold-out	94.17	98.99	89.33	-	-	-	-	-
Zou et al. (2018)	23	-	-	CNN	Leave-one-record-out reduced cross-validation	-	99.46	-	-	-	-	0.12/h	8.08

Note. Features refer to a method to extract features and may have multiple levels depending on number of channels or number of decomposition. The best performing version of a pipeline in a paper is chosen to save on space.

Table 2.A.2: Current and previous commercially available seizure detection systems.

Name	Features	Classification	Expert System	Papers
Monitor	Relative amplitude to past averages Coefficient of variation Average duration	Manual Threshold	Detections required in more than one channel & epoch	Gotman (1982) Gotman (1990)
CNet	Difference from background using a Fourier transform	Manual Threshold Self-Organising Map	Groups of averaged channels intially evaluated before individual channels Epochs rejected based on EEG amplitudes relative to past and future epochs Detections required to occur at least twice in 15 seconds	Gabor et al. (1996) Gabor (1998)
Reveal	Amplitude, duration, and frequency of Gabor atoms	Artificial Neural Network	Accounting for the temporal and spatial context?	Wilson et al. (2004)
Persyst	Background activity, Rhythmicity, Amplitude, ...	Various Neural Networks	Accounts for artifactual components	Wilson (2005) Wilson (2006)

Table 2.A.3: Seizure onset and detection research papers using TUH dataset

Reference	Subjects	Features	Feature Selection/Extraction	Classifier	Evaluation Method	ACC	SEN	SPEC	PREC	F1-score	AUC	FPR
Golmohammadi et al. (2017)	246	3 Frequency	-	CNN/GRU	25% Holdout	-	30.83	91.49	-	-	-	0.88/h
			-	CNN/LSTM		-	30.83	97.1	-	-	-	0.25/h
Shah et al. (2017)	246	3 Frequency	-	CNN/LSTM	25% Holdout	-	39.15	90.37	-	-	-	0.95/h
Golmohammadi et al. (2018)	159	3 Frequency	HMM	DAE	60% Holdout	-	35.35	73.35	-	-	-	3.21/h
			HMM	LSTM		-	30.05	80.53	-	-	-	2.5/h
			PCA	LSTM		-	32.97	77.57	-	-	-	3.04/h
			-	CNN/MLP		-	39.09	76.84	-	-	-	3.21/h
			-	CNN/LSTM		-	30.83	96.86	-	-	-	0.29/h
Zhang et al. (2018)	27	23 Time	Small standard deviations removed	SVM	10-Fold Cross-Validation	100.00	100.00	99.00	-	-	-	-
		2 Frequency	Recursive feature elimination SVM	LR		98.00	95.00	98.00	-	-	-	-
		23 Time-Frequency	Backwards feature selection	RF		81.00	25.00	97.00	-	-	-	-
				Gboost		83.00	40.00	96.00	-	-	-	-
				NB		60.00	75.00	56.00	-	-	-	-
				KNN		80.00	43.00	90.00	-	-	-	-
Asif et al. (2019)	246	1 Frequency	-	CNN	5-Fold Cross-Validation	-	-	-	-	88.01	-	-
Golmohammadi et al. (2019)	370	3 Frequency	Manual Feature Selection	HMM	30% Holdout	-	86.78	82.3	-	-	-	-
			PCA	HMM/DAE		-	78.93	95.6	-	-	-	-
			PCA	HMM/DAE/SLM		-	90.1	95.12	-	-	-	-
Iešmantas and Alzbutas (2020)	246	1 Time	-	SVM	25% Holdout	-	-	-	-	-	64	-
		1 Frequency	-	CNN		-	-	-	-	-	74	-
		1 Phase										
Ramadhani et al. (2019)	??? (210 Signals)	3 Time 1 Frequency	Independent Component Analysis	SVM	40% Holdout	91.4	90.25	97.83	-	-	-	-
Roy et al. (2019a)	246	3 Frequency	-	KNN	5-Fold Cross-Validation	-	-	-	-	90.70	-	-
			-	SGD		-	-	-	-	77.50	-	-
			-	XGBoost		-	-	-	-	79.60	-	-
			-	Adaboost		-	-	-	-	70.70	-	-
			-	CNN		-	-	-	-	72.30	-	-
Vanabelle et al. (2020)	36	10 Time 12 Frequency	-	XGBoost	Leave-One-Patient-Out Cross-validation	-	78.72	76.41	33.54	47.04	-	-
George et al. (2020)	??? (150 files/24608s each for generalized, focal, normal)	3 Entropy/Time-Frequency	Particle Swarm Optimization	ANN	33% Holdout	(normal-focal) 95.10	-	-	-	-	-	-
						(normal-generalised) 97.4	-	-	-	-	-	-
						(normal-focal + generalised) 96.2	-	-	-	-	-	-
						(normal-focal-generalised) 88.8	-	-	-	-	-	-
Li et al. (2020)	?? (1983 Seizures)	1 Time-Frequency	-	SVM	5-Fold Cross-Validation	85.69 (0.73)	-	-	-	86.85 (0.87)	-	-
		1 Time-Frequency		SVM		90.89 (0.35)	-	-	-	91.90 (0.70)	-	-
				ResNet18		79.58 (0.68)	-	-	-	75.36 (1.51)	-	-
				ConvNet		88.03 (0.68)	-	-	-	89.40 (0.41)	-	-
				CE-stSENet		92.00 (0.15)	-	-	-	93.69 (0.33)	-	-
Liu et al. (2020)	314	1 Frequency	-	CNN	Stratified 5-Fold Cross-Validation	-	-	-	-	95.5	-	-
				RNN		-	-	-	-	95.8	-	-
				B-CNN		-	-	-	-	96.7	-	-
				B-RNN		-	-	-	-	96.9	-	-
				Hybrid		-	-	-	-	97.4	-	-
Zhang et al. (2020)	14	-	-	SVM	Leave-One-Patient-Out Cross-validation	64.3	-	-	-	-	-	-
				RF		61.9	-	-	-	-	-	-
				KNN		69.9	-	-	-	-	-	-
				CNN		80.5	-	-	-	-	-	-

Table 2.A.4: Abbreviations for tables 2.A.1 & 2.A.3.

AB	Ansari-Bradley Test	ANOVA	Analysis of Variance
BiLSTM	Bidirectional Long Short-Term Memory	BPNN	Back-Propagation Neural Networks
CMIM	Conditional Mutual Information Maximization	CNBC	Collective Network of (Evolutionary) Binary Classifiers
CNN	Convolution Neural Network	CP	Changepoint
CSP	Common Spatial Patterns	DT	Decision Tree
DAE	Denoising Autoencoders	DMD	Dynamic Mode Decomposition
ELM	Extreme Learning Machine	FC	Fuzzy Classifier
FR	Feature-Ranking	FRB	Fuzzy-Rules-Based Feature Selection
FS	Forward Selection	FT	Functional Tree
GA	Genetic Algorithms	GMM	Gaussian Mixture Model
GS	Gram-Schmidt Analysis	HMM	Hidden Markov Model
ICA	Independent Component Analysis	KNN	K-Nearest Neighbour
LDA	Linear Discriminant Analysis	LDAB	Linear Discriminant Analysis (Backwards Search)
LDAF	Linear Discriminant Analysis (Forward Search)	LDAG	Layered Directed Acyclic Graph
LDAi	Independent Search	LSTM	Long Short-Term Memory
LW	Lambda of Wilks	POLYC	Polynomial Classifier
LC	Linear Classifier	LDA	Linear Discriminant Analysi
LOG	Logistic Regression	MLP	Multilayer Perceptron
MOEA	Multi-Objective Evolutionary Algorithm	MV	Majority Vote
NB	Naive-Bayes	NN	Neural Network
PARZEN	Parzen Classifier	PCA	Principal Component Analysis
PRNN	Pattern Recognition Neural Network	PSO	Particle Swarm Optimisation
QC	Quadratic Classifier	QDA	Quadratic Discriminant Analysis
QDC	Quadratic Discriminant Classifier	RBF	Radial Basis Kernel
RCNN	Recurrent Convolutional Neural Network	RF	Random Forest
RFE	Recursive Feature Elimination	RNN	Recurrent Neural Network
RVM	Relevance Vector Machine	RIPPER	Repeated Incremental Pruning to Produce Error Reduction
SA	Stacked Autoencoders	SAE	Sparse Autoencoder
SG	Skip-Gram	SLM	Statistical Language Modeling
SS	Statistical Significance	SSAEs	Stacked Sparse Autoencoders
SVM	Support Vector Machine	T-Test	Student's T-Test
TTL-FSs	Transductive Transfer Learning Fuzzy Systems	UDC	Uncorrelated Normal Density Classifier
W-FPE-F	K-Means Feature Weighting and FPE-Complexity Degree	WSR	Wilcoxon Signed-Rank Test

Chapter 3

Automatic Detection of Absence Epilepsy Seizures in Paediatric Electroencephalography Records

3.1 Introduction

Epilepsy is the tendency to have unprovoked and recurrent seizures, which are often accompanied by an alteration of consciousness (Giourou et al., 2015; Krumholz et al., 2007). Clinical manifestations of epilepsy are dependent on several factors; such as the particular epilepsy syndrome, patient age, the brain area that generates seizures, and if discharges remain local or propagate to other brain areas (Giourou et al., 2015). Whilst all seizures result from an increase in cellular excitability, the mechanisms of synchronization differ between seizures, broadly categorising them as focal or generalized epilepsies. This research focuses on the detection of generalized epileptiform discharges using in-clinic diagnostic assessment records from patients diagnosed with absence epilepsy. Absence epilepsy constitutes around 10% of paediatric epilepsy patients (Hughes, 2009; Tanaka et al., 2008), and is characterised by 9- to 12-seconds bilateral 3Hz generalized spike-and-slow wave discharges of electrical activity generated from firing neurons (Hughes, 2009). Absences may be termed "frontal absences" to acknowledge the fast propagation of a seizure generated from a frontal fo-

cus (Holmes et al., 2004); as although historically categorised as "primarily generalized" in nature, there is an increasing acceptance that seizures originate in local ictogenenic microcircuits which propagate to other areas (Paz and Huguenard, 2014; Holmes et al., 2004). The clinical symptoms include a blank stare, interrupted activities, slowed speech, and upward rotation of the eyes (Sakkalis et al., 2013). Hyperventilation activates a seizure in a majority of childhood patients, whereas phoneic stimulation is only associated with a minority (Durá Travé and Yoldi Petri, 2006). Particularly when the onset is in early adolescence, generalized tonic-clonic seizures (GTCS) may also occur, with the occurrence of these seizures associated with a worse prognosis (Tovia et al., 2006). Furthermore, sometimes polyspikes appear before the onset of classic 3Hz generalized spike-and-waves; with these patients suggested to be associated with an intermediary form of idiopathic generalized epilepsy that may be drug resistant (Tatum et al., 2010).

The diagnosis of epilepsy relies on the identification of clinical features specific to a particular epilepsy syndrome. Electroencephalography (EEG), magnetic resonance imaging (MRI), and verbal descriptions of seizures are the most commonly available information to neurologists; with hospital records, seizure diaries, and videos of patient events desirable (Bidwell et al., 2015). In-clinic scalp EEG is useful for diagnosis as it provides a non-invasive method to characterise the mean electrical activity generated by the synchronous firing of open field neurons at a high temporal resolution. Typically, in the UK National Health Service (NHS), patients have an approximately 30-minute scalp EEG assessment, during which the patient may be asked to hyperventilate and exposed to photic stimulation to provoke a seizure. If a diagnosis is suspected, but not gained, a patient may then have a longer EEG assessment. Human experts, trained to qualitatively assess EEG records, will look for seizure-like oscillations that occur for a long duration and over a number of channels; however this manual review of EEG is time consuming, expensive, and prone to error (Varsavsky et al., 2011a). Indeed, Wilson et al. (2003) found less than 80% of events were similarly identified between two or more experts on a previously marked EEG record and misdiagnosis rates generally are estimated to be between 20-30% in developed countries (NICE Clinical Guidelines and Evidence Review for the Epilepsies, 2004).

The automation of seizure detection in EEG records has been investigated since the early

1970s (Tzallas et al., 2012), with modern approaches relying less on heuristic strategies with a select number of features and more on a variety of signal features with machine learning classification strategies. The majority of feature extraction methods for EEG signals can be segmented into categories of time, frequency, and time-frequency analysis methods. In time domain analysis, the mean (Mitha et al., 2014; Harpale and Bairagi, 2018), variance (Kiranyaz et al., 2014; Tsiouris et al., 2017), kurtosis (Fergus et al., 2015; Awan et al., 2016), and skewness (Elmahdy et al., 2015; Hamdan et al., 2015), are often used as basic statistics for defining the characteristics of EEG waveforms. Frequency domain characteristics are used to describe the power variations in brain waves at different frequencies, and are often reduced to mean (Pramod et al., 2014; Van Esbroeck et al., 2016), median (Zabihi et al., 2013; Fergus et al., 2016), and peak frequency (Hamdan et al., 2015; Fergus et al., 2016) features. Time-frequency analysis simultaneously extracts time and frequency domains to more effectively characterise non-stationary EEG signals. Similarly, these are often reduced to basic statistics such as mean (Javaid et al., 2015; Alickovic et al., 2018), standard deviation (SD; Ibrahim and Majzoub, 2017), and energy (Rafiuddin et al., 2011; Chandel et al., 2017). These features are often calculated in windowed sections of the data, sometimes followed by dimension reduction and extraction techniques, to reduce the complexity of the non-linear EEG data for classification.

As well as feature extraction, there has also been a broad range of classification approaches that have been applied (for review see Varsavsky et al., 2011a; Roy et al., 2019b). However, there is little consensus on the best features and classification pipeline to use, particularly for specific seizure types. Features commonly differ between papers, however it is increasingly common that wavelets are used (e.g. Alickovic et al., 2018). Feature selection or dimension reduction techniques are not commonly used but, where present, the features that were selected are often not reported or investigated. It is also common that one type of classifier is predominately focused upon, with some papers failing to provide adequate performance comparisons for other classifiers using the same features, data, and evaluation methods. Evaluation methods can vary between papers, with differing amounts of data held-out or cross-validation folds, or whether complete patient records are separated into different training and test sets or mixed and separated based on the proportion of seizures

in the data; with the former approach typically leading to worse performance but a more naturalistic testing environment.

This research compares three common "classical" classifiers (k-nearest neighbors, random forest, support vector machine), separately and grouped in an ensemble. Each classification pipeline has the same features extracted for each record, but can vary based on if there are preceding feature selection or dimension reduction steps before the classifier. Bayesian optimisation is used both to search for optimal components of the classification pipeline, and for optimal hyperparameters of these components. This is because often hyperparameters are chosen based on manual model test runs, previous literature that is not well documented or justified, computationally expensive grid search methods. By finding optimal hyperparameters, comparisons between models become more objective. This research therefore aims to provide a clearer understanding of features and classification pipelines that provide the best performance for seizure detection on real-world data, tested using a method similar to how they would be used in practice on unseen patient records (leave-one-patient-out cross-validation).

This chapter is structured as follows: in section 3.2 we describe how the NHS records were prepared for feature extraction and subsequent classification. Section 3.3 then presents the features extracted in each EEG channel, the machine learning classifiers investigated, the hyperparameter optimisation method, how model performance was assessed, and additional data handling information. Section 3.4 subsequently describes the validation and test set results, examines the best performing classifiers and the important features used for classification, and finally how performance can be improved using prediction post-processing. We then examine whether performance can be improved by training multi-class, as opposed to binary ictal/interictal classifiers, to also identify artefact labels. Finally, sections 3.5 and 3.6 discuss our findings and present our conclusions.

3.2 Data Preparation

EEG records from 21 paediatric patients (aged 4-13 yrs, mean age = 8.6) diagnosed with absence epilepsy were obtained from Royal Preston Hospital in the UK. The EEG is mostly

recorded at 256 samples per second (512Hz for P5 and P19) using an Xltek 18 channel system, with electrodes configured using the International 10-20 system. During data collection, a common reference was used for all electrodes (monopolar montage), with the reference electrode placed on the midline either between Cz and Fz or between Cz and Pz (see figure 3.2.1). A low impedance high conductive gel (Nuprep) was used to gently scarify the skin and a conductive paste (Ten20) applied to hold the electrodes in place. Patients underwent a routine clinical EEG assessment, lasting approximately 30 minutes, and were asked to hyperventilate and exposed to photic stimulation to provoke a seizure. Data from these sessions were anonymised and burned to a CD by a clinical physiologist after being used for diagnostic purposes. The raw data, with accompanying clinical physiologist notes, were accessed using a portable version of Natus NeuroWorks EEG software. The data was exported into .txt files to be processed, and later analysed, in Jupyter Notebooks (5.7.0) using Python 3 (3.6.8). Signals were loaded into an `MNE` (0.17.0; Gramfort et al., 2013; Alexandre Gramfort et al., 2014) object and key terms in the physiologist notes, such as "unresponsive" and "absence", were used to aid visual labelling of the EEG record appropriate for training machine learning models. The duration and labels associated with the visually assessed data segments were created by the researchers, and reviewed by a Consultant Neurophysiologist, using both the average of all the electrodes and A-P bipolar as re-reference montages (see table 3.2.1).

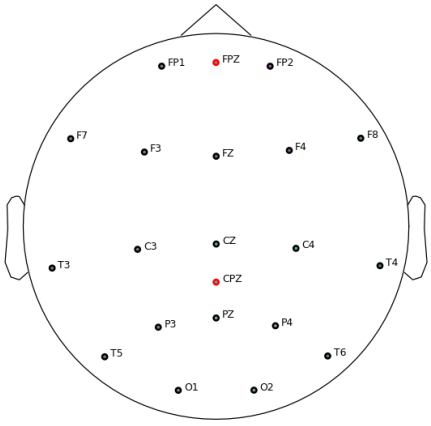
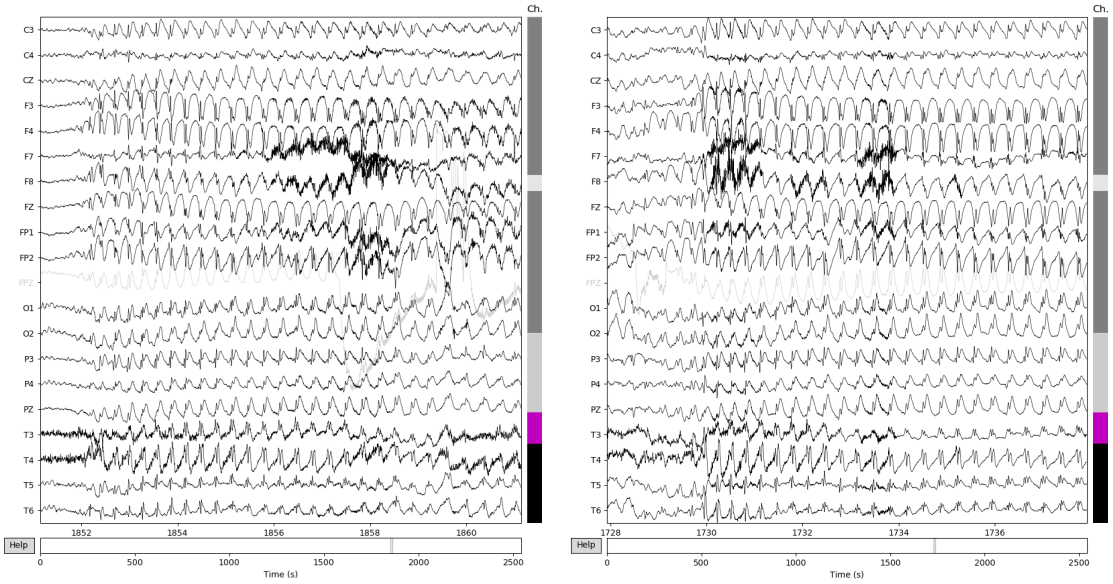


Figure 3.2.1: EEG channel locations for NHS diagnostic procedure.
Note. CPZ was the recording reference and FPZ was not used.

Table 3.2.1: A-P EEG bipolar montage

FP2-F8	F8-T4	T4-T6	T6-O2
FP2-F4	F4-C4	C4-P4	P4-O2
FP1-F3	F3-C3	C3-P3	P3-O1
FP1-F7	F7-T3	T3-T5	T5-O1

Data segments, of varying length, were assigned one of five labels; "Generalized Epileptiform Discharge" (576 secs; 1.45%), "Notched Rhythmic Waveforms" (213 secs; 0.54%), "Spikes" (14 Secs; 0.04%), "Artefact" (9,123 secs; 22.96%), and "AMPSAT" (10,891 secs; 27.41%). All data that was not marked, was assigned a "Baseline" label (18,911 secs; 47.60%) to represent interictal EEG with no content of interest (see tables 3.A.1 and 3.A.2). Labelling was not conducted on a second-by-second basis, instead sections of the data that fell into the above categories were labelled for their onset and offset to the nearest millisecond to provide a more naturalistic labelling than binned windows. "Generalized Epileptiform Discharge" encompass EEG data where there are spike-and-wave discharges, sometimes proceeded by polyspikes (see figure 3.2.2). Segments marked as "Notched Rhythmic Waveforms" represent benign EEG activity, likely a result of the patient being in a state of drowsiness (Britton et al., 2016). "Spikes" represent events that in isolation would be unlikely to be used as a diagnostic marker, as they could be epileptiform activity or just benign EEG. "Artefact" segments represent data likely representing physiologic or extraphysiologic electrical phenomena which distorts the neural signal; from respiratory, eye movement, muscle, or environmental sources. Segments of the data with amplifier saturation ("AMPSAT") resulted in missing data or extremely high amplitude signals. These sections of data, typically at the start of the recording, are caused by the technician adjusting the electrodes or inputting a small measurement voltage into the EEG device in order to determine the signal quality of the electrodes (impedance test). As these segments are not representative of electrical signals produced by the brain, or part of the data assessed for seizures by a physiologist as they reflect the technician configuring the diagnosis equipment, these were all filled in with missing data (`NumPy` NaN) to keep the time series consistent throughout the record. After feature extraction, sections of "AMPSAT" were removed before training the classification pipeline.



(a) Generalized seizure

(b) Generalized seizure with preceding polyspikes

Figure 3.2.2: Generalized epileptiform discharges in the P4 record.
Note. Plotted using an average reference with the `MNE` package.

The `MNE` objects were converted to `Pandas` (0.24.2; McKinney, 2010) dataframes and separated into `NumPy` (1.15.4) arrays for the raw signal data, labels, and a list of feature names. The events in each dataset were assigned an integer. For binary classification, "Notched Rhythmic Waveforms", "Spikes", "Artefact", and "Baseline" data labels were encoded as 0, to represent interictal periods, and "Generalized Epileptiform Discharge" as 1 (ictal). "Notched Rhythmic Waveforms" and "Spikes" were included with the baseline data due to the limited number of events (see table 3.A.1).

3.3 Methods

In this section, we start by describing the features extracted in each EEG channel. We then describe how Bayesian optimization is used to search over pipeline components and model hyperparameters. Subsequently, a description of how performance was assessed during training and on left-out patient datasets is given. Finally, we give a description of the hardware and software used to implement the training and testing paradigm.

3.3.1 Feature Extraction

For this study, the data was epoched into window sizes of 2 seconds with a 1 second overlap (e.g. Henriksen et al., 2010; Duun-Henriksen et al., 2012b), to create quasi-stationary segments of EEG. For each window, a number of features were calculated for each electrode (see table 3.3.1). Firstly, the median power in several frequency bands (1-4Hz, 4-8Hz, 8-12Hz, 12-30Hz, 30-70Hz) was calculated using the Welch method for spectral density estimation (Welch, 1967) from the `SciPy` library (1.2.1; Virtanen et al., 2020). Welch's method divides the data into overlapping segments and computes an average periodogram across the segments using a fast Fourier transform (FFT). We used a Hann window with a length of 1 second to ensure there was at least 2 cycles of the lowest frequency (1Hz) in the epoch. We also used this method to get a general power measure for the 1-30Hz band, as well as the relative power between the 3-12Hz and 1-30Hz bands (Kjaer et al., 2017). The `PyWavelets` package (1.0.1; Lee et al., 2019) was used to apply a discrete wavelet transform, using the Daubechies 4 (db4) wavelet family, to split data into 6 sub-bands: d1, 128-64 Hz (Gamma); d2, 64-32 Hz (Gamma); d3, 32-16 Hz (Beta); d4, 16-8 Hz (Alpha); d5, 8-4 Hz (Theta); d6, 4-2 Hz (Delta). A wavelet transform projects the data onto several oscillatory kernels to gain different frequency components which can be analysed in respect to their scale (Kiymik et al., 2005; Sakkalis et al., 2008, 2006). Wavelets are scaled and shifted temporally

Table 3.3.1: Features extracted from patient records in multiple domains, frequencies, and channels.

Time	**Frequency**		**Time-Frequency**	
Feature	*Frequency (Hz)*	*Feature*	*Frequency (Hz)*	*Feature*
Correlation Coefficients	-	Correlation Coefficients	$2-4$	Mean
Eigenvalues	-	Eigenvalues		Standard Deviation
	$1-30$	Median Power		Log Sum
	$1-4$	$\cdots$		Mean Absolute
	$4-8$	$\cdots$	$4-8$	$\cdots$
	$8-12$	$\cdots$	$8-16$	$\cdots$
	$12-30$	$\cdots$	$16-32$	$\cdots$
	$30-70$	$\cdots$	$32-64$	$\cdots$
	$3-12/1-30$	Relative Power	$64-128$	$\cdots$
			$2-4/4-8$	Ratio
			$4-8/(\frac{2-4+8-16}{2})$	$\cdots$
			$8-16/(\frac{4-8+16-32}{2})$	$\cdots$
			$16-32/(\frac{8-16+32-64}{2})$	$\cdots$
			$32-64/(\frac{16-32+64-128}{2})$	$\cdots$
			$64-128/32-64$	$\cdots$

so are often used to characterise non-stationary signals (Varsavsky et al., 2011a). In each sub-band, the mean and SD of the coefficients, the mean absolute power, and the ratio of the mean absolute values of adjacent sub-bands were calculated (Subasi, 2007b; Alickovic et al., 2018); as well as the log-sum energy of the sub-band coefficients (Petersen et al., 2011; Duun-Henriksen et al., 2012b; Kjaer et al., 2017). Additionally, a FFT was applied to get the frequency magnitudes in the 1-47Hz range, for which correlation coefficients and eigenvalues were calculated in the time and frequency domains (Baldassano et al., 2017; Roy et al., 2019a) using functions in the `NumPy` library (Van Der Walt et al., 2011). For each patient, all the above features were combined and stored in a .hdf5 file with patient identifiers used as keys.

3.3.2 Signal Classification

Three "classical" machine learning classifiers were investigated to separate the windowed EEG signals into ictal and interictal classes; k-nearest neighbour (subsubsection 2.6.1), support vector machine (subsubsection 2.6.1), and random forest (subsubsection 2.6.2). A dummy classifier was also trained as a simple baseline for comparison, although results from this model are only subsequently presented visually. This dummy model makes a prediction of a class label based on a simple rule, the training set's class distribution, therefore is not considered a "learning" algorithm.

3.3.3 Optimisation and Cross-Validation

A Bayesian optimisation method, using the `fmin` function from the `Hyperopt` package (0.2; Bergstra et al., 2013), was used to search over classification pipeline components and model hyperparameters for each classifier. We used this approach to address the lack of consensus regarding optimal pipeline components or hyperparameter values in the current literature. The search space (see table 3.3.2) begins with a random combination of components and hyperparameters, which are optimised over 1000 iterations. The objective function, the mean F1-score from a stratified 5-fold cross-validation, was used at each iteration to update a prior from a history of model configuration and score pairs. For Bayesian optimisation, a probability model $P(score|configuration)$ is used to search for the most promising candi-

Table 3.3.2: Hyperparameter spaces for different pipeline components.

Pipeline Step	Algorithm	Hyperparameter	Parameter Space
Feature Selection	None	-	-
	SelectFromModel(RF)	Max Features	randint(1, 541)
Dimension Reduction	None	-	-
	PCA	Number of Components	uniform(0.05, 1.0)
Classification	Dummy Classifier	-	-
	KNN	Nearest Neighbors	randint(1,10)
		Algorithm	choice(ball tree, kd tree, brute)
		p	randint(1,10)
		Leaf Size	normal(m=30, sd=8)
	SVM	C	uniform(0.05, 8)
		Kernel	choice(linear, rbf)
		Gamma	uniform(0.005, 2)
	RF	Number of Estimators	normal(m=2000, sd=500)
		Criterion	choice(Gini, Entropy)
		Max Depth	choice(None, randint(1, 50))
		Min Samples Split	uniform(0.01, 1.)
		Max Features	uniform(0.01, 1.)

Note. choice: choose one; randint: random integer; normal: normal distribution; uniform: value selected randomly between lower and upper bounds; lognormal: exponential of the normal distribution so the logarithm of the return value is from a normal distribution.

dates and is therefore quicker than evaluating all possible combinations (e.g. grid search). Gaussian processes (Williams and Rasmussen, 2006) or regression models, such as decision trees (Bergstra et al., 2012), can be used for modelling the probability; here we use a Tree of Parzen Estimators (TPE; Bergstra et al., 2011) algorithm. The TPE fits a Gaussian Mixture Model (GMM; $l(x)$) to parameters associated with the smallest loss function values, and another GMM, $g(x)$, to the remaining values to choose a parameter value, x, that maximises the ratio $l(x)/g(x)$ (Bergstra et al., 2013). The model implementation weights recent trials more than older trials and varies the fraction of trials used to estimate $l(x)$ and $g(x)$. The TPE algorithm provided by the `Hyperopt` package was chosen as it can handle real-valued, discrete and conditional variables, as well as being able to optimize large-scale hyperparameter optimization problems (Bergstra et al., 2015). Furthermore, TPEs allow for tree-structured dependencies for hyperparameters; for example, a Gamma hyperparameter value can only be selected if the SVM kernel is chosen to be a RBF rather than linear (NeuPy, 2019).

Separately for each classifier, at each step of the Bayesian optimisation training, features could be selected using a RF (model stacking), extracted using PCA, or both. Model stacking, where the input to one model is the output of another, can be used to capture non-linearities in a complex model which is followed by a more efficient linear classifier. Model stacking was implemented for feature selection by using the ranked importance of

features; calculated using the average impurity decrease from a RF model. Although the number of features to be selected from this RF model could vary between 1 and half the available features, the rest of the hyperparameter space for the RF when used for feature selection was fixed to be the same as Birjandtalab et al. (2017); Gini impurity as the criterion for splits, 1000 separate trees, and the maximum features selected at each node being set to the square root. For models using SVM as a classifier, the provided range of values for the penalty (C), kernel, and gamma hyperparameters were based on previous research (see Kjaer et al., 2017). Most other hyperparameters across the PCA, KNN, and RF pipeline components were set to cover all available options or centred around the software default; with the exception being the number of estimators for RF models, which were based on models in Baldassano et al. (2017) when used in the classifier step of a pipeline. The data was standardised to mean 0 and variance 1, to remove the mean and scale to a unit variance, if PCA, SVM, or KNN were in the model. At each Bayesian optimisation iteration and stratified k-fold, pipelines were trained on an undersample of the data to balance the number of ictal and inter-ical data during training (e.g. Roy et al., 2019a; Iešmantas and Alzbutas, 2020; Zhang et al., 2019). Indeed, it has been demonstrated that undersampling tends to outperform more advanced methods for dealing with extreme class imbalance (e.g. SMOTE; Wallace et al., 2011; Hulse and Khoshgoftaar, 2007). Reducing the proportion of ictal and intra-ictal activity is required for most models, not just because of its effects on decision boundaries, but also as models are slower to train as the number of training examples, n, increases; particularly for SVM models (Bottou and Lin, 2007). For each training data set, the Bayesian optimisation process was run 2 times to test if similar models performed best using different undersampled data, random states, and starting parameters (see figures 3.A.1-3.A.4). By changing the starting values, so they begin at different areas in the parameter space, we could check on chain convergence.

3.3.4 Performance Evaluation

Models were trained and tested using a leave-one-patient-out cross-validation scheme, which trained models on data from all the patients except one. Performance is measured on the patient that was left out to replicate how the models would be used in practice. The best

pipelines for each classifier (SVM, RF, KNN) on each held-out dataset were selected and re-trained on the entire randomly undersampled training data it was previously cross-validated on, as during cross-validation it was only trained on 4/5th of training data at each fold. These models were then also grouped into a soft voting ensemble (SVE), which votes using class probabilities, and re-trained on new undersampled training data. This is to check if the combination of these models was better than each model individually, which can occur when models in an ensemble are diverse. A SVE specifically often performs better than a hard voting ensemble because more weight is given to cases where models are more certain (Géron, 2019). A number of evaluative metrics were used to assess the final performance of each model on each held-out patient test set:

Accuracy is used to give general performance information regarding the number of all correct (True Positive, TP; True Negative, TN) or false (False Positive, FP; False Negative, FN) predictions comparative to the total number of predictions. Positive in the context of binary seizure classification refers to ictal (seizure) activity, whereas negative refers to interictal activity.

$$Accuracy = 1 - \frac{FP + FN}{FP + FN + TP + TN} \tag{3.1}$$

Accuracy is a common metric for supervised classification to reflect general model performance, however is less focused upon for imbalanced datasets as it will likely over-represent the true positives for the negative class. For example, a seizure detection model may have a high accuracy but only produce "naive behaviour" by always labelling each window as interictal. So despite often being reported (e.g. Alkan et al., 2005; Subasi, 2007a; Liang et al., 2010; Zeng et al., 2016), it should be viewed as complimentary to other metrics in this thesis.

Precision compares the total number of positive predicted labels from a model to the number of true positives.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

Precision is a good measure to use when the costs of a false positive is high. Therefore in a clinical decision support system, where false positives could be identified by a human interpreter (physiologist) reviewing a models detections, it is less central than other metrics.

Sensitivity (or True Positive Rate/Recall) calculates how many true positives a model correctly labelled.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.3}$$

This is particularly useful when the fraction of correct or misclassified samples in the positive class are of interest. This is the most commonly reported metric for seizure detection (e.g. Petersen et al., 2011; Duun-Henriksen et al., 2012b) as it focuses on the ability of the model to detect seizures correctly.

Specificity (or True Negative Rate) calculates how many of the actual negatives our model correctly labelled.

$$Specificity = \frac{TN}{TN + FP} \tag{3.4}$$

Although this metric focuses on the class of less interest (interictal), in the context of seizure detection, it is often reported to compliment the sensitivity metric.

F1-score is a combination of sensitivity and precision metrics.

$$F1 = 2 * \frac{PRE * REC}{PRE + REC} \tag{3.5}$$

F1-score is commonly used when there is an uneven class distribution due to a large number of true negatives and where false negatives and false positives are focused upon. However, despite its clear applicability to seizure detection, it is rarely reported.

ROC AUC is an abbreviation for the area under the curve (AUC) of a Receiver Operating Characteristic Curve (ROC). ROC AUC plots the true positive rate against the false positive rate (FPR) at different thresholds.

$$FPR = \frac{FP}{FP + TN} \tag{3.6}$$

ROC AUC is useful for summarising the overall accuracy of the model. As it focuses on the accuracy of the model, it is also a less useful metric in this case, with mixed reporting in the literature.

False detection rate (FDR/h) is a commonly reported metric in the seizure detection literature, representing the number of false positives that occur per hour. Because the recordings in the dataset used in this research are typically shorter than an hour, we first calculate the proportion of an hour the length of the data is and divide it from the number of false positives.

$$FDR/h = \frac{FP}{SECS/3600} \tag{3.7}$$

FDR/h is the only metric not bounded between 1 (best) and 0 (worst).

3.3.5 Data Handling

Feature selection, extraction, and classification pipeline steps all used functions from the `Scikit-learn` (0.20.2; Pedregosa et al., 2011) and `Imblearn` (0.4.3; Lemaitre et al., 2017) packages. Feature extraction and test set predictions were made in serial on a Dell XPS 13 9370 laptop using an Intel® Core™i7-8550U CPU with 16GB RAM. For each model classification pipeline (Dummy, KNN, SVM, and RF), on each hold-out training dataset, serial model training occurred on the Lancaster High End Computing cluster (HEC) using a single core and 2 gigabytes of memory per core, running separately and simultaneously across many cores.

Feature extraction for records was relatively quick considering the total number of features extracted per record (1083), due to the simplicity of the features chosen to be calculated (mean = 19.57s, SD = 6.68s), and the short length of each recorded session (mean = 31.53 mins, SD = 10.29 mins). Due to 4 models being trained on each of the 21 patients separately (leave-one-patient-out), with 1000 Bayesian hyperparameter optimisations, each configuration assessed using the mean F1-score from a stratified 5-fold cross-validation, and 2 separate iterations of the whole training paradigm with different random states, there were a total of 840000 models trained for binary classification (168,000 unique pipeline/hyper-parameter configurations).

3.4 Results

In this section we start by describing the validation results gained during training. We then look at the performance of the best pipelines for each classifier across held-out patient datasets. For each pipeline that contained a feature reduction step, we then examine the most commonly selected features used to aid classification. We then look at how performance can be improved using post-processing, so that seizure predictions are only kept if they occur sequentially for a given length of time. As the most common cause of false positives was artefactual EEG, we then examine whether seizure detection performance is improved by training multi-class classifiers that separate features into ictal, interictal, and artefact labels. Patient results are typically displayed in average, with further patient-by-patient details available in the supplementary information document (https://bit.ly/3bZQxop).

3.4.1 Binary Classification

The highest scoring pipelines on the validation data, as measured by the F1-score during training optimisation, ended with a KNN classifier (see figure 3.4.1). Pipelines ending with a KNN classifier were also the most consistent across the two separate Bayesian optimisation iterations, both in terms of pipeline components (see figure 3.4.1) and optimal hyperparameters (see figure 3.A.2). Conversely, pipelines using a RF classifier were the lowest scorers and those ending in SVM classifiers had the most variation for pipeline components (see

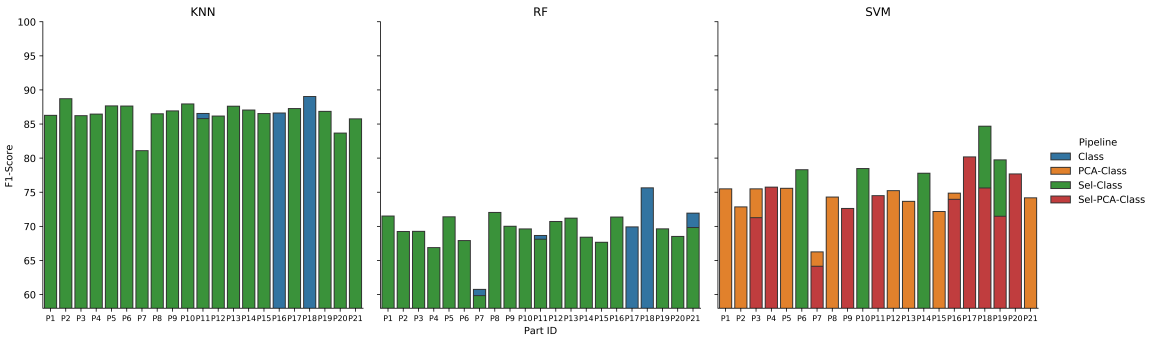


Figure 3.4.1: Best validation F1-score for binary classification when individual patient datasets were left out.
Note. Overlapping bars are plotted where different pipelines were optimal between the two Bayesian optimisation iterations.

figure 3.4.1) and optimal hyperparameters (see figure 3.A.3).

As can be seen in table 3.4.1, across the held-out patient test sets, pipelines with a KNN classifier generally scored better across performance metrics comparative to other pipelines. FDR/h was particularly high for pipelines with RF classification comparative to other models, and is high compared to the broader literature (see table 3.5.1). Relative to KNN, only sensitivity, AUC, and time taken to predict test set labels were better in other pipelines.

For each best pipeline that used feature extraction as a step, the features that were selected to reduce the data dimensionality were collected. On average, 325.23 (SD = 177.77) out of the 1083 features were selected across the models. Pipelines that had KNN classification selected the most features and SVM the least (see table 3.4.2); indeed three "optimal" SVM models used only 2 features (see figure 3.A.9). To investigate which features were commonly selected across the different pipelines, we first added together the occurrence of each feature to get the number of times it was selected. To reduce this further, and focus on the most commonly selected features, we used an implementation of `cPOP` (Fearnhead et al., 2019), using `rpy2`, to determine a threshold beyond which the feature counts significantly reduce in times selected (see figure 3.4.2a). `cPOP` aims to detect multiple changes in slope by finding the "best" continuous piecewise linear fit of the data by minimizing a square error loss plus an L_0 penalty. Although the original paper found a penalty of $2logn$ worked well for simulations, we used $8logn$ to reduce the number of changes detected even further and to account for the autocorrelation between features selected. The smallest threshold, determined by the smallest changepoint, was selected in order to find the best features and channels. As displayed in figures 3.4.2b and 3.4.2c, features predominately in frontal channels, and covering slower oscillation frequencies, were more commonly selected. However, these were not restricted to the frequency range associated with absence seizures (3/4Hz), demonstrating a broad range of frequencies may be useful for classifying absence seizures. They also appear to be predominately time-frequency (mean of the wavelet coefficients or mean absolute power) or frequency (median power) based features.

The majority of performance metrics can be improved with post-processing of the predictions. Post-processing is often used to reduce the number of false positives generated from

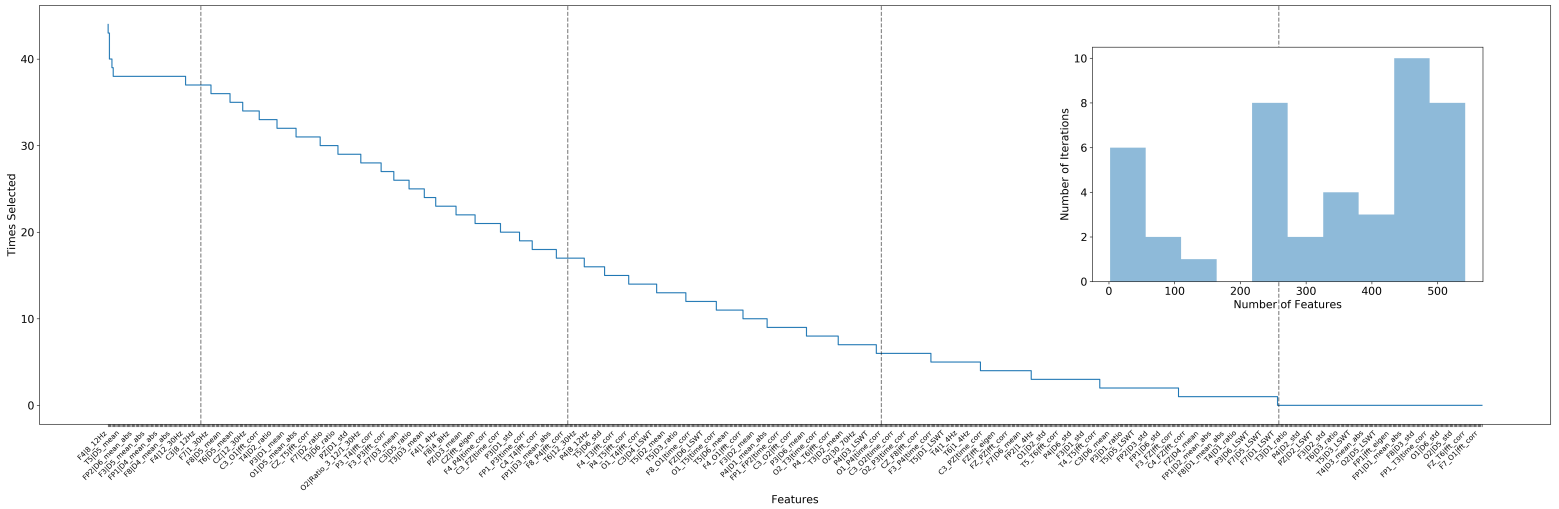
Table 3.4.1: Average (and standard deviation) test scores for binary classification.

Classifier	Accuracy		Sensitivity		Specificity		Precision		F1-score		AUC		FDR/h		Prediction Time (secs)	
	Mean	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***
KNN	**98.73**	(2.26)	91.49	(7.94)	**98.84**	(2.32)	**85.12**	(15.61)	**86.91**	(7.94)	96.74	(2.67)	**40.91**	(82.89)	1.79	(0.54)
RF	92.63	(21.55)	80.95	(36.19)	93.38	(21.59)	55.23	(34.71)	62.87	(33.27)	91.26	(18.38)	237.21	(777.42)	0.44	(0.27)
SVM	97.94	(3.2)	91.37	(7.97)	98.07	(3.24)	77.03	(26.87)	79.74	(22.18)	**99.22**	(1.09)	68.61	(115.92)	**0.07**	(0.04)
SMV	98.14	(3.16)	**94.96**	(4.74)	98.21	(3.24)	79.68	(23.6)	84.23	(16.86)	98.98	(1.56)	63.5	(115.4)	2.29	(0.57)

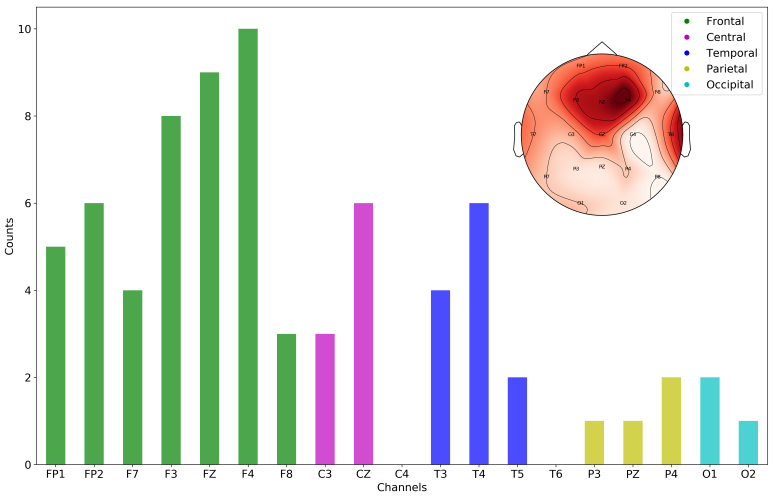
Note. The best average score for each metric, across classifiers, are in bold.

Table 3.4.2: Most common pipeline steps for binary classification across all datasets with either the most common categorical or average (and standard deviation) hyperparameter value.

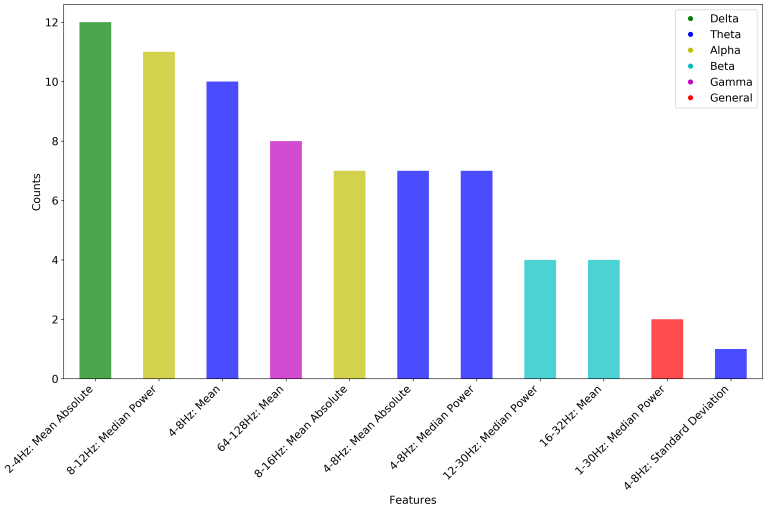
KNN			**RF**			**SVM**		
Steps	*Hyperparameters*	*Average Value (SD)*	*Steps*	*Hyperparameters*	*Average Value (SD)*	*Steps*	*Hyperparameters*	*Average Value (SD)*
1. Random Undersample	-	-	1. Random Undersample	-	-	1. Random Undersample	-	-
2. Feature Selection	Num Features	469.94 (52.27)	2. Feature Selection	Num Features	276.62 (106.90)	2. Feature Selection	Num Features	142.5 (208.79)
3. Scale	-	-	3. Classifier	Criterion	Entropy	3. Scale	-	-
4. Classifier	Algorithm	k-d tree		Number of Estimators	889.81 (651.27)	4. Classifier	Kernel	Linear
	Leaf Size	41.19 (25.49)		Max Depth	15.43 (15.32)		C	2.97 (2.75)
	Nearest Neighbours	2 (0)		Minimum Samples Split	0.21 (0.27)		Gamma	0.09 (0.07)
	Distance Metric	Manhattan		Max Features	0.45 (0.35)			



(a) Number of times features were selected across the best models with cPOP slope change detections.



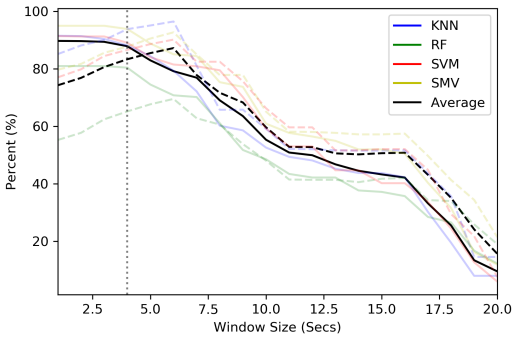
(b) Number of selected features in each EEG channel with topoplot.



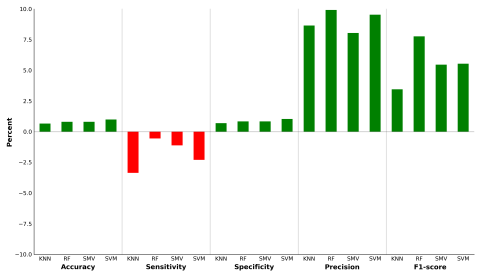
(c) Number of times a feature was selected across channels.

Figure 3.4.2: Most common EEG channels and features selected for when a feature selection component was in a binary classification pipeline.

Note. Channels and features were reduced using the smallest cPOP threshold.



(a) Post-processing window length sensitivity (solid) and precision (dashed) scores for binary classification predictions.



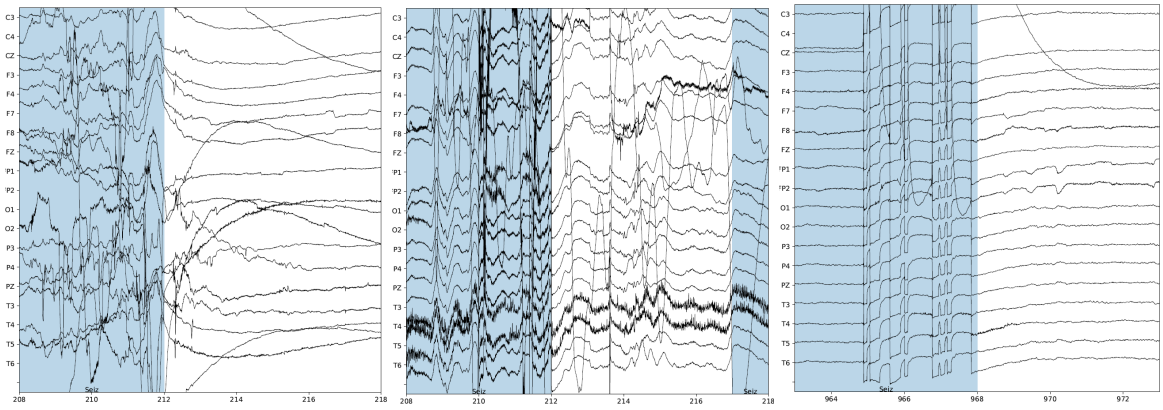
(b) Average test set score change (%) from a four second long post-processing window on binary model predictions.

Figure 3.4.3: Affect of prediction label post-processing on binary classification performance metrics.

a models predictions (e.g. Duun-Henriksen et al., 2012b). In this work, to post-process the pipeline predictions, we kept predictions only when they consecutively predicted a seizure each second for the length of a given window size. Figure 3.4.3a depicts the relationship between the minimum size of this window and the tradeoff between sensitivity and precision. The optimal window size of 4 seconds was determined by finding the value between 2 and 20 seconds that gave the maximum mean precision, sensitivity, specificity, and F1-score across participants. Using this post-processing window size increased precision and F1-score across all models (see figure 3.4.3b), as well as reduced the number of false positives per hour (see figure 3.4.5). The most common cause of these false positives appeared to be physiologic or extraphysiologic electrical artefacts (see figure 3.4.4).

3.4.2 Multi-Class Classification

It was hypothesised that training the models to also distinguish artefactual labels separately from the ictal and interictal classes could improve seizure detection performance. As such, the same data and analysis process for binary classification was used to train new pipelines; differing only in that "Artefact" labels were now separately encoded as 2, keeping "Notched Rhythmic Waveforms", "Spikes", and "Baseline" data labels as 0, and "Generalized Epileptiform Discharge" as 1. SVM was fitted with the (OvR) strategy, with the other classifiers already being able to handle multi-class labels.



(a) P16 (b) P17 (c) P18

Figure 3.4.4: Examples of misclassified segments of patient EEG records

The average F1-score on the validation set during training, for the different classifier types, were similar across patient datasets and the two iterations; KNN (mean = 82.25, SD = 0.53), RF (mean = 82.99, SD = 0.49), SVM (mean = 83.17, SD = 0.70). Only pipelines ending with SVM classification had different pipeline components performing best between the two different runs; varying between feature selection before classification and both feature selection and PCA before classification. All optimal pipelines ending in KNN or RF classifiers had preceding feature selection.

Multi-class test set performance is reported in table 3.4.3, both for metrics based on the seizure class compared to the baseline and artefact labels grouped (one-against-all), and the weighted metrics, which account for label imbalances across all metrics. It is worth noting that categorisation of baseline and artefact labels was poorer than with seizures, as evidenced by the lower scores in the weighted metrics. However, as identifying absence seizures is the main focus of this research, rather than the models ability to distinguish baseline and artefact segments, metrics from grouped predictions using an one-against-all approach are more relevant to our aims and subsequently focused on. Furthermore, these metrics allow for a direct comparison between the binary and multi-class pipelines.

As can be seen in figure 3.4.6, the gains from training models on the multi-class data are mainly for pipelines ending with RF models. The most affected metric is FDR/h, where both RF and SVM see a decease in average false positives (see figure 3.4.5), although there is a slight increase for KNN and SMV. As can be seen in the ROC plots (figure 3.A.6), models

Table 3.4.3: Average (and standard deviation) test scores for multi-class classification when viewed as between the seizure class and the rest (One-Against-All) or weighted metrics.

		Accuracy		**Recall**				**Precision**		**F1-score**		**AUC**		**FDR/h**		**Prediction Time**	
				Sensitivity		***Specificity***											
		Mean	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***	**Mean**	***(SD)***
Seizure (One-Against-All)	**KNN**	98.23	(2.74)	92.3	(6.24)	98.33	(2.79)	**82.48**	(20.96)	**85.44**	(13.69)	97.94	(1.75)	59.2	(99.7)	2.25	(0.77)
	RF	98.23	(2.47)	93.66	(5.0)	98.31	(2.53)	73.19	(21.43)	80.42	(14.93)	98.85	(1.56)	59.91	(90.38)	**0.39**	(0.2)
	SVM	**98.52**	(1.53)	86.69	(16.77)	**98.83**	(1.37)	77.99	(23.82)	79.26	(19.11)	98.06	(1.97)	**41.67**	(48.76)	0.53	(0.34)
	SMV	98.07	(2.68)	**94.13**	(4.23)	98.15	(2.72)	75.99	(23.32)	81.98	(17.19)	**98.92**	(1.36)	65.66	(97.49)	3.23	(1.05)
Weighted Metrics	**KNN**	77.37	(9.17)	77.37	(9.17)	···	···	79.43	(8.06)	76.53	(11.41)	-	-	-	-	···	···
	RF	79.09	(9.12)	79.09	(9.12)	···	···	82.68	(5.56)	78.37	(11.7)	-	-	-	-	···	···
	SMV	79.39	(8.79)	79.39	(8.79)	···	···	82.4	(5.59)	78.7	(11.26)	-	-	-	-	···	···
	SVM	78.16	(8.93)	78.16	(8.93)	···	···	80.32	(7.96)	77.28	(11.28)	-	-	-	-	···	···

Note. The best average score for each metric, across classifiers and method of reporting (One-Against-All and Weighted), are in bold. Sensitivity and specificity are typically reported separately for binary classification, but are covered by a single metric (recall) for multi-class classification. Prediction time is the same for both methods of reporting as they reflect the same models, with post-processing occurring after predictions have been made.

Table 3.4.4: Most common pipeline steps for multi-class classification across all datasets with either the most common categorical or average (and standard deviation) hyperparameter value.

KNN			**RF**			**SVM**		
Steps	*Hyperparameters*	*Average Value (std)*	*Steps*	*Hyperparameters*	*Average Value (std)*	*Steps*	*Hyperparameters*	*Average Value (std)*
1. Random Undersample	-	-	1. Random Undersample	-	-	1. Random Undersample	-	-
2. Feature Selection	Num Features	222.1 (45.32)	2. Feature Selection	Num Features	423.1 (59.72)	2. Feature Selection	Num Features	225.9 (88.57)
3. Scale	-	-	3. Classifier	Criterion	Entropy	3. Scale	-	-
4. Classifier	Algorithm	Brute		Number of Estimators	1517.86 (532.34)	4. Classifier	Kernel	RBF
	Leaf Size	37.1 (32.3)		Max Depth	26.86 (13.87)		C	4.54 (1.69)
	Nearest Neighbours	6.95 (1.02)		Minimum Samples Split	0.01 (0)		Gamma	0.02 (0.02)
	Distance Metric	Manhattan		Max Features	0.17 (0.13)			

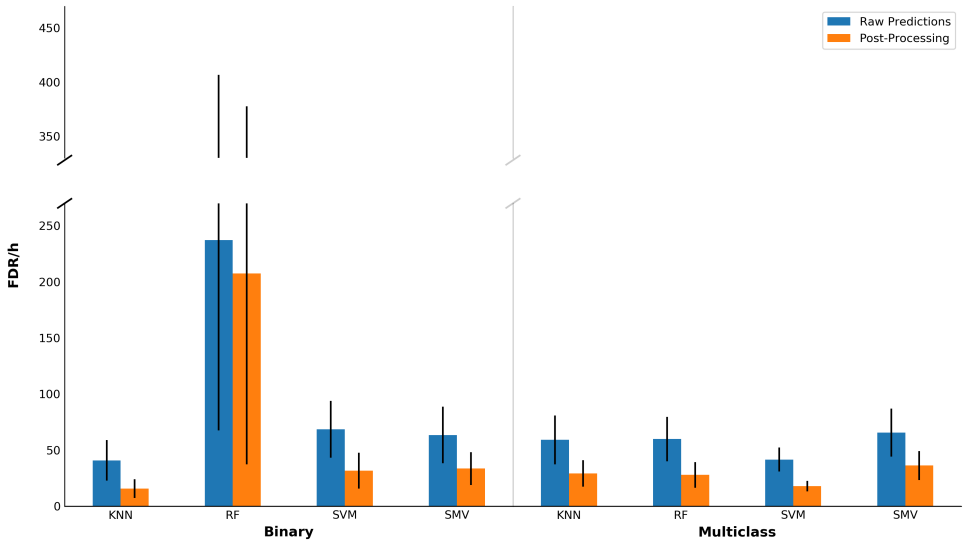


Figure 3.4.5: Average (with standard error of the mean) false positive rate for both the binary and multi-class pipelines before and after prediction post-processing.

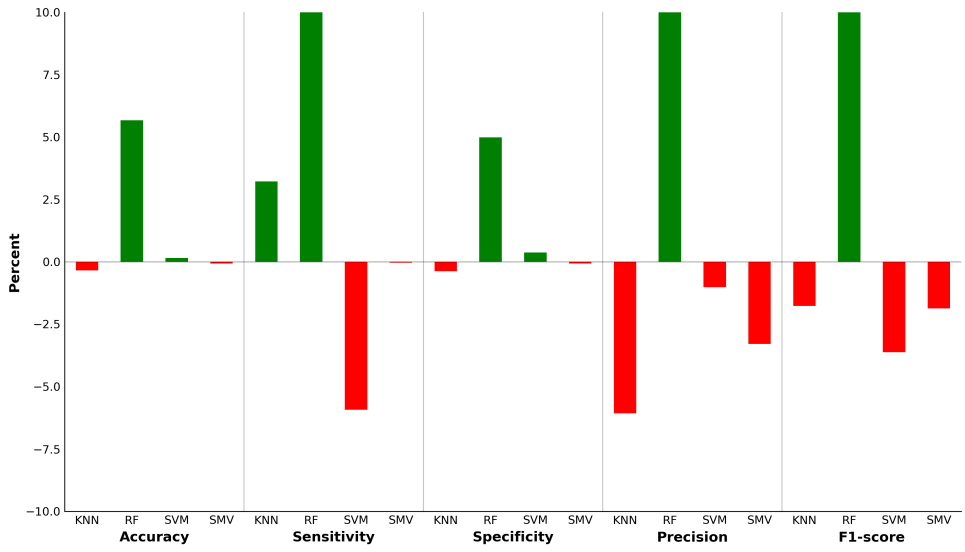
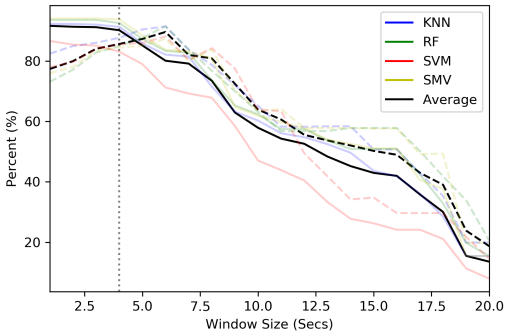
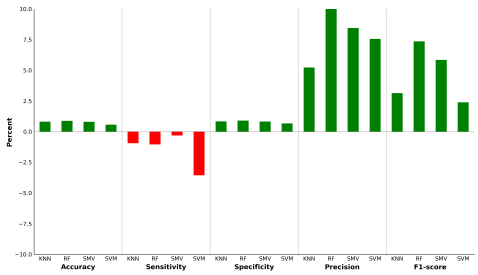


Figure 3.4.6: Average test set score change (%) between binary and multi-class classification predictions.



(a) Post-processing window length sensitivity (solid) and precision (dashed) scores for multi-class classification predictions.



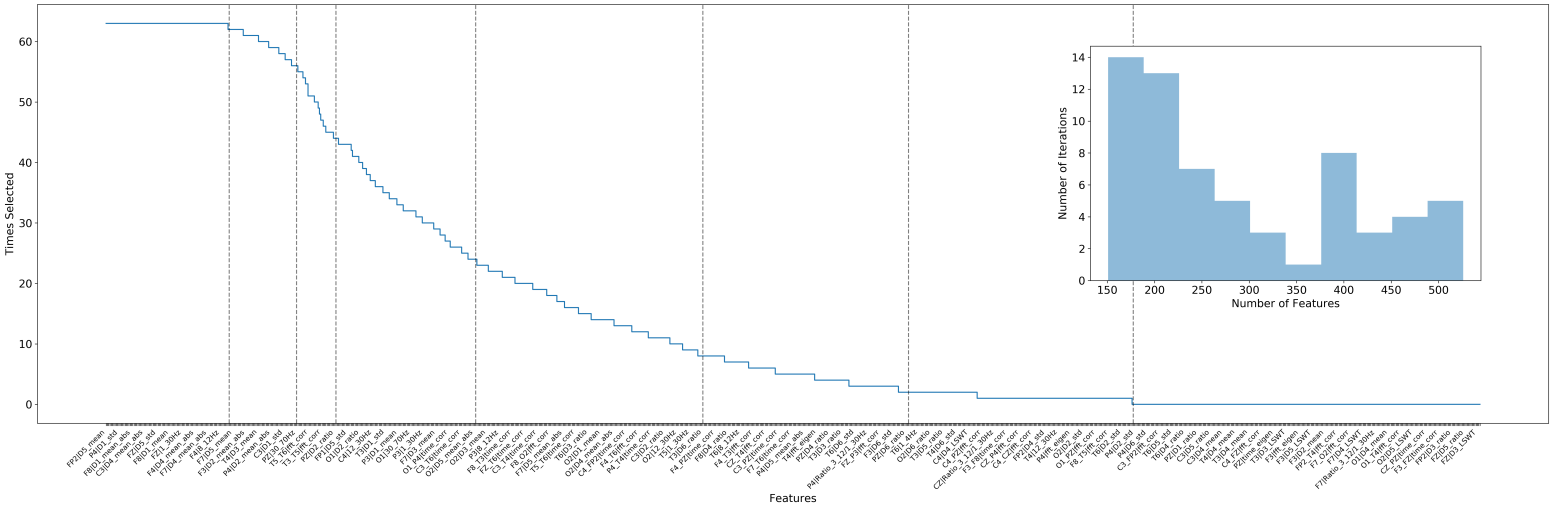
(b) Average test set score change (%) from a four second long post-processing window on multi-class model predictions.

Figure 3.4.7: Affect of prediction label post-processing on multi-class classification performance metrics.

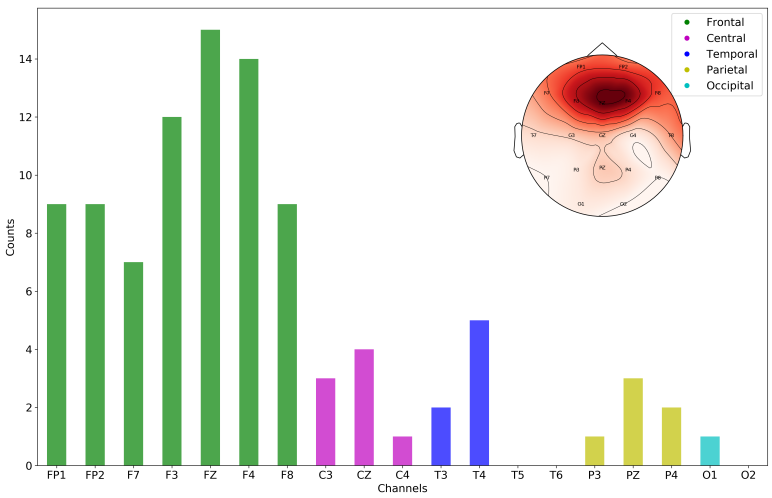
are generally more similar for multi-class predictions than binary; with RF having a greater AUC than KNN and SVM for multi-class labelling and the inverse for binary labelling.

In general, the best pipelines for the multi-class models had more complex classifiers (see table 3.4.4) than the binary pipelines (see table 3.4.2). For example, on average there were more trees, which were each deeper, in RF models, and RBF kernels were more commonly used than linear kernels for SVM models. This no doubt reflects the added complexity of the classification problem when more classes are introduced to a model.

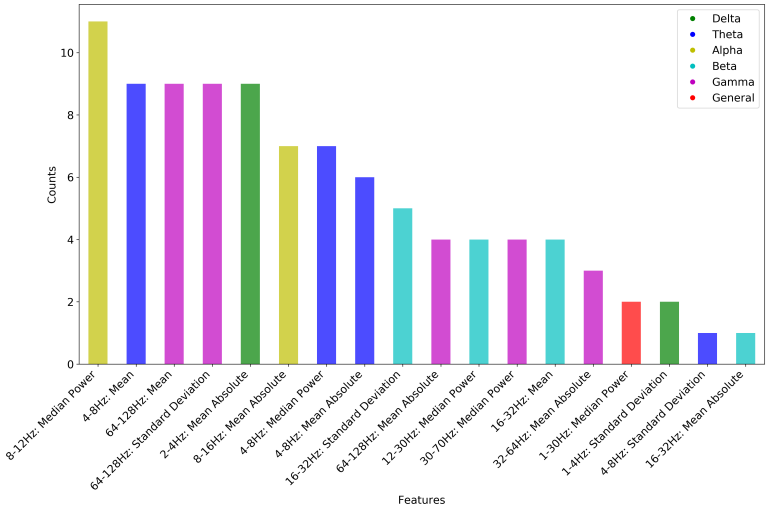
Similar to the binary pipelines, where a feature selection component was present, features in the frontal channels were most commonly selected. However, the features generally covered a broader range of frequencies (see figure 3.4.8). The average number of features selected was also smaller in multi-class pipelines (mean = 290.37, SD = 115.31) than binary (mean = 325.23, SD = 177.77). Also similar to binary classification pipelines, the best window size for post-processing was also 4 seconds (see figure 3.4.7a). The effect of this windowing on predictions is also similar for both multi-class and binary pipelines (see figure 3.4.7b); although there is a smaller overall reduction in sensitivity and larger reduction in the FDR/h (see figure 3.4.5).



(a) Number of times features were selected across the best models with cPOP slope change detections.



(b) Number of selected features in each EEG channel with topoplot.



(c) Number of times a feature was selected across channels.

Figure 3.4.8: Most common EEG channels and features selected for when a feature selection component was in a multi-class classification pipeline.

Note. Channels and features were reduced using the smallest cPOP threshold.

3.5 Discussion

We aimed to assess a number of machine learning pipelines for the automatic detection of absence epilepsy seizures in NHS records using Bayesian optimisation. Similar to other papers focusing specifically on absence epilepsy seizures (see table 3.5.1), we developed and tested models on a novel dataset. This dataset reflects records gained from clinical practice, wherein an investigation of an epilepsy diagnosis is being made, with this research being the first to use NHS diagnostic records for automated seizure detection. The models reported from this research have greater accuracy, specificity, and precision than most previous research, where reported (see table 3.5.1). This is notable due to a number of authors specifically removing artefacts from the data before training and testing models (e.g. Zeng et al., 2016). Through the combination of a large feature set and feature reduction techniques, we were also able to identify components of the EEG record that are useful for identifying absence epilepsy seizures. To our knowledge, this work also provides the first use of Bayesian optimisation methods to find optimal hyperparameters and pipeline configurations for absence epilepsy detection. Finding optimal hyperparameters is important not only in improving the fit of models, but also to ensure differences between tested models are not just a result of default/selected parameters rather than the actual behaviour of the models or features (Zheng and Casari, 2018).

Machine learning algorithms could assist with collecting longer EEG records from patients with epilepsy, due to the current bottleneck of clinical time required for manual marking. Indeed, using a new algorithmic approach would no doubt have benefited the $\sim$ 40% of patients in our dataset who did not have any identifiable generalized epileptiform activity in the 30 minute records provided, but were later diagnosed with absence epilepsy. Indeed, it has been shown that 30% of 451 children with absence seizures who had no clinically detected seizures in a standard recording procedure had them detected in 1 hour EEG recordings (Glauser et al., 2013; Ulate-Campos et al., 2016). Algorithms would enable longer recordings with such patients without as large an increase in the marking and reporting time burden on clinical psychologists this imposes. The algorithms detailed in this work have a high false positive rate, which would be an issue in a remote patient monitoring

Table 3.5.1: Comparison of metrics from our approach after post-processing predictions, using a four second window, to previous research.

Reference	Participants	Data Length	Seizures	Channels	Features	Label Classification	Classifier	Evaluation Method	ACC	SEN	SPEC	PREC	F1	FPR/h
Alkan et al. (2005)	5 In-Clinic Patients	-	20	4	3 Frequency	Binary	Logistic Regression	40% Hold-out	90.5	87.9	92.6	-	-	-
	7 Controls	-	N/A				Multilayer Perceptron		92	90	93.6	-	-	-
Subasi (2007a)	5 In-Clinic Patients	-	20	128	4 Time-Frequency	Binary	Multilayer Perceptron	40% Hold-out	92	92	91.9	-	-	-
	7 Controls	-	N/A				Adaptive Neuro-Fuzzy Inference System		94	94.3	93.7	-	-	-
Liang et al. (2010)	3 Rats	1288 secs	44%	1 (Inter-cranial)	2 Time 1 Frequency	Binary	Linear Least Squares	Leave-One-Out Cross-Validation	95.39	90.33	99.07	-	-	-
							Linear Discriminate Analysis		95.81	91.37	98.93	-	-	-
							Backpropagation Neural Network		97.37	96.8	97.83	-	-	-
							Support Vector Machine		97.5	97.03	97.83	-	-	-
Petersen et al. (2011)	19 In-Clinic Patients	11hrs 48m	111	18 (F7-FP1)	1 Time-Frequency	Binary	Support Vector Machine	Leave-One-Out Cross-Validation	-	99.1	-	94.8	-	0.5
Duun-Henriksen et al. (2012b)	20 In-Clinic Patients 1 Ambulatory Patient	11hrs 23m 4 Days	125 -	19 4 (F7-FP1)	2 Time 1 Time-Frequency	Binary	Support Vector Machine	Triple-Repeated Fivefold Cross-Validation (1 Day Training, 3 Test)	- -	97.2 95	- -	- -	- -	0 0.037
Zeng et al. (2016)	9 In-Clinic Patients	600 secs	33%	19	3 Time 1 Frequency 1 Time-Frequency	Multi-class (Ictal, Interictal, Pre-Ictal)	Decision Tree	10-Fold Cross-Validation	71.8	-	-	-	-	-
							K-nearest Neighbor		72.1	-	-	-	-	-
							Discriminant Analysis		76.7	-	-	-	-	-
							Support Vector Machine		74.3	-	-	-	-	-
Kjaer et al. (2017)	6 Ambulatory Patients	96hrs	-	3 (F7-FP1)	1 Time 4 Frequency 1 Time-Frequency 1 Phase	Binary	Support Vector Machine	5-Fold Cross-Validation	-	98.4	100	87.1	-	0.23
Our Approach	21 In-Clinic Patients	11hrs	53	19	2 Time 4 Frequency 5 Time-Frequency	Binary	K-Nearest Neighbors	Leave-One-Out Cross-Validation	**99.39**	88.14	**99.55**	**93.78**	**90.36**	**15.77**
							Random Forest		93.44	80.39	94.22	65.15	70.64	207.49
							Support Vector Machine		98.93	89.07	99.11	86.57	85.28	31.65
							Soft Majority Vote		98.95	**93.85**	99.05	87.73	89.69	33.63
						Multi-class (Ictal, Interictal, Artefact)	K-Nearest Neighbors	Leave-One-Out Cross-Validation	99.05	91.36	99.17	87.71	88.59	29.21
							Random Forest		99.11	92.62	99.21	85.02	87.78	27.97
							Support Vector Machine		99.09	83.14	99.49	85.55	81.66	17.94
							Soft Majority Vote		98.88	93.82	98.98	84.44	87.83	36.3

Note. The best average score for each metric, across classifiers and label classification, for our approach are in bold.

system. However, if ever practically implemented into a clinical environment, the classification predictions from these algorithms would be reviewed by a qualified physiologist; so there is a preference for false positives over false negatives. Indeed, despite the number of false positives, the best binary algorithms in this work never missed marking a seizure, and in most cases, accurately marked the full duration of seizures where present. Therefore, in practice the use of the binary KNN classifier could result in reducing reviewing the full 11 hours of EEG across all 21 patients down to only 14 minutes of EEG segments identified by the algorithm.

As well as increasing the speed of assessing an EEG record, the best algorithms in this research have the potential to improve diagnostic accuracy of epilepsy seizures above the 70-80% rates currently estimated to occur in developed countries (NICE Clinical Guidelines and Evidence Review for the Epilepsies, 2004). Due to the nature of the diagnostic assessment, although collected in controlled environment (NHS clinics), the records are often contaminated with activity that could be mistaken for epileptic seizures. The sources of this activity can be rhythmic brain activity from drowsiness, caused by asking patients to lie down and close their eyes during the assessment, or respiratory artefacts from heavy breathing. It is key that these EEG segments are not mistaken for epileptic EEG in order to ensure a misdiagnosis is not made. Indeed, it is more difficult to both visually and automatically accurately classify data with significant numbers of artefacts comparative to relatively "clean" records. Furthermore, an algorithm trained on multiple types of seizures, or ensemble of algorithms that each focus on specific seizure types, could also improve diagnostic accuracy when assessing a patient for multiple types of epilepsy. Indeed, the type of epilepsy to be investigated is often not clear before assessment and epileptic patients are at risk of having co-morbidities and experience multiple seizure types.

A limitation to the current automated absence epilepsy seizure classification literature, is the difficulty in determining the context recordings were taken in beyond "routine examination", as well as their content; due to most authors using private datasets and lack of data description. In this work we demonstrate the application of machine learning methods for seizure detection on routine EEG data where a fifth is known to contain artefacts; a substantial proportion of which is due to the procedure including sustained heavy breathing.

This could be one of the causes of increased instances of false positives in this particular dataset comparative to other authors, or that these metrics are reported on a second-by-second basis rather than detections grouped together. The artefacts present in the data may also have influenced the larger optimal window size for post-processing than other authors (e.g. 2; Duun-Henriksen et al., 2012b; Petersen et al., 2011). Post-processing was reported separately to the output of the models, as if such models were implemented into practice, it would likely be useful to manually change the post-processing window size on a record-by-record basis; dependent on the type of epilepsy being investigated, the number of short seizure predictions made, and the data quality. Regardless of post-processing, the models predict seizures on a second-by-second basis, as it is still important to identify short seizures (Browne et al., 1974; Duun-Henriksen et al., 2012b), even if these brief seizures are unlikely to affect complex tasks (Goode et al., 1970; Opp et al., 1992). The shortest seizures present in this dataset were 3/4 seconds (see table 3.A.2), which is likely the main reason why 4 seconds was found to be an optimal window size, as seizures begin to be missed as window sizes increase beyond this.

A limitation to the labelling of the EEG dataset used in this research, was that no personally identifiable data, such as videos, were provided to the researchers. Although written notes about patients seizures and movements, coded alongside video recordings during and after the assessment by clinical physiologists, were used to aid visual coding by the researchers, access to the video would have enabled clarification of some EEG features which were not immediately clear as to their origin. Cautious interpretation of the EEG was therefore given to the mark-up, with clearly diagnostic or benign features given the categories of "Generalized Epileptiform Discharge" or "Notched Rhythmic Waveforms", and the other more subtle phenomena being categorised as "Spikes" or "Artefact"; these all reviewed by a Consultant Neurophysiologist. Conversely this does mean the labelling is focused specifically on the electrical phenomena present within the EEG, which the machine learning models were trained on. Furthermore, as the data was limited to recording patients in the clinic of a hospital, these records were shorter and with a narrower range of movement artefacts present than would be expected in typical ambulatory paediatric measurements. However, a benefit to this dataset was that it was marked freely, in that segments were

not marked on a second-by-second basis; instead start and end times were recorded using indexing to the nearest millisecond. This means the labels reflect the data more closely than if it was segmented into 1 or 30 second bins, and also does not artificially improve performance of methods that window data into segments that correspond to how it was marked.

Although childhood absence epilepsy is generalized in its presentation, focal paroyms have also been observed to occur in 38% of cases, albeit predominantly in the frontal lobes (Mariani et al., 2011). Therefore approaches which aim to limit the electrodes assessed (e.g. Duun-Henriksen et al., 2012b), mean some focal seizures may not be detected. Nevertheless, dimensionality reduction methods can be used to best identify features and channels for seizure detection for faster run time, lower power consumption, and increased accuracy. Indeed, dimensionality reduction has been used previously by Birjandtalab et al. (2017) to find the best few EEG channels for seizure detection using the same spectral feature set in each channel, with a potential future use in flexible and personalised electrode positioning. The features selected by random forests before classification models in this work do arguably best reflect the presentation of absence seizures, as they were predominately in frontal channels and focused on slower oscillation frequencies. However, it is of note that a broad range of frequencies, predominately around alpha bands and below, as well as the 2-4Hz range, were selected for classification; suggesting other frequency ranges beyond the typical 3/4Hz are still useful in determining the occurrence of absence seizures. Indeed respiratory and eye movement artefacts typically overlap with the slower frequency ranges (>15Hz), meaning a broader range of slow frequencies may be required for 3Hz seizure identification to account for this (Duun-Henriksen et al., 2012b). This also lends credence to the idea that feature selection based on tree classifiers may enable a seizure specific EEG channel profile based on the focal area of seizures, and with enough data from a patient, enable patient specific limited channel EEG for long term monitoring based on their unique seizure topography. However, these inferences should be taken with a degree of caution, as the features in channels are highly correlated (see figures 3.A.7 & 3.A.7), so feature ranking is unlikely to be able to capture the information of all features fully (Raschka and Mirjalili, 2019). Indeed, future investigation into the high correlations of features may be served by grouping features based

on correlation structure and investigating group selection. Furthermore, future work needs to confirm chosen channels are indeed the best for classifying a broad range of seizure types and that these channels are not just selected due to just having less noise/artefacts present. Additionally, similar to Fearnhead et al. (2019), the penalty value selected for reducing the feature set currently used no theory to support the choice, with future work recommended to investigate a range of penalties using methods such as CROPS (Haynes et al., 2017).

A few other papers have also focused on seizure-event detection algorithms to detect childhood absence seizures (see table 3.5.1). Although difficult to compare to other research, this work is broadly similar to Petersen et al. (2011) in number of patients, data length, features (log-sum of wavelet transform), and evaluation method. Although their reported metrics of sensitivity, precision, and false positives per hour are generally better than the metrics from this work (see table 3.5.1), their models were trained on individual channels with fewer features; meaning models were less likely to overfit, at the expense of reporting different detections in different channels to varying accuracy. Duun-Henriksen et al. (2012b) has the second highest sensitivity with the lowest false detection rate, comparative to the previous absence epilepsy literature. However, they also limit the electrodes used, meaning some focal seizures may not be detected. Furthermore, both papers did not test algorithms on data without seizures present (e.g. Alkan et al., 2005; Subasi, 2007b) to get a better representation of the FPR/h. As previously mentioned, $\sim 40\%$ of patient records in this work did not have any identifiable generalized epileptiform activity. Interestingly, model performance tended to be worse on these patients comparative to those that had evident seizures; suggesting future work should test algorithms on records collected with patients without a history of seizures and potentially on records from epileptic patients without seizures in.

Some limited comparisons regarding model parameter values between this and other papers can also be drawn; although it is worth noting optimal pipeline components and hyperparameter values varied between held-out datasets, as well as between and binary and multi-class labelled data. For our best performing classifier, KNN, the hyperparameter k is typically set higher than 2 (e.g. Zeng et al., 2016; Polat and Ozerdem, 2016), as was found for our binary models. Distance measures are not always reported, however we found

Manhattan distance was the most commonly used in our optimal models, different from the Euclidean distance which is often used. Regarding SVM, Liang et al. (2010) similarly found increased performance with a proceeding PCA step. The value of C, between 3 and 5 for the binary and multi-class models, is broadly similar to other authors (e.g. Liang et al., 2010; Petersen et al., 2011), but the gamma parameter was much lower; for example Liang et al. (2010) used a value of 0.5 whereas values were typically lower than 0.1 in this work. However it is important to note, model architecture and hyperparameters tend to be manually set by authors rather than found through a specific search method; although some authors mention testing different parameters without providing details as to the methods (e.g. Petersen et al., 2011). Specific to the absence epilepsy literature, only Duun-Henriksen et al. (2012b) and Kjaer et al. (2017) detail their use of a grid search method. These papers used a hyperparameter search space that was similar to this work, although they do not provide the final optimised hyperparameter values found for comparison. As previously mentioned, finding optimal hyperparameters is important for improving model fit and ensuring differences between models are not just a result of default/selected parameters (Zheng and Casari, 2018). It is however also worth noting that the sensitivity to hyperparameter settings do change depending on the specific classifier.

As training took place on multiple single core CPUs, and the pipelines searched over could be complex, training time was long (particularly for random forests). However, prediction of class membership was quite quick for most models after features had been extracted from the data (see tables 3.4.1 & 3.4.3), with binary models generally being quicker. This means the majority of the prediction time in practice would be spent on extracting features for the models. SVM were the fastest for making predictions, likely due to predominately using linear kernels for binary classification and selecting fewer features than other models. Indeed, for SVM models with P6, P10, and P20 records held-out, only 2 features were selected for classification (see figure 3.A.9). Such simple models, although not providing the best performance, are certainly interpretable in their classifications, so would have advantages in medical practice where decisions need to be justifiable. As SVMs seem to favour more simple models, it is unsurprising they are common in both the offline and online detection literature. Still, it is common to use a RBF (e.g. Kjaer et al., 2017; Perera et al.,

2017) rather than a linear kernel, to compensate for the complexity of the data. This suggests that with sufficient dimensionality reduction, linear models may still be able to be used in practice. However, there was a lot of variation on the best dimensionality reduction approach and hyperparameters for each participant hold-out dataset, and with different random states, for the SVM pipelines comparative to other models.

RF models performed particularly poorly on data with one class (without seizures), which lead to improvements in its application to data with artefacts also labelled. RF models are known to be more stable with larger datasets and linear approaches better in their application to smaller datasets. As training balanced ictal and interictal data, the data size was substantially reduced, which may have lead to problems of overfitting. Furthermore, RF models did not have the data scaled before classification, a common practice, but one which could have lead the models to be influenced by changing data scales between patients that occur as part of EEG measurement. In general for other models, there was limited benefit to training data with artefacts additionally labelled over the more common binary ictal/interictal labels. The main benefit appears to be that model pipelines and optimal hyperparameters were more similar between models trained on different hold-out datasets when trained on the three class data comparative to the binary data labels. Explicit methods to handle artefacts before EEG machine learning classification, beyond basic filtering, downsampling, and re-referencing, is not always used; suggesting this is not always required to gain meaningful results (Roy et al., 2019b). Nevertheless, this research did not create features explicitly demonstrated to separate artefactual data from "normal" EEG, instead focusing on those known to classify ictal/interictal labels, and did not employ common automatic/semi-automatic artefact removal methods, such as ICA (Delorme et al., 2012; Albera et al., 2012; Urigüen and Garcia-Zapirain, 2015). Instead models could employ a PCA step to reduce data's dimensionality, although this was only found to be optimal for SVM models. The use of PCA in its application to EEG data has been questioned as the assumption of orthogonality between neural activity and typical physiological artefacts, required for effective PCA application, is not often supported (James and Hesse, 2005; Jung et al., 2000; Vigário, 1997; Choi et al., 2005). Other methods for handling artefacts could be investigated in the future to see if it improves performance, such as setting thresholds on

the number of zero crossings and absolute amplitude before classification (Duun-Henriksen et al., 2012b).

3.6 Conclusion

We successfully assessed a number of machine learning pipelines for the automatic detection of absence epilepsy seizures in NHS records. The models reported from this research have greater accuracy, specificity, and precision than most previous research. Through the combination of a large feature set and feature reduction techniques, we were able to identify components of the EEG record that are useful for identifying absence epilepsy seizures. To our knowledge, this work also provides the first use of optimal hyperparameters and pipeline components found through Bayesian optimisation methods for absence epilepsy detection. In the future, such pipelines could reduce the current bottleneck of clinical time required for manual marking of EEG records. Although there is currently a high false-positive rate, the best binary algorithms in this work never missed marking a seizure and, in most cases, accurately marked the full duration of seizures where present. This is promising for later integration into a full system which provides a preliminary marked record to be reviewed by a qualified physiologist.

3.A Appendix B

Table 3.A.1: Length of time, rounded to nearest second, of classification labels in each NHS patient.

	AMPSAT	Artefact	Baseline	Generalised Epileptiform Discharge	Notched Rhythmic Waveforms	Spikes	Total
P1	1025	327	925	18	0	0	2294
P2	447	480	696	7	0	2	1632
P3	0	508	769	32	0	1	1311
P4	1126	392	955	69	0	0	2542
P5	0	323	913	0	0	0	1237
P6	1268	349	954	22	0	1	2595
P7	1389	264	836	167	0	3	2660
P8	1118	546	669	0	7	0	2340
P9	1384	769	628	0	8	0	2789
P10	0	361	842	0	0	0	1203
P11	0	349	833	0	105	0	1287
P12	1008	501	558	35	12	0	2115
P13	262	423	854	0	67	4	1609
P14	69	73	2920	67	3	2	3134
P15	0	566	625	18	0	0	1209
P16	148	622	631	0	0	0	1401
P17	190	552	525	65	0	0	1333
P18	786	382	1039	0	0	0	2206
P19	639	206	1141	16	0	0	2003
P20	14	685	583	53	0	0	1335
P21	18	444	1015	7	10	1	1495
All	**10891**	**9123**	**18911**	**576**	**213**	**14**	**39729**

Table 3.A.2: Length of each seizure, rounded to nearest second, for each NHS patient.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**Sum**
P1	18																18
P2	7																7
P3	7	10	6	9													32
P4	13	15	20	11	10												69
P5																	0
P6	8	4	3	3	3												22
P7	18	21	27	20	26	15	21	20									167
P8																	0
P9																	0
P10																	0
P11																	0
P12	17	17															35
P13																	0
P14	5	4	6	4	5	6	4	4	4	4	4	3	4	3	5	3	67
P15	18																18
P16																	0
P17	15	11	11	13	16												65
P18																	0
P19	16																16
P20	18	18	18														53
P21	7																7

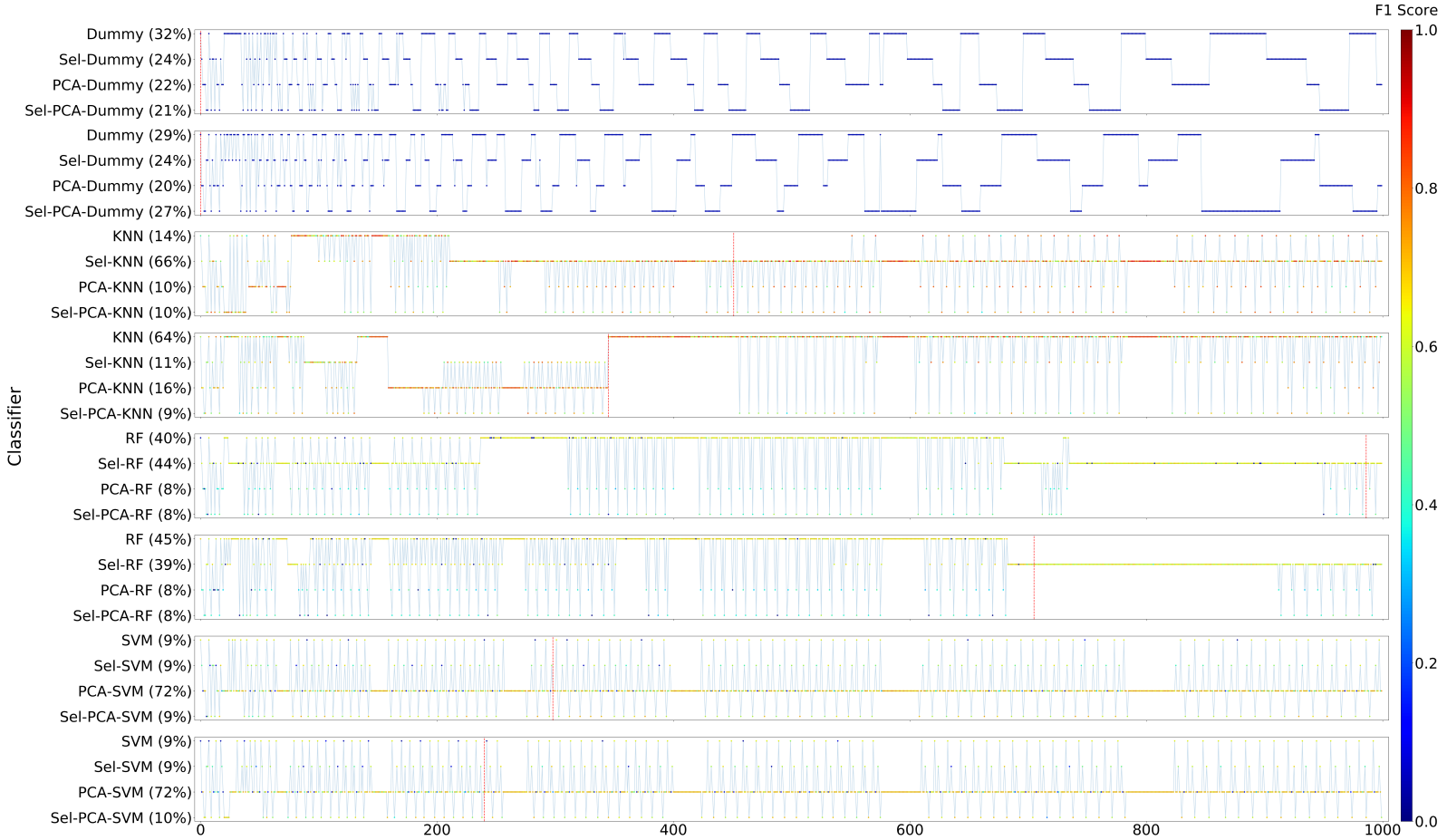
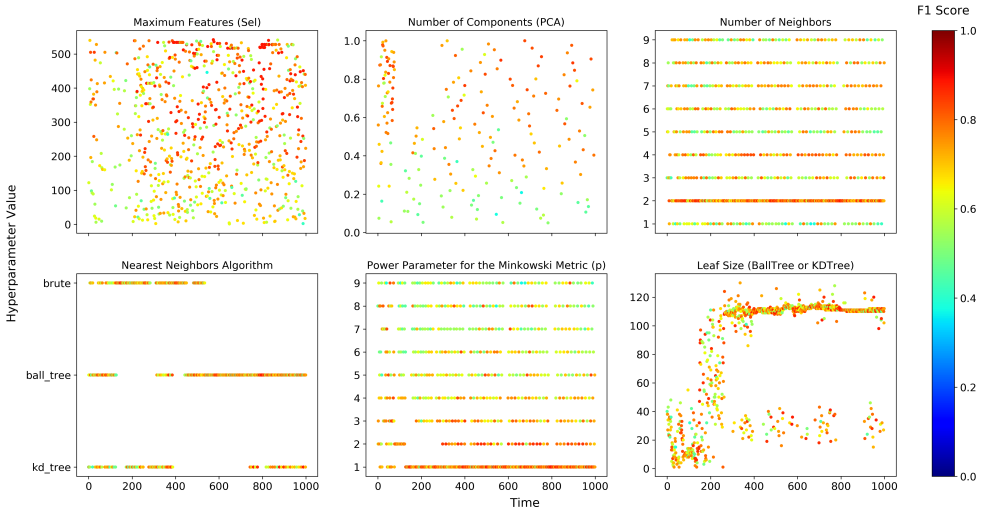
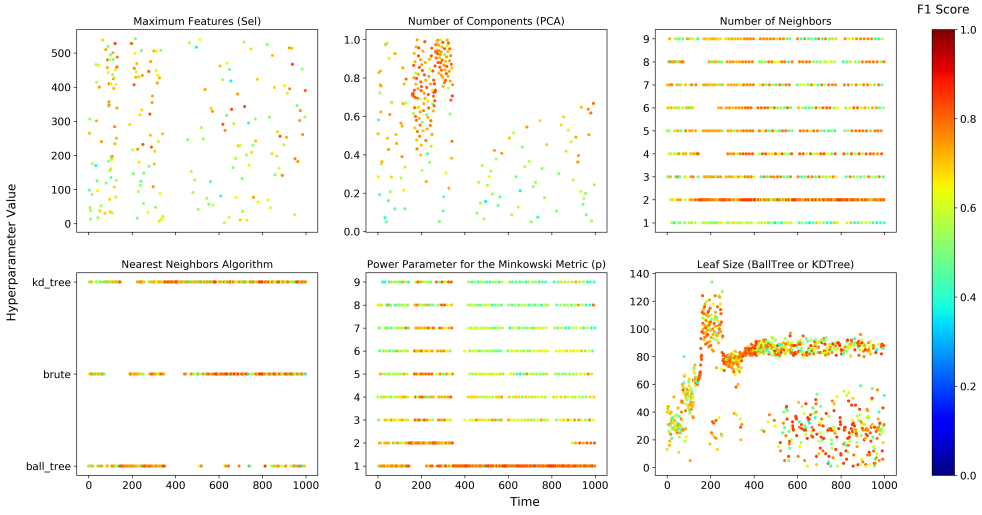


Figure 3.A.1: Progression of Bayesian optimisation over binary classifiers when P2 was held-out.

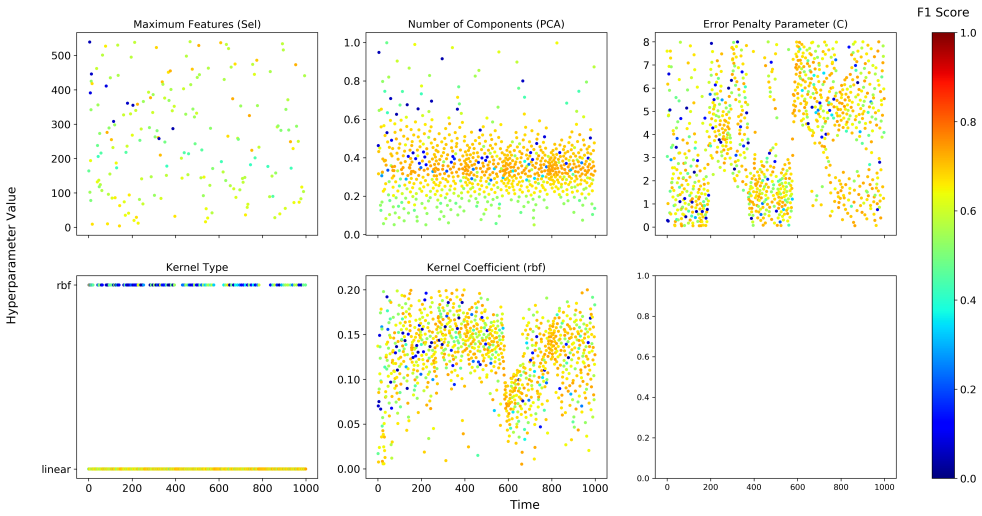


(a) Random state 1.

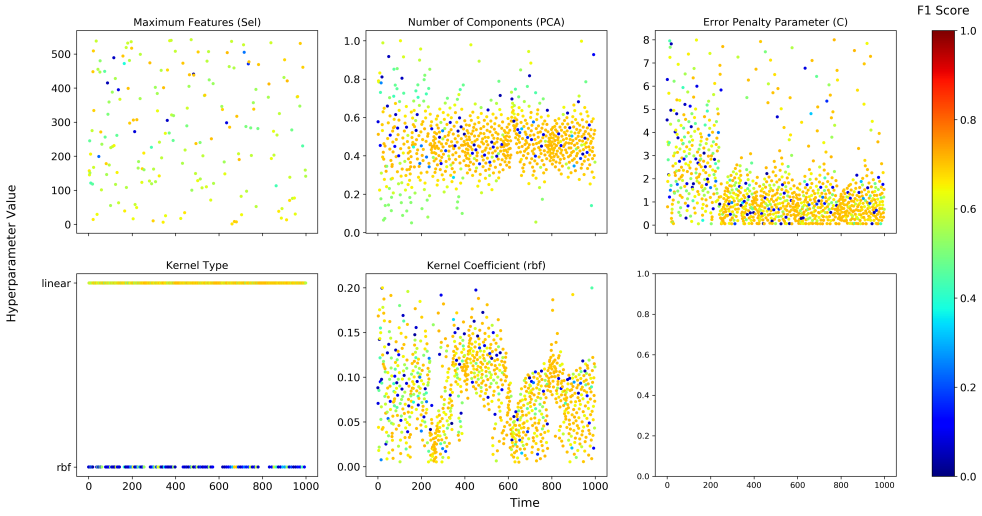


(b) Random state 2.

Figure 3.A.2: Binary classification hyperparameter values, and F1-score on the validation set, during training pipelines ending in KNN models when P2 was held-out.

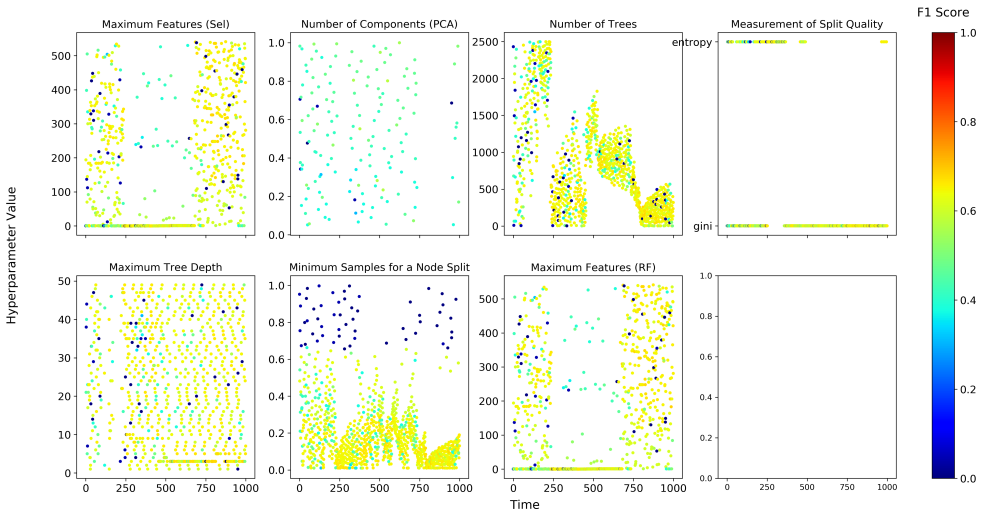


(a) Random state 1.

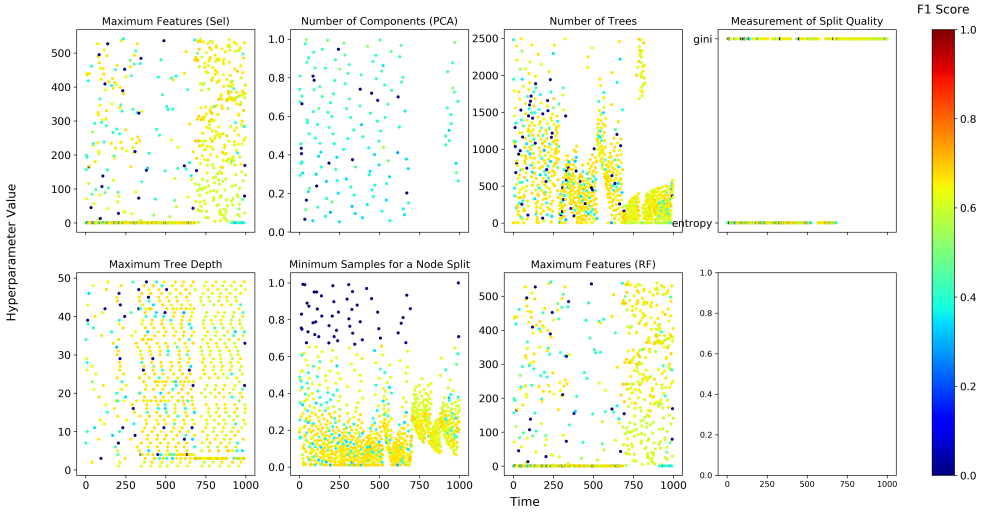


(b) Random state 2.

Figure 3.A.3: Binary classification hyperparameter values, and F1-score on the validation set, during training pipelines ending in SVM models when P2 was held-out.

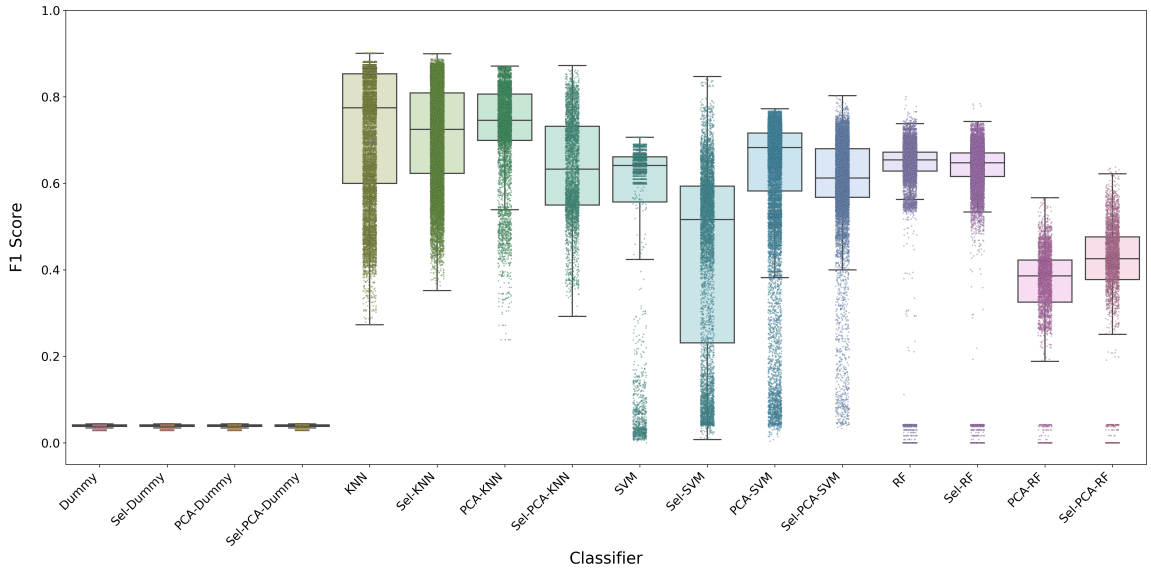


(a) Random state 1.

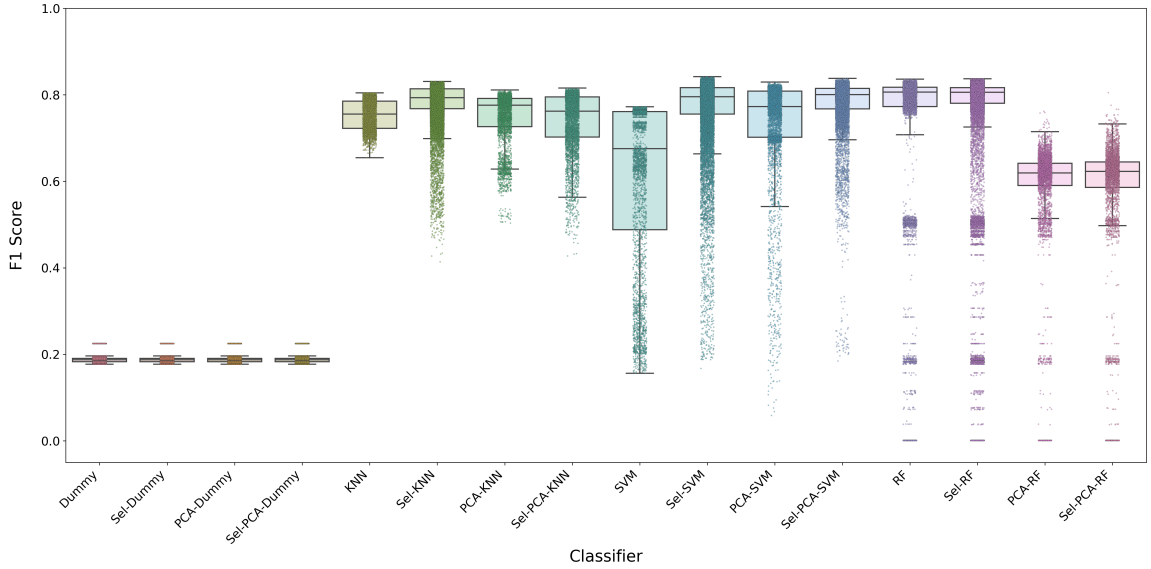


(b) Random state 2.

Figure 3.A.4: Binary classification hyperparameter values, and F1-score on the validation set, during training pipelines ending in RF models when P2 was held-out.

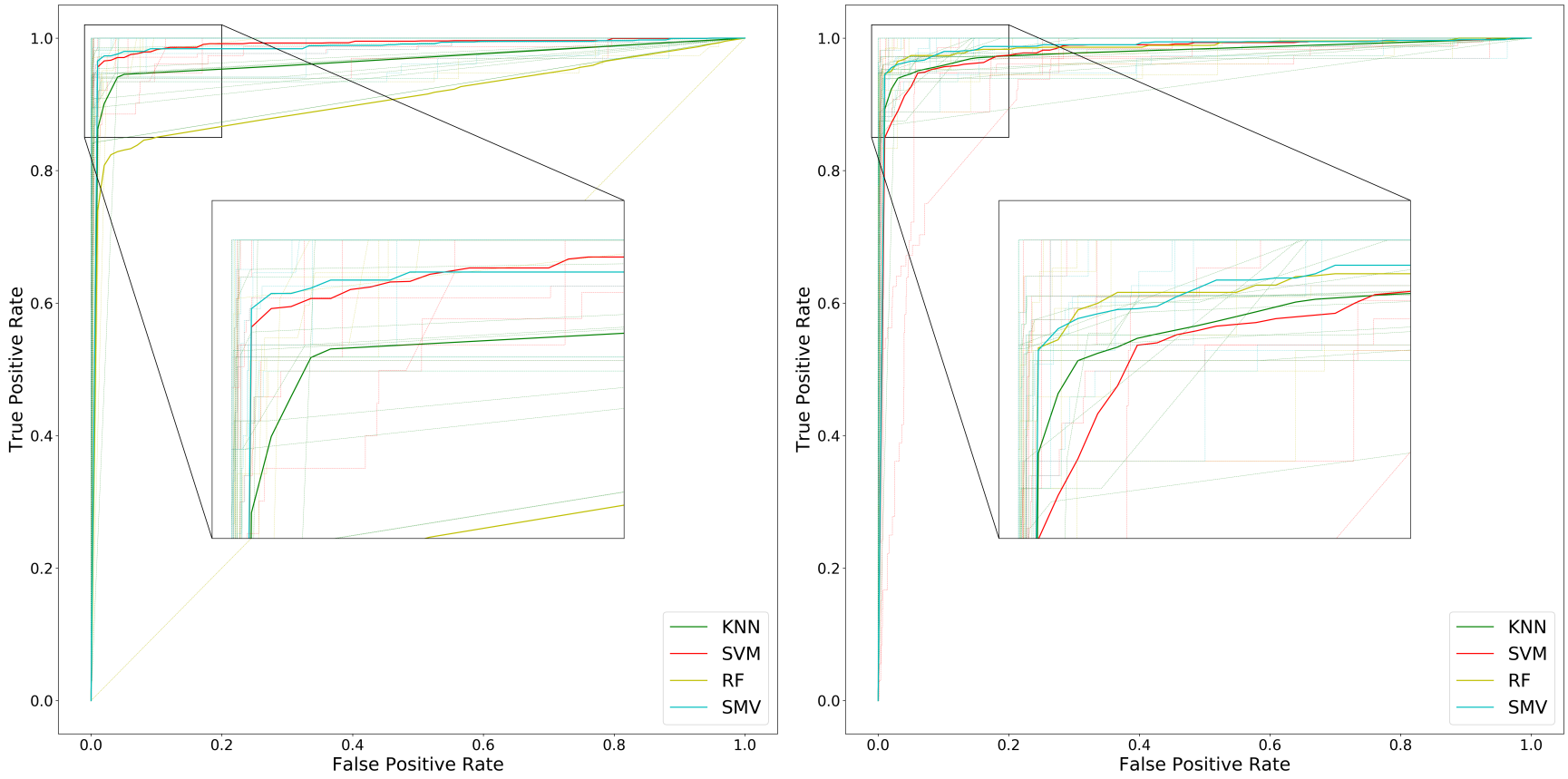


(a) Binary Labels



(b) Multi-class Labels

Figure 3.A.5: F1-scores on the validation set during training across classifiers and datasets with patients held-out.



(a) Binary Labels

(b) Multi-class Labels

Figure 3.A.6: Pipeline Receiver Operating Characteristic Curves (ROCS) for test set performance.

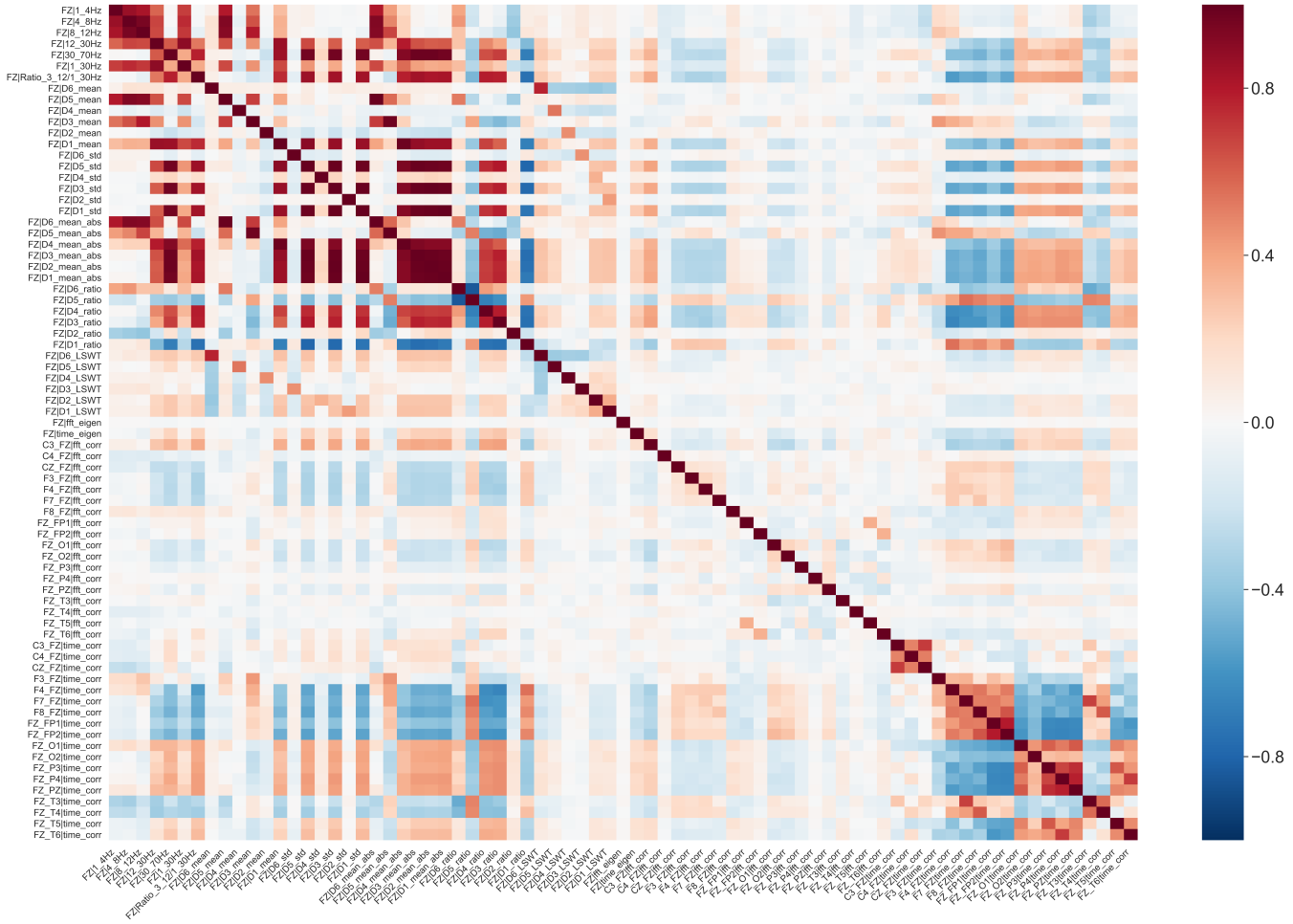


Figure 3.A.7: Correlations between features in the same channel across the full P19 record.

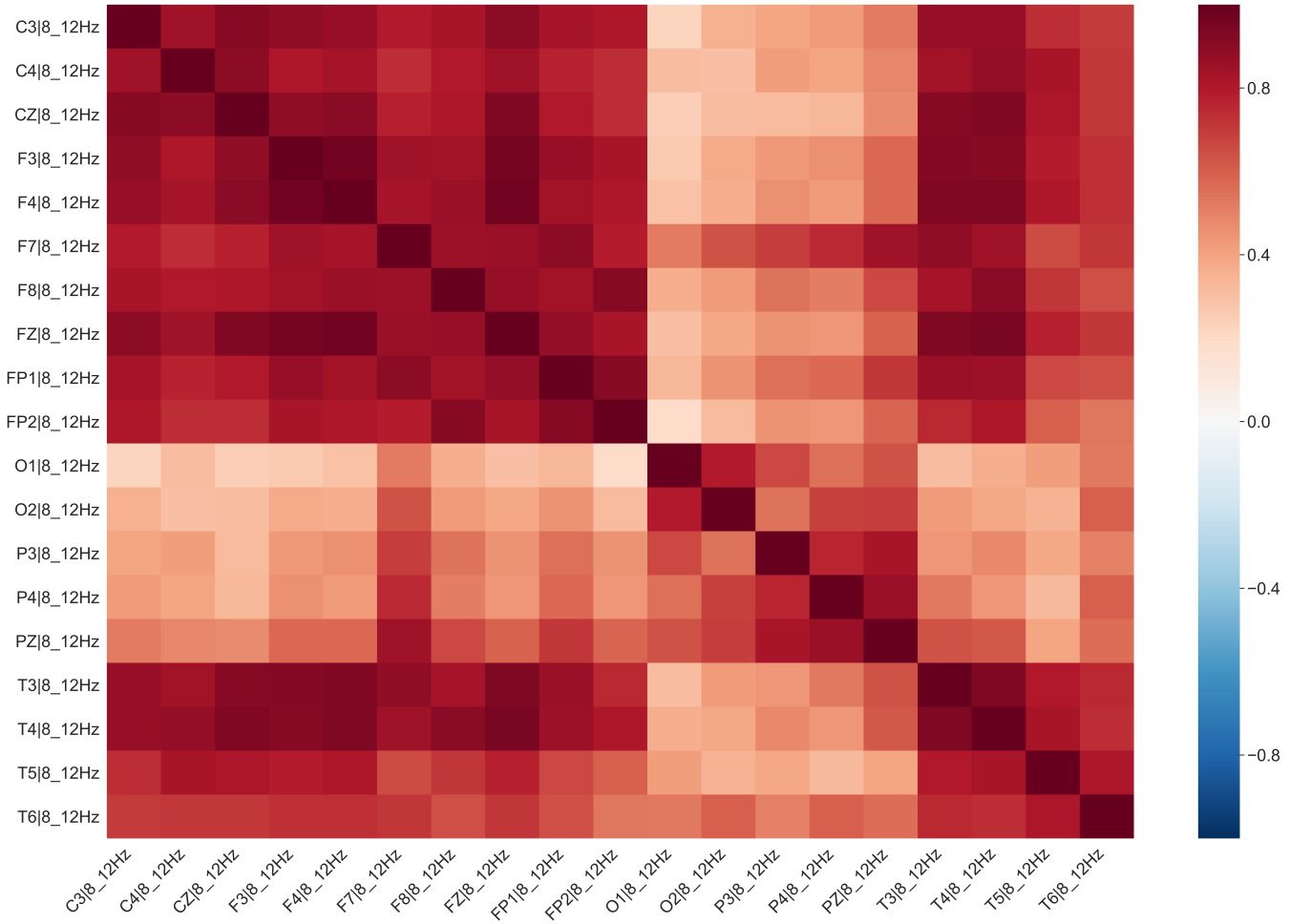
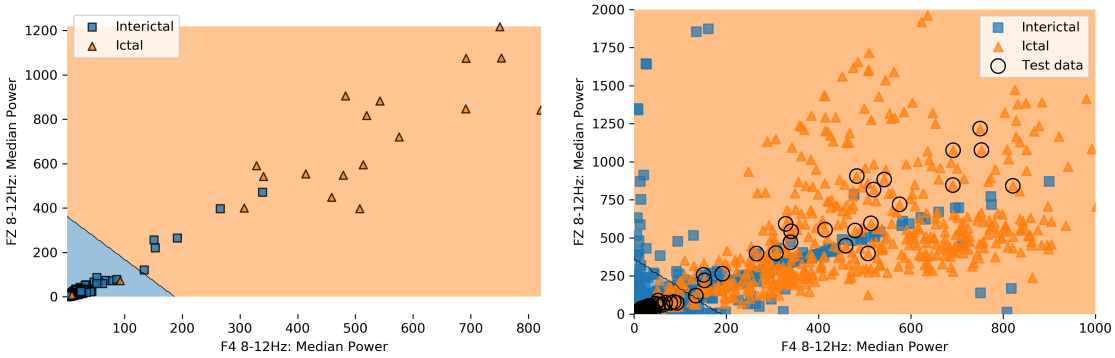
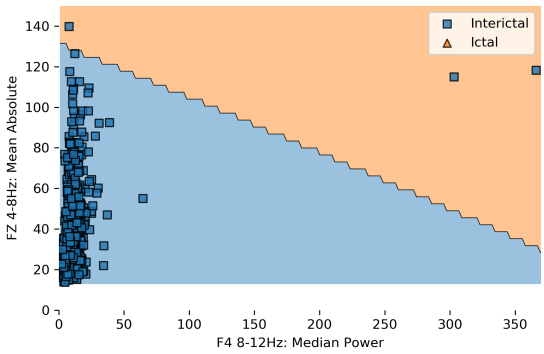


Figure 3.A.8: Correlations of the same feature between channels across the full P19 record.

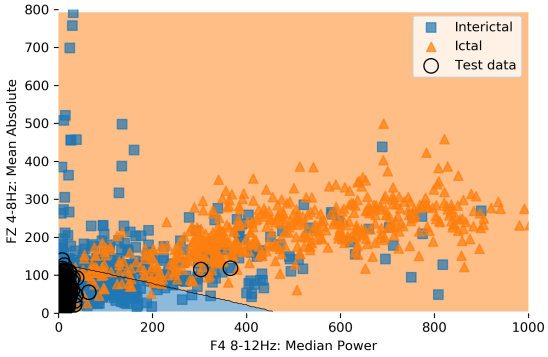


(a) P6 test data

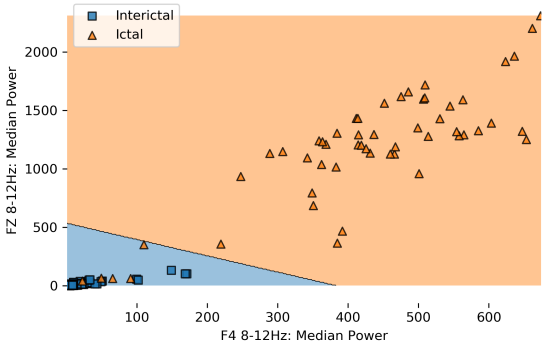
(b) Training/validation and P6 test data (circled)



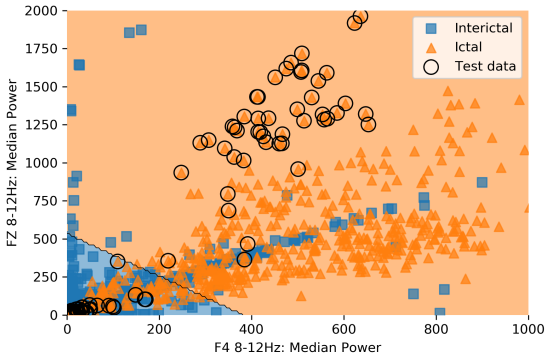
(c) P10 test data



(d) Training/validation and P10 test data (circled)



(e) P20 test data



(f) Training/validation and P20 test data (circled)

Figure 3.A.9: SVM decision boundaries on held-out test records where optimal models only used 2 features.

Chapter 4

Ensemble Classification of Absence Epilepsy Seizures in Electroencephalography Records

4.1 Introduction

In chapter 3 we assessed a number of "classical" machine learning models for their ability to detect generalized absence seizures. We found models with good overall performance at the expense of a high false positive rate. Furthermore, class imbalances inherent in the data were a challenge for effective and consistent model training. A number of methods are available to address training models on imbalanced datasets; such as increasing the training weights of the minority class (e.g. Yuan et al., 2017a), undersampling the majority class (e.g. Roy et al., 2019a), or oversampling the minority class with interpolation (e.g. de la Cal et al., 2018). Changing the training class distribution has previously been demonstrated to be important for achieving good model performance (e.g. Zou et al., 2018); with different sub-sampling ratios found to improve seizure detection performance to different degrees (Alkanhal et al., 2018). Similarly, classification models which use a hybrid method of sampling and boosting (e.g. RUSBoost, Adaboost, XGBoost), have also previously been applied to seizure detection (Seiffert et al., 2008; Roy et al., 2019a). These models have a low com-

putational cost comparative to deep learning models and high performance comparative to "classical" models (e.g. support vector machines; Solaija et al., 2018; Amin and Kamboh, 2016); although are not as commonly used in the seizure detection literature as they are in other applied domains.

The *replicability* (see section 2.7 for definition) of results from machine learning pipelines in chapter 3 varied according to a number of factors; such as which patient was left-out of model training, the classification model used at the end of the pipeline, and the starting values of an optimisation search space. Another likely contributing factor to the differences in pipeline performance, across repetitions of the same optimisation method with different random states, was that pipelines were only trained on one undersample of the data. Therefore, different samples of interictal data used when repeating training could have lead to variations in "optimal" pipeline design. However, balanced ensemble methods allow for individual models in an ensemble to each be trained on different resamples of the data, so collectively they are trained using more than just one undersample of the data. For the case of undersampling epileptic EEG data, this means each model in an ensemble would train on the same ictal samples but different inter-ictal samples.

Seizure detection classifiers also have a *generalisability* problem due to the lack of multi-institution datasets used or compared in published research; a broader issue affecting most research into machine learning for healthcare applications (see McDermott et al., 2019). This is in part due to health data being privacy sensitive, therefore it is difficult to release data openly without de-identification techniques that could affect the utility of the data (Dwork and Ullman, 2018). What limited datasets are available, are frequently used (see section 2.7), which leads to a risk of dataset-specific over-fitting in the literature (McDermott et al., 2019). It is well established that developing models which generalise over care practices or data formats is challenging (Gong et al., 2017; Nestor et al., 2018), not least due to differences in data collection and deployment environments as changes to care patterns evolve (McDermott et al., 2019; Caruana et al., 2015). In this chapter we train and test models on multiple datasets from different health organisations. We analyse NHS EEG records from two hospitals in the UK, Royal Preston and Leeds Teaching Hospitals, and one hospital in the US, the Temple University Hospital. Furthermore, as it has been shown

that increasing the size of a dataset typically increases the replicability of machine learning pipelines (Bouckaert, 2005), we examine the differences in model performance in each dataset separately, as well as when datasets are combined.

Additionally, we aim to further explore the features that are important for absence seizure detection in EEG records. In chapter 3 we found slow frequency frontal channel features were predominately selected when using random forest models in a classification pipeline; which are similar to the properties looked at by physiologists. Therefore, we examine whether this finding is replicable using different ensemble models, examining the feature importances directly instead of feature selection counts. This is important as future clinical adoption of an algorithm depends on its usefulness and trustworthiness; with the implication that a model's processing pipeline needs to be explainable and justifiable (Vollmer et al., 2020). Beyond trust, recent legislative changes (e.g. the EU General Data Protection Regulation) has created a legal requirement to make clear the existence of automated decision-making, provide insight into the decision making process, and explain the significance and the envisaged consequences of such processing for the data subject (European Parliament, 2016).

This chapter is structured as follows: Section 4.2 describes how the datasets were prepared for feature extraction. Section 4.3 introduces the extracted features, machine learning classifiers, and hyperparameter optimisation method used. Section 4.4 describes the validation and test set results, then examines the best performing classifiers and the important features used for decision making, and finally how performance can be improved using prediction post-processing. Finally, sections 4.5 and 4.6 discuss our findings and present our conclusions.

4.2 Data Preparation

In addition to the previously described NHS dataset introduced in chapter 3, hereby refereed to as NHS (Preston), two additional datasets were used in this study; NHS (Leeds), and the Temple University Hospital (TUH) EEG Seizure Corpus (v1.5.0; Shah et al., 2018). This allows us to compare ensemble models to classical models in chapter 3 using the NHS

(Preston) data, as well as also compare performance across datasets from different health organisations.

NHS (Leeds) is a collection of 20 EEG records from 16 paediatric patients (aged 3-10 yrs, mean age = 6.2), diagnosed with absence epilepsy, from Leeds Teaching Hospitals NHS Trust. The EEG signals were mostly sampled at 512Hz, with one record from P2 recorded at 200Hz. As in NHS (Preston) records, EEG was collected using the international 10-20 EEG electrode system with a monopolar recording montage using a central midline reference electrode. Most records are collected using the same diagnostic routine as previously described in section 3.2, although one record from P2 was collected during longer term in-clinic monitoring. To prepare the datasets for training, they were first re-referenced to the average, and key terms in the physiologist notes used to aid visual labelling of the EEG record in all but 2 records where labels were unavailable (P2 & P5). These records were still included in the research, with labelling in these cases solely focusing on the EEG data. The duration and labels associated with the visually assessed data segments were created by the researchers and reviewed by a Consultant Neurophysiologist. Similar to NHS (Preston), data segments were assigned one of four labels (see tables 4.A.1 & 4.A.2); Generalized Epileptiform Discharge (1498.11 secs; 2.85%), Spikes (386.88 secs; 0.74%), AMPSAT (780.85 secs; 1.48%), or Baseline (49929.16 secs; 94.93%). As spikes are not always a clear diagnostic marker, and this research focuses on binary classification, Spikes and Baseline data labels were encoded as 0 (interictal) and Generalized Epileptiform Discharge as 1 (ictal; see table 4.2.1). AMPSAT segments were treated similar to chapter 3 and removed before training the classification pipeline.

We also created a dataset from a subset of patients in the TUH EEG Seizure Corpus (TUHS); a subset of the TUH EEG corpus (Harati et al., 2014; Obeid and Picone, 2016), the world's largest publicly available corpus of clinical EEG data. To prepare the data for classification, the TUHS dataset was firstly downloaded from the project page on the Institute for Signal and Information Processing website (Picone and Obeid, 2016) on 24/05/2019. Eleven patients (mean age = 9.8, 6 female) with 97 absence seizures were identified and used for the dataset hereafter referred to as the TUH (Absence) dataset. Although all patients are recorded using the same 10/20 channel configuration, the electrodes in the TUHS are

Table 4.2.1: Information on patient records used in each dataset for model training.

NHS (Preston)					NHS (Leeds)					TUH (Absence)				
Patient ID	**Age (Gender)**	**Seizure Events**	**Total Time (Seconds)**		**Patient ID**	**Age (Gender)**	**Seizure Events**	**Total Time (Seconds)**		**Patient ID**	**Age (Gender)**	**Seizure Events**	**Total Time (Seconds)**	
			Ictal	***Inter-Ictal***				***Ictal***	***Inter-Ictal***				***Ictal***	***Inter-Ictal***
P1	13 (NR)	1	17.80	2276.60	P1	7 (NR)	2	15.69	990.31	P1 (00000675)	4, 6 (F)	27	202.70	2279.29
P2	8 (NR)	1	6.65	1625.60	P2	5, 8 (NR)	46	577.28	26070.72	P2 (00001113)	20 (F)	14	83.37	2726.62
P3	7 (NR)	4	32.33	1279.15	P3	5 (NR)	7	46.49	1317.51	P3 (00001413)	10, 12, 14 (F)	11	80.16	3760.82
P4	11 (NR)	5	68.77	2473.08	P4	5 (NR)	0	-	1711.00	P4 (00001795)	9 (F)	2	46.26	1194.74
P5	4 (NR)	0	-	1236.54	P5	5 (NR)	7	61.98	1255.02	P5 (00001984)	6 (M)	9	83.90	1375.10
P6	10 (NR)	5	21.55	2573.02	P6	9 (NR)	3	25.59	1177.41	P6 (00002448)	4 (M)	10	119.96	2101.02
P7	9 (NR)	8	167.14	2492.80	P7	10 (NR)	10	60.44	1294.56	P7 (00002657)	5 (M)	10	133.98	2540.01
P8	5 (NR)	0	-	2340.11	P8	3 (NR)	4	62.76	1136.24	P8 (00003053)	5 (F)	1	16.45	1454.55
P9	9 (NR)	0	-	2789.27	P9	6 (NR)	5	109.55	1147.45	P9 (00003281)	13 (M)	2	19.81	1293.18
P10	11 (NR)	0	-	1203.35	P10	6 (NR)	2	35.53	1230.47	P10 (00003306)	13 (F)	4	31.51	1394.48
P11	7 (NR)	0	-	1287.10	P11	5 (NR)	7	152.44	1258.56	P11 (00003635)	6 (M)	7	19.19	1598.80
P12	4 (NR)	2	34.66	2079.85	P12	4 (NR)	6	28.99	1100.01					
P13	7 (NR)	0	-	1609.09	P13	7 (NR)	5	67.63	1182.37					
P14	12 (NR)	16	67.33	3066.19	P14	7, 9 (NR)	4	32.36	3677.64					
P15	9 (NR)	1	18.07	1190.44	P15	5 (NR)	16	193.79	5370.21					
P16	12 (NR)	0	-	1400.93	P16	6 (NR)	4	27.58	1177.42					
P17	5 (NR)	5	65.28	1267.61										
P18	12 (NR)	0	-	2206.33										
P19	11 (NR)	1	16.35	1986.17										
P20	5 (NR)	3	52.93	1281.71										
P21	11 (NR)	1	7.24	1487.88										
Total	**N/A**	**53**	**576.08**	**39152.80**	**-**	**N/A**	**127**	**1498.09**	**51096.91**	**-**	**N/A**	**97**	**837.31**	**21718.61**

either referenced to the average or linked ears montage. Therefore, although one additional patient was also identified with absence seizures, we removed them as they had a different EEG reference montage (average rather than linked mastoids). All records were recorded at 250Hz and all but one patient (P2) was examined routinely; with the exception under long-term-monitoring. As models were assessed using a patient-specific leave-one-out cross-validation scheme, the pre-determined groups of training and test sets found in the TUHS were not used, and session records from the same individual patients were combined.

A combined dataset, consisting of both the NHS (Preston) and NHS (Leeds) datasets, was also created to see if the model performance could be further improved with more available data across different institutions. The TUH (Absence) dataset was not included in this combined set due to differences in the reference montage.

4.3 Methods

In this section we start by describing the features extracted in each EEG channel in subsection 4.3.1. In subsection 4.3.2 we then give an overview of the four machine learning classifiers used to separate features into ictal and interictal classes. Subsection 4.3.3 then briefly describes how Bayesian optimization was used to search over model hyperparameters, focusing on differences from chapter 3. For a description of how performance was assessed during training and on left-out patient datasets, see subsection 3.3.4.

4.3.1 Feature Extraction

Features were chosen based on those most commonly selected for classification from the random forest model stacking in chapter 3. Firstly, we calculated the median power of frequency bands using a fast Fourier transform, in 1-4Hz, 4-8Hz, 8-12Hz, 12-30Hz, and 30-70Hz ranges. We also calculated the mean and mean absolute amplitude of 2-4Hz, 4-8Hz, 8-16Hz, 16-32Hz, 32-64Hz, and 64-128Hz frequency bands, using an undecimated wavelet transform (UDWT) with the db4 wavelet family. UDWT was used rather than a decimated wavelet transform (DWT), as although it takes longer to calculate, it has been shown to result in better discrimination between noise and activity and has a more precise frequency

localisation (see chapter 2 for details; Mamun et al., 2013). A common feature, that was not used in chapter 3, was the entropy of a signal. Entropy-based approaches are commonly used for seizure detection (e.g. Orellana and Cerqueira, 2016; Zhu et al., 2017) as they quantify the regularity and unpredictability of a non-stationary EEG signal (see chapter 2 for details). Entropy can be applied in a number of ways, with this study using spectral and sample entropy, due to research showing these are better suited to physiological time-series than approximate entropy (Richman and Moorman, 2000). Phase can also be calculated (e.g. Parvez and Paul, 2016), however it is a computationally intensive calculation and, given the volume of data, was therefore not used. All data was scaled on a patient-by-patient basis by removing the mean and scaling to a unit variance of 1. Although most models trained in this research are based around tree classifiers, which do not require data to be scaled, scaling was conducted because EEG data scales vary according to numerous factors; such as amplifier gain and electrode contact to the skin. Scaling was completed separately for each feature in each channel, apart from features in the frequency domain, where scaling was completed for each type of feature across all frequency bands for each channel (e.g. C3 Mean 2-4Hz, 4-8Hz, 8-16Hz, 16-32Hz, 32-64Hz, 64-128Hz). Scaling across frequency bands was used to ensure scale relationships between frequency bands were not altered.

4.3.2 Signal Classification

In chapter 3 we found that imbalances in the data were a challenge for effective and consistent model training. As such, this study focuses on ensemble models that have been shown to be effective with imbalanced data. We choose to focus on two balanced bagging methods; a balanced random forest (BRF) and a balanced bagged ensemble of k-nearest neighbours (BKNN) models, as well as two balanced boosted methods; RUSBoost and LightGBM.

Balanced Bagging Ensembles individually train multiple base classifiers on a random undersample of data without replacement, which then are used in an ensemble to vote on the class of unseen data. In other words, instead of training on just one random undersample of data, this enables the training of individual models on multiple undersamples of data. Indeed, bagged classifiers over balanced bootstrapped samples have been recommended for

imbalanced datasets by many authors (e.g. Wallace et al., 2011; Hido et al., 2009; Liu et al., 2008). Balanced bagging can be applied to most classifiers, although are commonly used to create a BRF from multiple tree classifiers (Chen et al., 2004). In this research we used the `BalancedRandomForestClassifier` from `imbalanced-learn` (0.5.0; Lemaitre et al., 2017). As KNN was the best performing model in chapter 3, we also trained a bagged ensemble of KNN models using the `BalancedBaggingClassifier` function from `imbalanced-learn` with the `KNeighborsClassifier` from `Scikit-learn` (0.21.3; Pedregosa et al., 2011) as the base estimator. To our knowledge, both variants have not been applied to EEG seizure detection, although BRF has been applied to detect tonic seizures using surface electromyography (Larsen et al., 2014).

Balanced Boosted Ensembles are typically comprised of weak estimators that are sequentially built so that each estimator attempts to reduce the bias of the previous model (Géron, 2019). In this research we use RUSBoost, which alters AdaBoost (subsubsection 2.6.2) by adding data sampling into the algorithm, so that examples from the majority class are balanced and randomly sampled from the full data each iteration (Seiffert et al., 2008). RUSBoost has been shown to perform comparably to SMOTEBoost, an oversampling method, but is simpler and faster to implement (Seiffert et al., 2009). RUSBoost has been applied to seizure detection using spectral, spatial, temporal (Amin and Kamboh, 2016), and dynamic mode decomposition (Solaija et al., 2018) features; achieving good sensitivity with low computational cost. In this work we used the `RUSBoostClassifier` from `imbalanced-learn` (0.5.0; Lemaitre et al., 2017). Currently, the `KNeighborsClassifier` from `Scikit-learn` cannot be used with a boosting model as it is unable to change the sample weight at each fit. Compared to bagging models, boosting generally leads to a decrease in bias, however it is prone to overfitting to the training data (high variance; Raschka and Mirjalili, 2019). Furthermore, some implementations of boosted trees can be slower than bagged trees as it is easy to parallelise bagging models comparative to boosting. However, an efficient boosting model is LightGBM (Ke et al., 2017), which has a number of improvements upon basic gradient boosted decision tree (GBDT) algorithms (see subsubsection 2.6.2). LightGBM is often compared with XGBoost (Chen and Guestrin, 2016),

which has previously been used in combination with Bayesian hyperparameter optimisation to classify seizures in the TUHS dataset, where only KNN was found to surpass its performance (Roy et al., 2019a). In their application to imbalanced datasets, at the time of writing, both XGBoost and LightGBM can scale the positive weight and subsample the data, but only LightGBM can specifically subsample the negative (interictal) or positive (ictal) samples. Due to this, and their similarities, we chose to focus only on `LightGBM` (2.2.3) as a balanced gradient boosting ensemble, rather than both.

4.3.3 Optimisation and Cross-Validation

Similar to chapter 3, a Bayesian optimisation method, using the `fmin` function from the `Hyperopt` package (0.2; Bergstra et al., 2013), was used to search over model hyperparameters for each classifier. The search space (see table 4.3.1) begins with a random combination of hyperparameters, which were optimised over 1000 iterations. The objective function, the mean F1-score from a stratified 3-fold cross-validation, was used at each iteration to update a prior from a history of model configuration and score pairs. Generally, the search spaces for the hyperparameters were set to cover a range of potential values around either the default, prior work, or encompassing all available options for the hyperparameter. Each model had different numbers of available parameters to optimise, with bagged models having less than boosted models; BKNN (7), BRF (8), RUSBoost (9). LightGBM (12). Having more hyperparameters to fine-tune can be a limitation as it makes it less clear what the effect each hyperparameter has on the outcome. Furthermore, for Bayesian optimisation, it means more runs are required to start finding parameters based on a model; as the optimisation starts by randomly testing parameters in the search space, more random tests are required the more hyperparameters searched over. For each training data set, the Bayesian optimisation process was re-run 2 times to test if similar models performed better using different random states and starting parameters.

Table 4.3.1: Hyperparameter search spaces for different classifiers.

Algorithm	Hyperparameter	Parameters
BKNN	Nearest Neighbors	randint(1,10)
	Weights	choice(uniform, distance)
	Algorithm	choice(ball tree, kd tree, brute)
	Metric	minkowski
	p	randint(1,10)
	Leaf Size	normal(m=30, sd=8)
	Number of Estimators	randint(1,10)
	Max Features	uniform(0.01, 1.0)
	Bootstrap Features	TRUE
BRF	Number of Estimators	randint(1, 2500)
BRF/RUSBoost	Criterion	choice(Gini, Entropy)
	Max Depth	choice(None, randint(1, 50))
	Min Samples Split	randint(2,10)
	Min Samples Leaf	randint(1, 10)
	Max Leaf Nodes	choice(None, randint(2, 40))
	Max Features	uniform(0.01, 1.0)
	Minimum Impurity Decrease	uniform(0.00005, 0.01)
RUSBoost/LightGBM	Learning Rate	loguniform(log(0.01), log(0.2))
	Number of Estimators	randint(1, 200)
LightGBM	Boosting Type	choice(gbdt, goss, dart)
	Num Leaves	randint(2, 40)
	Max Depth	choice(None, randint(1, 25))
	Subsample For Bin	200000
	Objective	binary
	Min Split Gain	0
	Min Child Weight	uniform(0.001, 5.)
	Min Child Samples	randint(1, 30)
	Subsample	1
	Subsample Freq	1
	Colsample By Tree	uniform(0.1, 1.)
	Reg Alpha	uniform(0., 1.)
	Reg Lambda	uniform(0., 1.)
	Importance Type	split
	Scale Pos Weight	uniform(0., 10.)
	Neg Bagging Fraction	uniform(0.01, 1.0)

Note. Some hyperparameters are shared between classifiers.
choice: choose one; randint: random integer; normal: normal distribution; uniform: value selected randomly between lower and upper bounds; loguniform: a log-uniform distribution.

4.4 Results

In this section, we start in subsection 4.4.1 by describing the validation results gained during training. Subsection 4.4.2 then looks at the performance of the best classifiers on each respective held-out patient test set. In subsection 4.4.3 we examine the most important features used in the ensembles for classification. Finally, similar to chapter 3, Figure 4.4.4 looks at how performance can be improved using post-processing of the predictions. Similar to chapter 3, patient results are typically displayed in average, with further patient-by-patient details available in the supplementary information document (https://bit.ly/3bZQxop).

4.4.1 Validation Scores

LightGBM classifiers were the highest scoring models on the validation data, as measured by the average maximum F1-score gained during training optimisation across each training dataset where a patient was left-out (see figure 4.4.1). The ordering of the methods, for both the training times and average maximum validation scores, were similar across the different datasets. BRF models generally had the lowest maximum validation score, followed by BKNN, RUSBoost, and LightGBM performing best. The boosting methods (RUSBoost and LightGBM) generally had similar performance, but LightGBM had a consistently higher average score, smaller standard deviation of scores, and was faster to train (see table 4.4.1). All optimal models had similar validation scores between re-runs with different random states.

The TUH (Absence) dataset had the highest maximum validation F1-score across each model comparative to other datasets. This could be due to a number of potential factors. A simple explanation is that there are fewer patients in the sample, meaning there are fewer folds in the leave-one-patient-out cross-validation and less variability of EEG data comparative to other datasets. Another explanation could be that all files in the TUHS are pruned versions of original EEG recordings; meaning unlike the other two NHS datasets, there is a lack of continuous EEG data. This also means these records have fewer artefacts, although this can only be assessed visually as no artefact labels were provided with the records. Furthermore, hyperventilation and photic stimulation only occurred in 64% of TUH (Absence)

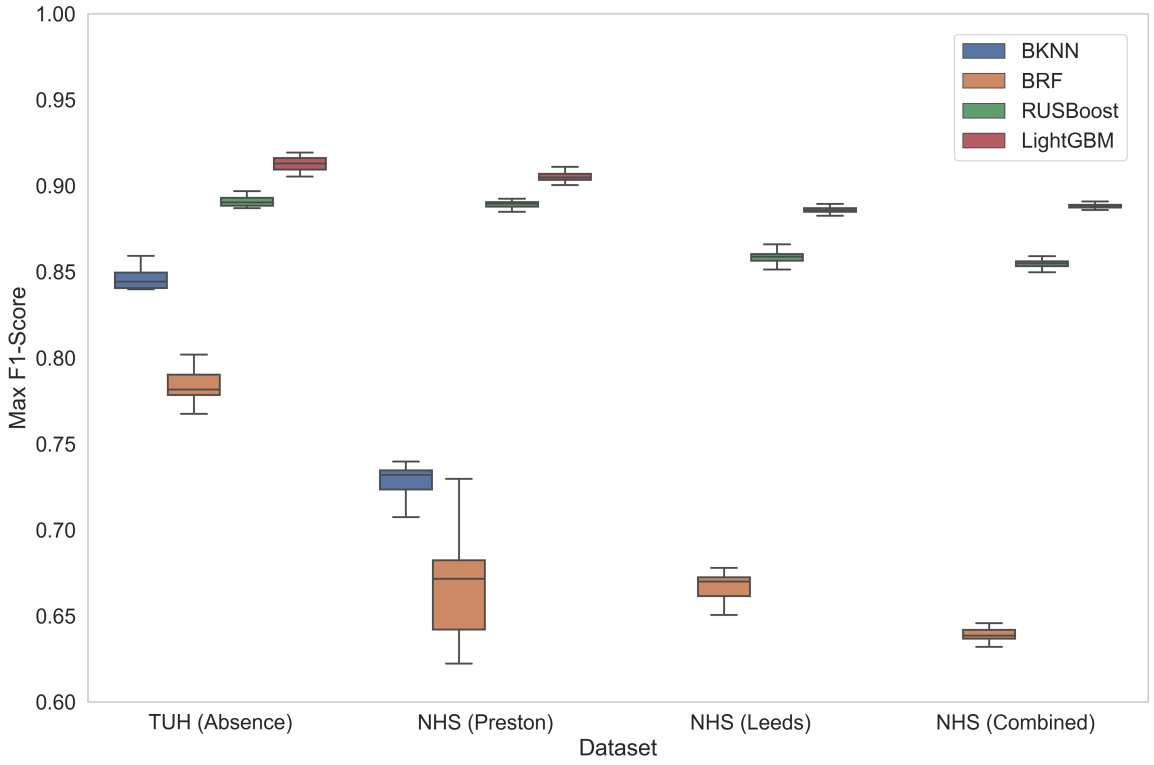


Figure 4.4.1: Boxplot of maximum validation scores across each training dataset where a patient was held-out.

Table 4.4.1: Average and total training times across each held-out training data.

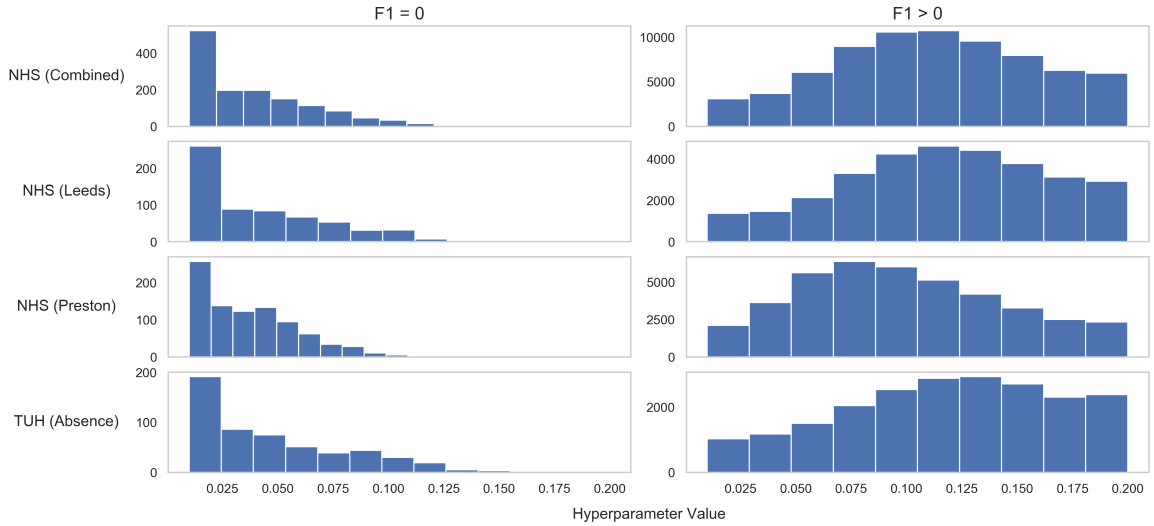
Data Name	**Classifier**	**Training Time (hrs)**			
		1 Bayesian Iteration		**Total**	**(Per EEG hr)**
		Mean	***(SD)***		
TUH	Bagged KNN	0.15	(0.29)	3309.56	528.22
	Balanced RF	0.08	(0.10)	1787.76	285.33
	RUSBoost	0.03	(0.02)	584.84	93.34
	LightGBM	**0.005**	**(0.003)**	**115.91**	**18.50**
NHS (Preston)	Bagged KNN	0.15	(0.27)	6235.53	565.03
	Balanced RF	0.10	(0.08)	4048.79	366.88
	RUSBoost	0.01	(0.01)	604.74	54.80
	LightGBM	**0.005**	**(0.002)**	**196.64**	**17.82**
NHS (Leeds)	Balanced RF	0.14	(0.13)	4583.53	313.73
	RUSBoost	0.03	(0.02)	890.20	60.93
	LightGBM	**0.005**	**(0.003)**	**170.81**	**11.69**
NHS (Combined)	Balanced RF	0.48	(0.37)	35156.36	1370.86
	RUSBoost	0.06	(0.04)	4326.86	168.72
	LightGBM	**0.01**	**(0.005)**	**798.84**	**31.15**

Note. Average and standard deviation training time of 1 iteration of the Bayesian optimisation is calculated across all training folds of the dataset. Total training time is the total CPU hours needed to train all optimisation iterations on all training folds. Total (Per EEG hr) is the total training time divided by the total number of hours of EEG in the dataset.

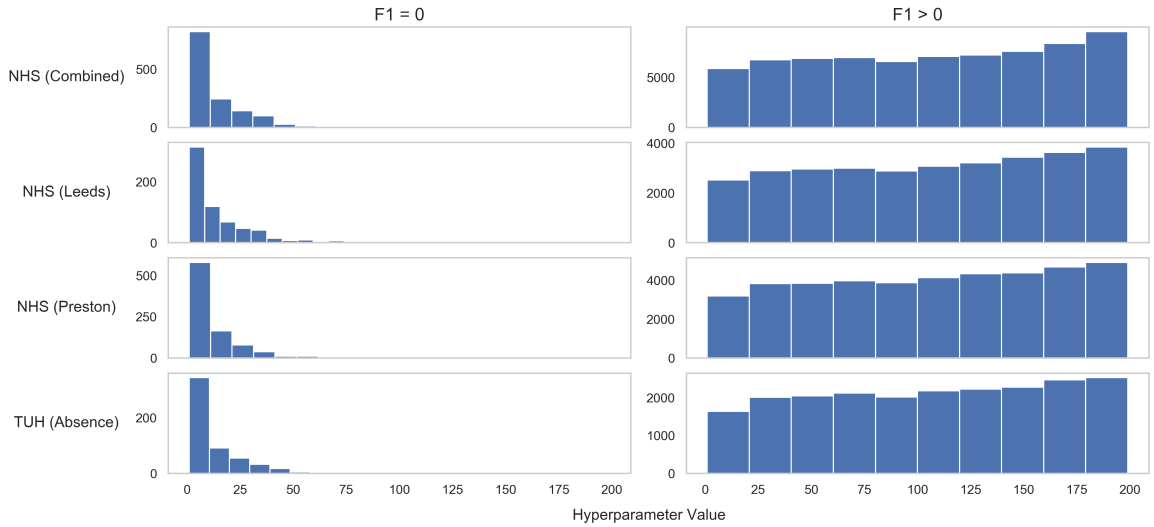
records, compared to 100% of NHS (Preston) and 95% of NHS (Leeds) records. Similarly, there are more records where the patient is sleeping in the TUH (Absence) dataset (29%), compared to NHS (Preston) and NHS (Leeds); 0% and 5% respectively. This is another reason why EEG records in the TUH (Absence) dataset may be less affected by artefacts, as they are less common in EEG recorded while a patient is sleeping or hyperventilating.

NHS (Preston) had the highest variation in validation F1-scores across models and datasets. This could be due to this dataset having the most records without any seizures (see table 4.2.1). Indeed, in chapter 3 and this work (see figure 4.A.1), performance generally was lower on records with no seizures present. Similar to models trained on the TUH (Absence) dataset, NHS (Preston) bagged models have an order of magnitude higher training time comparative to boosted models. Indeed this is the reason only one of the two bagged models, BRF, was subsequently run on the NHS (Leeds) and NHS (Combined) datasets; which both have increased data sizes. Models trained on the NHS (Leeds) and NHS (Combined) datasets have comparable results for the maximum F1-score on the validation set, although combining the datasets did reduce the standard deviation (SD), meaning the performance across patient datasets was less variable. Training time was unsurprisingly the largest for models on the NHS (Combined) dataset due to the increased size of the training data. However, the increase in average training time from combining the datasets was larger than the sum of each NHS dataset separately. Although, this increase in training time was considerably smaller for LightGBM models than BRF and RUSBoost models.

Despite LightGBM having the most hyperparameters available to tune, it was generally the least sensitive model to changes in hyperparameter values during optimisation. However, some LightGBM model configurations did get an F1-score of 0 during optimisation, reflective of a model which only predicted inter-ictal labels. This was likely due to low hyperparameter values for the learning rate and number of estimators for these poor performing models (see figure 4.4.2). Although generally performing worse, the BRF was also relatively insensitive to parameter values. Conversely, both RUSBoost and BKNN models had a greater variation in score according to hyperparameter values. The average optimal model hyperparmeters were generally similar across datasets for each model, with the maximum number of features for BRF being the notable exception (see table 4.A.3).



(a) Learning Rate



(b) Number of Estimators

Figure 4.4.2: Histogram of hyperparameter values for LightGBM models when the validation F1-score score was either equal to zero or higher.
Note. The distribution of other hyperparameter values were not substantially different.

4.4.2 Test Scores

Similar to the validation scores, where LightGBM had the best average maximum F1-score and lowest training time, LightGBM generally had the best average performance across the held-out patient test records on all metrics except sensitivity; as well as F1-score and AUC on some datasets (see table 4.4.2). In the TUH (Absence) dataset, LightGBM consistently scored the best across all performance metrics, except sensitivity where BRF scored the

Table 4.4.2: Average (and standard deviation) test scores across patient held-out datasets.

Dataset	Classifier	Combined	Accuracy		Sensitivity		Specificity		Precision		F1-score		AUC		FP/h		Pred Time	
			Mean	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*
TUH (Absence)	BKNN	False	98.05	(1.51)	84.71	**(6.75)**	98.73	(1.37)	74.61	(16.58)	78.29	(10.42)	95.8	(2.98)	43.16	(46.28)	5.76	(3.27)
	BRF		97.3	(1.7)	**87.01**	(7.14)	97.86	(1.78)	65.07	(22.27)	71.95	(16.03)	97.47	**(1.71)**	73.73	(61.69)	0.56	(0.34)
	RUSBoost		98.43	(1.1)	81.65	(9.0)	99.3	(0.96)	82.09	(19.77)	79.99	(12.47)	97.05	(2.33)	24.26	(32.56)	1.41	(0.51)
	LightGBM		**98.63**	**(1.01)**	81.35	(8.5)	**99.46**	**(0.83)**	**88.21**	**(12.29)**	**83.92**	**(6.96)**	**97.76**	(2.33)	**18.65**	**(28.0)**	**0.11**	**(0.07)**
NHS (Preston)	BKNN	False	97.61	(2.26)	81.76	(17.38)	98.16	(2.19)	69.41	(21.44)	72.3	(16.78)	95.43	(3.88)	65.56	(78.84)	2.29	(0.64)
	BRF	False	97.05	(2.63)	76.68	(21.78)	97.58	(2.83)	63.98	(26.66)	66.28	(22.4)	96.72	**(2.22)**	86.43	(102.14)	0.49	(0.33)
	RUSBoost	True	96.12	(3.79)	**86.62**	(16.13)	96.33	(3.93)	56.87	(23.37)	63.88	(15.45)	**97.21**	(2.23)	130.25	(141.29)	0.28	(0.18)
		False	98.82	(1.73)	65.13	(29.74)	99.68	(1.06)	87.4	(27.98)	71.52	(29.26)	96.46	(3.26)	11.37	(38.22)	0.86	(0.2)
		True	98.72	(2.12)	79.02	**(6.67)**	99.21	(2.1)	91.18	(11.87)	**84.09**	**(6.63)**	96.14	(3.55)	28.11	(75.6)	0.43	(0.13)
	LightGBM	False	98.91	**(1.54)**	63.87	(31.83)	**99.77**	**(0.71)**	90.61	(27.42)	71.24	(33.3)	95.99	(2.78)	**8.27**	**(25.62)**	0.1	(0.06)
		True	**99.03**	(1.66)	74.59	(22.28)	99.48	(1.67)	**94.59**	**(6.15)**	80.48	(21.79)	96.76	(3.18)	18.67	(59.96)	**0.03**	**(0.01)**
NHS (Leeds)	BRF	False	95.09	(3.47)	**86.23**	**(8.97)**	95.6	(3.91)	57.0	(29.77)	63.97	(22.77)	96.09	(3.3)	151.8	(135.34)	0.28	(0.45)
	RUSBoost	True	96.43	(2.84)	85.41	(13.25)	96.98	(3.05)	63.19	(27.65)	69.73	(22.82)	96.67	(3.2)	103.88	(105.96)	0.44	(1.17)
		False	97.72	(1.58)	76.23	(13.52)	98.9	(1.52)	78.64	(25.2)	75.28	(17.04)	96.27	(3.07)	37.78	(52.33)	0.74	(2.04)
		True	97.72	(1.54)	66.16	(19.71)	99.43	(0.71)	82.65	(21.11)	71.78	(18.78)	95.38	(4.17)	19.77	(24.55)	0.64	(1.68)
	LightGBM	False	98.11	**(0.99)**	75.65	(14.48)	99.31	(0.76)	81.74	(21.47)	**77.35**	**(15.94)**	**96.96**	**(2.43)**	23.96	(26.42)	**0.04**	**(0.09)**
		True	**98.21**	(1.18)	68.84	(17.5)	**99.75**	**(0.35)**	**89.41**	**(17.02)**	76.55	(16.32)	96.41	(3.14)	**8.64**	**(12.41)**	0.06	(0.14)
NHS (Combined)	BRF	True	96.26	(3.38)	**85.97**	**(14.39)**	96.61	(3.54)	60.26	(25.48)	67.02	(19.62)	**96.92**	**(2.76)**	118.85	(126.27)	0.35	(0.77)
	RUSBoost		98.29	(1.93)	72.13	(16.24)	99.31	(1.63)	86.61	(17.68)	77.5	**(15.54)**	95.73	(3.84)	24.5	(58.68)	0.53	(1.09)
	LightGBM		**98.67**	**(1.51)**	71.51	(19.69)	**99.6**	**(1.27)**	**91.81**	**(13.19)**	**78.37**	(18.79)	96.57	(3.11)	**14.34**	**(45.68)**	**0.04**	**(0.09)**

Note. The best average score for each metric, across classifiers, are in bold. Results are shown both for when datasets from different NHS sites were trained separately and when they were combined.

highest. However, this higher sensitivity is due to BRF having the highest false positives per hour (FP/h), on average over four times higher than LightGBM models. For the NHS (Preston) dataset, the accuracy, specificity, and F1-score were comparable between boosting classifiers; with both generally performing better on these metrics than the bagged classifiers. However, what separates the two boosted models is the lower average false positives and prediction time for LightGBM models. Similar to the TUH (Absence) dataset, bagging classifiers provided better sensitivity at the expense of a greatly increased false positive rate. Comparative to chapter 3, the sensitivity across all models in this chapter is lower at the expense of having better classification of inter-ictal data segments. Therefore, as there are fewer false positives, the accuracy of models are generally higher in this chapter before prediction post-processing (discussed in Figure 4.4.4). NHS (Leeds) also demonstrates a similar pattern to both NHS (Preston) and the TUH (Absence) datasets; with LightGBM scoring the best on most metrics, except sensitivity.

Combining the NHS (Preston) and NHS (Leeds) data together had different influences on model performance for each dataset than when models were trained on each separately (see table 4.4.2). Firstly, sensitivity for all models and F1-score for boosted models are improved for NHS (Preston) records when models are trained on a combined dataset compared to independently, and worse for the records in the NHS (Leeds) dataset. Conversely, the specificity of all models and FP/h were improved on records in the NHS (Leeds) dataset and reduced in performance for those in NHS (Preston). Therefore, generally this reflects models having a better identification of seizures in the NHS (Preston) set, at the expense of increased false positives, and conversely, reduced false positives in the NHS (Leeds) dataset, at the expense of marking fewer seizure epochs. This could be due to NHS (Leeds) having over twice as many seizure events than NHS (Preston), so this improves the ability of the models to identify them in NHS (Preston) records when combined. Similarly, the improvement in specificity for the NHS (Leeds) records could be due to some NHS (Preston) records having no seizures throughout the record, improving the model's false positive rate.

Examining model performance on individual records reveals that boosted methods are generally more conservative and often correctly label epochs in the middle of seizures rather than towards the onset or offset (see figure 4.A.3). This is unlike chapter 3, where classical

models trained on the NHS (Preston) data generally marked the full seizure. However, boosted models have many fewer false positives, particularly on data with no seizures in the whole record (see figure 4.A.2).

F1-scores for the validation set were generally better than those from the left-out test set (mean difference of 7.56%). However, this effect was different for bagged and boosted models, as bagged models had a smaller difference between validation and test sets (BKNN, 3.19%; BRF, 2.56%; RUSBoost, 11.90%; LightGBM, 12.02%). This difference was generally consistent across datasets, apart from TUH (Absence) where the difference was more similar across models (see figure 4.4.3). This general improvement of F1-scores on the validation data, compared to the test set, could be due to training and validation sets being split to preserve the percentage of samples for each class (stratified cross-validation), rather than separating patient records so different segments from the same patient do not appear in both sets. Stratified cross-validation was specifically used as a simple k-fold cross-validation might result in training subsamples with no or insufficient instances of the minority class; resulting in an unrepresentative subsample to train with. For each left-out training dataset, no data from the test set patient record was contained, therefore this performance decrease on the test data may be due to models simply being worse at classifying EEG from unseen patients due to baseline differences in brain activity (for example see Deiss et al., 2018). However this could also reasonably reflect, or be contributed to by, optimisation overfitting to the training/validation data. Nevertheless, for real world applications of general models,

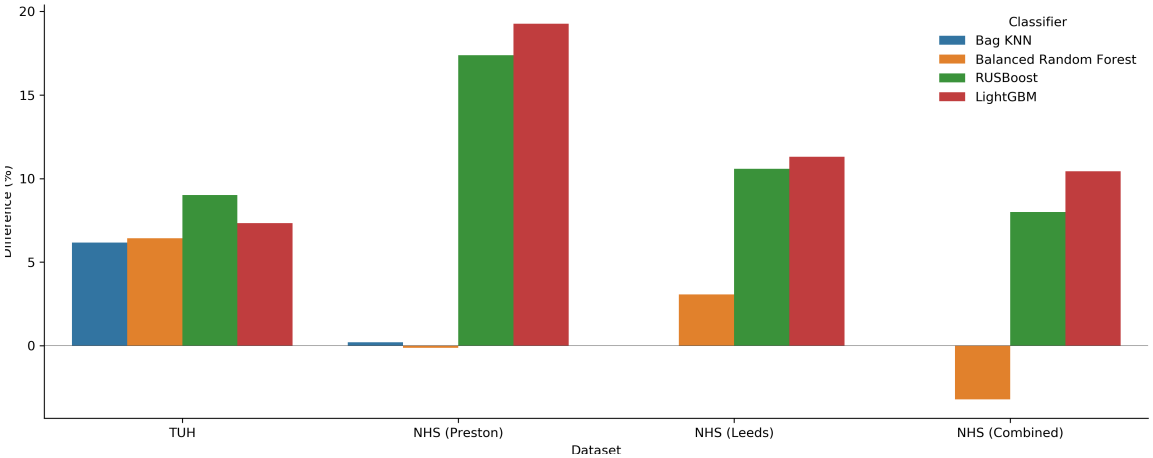
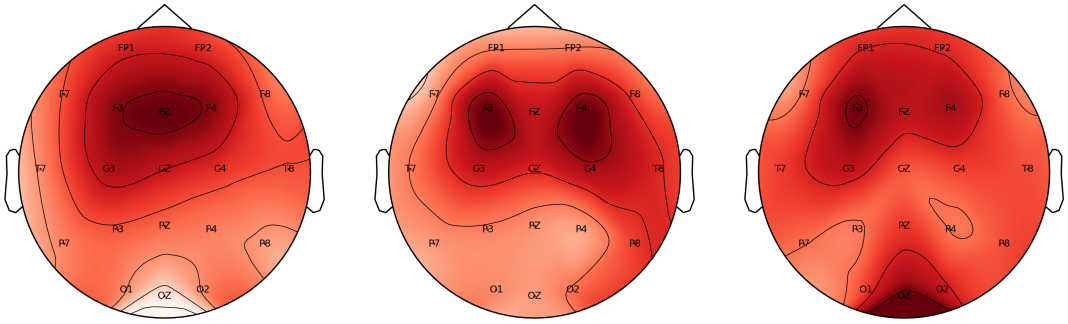


Figure 4.4.3: Average F1-score increase in the validation set comparative to the test set.

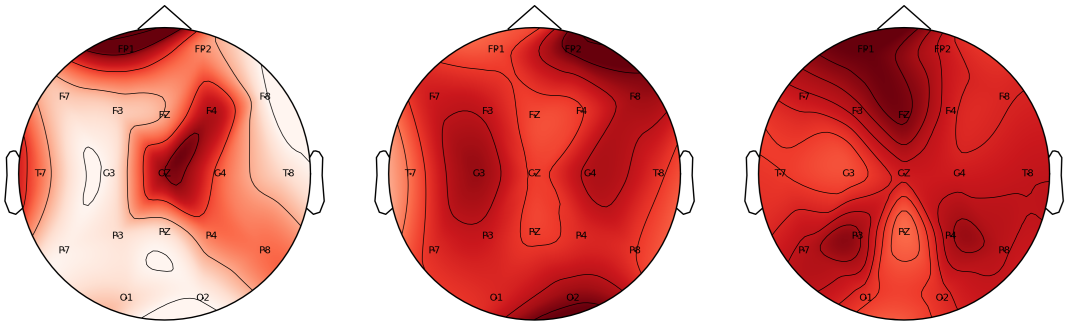
which are not trained to be specific to particular individuals, it is typically expected that there will be worse performance than those reported from models trained and tested on parts of the same EEG records (see section 2.7). In reality, training is unlikely to occur using part of an individual patients record; with training more likely to occur across a range of patient data external to the individual either at the hospital or trust level.

4.4.3 Feature Importances

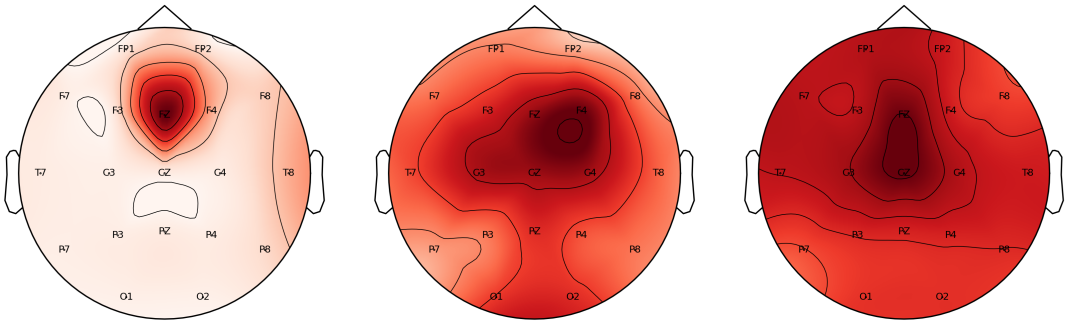
To investigate which features were the most important for seizure detection, we look at the average feature importance for the best BRF, RUSBoost, and LightGBM models across left-out patient datasets. The largest difference between models is that BRF models focused on more specific channel locations and areas of the frequency spectrum than RUSBoost and LightGBM models. As shown in figures 4.4.4 and 4.A.4, BRF models primarily use the frontal channels, as well as some temporal or central locations, for classification. Similar to findings in chapter 3, this arguably represents an absence seizure; which although primarily generalized, is more associated with frontal channels. The other two models still predominantly use the frontal and central channels, but to a much less prominent degree. Another main difference between models in this chapter is that BRF models generally use features in the slower frequencies, between the range of 4-16Hz, much more than features in the other frequencies or entropy measures (see figure 4.A.5). This pattern still occurs in RUSBoost models, with slower frequency components more important than faster frequencies, but to a smaller degree. LightGBM models appear to use features in different frequencies relatively similarly, although there is more use of entropy features than the other models. Focusing on differences between the datasets, although TUH (Absence) is broadly similar in important features, it differs slightly from the other datasets specifically for BRF models; where more features across the frequency range are used rather than being as selective. Its worth noting the TUH (Absence) dataset has a different reference to the NHS (Preston) and NHS (Leeds) datasets, as it has been re-referenced to the linked mastoids rather than the average, as well as the other differences outlined in subsection 4.4.1. Additionally, non-EEG channels are present in this data not found in the NHS data, which although are the least important for BRF and RUSBoost models, do still contribute to LightGBM models (further discussed in



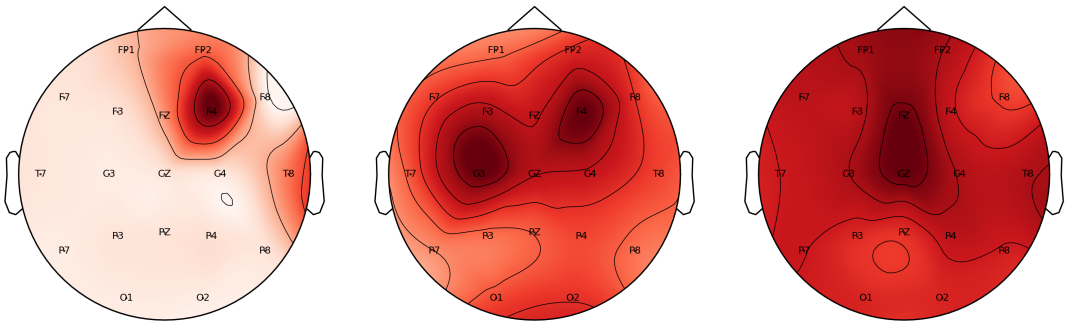
(a) TUH (Absence) - BRF (b) TUH (Absence) - RUSBoost (c) TUH (Absence) - LightGBM



(d) NHS (Preston) - BRF (e) NHS (Preston) - RUSBoost (f) NHS (Preston) - LightGBM



(g) NHS (Leeds) - BRF (h) NHS (Leeds) - RUSBoost (i) NHS (Leeds) - LightGBM



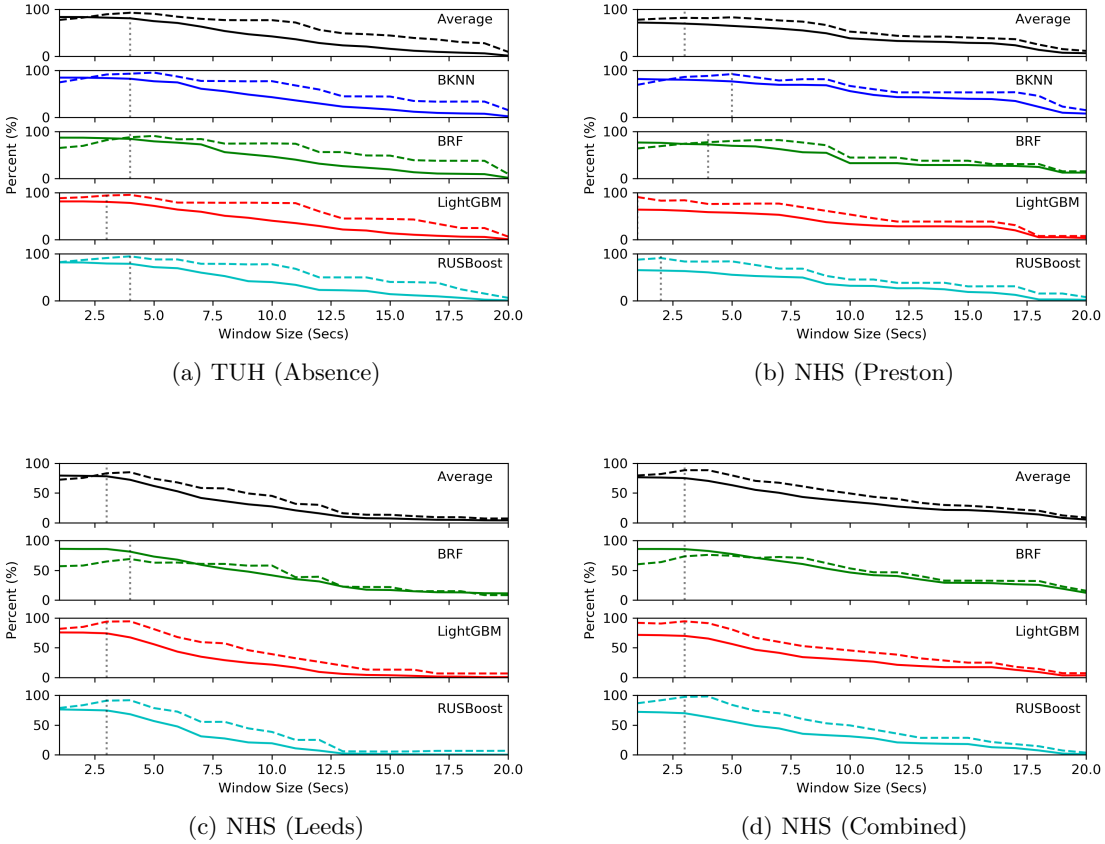
(j) NHS (Combined) - BRF (k) NHS (Combined) - RUSBoost (l) NHS (Combined) - LightGBM

Figure 4.4.4: Topoplots of average feature importance, according to electrode location, across features and models trained on different left-out patient training data.

section 4.5).

4.4.4 Prediction Post-Processing

The majority of performance metrics can be improved with post-processing of the predictions. This requires finding an appropriate threshold for the length of a seizure prediction, to remove short predictions. Similar to chapter 3, the best window sizes for this threshold varied between 3 and 5 seconds across datasets and classifiers (see figure 4.4.5). A notable exception to post-processing improving performance was to the LightGBM models in the NHS (Preston) dataset, where any post-processing had a detrimental effect on performance. This could be due to this model generally predicting short seizures that did not cover the full length of seizures. Therefore increasing the threshold for the length of a prediction removed the short positive seizure predictions, affecting performance more than in other



(a) TUH (Absence)

(b) NHS (Preston)

(c) NHS (Leeds)

(d) NHS (Combined)

Figure 4.4.5: Effects of post-processing window size on test set performance metrics.
Note. Thick line is sensitivity and the dashed line is precision.

models where predictions were more often longer and grouped.

As can be seen in figure 4.4.6, there is a greater change to performance metrics for the bagging models than boosting models due to post-processing. This change is more prominent in the smaller datasets (TUH, Preston) than the larger datasets (Leeds, Combined). Although generally boosting methods are still preferable, post-processing significantly reduces the difference in performance between bagging and boosting performance metrics (see table 4.A.4). This further emphasises the main performance limitation of bagging models compared to boosting models being the false positive rate rather than model sensitivity.

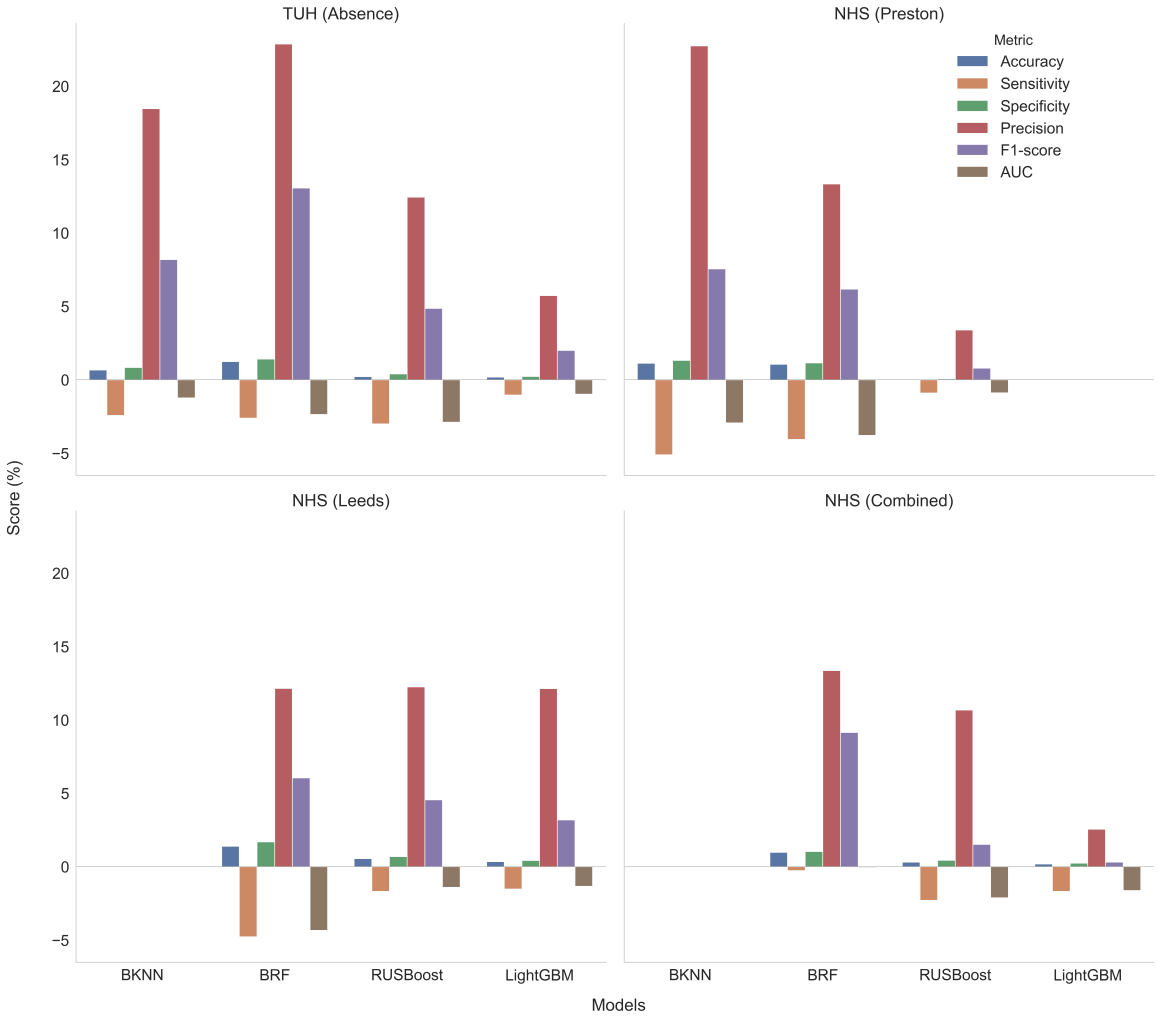


Figure 4.4.6: Average test set score change (%) due to post-processing on various performance metrics.
Note. LightGBM models on NHS (Preston) were not improved with any post-processing (see figure 4.4.5). Bag KNN was not run on NHS (Leeds) or NHS (Combined) datasets.

4.5 Discussion

In this chapter we assessed four balanced ensemble classifiers for the automatic detection of absence epilepsy seizures on three different datasets; making this the largest number of patients used for absence seizure detection in the current literature. Similar to chapter 3, optimal hyperparameters were found using Bayesian optimisation and datasets were chosen to reflect records gained during routine clinical practice. Balanced ensembles are not often applied to seizure detection, but are nevertheless useful for classifying data with large class imbalances. The models reported from this research have greater specificity and precision than most previous research (see tables 3.5.1 and 4.5.1) despite containing a significant amount of artefactual data. However, due to being more conservative than other models in chapter 3, they have comparatively poor sensitivity.

Balanced ensemble models generally were found to have a high specificity and precision, correctly marking inter-ictal segments, and a lower false positive rate comparative to classical models in chapter 3 (see table 4.5.2). This is likely due to each classifier in the ensemble training on different sampled inter-ictal epochs, meaning the ensemble overall used a broader range of the full training data than if one undersample was taken to train the whole model. However, the lower sensitivity in these models means less of a seizure was marked where detected than classical models. For example, although LightGBM models without post-processing on the NHS (combined) dataset marked at least 1 second within nearly every seizure (missing 1 seizure in P35 and 6 in P23), the full duration of seizures were often not marked. Electrical artefacts were still the leading cause of false positives; as shown by the similar misidentification of data segments in P18 in the NHS (Preston) dataset as in chapter 3. Nevertheless, for LightGBM models on the NHS (combined) dataset, 30% of patient records had less than 1 false positive per hour, and 49% less than 5 (see Supplementary Information), which is a vast improvement. Classical and ensemble methods therefore appear to describe the data in different, but complimentary, ways; as the more specific ensemble algorithms are more useful to determine the number of seizures present in a record, and the more sensitive classical methods on the duration of seizures.

In this chapter we found that bagged ensembles generally had better sensitivity than

Table 4.5.1: Comparison between average model performance reported in this paper (after post-processing), using the TUH (Absence) dataset, to other TUH papers which report absence seizure detection performance.

Paper	Classifier	Sensitivity	Specificity	F1-score	AUC
Iešmantas and Alzbutas (2020)	CNN	80.00	66.00	-	72.00
Liu et al. (2020)	CNN	-	-	58.60	-
	RNN	-	-	66.19	-
	Hybrid	-	-	67.70	-
This Paper	BKNN	82.28	99.57	**86.48**	94.58
	BRF	**84.4**	99.28	85.01	95.11
	RUSBoost	78.65	**99.7**	84.86	94.17
	LightGBM	80.31	99.69	85.92	**96.78**

Note. The best average score for each metric, across classifiers, are in bold. For the full post-processed scores see table 4.A.4.

Table 4.5.2: Table showing how the post-processed binary algorithms scores compare between chapters on the NHS (Preston) dataset.

Chapter	Classifier	Accuracy		Sensitivity		Specificity		Precision		F1-Score		FPR/h		Prediction Time (secs)	
		Mean	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*	*Mean*	*(SD)*
Chapter 3	**K-Nearest Neighbours**	**99.39**	**(1.06)**	88.14	(12.77)	**99.55**	**(1.07)**	**93.78**	**(8.77)**	**90.36**	**(9.17)**	**15.77**	**(38.52)**	1.91	(0.57)
	Random Forest	93.44	(21.65)	80.39	(36.05)	94.22	(21.66)	65.16	(34.14)	70.64	(33.39)	207.49	(779.98)	0.49	(0.28)
	Support Vector Machine	98.93	(1.99)	89.07	(10.57)	99.11	(2.03)	86.57	(22.04)	85.28	(17.88)	31.65	(73.23)	**0.08**	**(0.04)**
	Soft Majority Vote	98.95	(1.83)	**93.85**	**(6.25)**	99.05	(1.89)	87.73	(16.22)	89.69	(10.08)	33.63	(67.07)	2.49	(0.6)
This Chapter	Bagged K-Nearest Neighbours	98.74	(1.61)	**76.66**	**(24.97)**	99.48	(0.9)	**92.17**	**(13.14)**	**79.86**	**(22.5)**	18.58	(32.58)	2.29	(0.64)
	Balanced Random Forest	98.10	(1.91)	72.62	(28.13)	98.73	(2.0)	77.32	(30.3)	72.45	(26.85)	45.51	(72.08)	0.49	(0.33)
	RUSBoost	98.87	(1.68)	64.24	(30.52)	99.75	(0.88)	90.79	(27.51)	72.31	(30.59)	8.79	(31.64)	0.86	(0.2)
	LightGBM	**98.91**	**(1.54)**	63.87	(31.83)	**99.77**	**(0.71)**	90.61	(27.42)	71.24	(33.3)	**8.27**	**(25.62)**	**0.10**	**(0.06)**

Note. The best average score for each chapter and metric, across classifiers, are in bold. For the full post-processed scores see table 4.A.4.

boosted ensembles, but at the cost of a much higher training time and worse specificity/false positive rate. Indeed, for larger datasets, such as the NHS (Leeds) and NHS (Combined), the BKNN ensemble was not run due to a large estimated run-time. As available data increases for training, we would not recommend BKNN due to the limited performance gains compared to a single KNN model with prior feature reduction shown to be better overall in chapter 3. The boosted models (RUSBoost and LightGBM) performed similarly, although LightGBM generally had slightly improved performance and was faster to train, at the expense of having more hyperparameters to fine-tune. LightGBM, although not previously used for seizure detection, is generally a popular ML method due to model performance and execution speed (e.g. Tyralis and Papacharalampous, 2020; Iskandaryan et al., 2020). Although XGBoost is a more popular gradient boosted model, in this case LightGBM is currently recommended as it has more internal hyperparameters specific for imbalanced datasets not yet available in XGBoost.

Combining the two NHS datasets to make a larger dataset, collected at different NHS trusts, marginally improved accuracy and other model performance metrics. The main benefit of the increased size was for the boosted models, as it ensured that in two records where all seizures were missed entirely, P17 and P19 in NHS (Preston), they became at least partly marked. This therefore suggests training on a larger dataset removes some of the more extreme errors of models, likely due to the larger variety of data available to train on. Furthermore, it also suggests that despite some marginal differences in practices between NHS sites in their diagnostic procedure using EEG (e.g. data lengths, order of procedure, amount of AMPSAT present in records), the data can still be combined between trusts for improved machine learning performance. This is promising, as often it is difficult to generalise model performance across multiple healthcare institutions. Indeed in general, replicability of the results from the balanced ensemble models was improved comparative to classical models in chapter 3, as there was lower variation in scores between re-runs with different random states.

In this chapter we were able to replicate the finding that features from the slower frequency components were generally the most important for optimal random forest models to detect absence seizures. Both this chapter and chapter 3 also demonstrate the frontal,

central, and, to a smaller degree, temporal channels are important for these models to detect absence seizures. This is notable as this similarity exists despite using a different methodology and models to assess the most prominent features; this chapter using the average importance scores from balanced ensembles and chapter 3 using a threshold on forest feature selection counts across models. However, interestingly this effect was less prominent in boosted models, likely representative of these optimal models on average using a larger number of features for training (see table 4.A.3). The TUH (Absence) dataset also differed from these patterns specifically in the BRF models, where a broader range of electrodes and frequencies were used. This no doubt was affected by the much lower maximum number of features (1%) used for training each tree, than for the same model on different datasets. It is also worth noting that this dataset was re-referenced differently to the NHS datasets, via the linked mastoids rather than the average, as well as had more channels which could have affected training.

Comparisons between this chapter and chapter 3 should note the difference in the number of different feature types used in each channel, in order to reduce the time to train the models and chance of overfitting; 1 Time, 2 Frequency, and 2 Time-Frequency in this chapter compared to 2 Time, 4 Frequency, and 5 Time-Frequency in chapter 3. It is possible that the removal of some features, although shown not to be the most important for detecting seizures for classical models, were still beneficial to overall sensitivity performance. It could also have been affected by the different data scaling method using in this chapter, where we scaled across frequency bands. A limitation for comparisons between the TUH (Absence) dataset and NHS datasets was that upon investigation some channels in TUH (Absence) records were not EEG channels, despite being labelled as such; 2 Nasopharyngeal Electrodes, 1 ECG channel, and one misc channel "EEG 30-LE".

Comparisons between results from this chapter to other published papers should note that models trained only to detect childhood absence seizures often are more accurate than models that classify other seizure types. This is due to the EEG patterns being distinct, with little intra-patient and inter-patient variability or movement artefacts during the seizure (Baier et al., 2006). If we focus specificity on the TUH (Absence) dataset used in this work (see chapter 3 for a discussion of published results from other absence datasets), it

is initially apparent that there is currently limited use of the TUHS dataset for seizure detection comparative to older datasets (e.g. Bonn Epileptologie Database, CHB-MIT). What research has been conducted using the TUHS dataset typically aims to detect seizures, or the type of seizure contained in a record, across multiple different seizure types rather than focusing on one type as we did in this research. However, although trained on multiple seizure types, Iešmantas and Alzbutas (2020) does report detection metrics separately for each seizure type from a CNN model. Comparing our results, we find all our models perform better on all reported metrics on patients with absence seizures (see table 4.5.1). Similarly, Liu et al. (2020) also trained on multiple seizure types, and report lower F1-scores than found in this work across their tested models. Indeed, seizure detection performance was the lowest for absence seizures in their paper compared to other seizure types. This is likely due to absence seizures only constituting a small portion of both research papers full datasets (0.68% and 0.5% respectively). Nevertheless, this could be used to demonstrate the performance gains from focusing model training on specific seizure types rather than across multiple.

Other authors have reported metrics based on the detection of the broader category of "generalized" seizures in the TUHS dataset, but these are hard to directly compare to this current work as they cover a much larger range of seizure types (see table 2.A.3). Nevertheless, similar to our findings, boosted models typically perform favourably comparative to other model types (e.g. Vanabelle et al., 2020), with only KNN surpassing XGBoost in performance in Roy et al. (2019a). Looking at other common datasets, RUSBoost has been shown to be better than (Amin and Kamboh, 2016), or comparable to (Solaija et al., 2018), SVM on records in the CHB-MIT dataset. Furthermore, XGBoost has also been demonstrated to have improved performance comparative to a variety of other model types on this dataset, including KNN, SVM, LDA, and CNN (Wu et al., 2020). To our knowledge no previous research on seizure detection uses BRF or BKNN classifiers, as implemented in this research, despite BRF models shown to be effective on a number of highly imbalanced datasets (Chen et al., 2004).

4.6 Conclusion

Overall, this work is consistent in demonstrating the performance gains from using boosted ensemble models for seizure detection, which are often better or comparable to other model types, but with considerably lower training and prediction time. We suggest that boosted ensembles may be more useful for accessing the number of seizures present in a record, due to a low false positive rate and high precision/specificity, and classical models for the length of seizures where present. Furthermore, we have been able to demonstrate alpha frequencies in the frontal, central, and temporal channels as important for machine learning models to detect absence epilepsy seizures. This finding is clearly explainable and justifiable in the context of the data, so could be used as guidance for clinical staff and patients alike. We also found that merging datasets from multiple NHS sites could improve model performance and applicability if future adoption into practice were to occur. Despite future adoption requiring a more user friendly interface to be developed (e.g. Selvakumari et al., 2019), these models show promise for reducing the clinical time required for screening a potential epileptic patient's record; affording more of a physician's time to work on a patient focused treatment plan.

4.A Appendix C

Table 4.A.1: Length of time, rounded to the nearest second, of classification labels in each NHS (Leeds) patient.

	AMPSAT	Baseline	Generalised Epileptiform Discharge	Spikes	Total
P1	13	960	16	17	1006
P2	75	25789	577	207	26648
P3	5	1304	46	9	1364
P4	66	1635	0	10	1711
P5	27	1226	62	2	1317
P6	18	1152	26	7	1203
P7	0	1290	60	5	1355
P8	142	994	63	1	1199
P9	10	1138	110	0	1257
P10	8	1218	36	4	1266
P11	3	1255	152	0	1411
P12	8	1073	29	19	1129
P13	26	1127	68	29	1250
P14	27	3583	32	67	3710
P15	451	4910	194	9	5564
P16	1	1177	28	0	1205
All	**879**	**49831**	**1498**	**387**	**52595**

Table 4.A.2: Length of each seizure, rounded to the nearest second, for each NHS (Leeds) patient.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**Sum**
P1	8	8																																													16
P2	14	19	15	12	18	17	14	19	22	15	14	13	18	16	14	17	15	15	16	21	6	4	8	6	5	5	6	6	10	8	6	3	10	10	11	11	17	13	12	14	14	12	14	13	15	14	577
P3	9	7	9	4	4	8	5																																								46
P5	7	8	9	14	5	10	8																																							62
P6	9	8	9																																											26
P7	7	3	5	8	3	4	9	4	7	9																																					60
P8	13	16	13	20																																										63
P9	20	25	22	21	21																																										110
P10	17	18																																													36
P11	15	25	14	25	24	28	22																																							152
P12	6	5	4	5	5	5																																								29
P13	17	14	8	20	9																																										68
P14	14	8	4	6																																											32
P15	8	11	9	9	8	18	10	18	10	11	15	22	11	24	4	4																																194
P16	8	6	9	4																																											28

Table 4.A.3: The most common categorical or average hyperparameter value for each model across left-out patient training sets.

Classifier	Hyperparameter	TUH		NHS (Preston)		NHS (Leeds)		NHS (Combined)	
		Mean	***(SD)***	***Mean***	***(SD)***	***Mean***	***(SD)***	***Mean***	***(SD)***
BKNN	***Leaf Size***	34.0	*(18.22)*	31.29	*(18.17)*	-	-	-	-
	Nearest Neighbors	2.45	*(3.24)*	1.0	*(0.)*	-	-	-	-
	P	1.0	*(0.0)*	1.0	*(0.)*	-	-	-	-
	Max Features	0.18	*(0.06)*	0.22	*(0.09)*	-	-	-	-
	Number of Estimators	7.0	*(2.14)*	6.48	*(1.78)*	-	-	-	-
	Algorithm	Ball Tree	-	Ball Tree	-	-	-	-	-
BRF	***Max Features***	0.01	*(0.00035)*	0.49	*(0.26)*	0.43	*(0.33)*	0.61	*(0.2)*
	Number of Estimators	1211.91	*(786.21)*	1236.52	*(796.36)*	1052.63	*(758.59)*	1141.05	*(759.76)*
	Min Impurity Decrease	0.0061	*(0.0032)*	0.0037	*(0.003)*	0.0016	*(0.0021)*	0.00058	*(0.00046)*
	Min Samples Leaf	4.73	*(2.61)*	1.81	*(1.40)*	1.94	*(2.41)*	1.027	*(0.16)*
	Min Samples Split	6.91	*(1.92)*	4.67	*(2.39)*	3.25	*(1.73)*	2.57	*(0.73)*
	Criterion	Entropy	-	Entropy	-	Entropy	-	Entropy	-
RUSBoost	***Number of Estimators***	180.64	*(17.48)*	184.67	*(12.33)*	191.31	*(9.75)*	191.78	*(8.45)*
	Max Features	0.69	*(0.16)*	0.58	*(0.25)*	0.76	*(0.16)*	0.75	*(0.19)*
	Min Impurity Decrease	0.0016	*(0.0019)*	0.0013	*(0.0022)*	0.00099	*(0.0018)*	0.00025	*(0.00035)*
	Min Samples Leaf	5.64	*(2.20)*	4.71	*(2.51)*	4.81	*(3.23)*	4.05	*(2.65)*
	Min Samples Split	5.09	*(2.12)*	5.52	*(2.54)*	6.75	*(1.95)*	5.57	*(2.65)*
	Learning Rate	0.066	*(0.0076)*	0.11	*(0.03)*	0.098	*(0.27)*	0.095	*(2.28)*
	Criterion	Entropy	-	Entropy	-	Entropy	-	Entropy	-
LightGBM	***Number of Estimators***	163.18	*(38.52)*	162.48	*(29.36)*	167.69	*(27.38)*	182.70	*(15.89)*
	Max Depth	13.0	*(17.04)*	9.29	*(15.68)*	13.44	*(15.38)*	12.84	*(17.60)*
	Learning Rate	0.16	*(0.024)*	0.12	*(0.038)*	0.14	*(0.03)*	0.12	*(0.023)*
	Column Sample per Tree	0.49	*(0.21)*	0.63	*(0.2)*	0.63	*(0.23)*	0.61	*(0.24)*
	Min Child Samples	14.45	*(8.27)*	15.57	*(9.14)*	12.69	*(7.68)*	14.3	*(9.25)*
	Min Child Weight	2.21	*(0.90)*	1.93	*(0.89)*	2.99	*(1.36)*	3.24	*(1.04)*
	Number of Leaves	19.45	*(8.24)*	21.38	*(9.93)*	24.19	*(9.09)*	25.11	*(4.6)*
	Alpha	0.55	*(0.33)*	0.31	*(0.21)*	0.29	*(0.22)*	0.34	*(0.24)*
	Lambda	0.51	*(0.30)*	0.43	*(0.28)*	0.34	*(0.24)*	0.46	*(0.28)*
	Negative Class Subsample	0.76	*(0.15)*	0.57	*(0.20)*	0.50	*(0.15)*	0.67	*(0.16)*
	Boosting Type	Gradient	-	Gradient	-	Gradient	-	Gradient	-

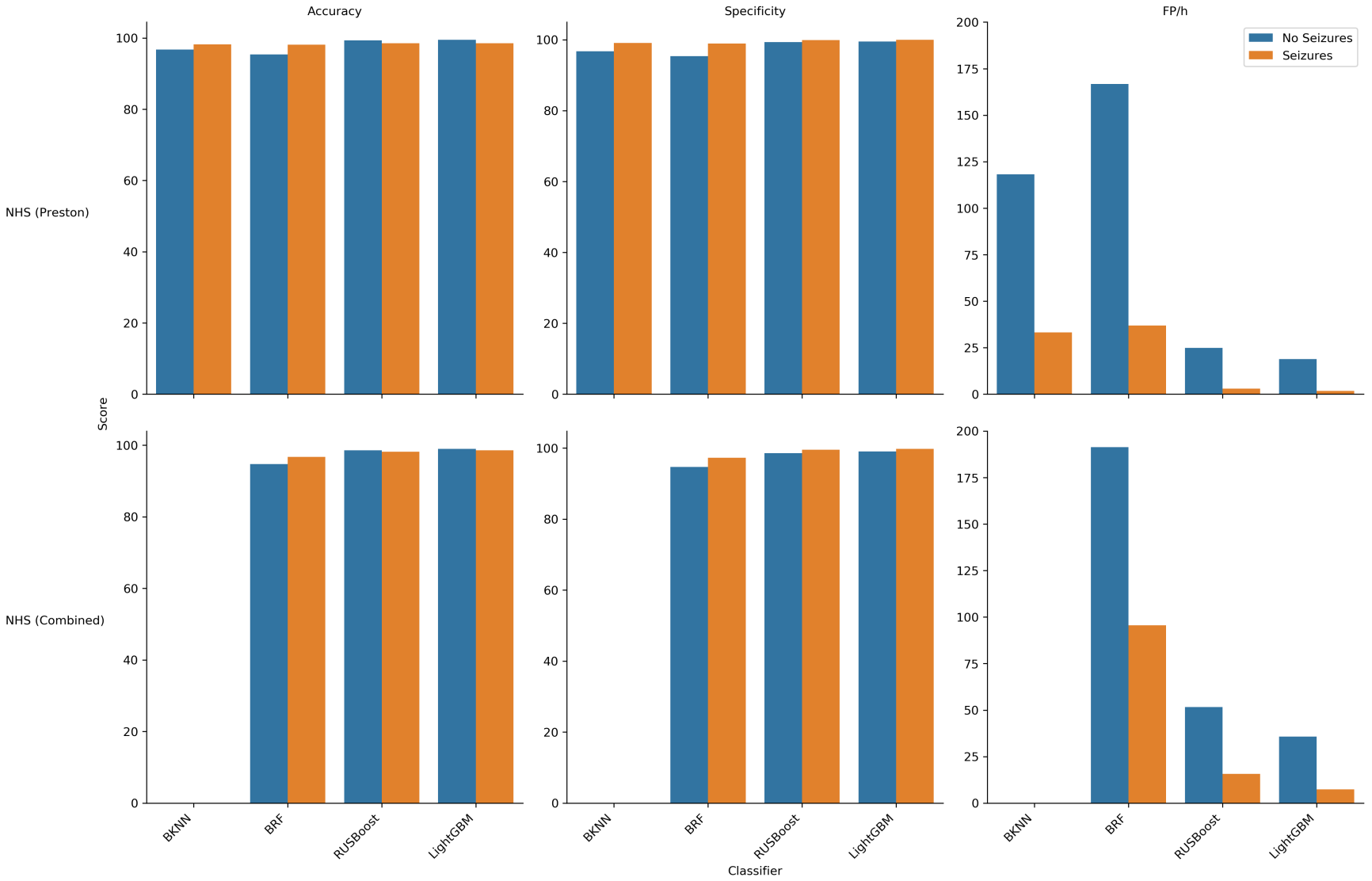


Figure 4.A.1: Difference between average scores, across left-out patient test sets, when records had seizures present or when absent.

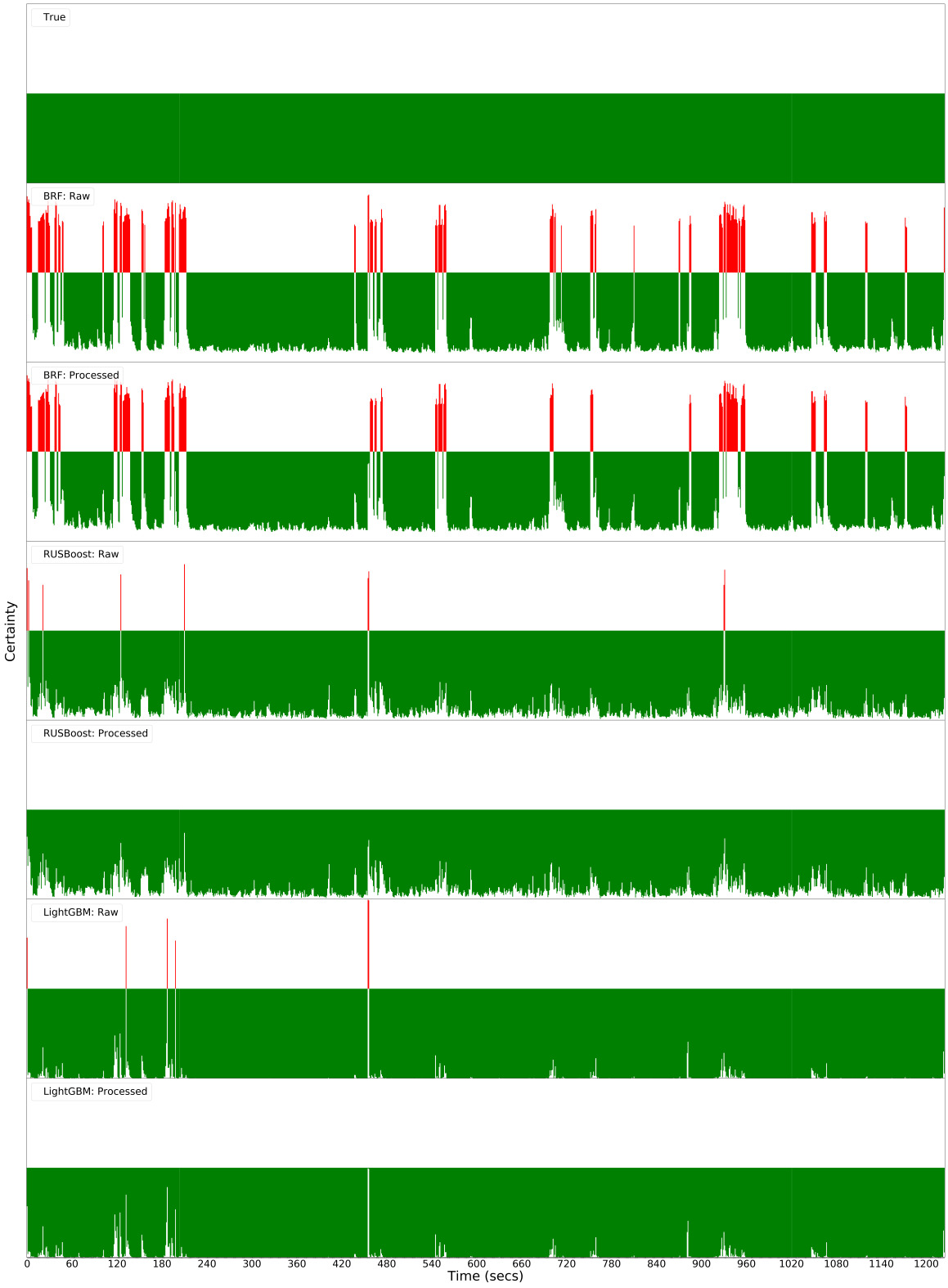
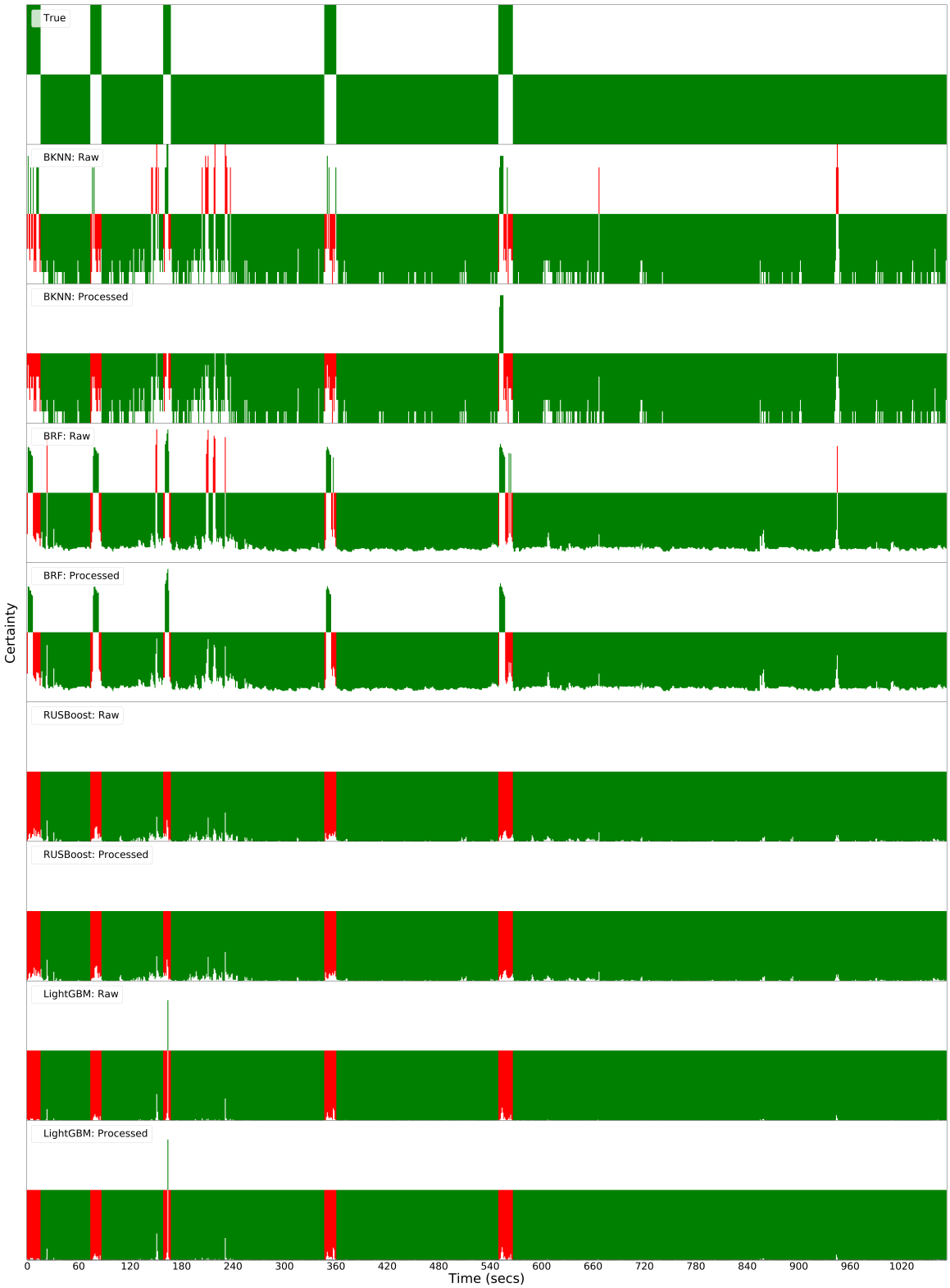
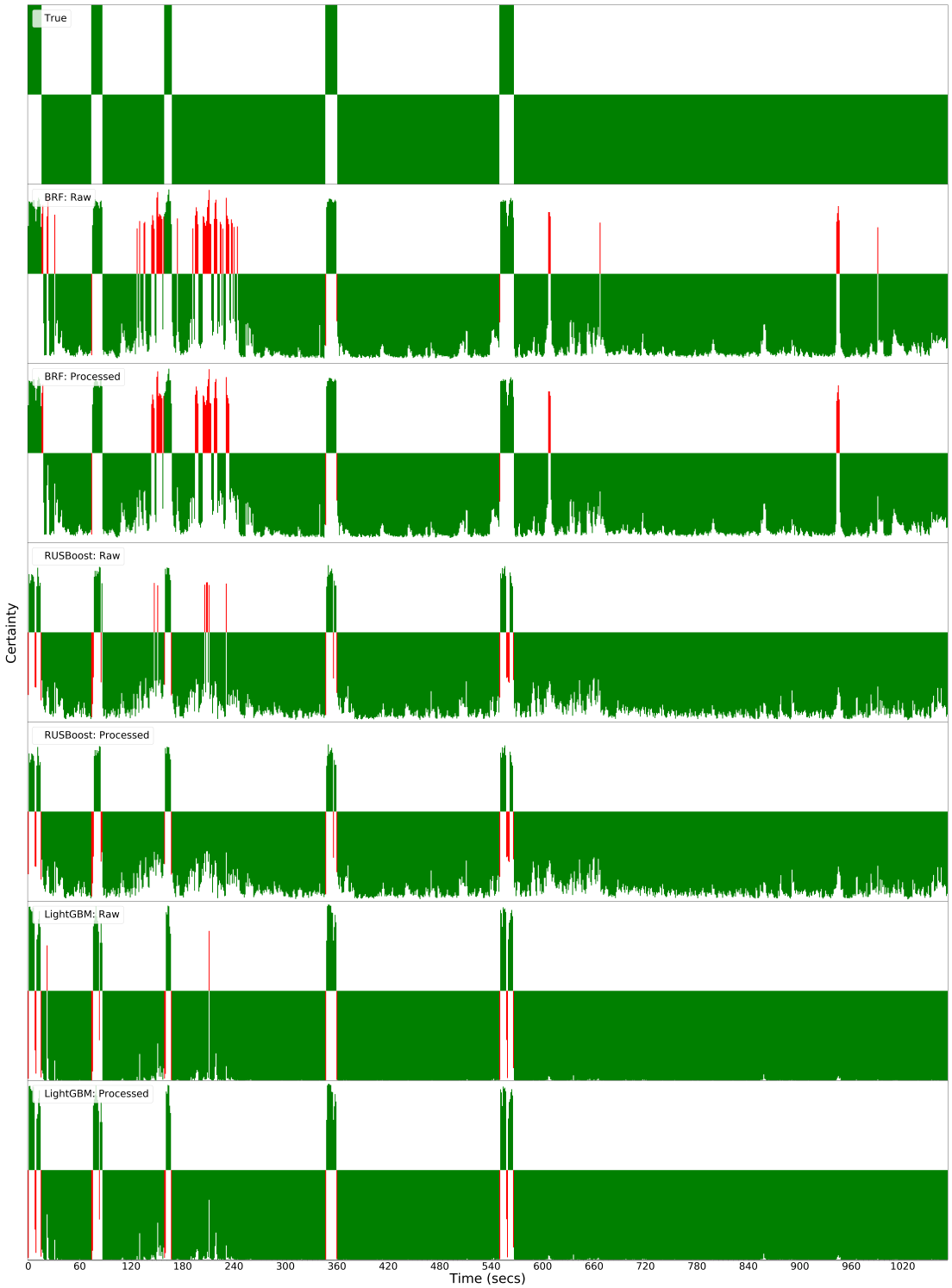


Figure 4.A.2: Example of the prediction certainty of models on a patient record with no seizures present in the whole record (P16).

Note. Models were trained on a combined dataset of NHS (Preston) and NHS (Leeds). The BRF model marks many false positive sections, but after post-processing, boosted models do not mark any.



(a) Models only trained on records in the NHS (Preston) dataset.



(b) Models trained on both records in the NHS (Preston) and NHS (Leeds) datasets.

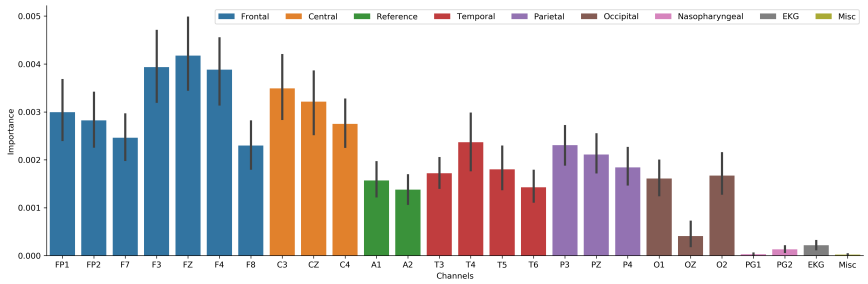
Figure 4.A.3: Prediction certainty of models on a patient record (P17) when records from one dataset or a combination were used for training.

Note. There is poor performance when only records from the Preston site were used for training, with bagged models only partially marking seizures and boosted models marking only a portion of one seizure. When trained on the combined dataset, the BRF model marks most of the length of seizures, but with many more false positives than the boosted models, which do not fully mark all of the seizures.

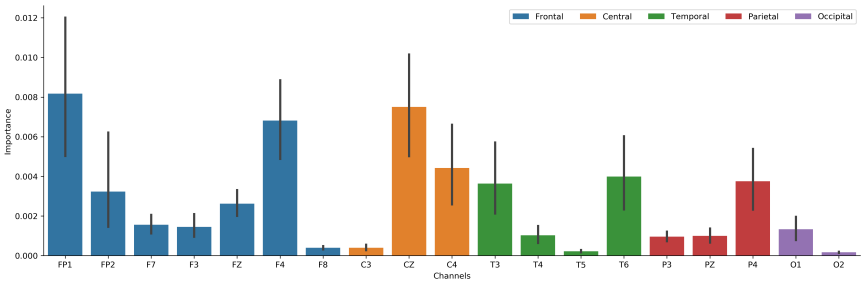
Table 4.A.4: Average (and standard deviation) post-processed test scores across patient held-out datasets.

Dataset	Classifier	Combined	Accuracy		Sensitivity		Specificity		Precision		F1-score		AUC		FP/h		Pred Time	
			Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*
TUH (Absence)	BKNN	False	98.73	(1.14)	82.28	(9.45)	99.57	(0.97)	93.08	(11.2)	**86.48**	(6.28)	94.58	(4.22)	14.62	(32.88)	5.76	(3.27)
	BRF		98.54	(1.13)	**84.4**	(10.19)	99.28	(1.01)	87.95	(13.69)	85.01	(7.39)	95.11	(5.58)	24.58	(34.22)	0.56	(0.34)
	RUSBoost		98.65	(1.14)	78.65	(11.94)	**99.7**	(0.72)	**94.53**	(9.92)	84.86	(7.08)	94.17	(5.56)	**10.09**	(24.53)	1.41	(0.51)
	LightGBM		**98.82**	***(0.91)***	80.31	***(8.9)***	99.69	***(0.68)***	93.95	***(9.43)***	85.92	***(5.4)***	**96.78**	***(3.22)***	10.63	***(22.96)***	**0.11**	***(0.07)***
NHS (Preston)	Bag KNN	False	98.74	(1.61)	76.66	(24.97)	99.48	(0.9)	92.17	(13.14)	79.86	(22.5)	92.5	(8.41)	18.58	(32.58)	2.29	(0.64)
	BRF	False	98.1	(1.91)	72.62	(28.13)	98.73	(2.0)	77.32	(30.3)	72.45	(26.85)	92.94	(6.74)	45.51	(72.08)	0.49	(0.33)
		True	97.11	(3.54)	**86.62**	(16.13)	97.33	(3.64)	71.59	(20.13)	75.2	(13.08)	**97.33**	***(2.24)***	94.94	(131.27)	0.28	(0.18)
	RUSBoost	False	98.87	(1.68)	64.24	(30.52)	99.75	(0.88)	90.79	(27.51)	72.31	(30.59)	95.58	(3.69)	8.79	(31.64)	0.86	(0.2)
		True	99.1	(1.48)	78.81	***(6.71)***	99.61	(1.38)	**97.12**	***(5.93)***	**86.75**	***(4.78)***	95.99	(3.52)	13.9	(49.59)	0.43	(0.13)
	LightGBM	False	98.91	(1.54)	63.87	(31.83)	**99.77**	***(0.71)***	90.61	(27.42)	71.24	(33.3)	95.99	(2.78)	**8.27**	***(25.62)***	0.1	(0.06)
		True	**99.31**	***(1.0)***	74.18	(23.65)	**99.77**	(0.91)	90.62	(27.45)	81.33	(24.95)	96.38	(3.09)	8.36	(32.84)	**0.03**	***(0.01)***
NHS (Leeds)	BRF	False	96.48	(3.27)	81.46	(15.22)	97.29	(3.72)	69.14	(28.77)	70.02	(20.01)	91.76	(9.64)	93.38	(128.28)	0.28	(0.45)
		True	97.42	(2.73)	**84.92**	***(13.39)***	98.06	(2.92)	75.39	(25.55)	77.0	(20.1)	**96.48**	***(2.9)***	66.75	(101.29)	0.44	(1.17)
	RUSBoost	False	98.28	(1.14)	74.56	(14.82)	99.6	(0.75)	90.9	(16.92)	79.84	***(11.71)***	94.87	(3.53)	13.89	(26.24)	0.74	(2.04)
		True	97.95	(1.74)	62.08	(24.7)	99.92	(0.23)	97.42	***(6.46)***	72.33	(21.55)	91.57	(8.08)	2.73	(7.93)	0.64	(1.68)
	LightGBM	False	**98.45**	***(1.08)***	74.14	(17.61)	99.74	(0.64)	93.88	(13.79)	**80.54**	(14.31)	95.63	(5.02)	9.22	(22.41)	**0.04**	***(0.09)***
		True	98.27	(1.45)	66.06	(21.91)	**99.95**	***(0.17)***	**97.62**	(7.92)	76.4	(19.93)	93.73	(6.5)	**1.77**	***(5.74)***	0.06	(0.14)
NHS (Combined)	BRF	True	97.24	(3.18)	**85.71**	***(14.47)***	97.65	(3.33)	73.63	(22.85)	76.16	(16.93)	**96.87**	***(2.6)***	82.75	(118.53)	0.35	(0.77)
	RUSBoost		98.61	(1.67)	69.85	(20.21)	99.75	(1.05)	**97.28**	***(6.11)***	**79.03**	***(17.45)***	93.63	(6.66)	9.07	(37.74)	0.53	(1.09)
	LightGBM		**98.86**	***(1.3)***	69.83	(22.68)	**99.85**	***(0.69)***	94.37	(19.49)	78.69	(22.11)	94.96	(5.29)	**5.51**	***(24.98)***	**0.04**	***(0.09)***

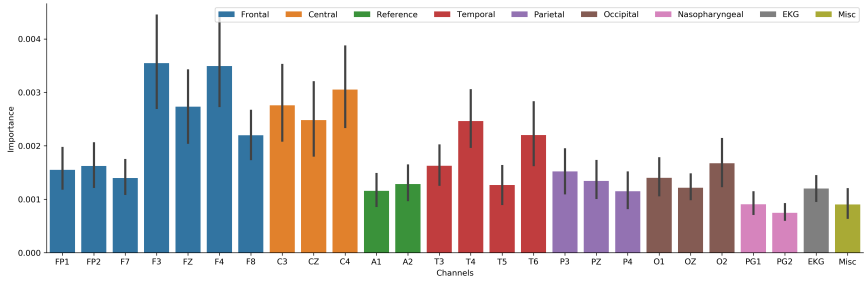
Note. The best average score for each metric, across classifiers, are in bold. Results are shown both for when datasets from different NHS sites were trained separately and when they were combined.



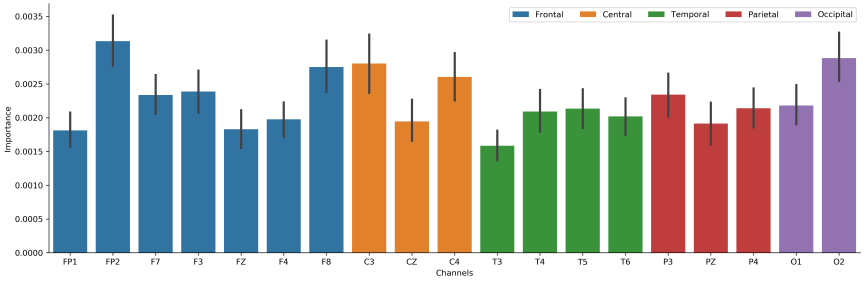
(a) TUH (Absence) - BRF



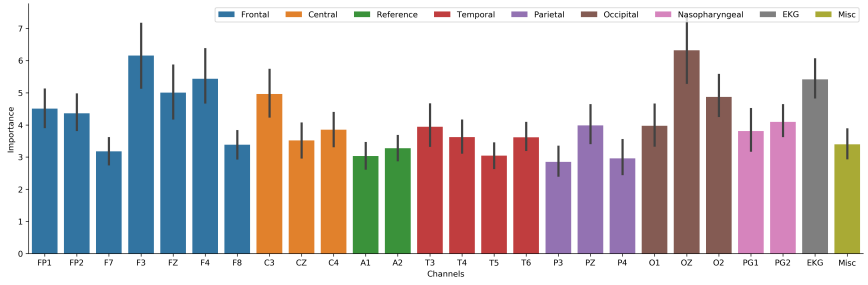
(b) NHS (Preston) - BRF



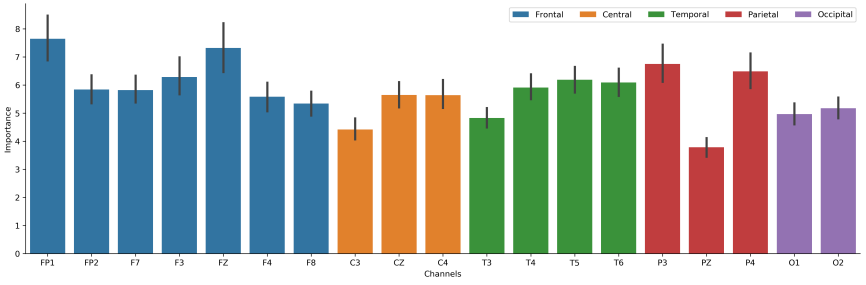
(c) TUH (Absence) - RUSBoost



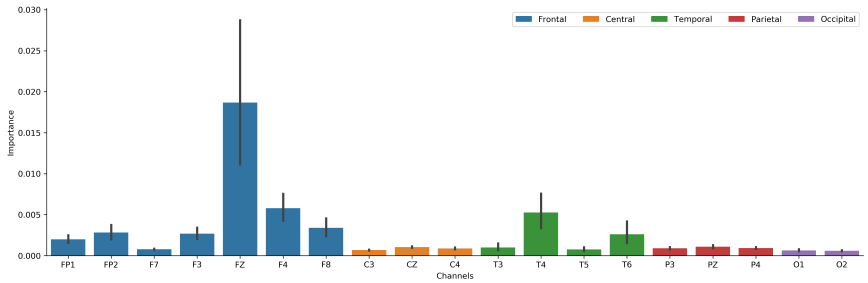
(d) NHS (Preston) - RUSBoost



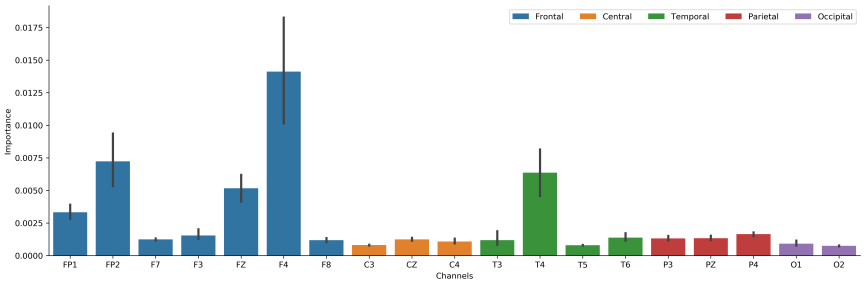
(e) TUH (Absence) - LightGBM



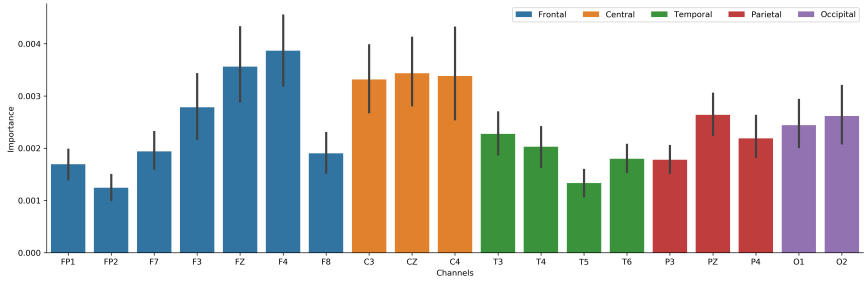
(f) NHS (Preston) - LightGBM



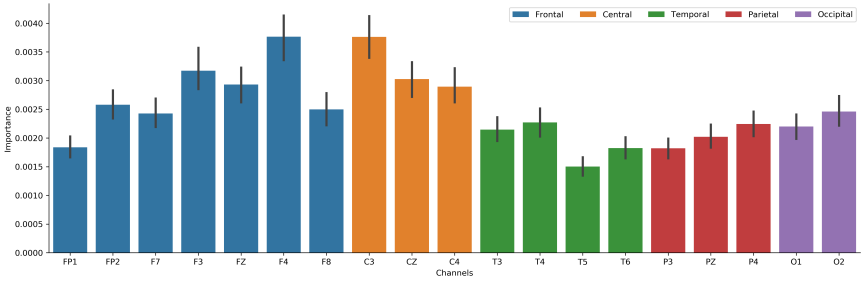
(g) NHS (Leeds) - BRF



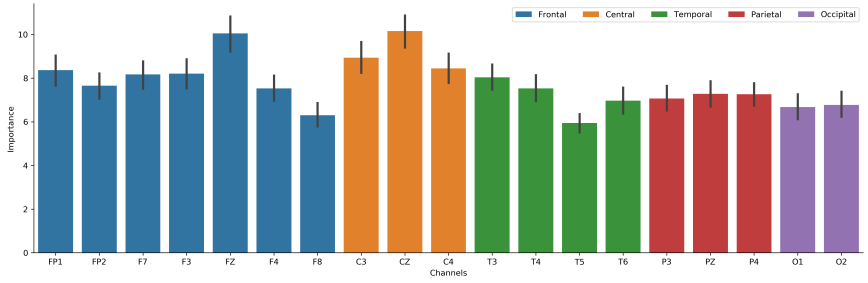
(h) NHS (Combined) - BRF



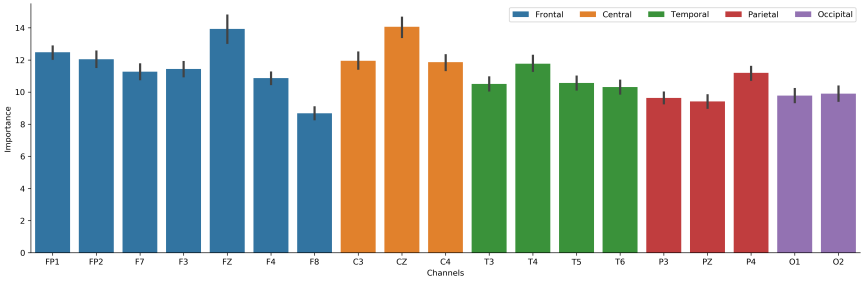
(i) NHS (Leeds) - RUSBoost



(j) NHS (Combined) - RUSBoost

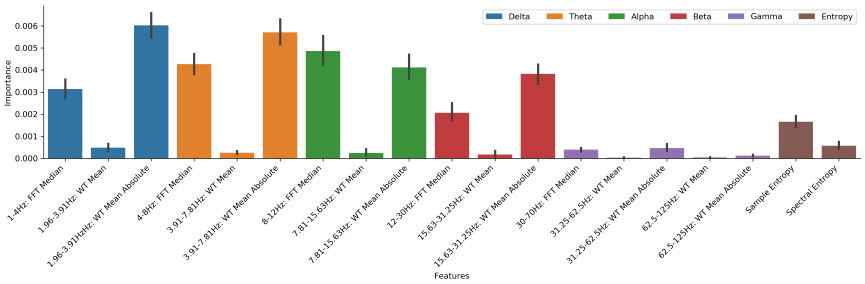


(k) NHS (Leeds) - LightGBM

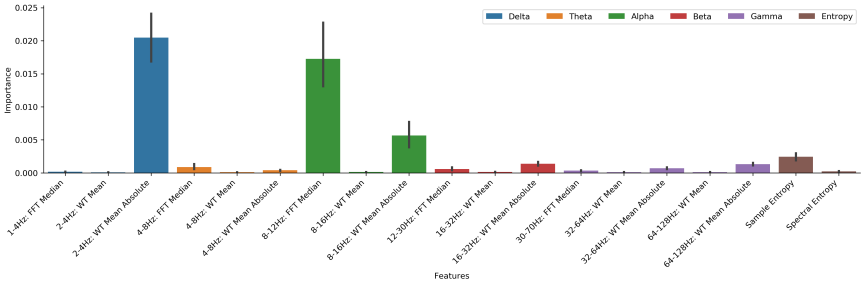


(l) NHS (Combined) - LightGBM

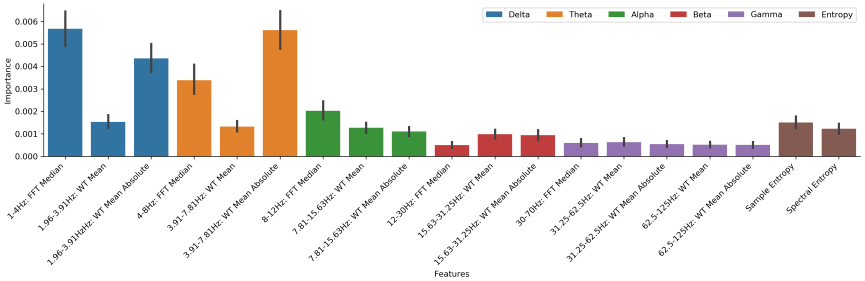
Figure 4.A.4: Bar graphs of average (and standard deviation) feature importances, across patient records and features, according to electrode channel.



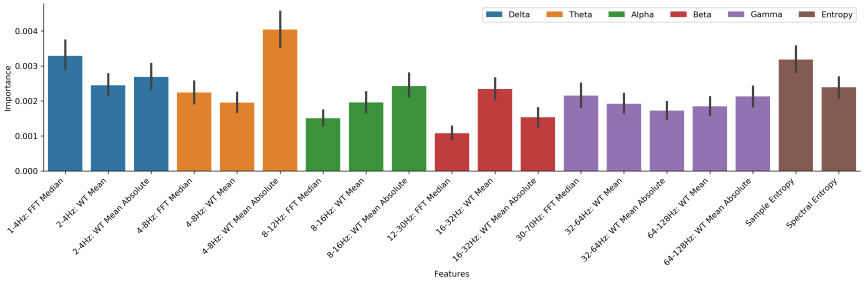
(a) TUH (Absence) - BRF



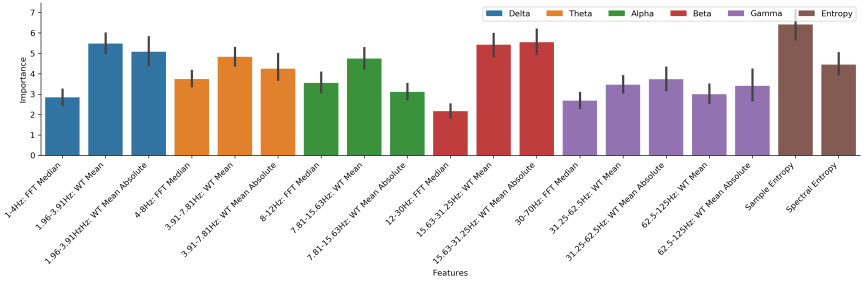
(b) NHS (Preston) - BRF



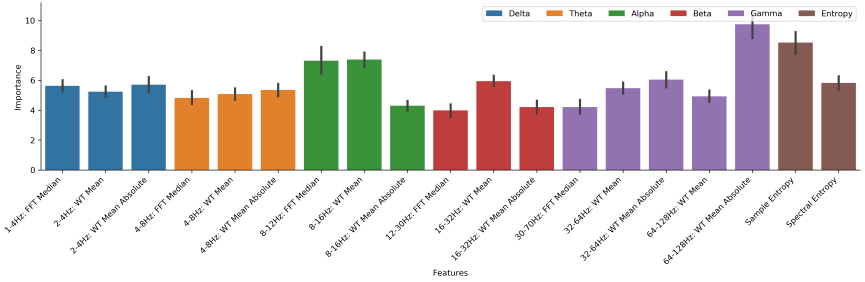
(c) TUH (Absence) - RUSBoost



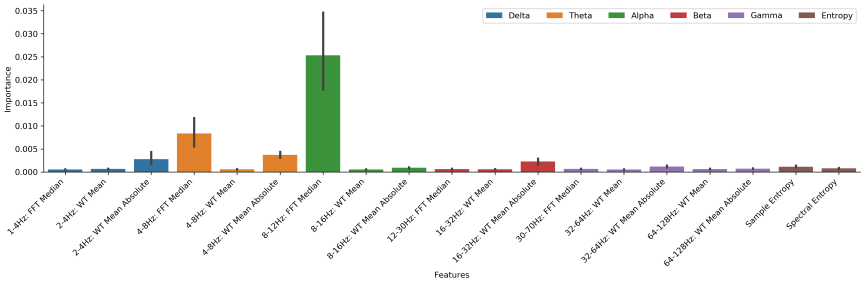
(d) NHS (Preston) - RUSBoost



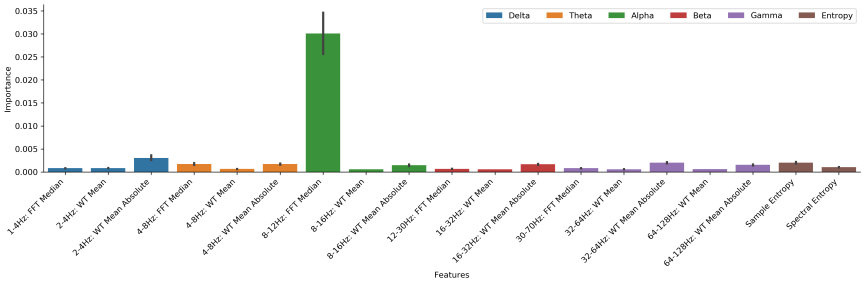
(e) TUH (Absence) - LightGBM



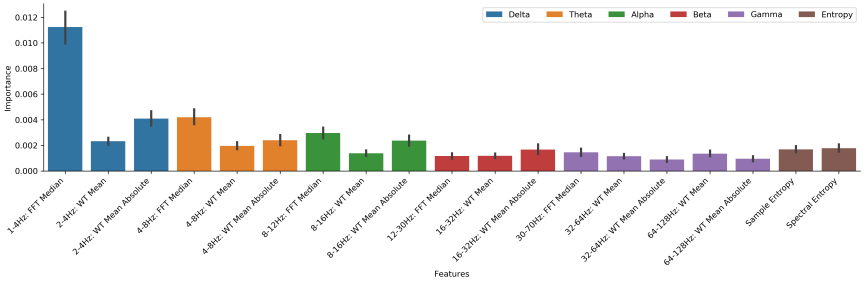
(f) NHS (Preston) - LightGBM



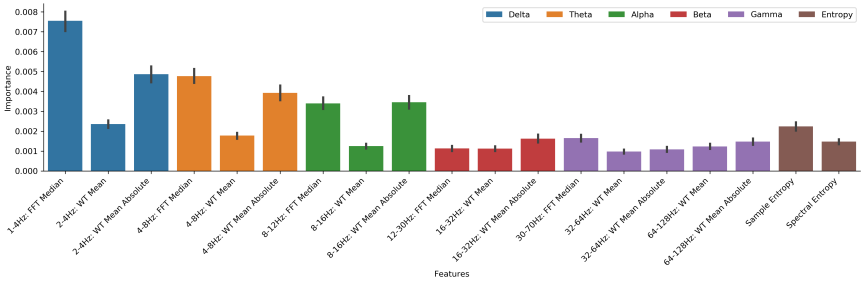
(g) NHS (Leeds) - BRF



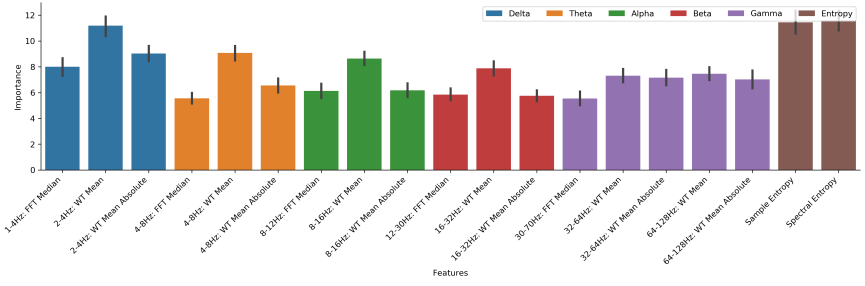
(h) NHS (Combined) - BRF



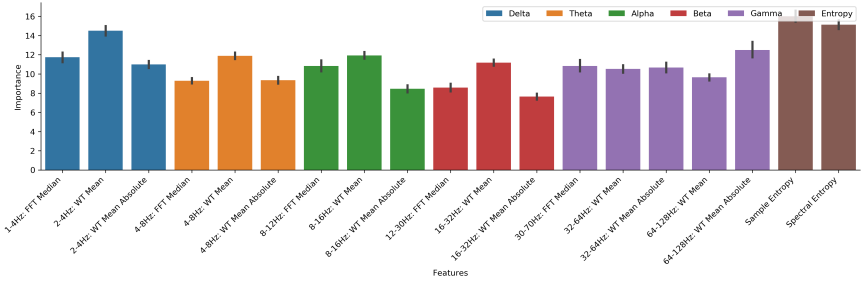
(i) NHS (Leeds) - RUSBoost



(j) NHS (Combined) - RUSBoost



(k) NHS (Leeds) - LightGBM



(l) NHS (Combined) - LightGBM

Figure 4.A.5: Bar graphs of average feature importances according to signal feature.

Chapter 5

Automatic Detection of Generalised and Intractable Epileptiform Discharges in Extra-cranial Electroencephalography using Tree-Based and Deep Neural Network Models

5.1 Introduction

There are many types of seizure, with behavioural effects reflecting the origin of the abnormal neuronal activity and if it propagates to both cerebral hemispheres (see subsubsection 2.1.2; Engel Jr, 2013). Seizures are broadly categorised as having a focal onset, generalized onset, or unknown onset (Fisher, 2017), with this thesis focusing on seizures with generalized onset. Generalized seizures encompass a range of seizures, associated with different symptoms,

ages, and potential co-morbidities. Although epilepsy is the most common cause of seizures, it is important to distinguish epileptic seizures from "nonepileptic" isolated events resulting from stressors such as drug withdrawal, head trauma, or infection. Although standard clinical evaluations such as blood tests, electroencephalography (EEG), computed tomography (CT), and magnetic resonance imaging (MRI) can lead to an accurate epilepsy syndrome diagnosis in most patients, around 20% of patients can remain unclassified (King et al., 1998; Mohanraj et al., 2006). Although these patients often have seizures which are identified in diagnostic imaging, enough information may not have been gained to classify them into a particular epilepsy sub-category.

This chapter aims to compare the generalization and flexibility of machine learning (ML) models by examining their performance to detect two different classifications of generalized seizure, with different intra-patient and inter-patient variabilities. The first seizure type are *generalized absence seizures*, as in chapters 3 and 4, which represent seizures which are distinct from background noise, with little variability, or movement artefacts during the seizure (Baier et al., 2006). We compare these models to those trained to detect *generalized non-specific* (GN) seizures, a label given to seizures which could not be categorised into a specific generalized seizure sub-category in the TUH EEG Seizure Corpus (TUHS; Shah et al., 2018). As such, these seizures cover a broad range of seizure etiologies with significant variation in appearance, length, and focality (see Ochal et al., 2020). The variability of EEG data has been shown to significantly affect the performance of "classical" machine learning classifiers, such as KNN, SVM, and LDA; with neural network/deep learning models potentially better able to account for this variability (Teo et al., 2018; Jana et al., 2019).

Deep learning (DL; Lecun et al., 2015) is often viewed as a simplified framework for end-to-end pre-processing, feature extraction, and classification, providing generalisable and flexible models with competitive performance (Roy et al., 2019b). DL is a class of machine learning algorithms that use multiple layers to create a neural network and typically have state-of-the-art performance on a variety of processing tasks for images, text, and audio signals (Lecun et al., 2015), as well as a number of successful applications to medical diagnostic imaging (e.g. Giger, 2018; Thomsen et al., 2020; Jang and Cho, 2019). DL is thought to have better generalisation than other ML model types (e.g. classical models) as these more

traditional models require the amount of information used for classification to be restricted, as performance deteriorates for higher dimensional inputs (Bengio et al., 2013; Faust et al., 2018). However, there are many reasons why DL may not currently be optimal for EEG processing, distinct from its other successful applications; including comparatively limited available data, a low signal-to-noise ratio, and little empirical work addressing class imbalance with DL models (Johnson and Khoshgoftaar, 2019). Furthermore, DL models generally have a high computational complexity which can lead to large financial, environmental, and productivity costs (Thompson et al., 2020).

This chapter focuses on comparing LightGBM and DL models for the detection of absence and GN seizures. LightGBM was chosen as a baseline comparison model as gradient boosting is another popular ensemble method found to have good performance on absence seizures in chapter 4, and similar to DL models, can be run on graphics processing units (GPUs). Although there has been a number of applications of DL to EEG classification (Roy et al., 2019b), comparisons between gradient boosting and DL model architectures for seizure detection are limited. Where comparisons have been made, they are currently on the University of Bonn dataset, a comparatively small and inter-cranial EEG dataset (see Andrzejak et al., 2001). Nevertheless, Liu et al. (2019) found a gradient boosting classifier gave the best performance for binary seizure classification on this dataset when compared to a number of classical and deep learning models. Sahu et al. (2020) also found gradient boosting models gave better classification accuracy than a number of classical models, but worse accuracy than a Convolutional Neural Network (CNN) classifier; similarly found by Chen et al. (2018) with an ensemble of boosted trees. Therefore, we predicted that a gradient boosted classifier or CNN classifier would provide better performance (e.g. accuracy) than other deep learning model configurations. For the models in these research papers, as well as DL research generally, model design was manually set rather than though the use of hyperparameter optimisation (e.g. Schwabedal et al., 2018; Stober et al., 2014, 2015). Therefore in this chapter, we also investigate the use of Bayesian optimisation to search for optimal model design and hyperparameters, to ensure objective comparisons between models.

This chapter is structured as follows: in section 5.2 we describe how the absence and

GN datasets were prepared for feature extraction. Section 5.3 describes the three different feature sets, used as input into different machine learning classifiers, and the hyperparameter optimisation method used in this chapter. Section 5.4 describes the model training and evaluation of absence seizure feature sets, followed by GN seizure feature sets. Finally, sections 5.5 and 5.6 discuss our findings and present our conclusions.

5.2 Data Preparation

The data we use in this chapter are subsets of the TUHS (1.5.0). As the details of the records from the TUHS containing absence seizures used in this chapter were outlined in section 4.2, hereafter referred to as the TUH (Absence) dataset, we will instead here focus on the records containing GN seizures, the TUH (Generalized) dataset.

102 recording sessions, across 65 patients (mean age = 50, 38 female), were identified with "Generalized Non-Specific Seizures" contained in their records (see table 5.2.2). Sessions would often be split into multiple records, with the TUH (Generalized) dataset consisting of 557 records in total. Sometimes sessions contained multiple seizure types, with all seizure types given the class "ictal", and baseline labels assigned the class "interictal" (see table 5.2.1). The majority of patients were recorded in the Intensive Care Unit (66%), with other recordings occurring in Inpatient (17%), Epilepsy Monitoring Units (11%), Outpatient (4%), and unknown (2%) environments. Most patients were under long-term monitoring (67%), but a number were routine assessments (33%). Patient's medical history was varied, however common medical conditions found across patients were Epilepsy, Hypertension, Stroke, HIV, Anoxic Brain Injury, Drug Abuse, and Diabetes (see table 5.A.3). Most records

Table 5.2.1: Time (in seconds) and proportion of each seizure type in the TUH (Generalized) dataset.

	Time	**(%)**
Background	299116.67	(83.88)
Generalized Non-Specific (GN) Seizure	45218.94	(12.68)
Focal Non-Specific (FN) Seizure	11832.17	(3.32)
Simple Partial (SP) Seizure	204.30	(0.06)
Complex Partial (CP) Seizure	122.52	(0.03)
Tonic-Clonic (TC) Seizure	96.40	(0.03)

Table 5.2.2: Information on patient records used in each dataset for model training.

TUH (Generalized)								TUH (Absence)							
Patient ID	**Group**	**Age**	**Gender**	**Seizure**		**Total Time (Seconds)**		**Patient ID**	**Group**	**Age**	**Gender**	**Seizure**		**Total Time (Seconds)**	
				Types	***Events***	***Ictal***	***Inter-Ictal***					***Types***	***Events***	***Ictal***	***Inter-Ictal***
P1 (00000492)	Training	54	M	GN	14	328.08	1711.92	P1 (00000675)	5	4	F	AB	27	202.70	2279.29
P2 (00000975)	Training	19	F	GN	1	26.44	3868.56			6		AB			
P3 (00002380)	Test	-	M	GN	4	366.26	2409.74	P2 (00001113)	2	20	F	AB	14	83.37	2726.62
P4 (00002521)	Training	27	F	FN	1	59.08	570.92	P3 (00001413)	2	10	F	AB	11	80.16	3760.82
				GN	2	419.00				12					
P5 (00002868)	Training	64	F	FN	3	243.54	1050.21			14					
				GN	2	134.25		P4 (00001795)	3	9	F	AB	2	46.26	1194.74
P6 (00002991)	Training	63	M	FN	21	496.30	3766.63	P5 (00001984)	4	6	M	AB	9	83.90	1375.10
				GN	1	27.07		P6 (00002448)	3	4	M	AB	10	119.96	2101.02
P7 (00003210)	Validation	29	F	GN	1	44.17	1366.83	P7 (00002657)	5	5	M	AB	10	133.98	2540.01
P8 (00004087)	Training	58	F	GN	2	226.23	1264.77	P8 (00003053)	4	5	F	AB	1	16.45	1454.55
P9 (00004456)	Validation	43	F	FN	11	45.55	4232.21	P9 (00003281)	1	13	M	AB	2	19.81	1293.18
		47		GN	11	138.24		P10 (00003306)	1	13	F	AB	4	31.51	1394.48
P10 (00004671)	Validation	22	M	GN	39	250.4	1679.6	P11 (00003635)	5	6	M	AB	7	19.19	1598.80
P11 (00005101)	Training	82	F	GN	1	53	2								
P12 (00005265)	Training	22	M	GN	1	569	2235.00								
P13 (00006107)	Training	89	M	GN	12	341.47	3192.53								
P14 (00006230)	Validation	29	M	GN	4	225.27	3018.73								
		32													
P15 (00006440)	Training	47	M	FN	16	1163.09	3642.88								
				GN	9	945.03									
P16 (00006520)	Training	20	F	GN	2	149.49	2752.12								
				TC	1	96.40									
P17 (00006546)	Test	38	M	CP	1	52.00	18982.43								
		40		FN	40	1787.57									
		41		GN	22	1144.69									
		42		SP	3	204.30									
P18 (00006563)	Training	55	F	FN	8	350.54	1031.96								
				GN	12	715.50									
P19 (00007032)	Validation	68	F	FN	10	958.09	2986.43								
				GN	5	2786.48									
P20 (00007170)	Test	80	F	GN	3	220.37	3905.63								
P21 (00007828)	Training	60	M	GN	1	26.18	1866.82								
P22 (00007936)	Training	55	F	GN	2	181.06	393.93								
P23 (00007937)	Test	38	M	GN	1	18.02	2392.98								
P24 (00008174)	Training	91	M	GN	6	495.5	784.5								
P25 (00008204)	Training	60	F	GN	3	53.03	1347.97								
P26 (00008295)	Test	22	F	CP	1	70.52	1539.80								
				FN	8	1757.34									
				GN	2	209.35									
P27 (00008303)	Training	55	F	GN	19	636.12	3656.88								
P28 (00008453)	Training	47	M	GN	2	100.88	5190.12								
		48													
		49													
P29 (00008479)	Training	59	M	GN	30	471.78	9699.22								
P30 (00008480)	Training	43	M	GN	1	350	5498.00								
P31 (00008512)	Test	60	M	FN	2	134.10	5545.12								
		61		GN	22	1271.79									
P32 (00008760)	Test	42	F	GN	1	44.1	1281.90								
P33 (00009104)	Test	60	M	FN	17	1144.49	48351.96								
		62		GN	9	845.55									
P34 (00009158)	Training	82	F	GN	2	72.24	1130.76								
P35 (00009162)	Validation	58	F	GN	2	49.71	4930.29								
P36 (00009231)	Training	-	F	GN	3	2258.00	2669.00								
P37 (00009232)	Training	49	M	GN	1	119.02	4356.98								
P38 (00009370)	Test	61	F	GN	2	497.1	3106.90								
P39 (00009540)	Validation	50	F	GN	1	1121.00	301.00								
P40 (00009623)	Validation	61	M	GN	3	229.58	5745.42								
P41 (00009839)	Training	64	M	FN	2	131.14	7240.91								
		65		GN	4	240.95									
P42 (00009852)	Training	39	M	GN	1	293.55	3251.45								
P43 (00009932)	Training	53	F	GN	1	99.32	4405.68								
P44 (00009934)	Training	24	M	FN	9	133.36	2936.80								
				GN	1	11.84									
P45 (00009994)	Training	29	M	GN	5	105.97	3967.02								
P46 (00010020)	Training	70	F	GN	71	1120.47	1047.52								
P47 (00010062)	Validation	39	F	FN	4	1394.91	10028.78								
				GN	28	9002.30									
P48 (00010106)	Training	53	F	GN	1	113.6	1245.40								
P49 (00010158)	Training	59	M	GN	5	241.75	3523.25								
P50 (00010418)	Test	66	F	FN	25	425.41	5932.57								
				GN	7	193.02									
P51 (00010421)	Test	42	F	GN	1	80.18	1798.82								
P52 (00010455)	Training	55	F	GN	2	1215.9	2297.10								
P53 (00010639)	Training	61	F	GN	18	308.35	1660.65								
		62													
P54 (00010760)	Test	61	F	GN	4	174.38	3763.62								
P55 (00010843)	Training	64	F	FN	3	184.55	2226.51								
		65		GN	5	434.94									
P56 (00010861)	Validation	57	F	GN	3	155.92	1047.08								
P57 (00011272)	Training	58	M	GN	1	51.1	4099.90								
P58 (00011580)	Training	-	F	FN	10	868.02	20025.36								
				GN	21	894.62									
P59 (00011870)	Training	88	F	FN	1	10.84	5367.11								
				GN	11	2717.05									
P60 (00011972)	Training	66	F	FN	3	318.37	7438.21								
				GN	9	894.42									
P61 (00011999)	Validation	40	M	FN	1	179.68	13533.64								
				GN	2	146.68									
P62 (00012046)	Validation	21	F	GN	13	158.13	2305.87								
P63 (00012707)	Training	28	F	FN	2	46.20	7466.40								
				GN	3	118.39									
P64 (00012940)	Test	20	M	GN	11	6657.00	6360.00								
P65 (00012941)	Validation	-	F	GN	9	1628.7	2686.30								
Total	**-**	**-**	**-**	**-**	**701**	**57474.37**	**299116.6**	**-**	**-**	**-**	**-**	**-**	**97**	**837.31**	**21718.61**

were recorded at sampling rates of either 256Hz (75.58%) or 250Hz (9.52%), with some records being recorded at 1000Hz (0.9%), 512Hz (1.44%), and 400Hz (12.57%), which were downsampled to ∼200Hz. An additional 22 sessions, from 18 patients, were also identified, but were excluded as they already used a linked mastoid re-reference as opposed to an average re-reference.

5.3 Methods

In this section, subsection 5.3.1 starts by describing the different feature sets extracted for the different classifiers. Then, in subsection 5.3.2, we give an overview of the ML classifiers used to separate features into ictal and interictal classes. Subsection 5.3.3 then briefly describes how Bayesian optimization was used to search over model hyperparameters, focusing on differences from chapters 3 and 4. For a description of how performance metrics were calculated, see subsection 3.3.4.

5.3.1 Feature Extraction

Three separate feature sets were created separately for both the TUH (Absence) and TUH (Generalized) datasets to reflect the typical input into different classifiers (see figure 5.3.1). Separately for the TUH (Absence) and TUH (Generalized) datasets, EEG channels that were common to all records were kept and all other channels were removed (see table 5.A.1). All feature sets were windowed into epochs of 512 samples in length (2 seconds at 256Hz) with a 256 sample overlap (1 second at 256Hz). The first feature set, used as an input into LightGBM and Multilayer Perception (MLP) models (see subsection 5.3.2), used the same features as detailed in subsection 4.3.1. The other two feature sets required less feature engineering, as commonly found in applications of DL. A filtered feature set, used for Recurrent Neural Network (RNN) models and 1-dimensional Convolutional Neural Network (CNN1D) models, was created using a digital butterworth filter of order 4 with a passband between 1 and 30Hz, applied both forwards and backwards to each electrode channel. The third feature set, used for 2-dimensional CNNs (CNN2D), represented EEG signals in windowed epochs as a spectrogram (see figure 5.3.2), created using an undecimated wavelet transform

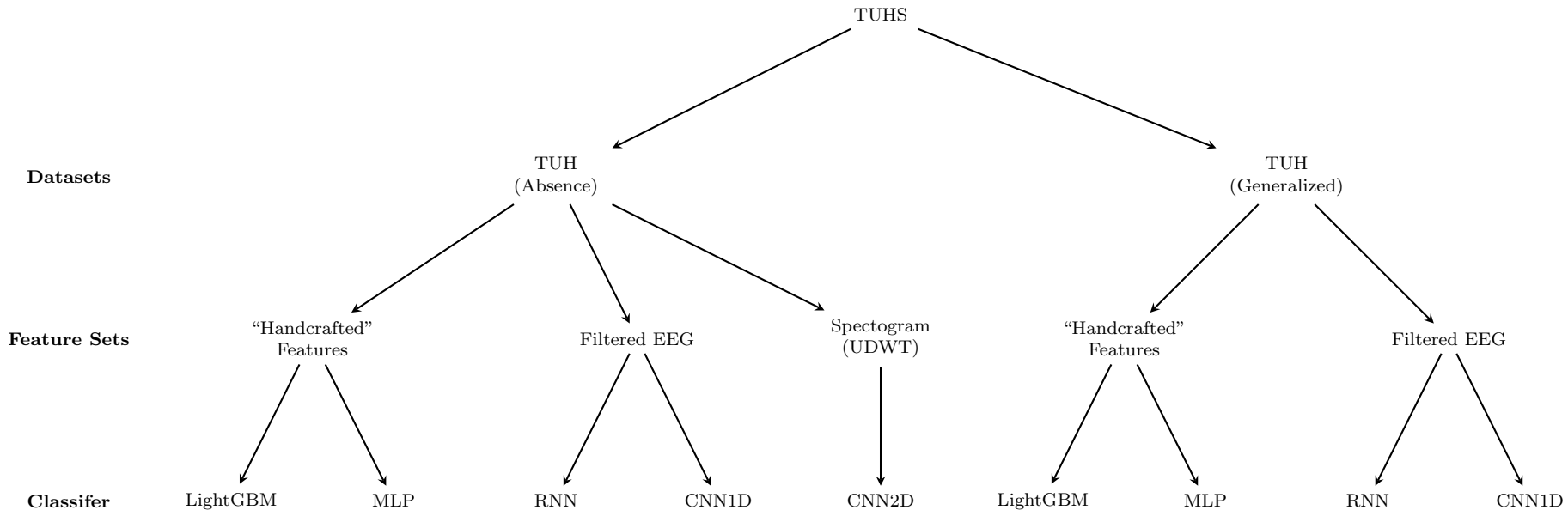


Figure 5.3.1: Classifiers used in this chapter and their associated features and datasets.

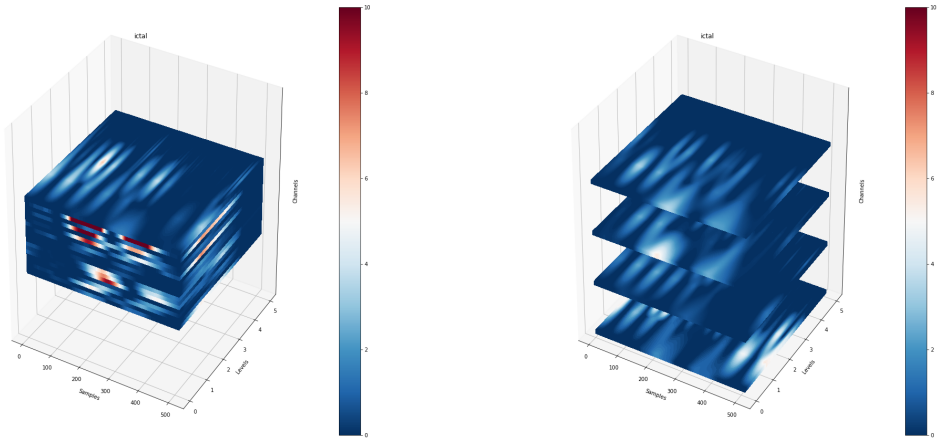


Figure 5.3.2: Example epoch of data used as an input to 2D CNN models.
Note. Illustrates a generalized non-specific seizure

(UDWT; discussed in subsections 2.4.3 and 5.3.2).

All data was scaled on a patient-by-patient basis by removing the mean and scaling to a unit variance of 1, identical to the pre-processing in subsection 4.3.1. For the filtered datasets, and where features were manually extracted, scaling was performed separately for each channel and each feature; except for features in the frequency domain, where scaling was done for each type of feature across all frequency bands for each channel (e.g. C3 Mean 2-4Hz, 4-8Hz, 8-16Hz, 16-32Hz, 32-64Hz, 64-128Hz). Scaling across frequency bands for each channel was also conducted for the UDWT feature set.

5.3.2 Signal Classification

The gradient boosting (LightGBM) and DL models chosen used GPU hardware for training and prediction to account for the size of the datasets and computational cost of DL models. Specifically, each model was trained using a NVIDIA® Tesla T4 GPU, attached to a virtual machine on the Google® Cloud Platform with various vCPUs and RAM sizes according to the dataset size. LightGBM (2.3.2; Ke et al., 2017), previously discussed in chapters 2 and 4, is a gradient boosting model which sequentially fits predictors to the residual errors of previous tree-like models. DL is another popular ensemble method, also discussed in more

detail in chapter 2, where layers of artificial neurons or other formulas are stacked on top of each other. These models reduce the amount of required *hard-coded* feature engineering as, although the input data is rarely completely "raw", generally there is less data transformation required outside the neural network. This is due to the model layers generally generating representations of the input which increase or decrease in complexity throughout the network layers. Neural networks can consist of various layer types, with common types including; Fully Connected (Dense) Layers, Convolutional Layers, and Recurrent Layers associated with global, local, or sequence pattern detection respectively. Different combinations and configurations of various layers are associated with different networks, with MLP, CNN, and RNN models currently the most common.

DL models were all created using `Tensorflow` (2.1.0/2.2.0; Abadi et al., 2016), with Intel® MKL-DNN/MKL and CUDA 10.1 (NVIDIA et al., 2019). All models were trained using a batch size of 32, a common size as larger batch sizes may lead to worse generalisation (Keskar et al., 2019). An Adam (Kingma and Ba, 2015) optimiser was used, with the learning rate reduced from the value chosen during optimisation (see subsection 5.3.3) to a minimum of 1e-4 when the validation loss had not improved in more than 2 epochs. All DL models ended with between 0 and 2 hidden dense layers, with Exponential Linear Unit (ELU) activation functions, before a final output layer with 1 output and a sigmoid activation function. MLP models consisted only of these hidden dense layers throughout, with batch normalisation (Ioffe and Szegedy, 2015) used before the activation function to standardise the outputs of each layer (Burkov, 2019). A 50% dropout rate (Hinton et al., 2012; Srivastava et al., 2014) was also used as a regularisation technique to prevent overfitting. A variant of dropout, Monte Carlo Dropout (MCD; Gal and Ghahramani, 2016), was used which keeps dropout on while making predictions so that models output predictions which better represent model uncertainty. MCD is useful in risk-sensitive medical systems as predictions, for both the positive and negative classes, should be made with more caution (Géron, 2019). For the recurrent networks, we used gated RNN (GRU) layers, to give the network "memory". RNNs are sometimes proceeded by CNN layers, as a 1D convolutional layer with strides above 1 downsamples the data. There are many different approaches to developing a CNN model, with this chapter using four different architectures which could be chosen

during optimisation, based on VGGNet, ResNet, Xception, and WaveNet (only available to 1-dimensional CNN models). Inputs into a 1-dimensional CNN model are 2-dimensional arrays (e.g. $time \times EEGchannel$), with 3-dimensional arrays (e.g. $time \times EEGchannel \times frequency$) input into 2-dimensional CNN models. The dimension of a CNN model describes how its kernel slides across the data, 1D CNN models therefore sliding across the one axis (e.g. $time$), and 2D CNN models sliding across two axes (e.g. $time$ & $EEGchannel$).

VGGNet (Simonyan and Zisserman, 2014) follows a typical CNN architecture, in that it stacks multiple convolutional and Rectified Linear Unit (RELU) activation layers together with occasional pooling layers. The number of filters applied to the data in each layer tends to increase as the other input dimensions get smaller. For 1D CNNs, the term *filter* and *kernel* can be used interchangeably to refer to a 2D array of weights. However, in 2D CNNs, a *filter* refers to a collection of unique kernels which output one "channel". Channels (or feature maps) is a term often used to describe the structure of a layer. An example could be an input layer of $time \times EEGchannel \times frequency$ which would have 3 channels if the inputs have 3 frequency band components; although channels do not always need to be last in the input. In 2D CNNs, 2D kernels are first convoluted separately to each input channel of the previous layer. The result of each separate convolution is then summed together to form a single channel, to which a bias gets added. In order for the next layer to have an input with multiple channels, multiple filters are applied to the input layer. VGGNet specifically uses a 3x3 kernel and double filters for each of the 5 stacks of convolutional layers from 64 to 512. In this work, the number of filters in the first layer, as well as the kernel size, could be optimised (see subsection 5.3.3). Furthermore, we added batch normalisation layers between convolution and activation layers, as adding these to the original VGGNet model design has been shown to improve performance (Santurkar et al., 2018).

ResNet (He et al., 2016) is variant of VGGNet, developed for the ILSVRC ImageNet challenge (Russakovsky et al., 2015). ResNet uses *skip connections* to add the input signal of a group of convolutional layers to the output. This leads to residual learning, as the layer now models the target function minus the input rather than just the target function.

Residual units are made up of two main parts. Firstly there are two convolutional layers that feed sequentially into each other, separated by batch normalisation and activation layers. The first convolutional layer has a stride of 1 or 2 depending on if there is an increase in the number of filters from the previous layer, so that the output of the layer is reduced similar to using pooling, and the second convolutional layer always has a stride of 1. The second part is a third separate convolutional and batch normalisation layer which takes the input for the first convolutional layer in the unit, runs it though a 1x1 kernel, and also applies a stride of 2 or 1 so that the output is the same as the other separate sequential convolutional layers. The unit then ends with an activation on the added output of the two paths in the unit.

Xception (Chollet, 2017b) merges ideas from GoogLeNet's (Szegedy et al., 2015) inception modules and ResNet. Xception (extreme inception) uses *depthwise separable* convolution layers to separate spatial and cross-channel pattern modelling. In this architecture, filters are similar to the convolution layers used above, as kernels are first separately convoluted for each input channel for 2D CNNs. However, instead of summing the resulting maps/channels to output one channel per filter, a *pointwise convolution* (1x1 convolution) is used. Essentially, convolution layers are split into a layer to filter the data and a layer to combine them. Depthwise separable convolution layers are less useful when the number of channels is low (Géron, 2019), therefore in Xception they are used after 2 regular convolutional layers to increase the channel number. The rest of the model then uses combinations of separable convolutional layers with skip connections, pooling, and final dense layers.

WaveNet (van den Oord et al., 2016), the final CNN model architecture used in this work, is only used with 2D data. WaveNet uses stacked 1D convolutional layers, doubling the *dilation rate* at each layer rather than increasing the number of filters. The dilation rate is the distance a neuron is in a layer, from the neuron feeding it an input from the previous layer. This means in the input layer there is a neuron for each sample in the data, in this case a sequence of 512 samples in a window, with this decreasing like a pooling or strided convolution at each layer. Therefore the initial layers can learn long-term patterns

and smaller patterns at lower levels. By zero padding the input sequences to each layer by the dilation rate, the sequence length of the input is preserved in the output of the model. Multiple dilated layers form a block, similar to a single convolutional layer with a kernel size of 1,024, which can be repeated and stacked on-top of each other to make a deeper model (Géron, 2019). Similar to ResNet and Xception, WaveNet also includes skip connections, however also includes Gated Activation Units (GAUs), which are similar to GRU cells found in RNNs.

5.3.3 Optimisation and Cross-Validation

BOHB, using `Hpbandster` (0.7.4 Falkner et al., 2018), was used to optimise model structure and hyperparameters separately for LightGBM, MLP, RNN, CNN1D, and CNN2D network models. BOHB is a combination of Bayesian and Hyperband optimisation methods, found to be faster, with better solutions, than Hyperband or random search alone (Falkner et al., 2018). The Bayesian part of the algorithm is based on Tree Parzen Estimators (Bergstra et al., 2011), which use a kernel density estimator to model the input configuration space to find optimal hyperparameters. Hyperband (Li et al., 2018) is a bandit strategy that randomly samples configurations of hyperparameters and removes poorly performing configurations using successive halving whilst allocating resources effectively.

For feature sets from the TUH (Absence) dataset, the BOHB algorithm had 200 iterations to determine the best parameters for each model based on the validation F1-score. This was lowered to 100 iterations for the models trained on the TUH (Generalized) feature sets to account for the increased computational cost associated with the larger amount of data. Bayesian optimisation does typically require fewer iterations than random search or gridsearch to get the optimal set of hyperparameter values, due to focusing on areas of the search space expected to generate a higher validation score (Falkner et al., 2018). However, the number of iterations chosen here is based on the available computational budget. Other published research using Bayesian optimisation for deep learning models (e.g. Wang et al., 2018; Nguyen et al., 2019; Liang, 2019) tend to use fewer iterations (18-90), however they tend to optimise fewer hyperparameters (3-6). The budget for the BOHB was set between 50 and 1000 estimators for each trained LightGBM model and between 5 and 30 epochs for

Table 5.3.1: Hyperparameter search spaces for different classifiers.

LGBM		**MLP**	
Hyperparameter	***Value***	***Hyperparameter***	***Value***
Min Split Gain	0	Batch Size	32
Subsample Frequency	1	Optimizer	Adam
Subsample For Bin	200000	ReduceLROnPlateau (Factor)	0.1
Objective	binary	ReduceLROnPlateau (Patience)	2
Min Split Gain	0	Dropout (Type)	MCDropout
Importance Type	split	Dropout (Rate)	0.5
Hyperparameter	***Search Space***	***Hyperparameter***	***Search Space***
Learning Rate	loguniform(0.01, 0.2)	Learning Rate	...
Negative Class Fraction	uniform(0.01, 0.5)	Negative Class Fraction	...
Scale Pos Weight	uniform(0., 10.)	Sample Weight	choice(True, False)
Boosting Type	choice(gbdt, goss, dart)	Kernel Regularization	choice(None, 'L2', 'Max Norm')
Num Leaves	randint(2, 40)	Number of Dense Layers	randint(1, 12)
Max Depth	choice(None, randint(1, 25))	First Layer Dense (Units)	randint(3, 300)
Min Child Weight	uniform(0.001, 5.)	Hidden Layer Dense (Units)	randint(2, 300)
Min Child Samples	randint(1, 30)		
Colsample By Tree	uniform(0.1, 1.)		
Reg Alpha	uniform(0., 1.)		
Reg Lambda	uniform(0., 1.)		
Hyperparameter	***Budget***	***Hyperparameter***	***Budget***
Number of Estimators	Min = 50; Max = 1000	Epochs	Min = 5; Max = 30
RNN		**1D CNN**	
Hyperparameter	***Value***	***Hyperparameter***	***Value***
Batch Size	...	Input Configuration	$Samples \times Channel$
Optimizer	...	Batch Size	...
ReduceLROnPlateau (Factor)	...	Optimizer	...
ReduceLROnPlateau (Patience)	...	ReduceLROnPlateau (Factor)	...
Dropout (Type)	...	ReduceLROnPlateau (Patience)	...
Dropout (Rate)	...	Dropout (Type)	...
Hyperparameter	***Search Space***	***Dropout (Rate)***	***...***
Learning Rate	...	***Hyperparameter***	***Search Space***
Negative Class Fraction	...	Learning Rate	...
Sample Weight	...	Negative Class Fraction	...
Kernel Regularization	...	Sample Weight	...
Number of Dense Layers	randint(0, 2)	Kernel Regularization	...
Conv (Filters)	randint(2, 128)	Number of Dense Layers	...
Conv (Kernel)	randint(2,6)	Conv (Filters)	...
Number of Conv Layers	choice(0, 1)	Conv (Kernel)	...
Number of Rec Layers	randint(1, 9)	Conv Layer Types	choice('VGGNet', 'ResNet', 'Xception', 'WaveNet')
Rec (Units)	randint(1, 128)	VGGNet (Number of Conv Blocks)	randint(1,4)
Dense (Units)	randint(3, 100)	VGGNet Dense (Units)	randint(3, 600)
		ResNet (Number of Conv Blocks)	randint(1,10)
		ResNet Dense (Units)	randint(3, 600)
		Xception (Number of Conv Blocks)	randint(1, 10)
		Xception Dense (Units)	randint(3, 1500)
		Wavenet (Number of Conv Blocks)	randint(1, 5)
		Wavenet Dense (Units)	***randint(3, 300)***
Hyperparameter	***Budget***	***Hyperparameter***	***Budget***
Epochs	...	Epochs	...
2D CNN			
Hyperparameter	***Value***		
Batch Size	...		
Optimizer	...		
ReduceLROnPlateau (Factor)	...		
ReduceLROnPlateau (Patience)	...		
Dropout (Type)	...		
Dropout (Rate)	...		
Hyperparameter	***Search Space***		
Learning Rate	...		
Negative Class Fraction	...		
Sample Weight	...		
Kernel Regularization	...		
Number of Dense Layers	...		
Conv (Filters)	...		
Conv (Kernel)	randint(4,12)		
Conv Layer Types	choice('VGGNet', 'ResNet', 'Xception')		
Input Configuration	choice($Freq \times Samples \times Channel$, $Channel \times Samples \times Freq$)		
VGGNet (Number of Conv Blocks)	...		
VGGNet Dense (Units)	...		
ResNet (Number of Conv Blocks)	...		
ResNet Dense (Units)	...		
Xception (Number of Conv Blocks)	...		
Xception Dense (Units)	...		
Hyperparameter	***Budget***		
Epochs	...		

Note. Some hyperparameters are shared between classifiers (...). choice: choose one; randint: random integer; normal: normal distribution; uniform: value selected randomly between lower and upper bounds; loguniform: a log-uniform distribution.

each trained DL model. Training epochs for the DL models in this chapter do not reflect a full training run on all the available training data, as a batch generator from `Imblearn` (0.7.0; Lemaitre et al., 2017) was used to randomly undersample the training data to create mini-batches of ictal and interictal data. All available ictal data was used for training, whereas the proportion of available interictal data for these batches could be optimised. The search space, created using `ConfigSpace` (0.4.13; Lindauer et al., 2019), consists of both hyperparameters and model layer configurations for DL models (see table 5.3.1). The model hyperparameters for LightGBM were determined based on previous results (see chapter 4). The defined search space for initial learning rate and whether to add a weighting to classes according to their proportion in the data were the same between all deep learning models. The heuristic used to calculate the class weights, $NumSamples/(NumClasses * bincount(y))$, is the same as implemented in `Scikit-Learn` (0.23.1; Pedregosa et al., 2011; King and Zeng, 2001).

For RNN and CNN models, the number of dense layers at the end of the model and number of filters for convolutional layers were the same. The number of dense, convolutional, and recurrent layers/blocks in most models were set to a maximum of 12, with the exceptions being Wavenet architectures (7) to ensure the data fitted into GPU memory, and VGGNet (4) as maxpooling could not halve the electrode channels beyond this if data was input as $(Freq \times Samples \times EEGChannel)$. Indeed, the UDWT feature set could be input into the CNN2D models as either $Freq \times Samples \times EEGChannel$ or $EEGChannel \times Samples \times Freq$. However, as illustrated in table 5.3.2, the layers as described above are not all of the layers in a model (e.g. input, reshape, batch normalisation, activation, dropout, global average pooling, and flatten layers); with convolutional blocks typically consisting of multiple layers. This means that models can be much deeper, or have more parameters to train, compared to other model types or chosen architectures. CNN models can, at the start of each iteration of model optimisation, choose one of three types of hidden blocks based on VGGNet, ResNet, and Xception, as well as Wavenet for 1D CNN models. Filters across the different layers/blocks in the CNN models increase according to the type of layers being used (see table 5.A.2); with 2D CNN models also halving the kernel value after the first block. The hidden dense layers at the end of RNN and CNN models similarly halve the

Table 5.3.2: Number of model layers and internal parameters if the maximum of all possible parameters/ hyperparameters were chosen during optimisation

Model Type		Max Layers	Max Params
MLP		49	1,160,101
RNN		30	927,273
CNN1D	***VGGNet***	46	7,898,977
	ResNet	25	30,167,521
	Xception	109	278,815,591
	Wavenet	176	8,835,625
CNN2D	***VGGNet***	47	177,567,073
	ResNet	26	344,932,321
	Xception	110	293,026,273

number of units at each subsequent layer before the final 1 unit output layer. Each type of DL model has a different maximum for the number of units for the hidden dense layers, to reflect that the proceeding number of outputs into those layers tend to be larger or smaller depending on the type of previous layers; for example, convolutional layers tend to have many outputs compared to recurrent layers.

5.4 Results

In this section we begin by describing the model training and evaluation on the feature sets generated from the TUH (Absence) dataset (subsection 5.4.1), followed by feature sets from the TUH (Generalized) dataset (subsection 5.4.2). Each subsection for the different seizure types begins by examining the validation scores gained during training, along with the time to train each model. We then look at the search space and optimal parameters for the different models found using BOHB optimisation. Subsequently, in each subsection, we examine the performance of the best classifiers for each model type on the test sets, followed by how these metrics can be improved using prediction post-processing. Similar to previous chapters, patient results are typically displayed in average, with further patient-by-patient details available in the supplementary information document (https://bit.ly/3bZQxop).

5.4.1 Absence Seizures

As discussed in subsection 5.3.1, three separate absence seizure feature sets were created to reflect the typical input into different classifiers; LightGBM and MLP models having "hand-crafted" features, RNN and CNN1D models using filtered EEG as an input, and CNN2D models using a spectrogram from an UDWT as an input. Patients were randomly allocated into one of five groups (see table 5.2.2). The model optimisation and evaluation then occurred within a 5-fold cross-validation scheme, where at each fold 3 groups of patients were used for training, 1 group for validation, and 1 for testing (see table 5.4.1). As can be seen in figure 5.A.1, this can lead to each fold having varying numbers of seizures in each set.

Validation Scores

Across folds, LightGBM models on average had the highest validation F1-score and were the fastest models to train (see table 5.4.2). CNN2D and MLP models had the lowest average F1-score, followed by CNN1D, RNN, and LightGBM. The validation F1-scores for all models across folds overlapped to some degree, when considering their standard deviation (SD), however there were clear differences in the training times; as LightGBM

Table 5.4.1: TUH (Absence) folds and associated groups.

	Training	**Validation**	**Test**
Fold 1	Group 1 Group 2 Group 3	Group 4	Group 5
Fold 2	Group 2 Group 3 Group 4	Group 5	Group 1
Fold 3	Group 3 Group 4 Group 5	Group 1	Group 2
Fold 4	Group 4 Group 5 Group 1	Group 2	Group 3
Fold 5	Group 5 Group 1 Group 2	Group 3	Group 4

Table 5.4.2: Average training F1-scores and total training times across all TUH (Absence) folds.

Model	Max F1-score		Training Time (secs)		Unique Configurations	Runs	Total Time (hrs)
	Mean	*(SD)*	Mean	*(SD)*			
LightGBM	89.0	(3.96)	7.43	(6.72)	5010	6685	13.79
MLP	85.93	(4.37)	57.63	(34.76)	3000	4340	69.47
RNN	87.24	(6.44)	387.19	(289.08)	3000	4340	385.79
CNN1D	87.11	(4.97)	459.18	(798.41)	3000	4340	553.56
CNN2D	85.86	(4.59)	1352.68	(2108.3)	3000	4340	1630.74

Note. Average and standard deviation training time represents time in seconds to train 1 iteration of the Bayesian optimisation across all folds. Total time is the total GPU hours needed to train all optimisation runs across all folds.

was significantly faster to train than MLP, RNN, CNN1D, and CNN2D DL models. The differences in training time also reflect the size of the dataset used to train each model, the two models (LightGBM, MLP) that used "hand-crafted" features (0.08 Gb) were the fastest, followed by the two models (RNN, CNN1D) which used just the filtered EEG (0.6 Gb), with CNN2D the slowest using a spectograms as the input (5.3 Gb). Although all models were set to 200 iterations to determine the best parameters, the actual number of unique configurations and runs conducted is determined by the minimum and maximum budget set. In this case, all DL models had the same number of configurations and runs for training as they had the same budget of between 5 and 30 epochs, whereas LightGBM used between 50 and 1000 estimators as the budget. In practice, this optimisation method samples model configurations and trains them on a smaller budget, and only trains certain configurations on higher budgets which are more computationally demanding, in this case more epochs or estimators.

Model Parameters

All models in this work have a lot of hyperparameters to tune, with LightGBM having the most. Some models had clear areas of the hyperparameter search space that improved model performance (e.g. MLP), whereas others were less clear (e.g. LightGBM). Furthermore, a variety of model configurations were found in the optimal models for each fold. Indeed, looking across the training and validation scores during optimisation (see figure 5.A.2), it is evident that changes to hyperparameters generally either make minor differences to F1-score in the majority of models or create models with very poor performance. Training fold 3

appeared to have the most variation in scores across models compared to other folds, which could be due to the testing group in this fold (group 5) being the largest, and therefore the models in this case were training and validating models on an overall smaller amount of data. This is because we randomly assigned patients in each group, so each had roughly the same number of patients, rather than controlling for the same number of seizures in each fold (see figure 5.A.1).

LightGBM appears to be the least sensitive model to particular hyperparameter changes when compared to the other models. It also appears to be the most prone to overfitting to the training data (see figure 5.A.2a). Looking at the hyperparameter search space (see figure 5.A.3), across folds, learning rate typically produced better models when it was below 0.05 and models were generally worse when less than 10% of available interictal data was sampled for training. Certain folds had other clear patterns, such as folds 2, 3, and 4 (to a smaller degree) where weights on the seizure labels were best below 2, and folds 2 and 4 where sampling below 40% of available features for trees produced better models. However, as these patterns in the search space were not present across all folds, these recommendations should be taken with caution. If we look across the folds at the hyperparameters of the optimal LightGBM models (models with the best validation score) we can see that there are quite a range of "optimal" model configurations (see table 5.4.3). It is interesting that the boosting type for LightGBM varied between a traditional Gradient Boosting Decision Tree (GBDT) and Gradient-based One-Side Sampling (GOSS) between training folds. GOSS is a novel sampling method available in LightGBM which retains instances with large gradients and performs random sampling on instances with small gradients so that Stochastic Gradient Descent converges faster/better during training. Other large differences in hyperparameters between optimal models are the number of features sampled, L2-regularization, and weight applied to seizure labels.

As mentioned above, MLP models had the clearest areas in the hyperparameter search space that produced better models, even if these were not always consistent across folds (see figure 5.A.4). Learning rate was generally best below 0.05 (apart from fold 3), a finding that was consistent across most folds in other DL models, and the sample of interictal data was generally best between 35 and 50%, although this was not the case in folds 1 and 5.

Table 5.4.3: Optimal TUH (Absence) seizure model hyperparameters for each fold.

Classifier	Type	Hyperparameter	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
LightGBM	Boosting Specific	Boosting Type	Gbdt	Goss	Goss	Goss	Gbdt
		Early Stopping Rounds	73	73	65	57	62
		L1 Regularization	0.41	0.85	0.37	0.58	0.31
		L2 Regularization	0.09	0.43	0.42	0.83	0.43
		Learning Rate	0.05	0.13	0.01	0.08	0.19
		Maximum Tree Depth	37	29	20	19	32
		Maximum Tree Leaves	19	8	10	37	11
		Minimal Sum Hessian in a Leaf	1.07	4.41	0.03	0.25	3.58
		Minimum Samples at a Node	7	14	2	3	7
		Number of Estimators	1000	1000	1000	1000	111
		Sample of Features	0.89	0.12	0.52	0.25	0.99
		Sample of Interictal Data	0.45	0.83	0.71	0.6	0.72
		Weight of Seizure Labels	3.6	0.2	0.03	5.32	1.77
MLP	Network General	Kernel Regularizer	-	Max Norm	-	-	Max Norm
		Learning Rate	0.01	0.1	0.11	0.05	0.05
		Loss Function Weighting	False	False	False	False	False
		Number of Epochs	30	30	30	30	30
		Sample of Interictal Data	0.27	0.45	0.47	0.45	0.45
	Network Specific	Dense Layers	3	1	10	11	7
		First Layer Neurons	204	185	207	298	295
		Hidden Layer Neurons	293	-	231	245	283
RNN	Network General	Kernel Regularizer	Max Norm	L2	Max Norm	-	Max Norm
		Learning Rate	0.01	0.01	0.01	0.01	0.02
		Loss Function Weighting	False	False	False	False	False
		Number of Epochs	30	30	30	30	7
		Sample of Interictal Data	0.34	0.42	0.42	0.39	0.36
	Network Specific	Convolutional Layer	False	True	False	False	False
		Filters	-	6	-	-	-
		Kernel	-	4	-	-	-
		Recurrent Layers	3	4	3	4	4
		Recurrent Neurons	113	86	28	39	14
		Dense Layers	0	1	1	2	0
		Dense Neurons	-	53	75	93	-
CNN1D	Network General	Kernel Regularizer	-	L2	-	-	-
		Learning Rate	0.07	0.02	0.12	0.02	0.02
		Loss Function Weighting	False	False	False	False	False
		Number of Epochs	30	30	30	30	30
		Sample of Interictal Data	0.19	0.33	0.29	0.49	0.41
	Network Specific	Convolutional Model	VGGNet	Xception	VGGNet	WaveNet	Xception
		Convolutional Layers/Blocks	1	2	3	2	9
		Filters	97	2	77	106	2
		Kernel	2	4	2	2	5
		Dense Layers	1	2	1	1	0
		Dense Neurons	64	1495	371	128	-
CNN2D	Network General	Kernel Regularizer	Max Norm	L2	L2	L2	-
		Learning Rate	0.12	0.01	0.02	0.18	0.01
		Loss Function Weighting	False	False	False	False	False
		Number of Epochs	30	30	30	30	30
		Sample of Interictal Data	0.28	0.33	0.48	0.31	0.16
	Network Specific	Convolutional Model	ResNet	Xception	Xception	Xception	VGGNet
		Input	$Chan \times Samp \times Freq$	$Freq \times Samp \times Chan$	$Chan \times Samp \times Freq$	$Chan \times Samp \times Freq$	$Chan \times Samp \times Freq$
		Convolutional Layers	6	3	10	6	1
		Filters	34	98	18	6	11
		Kernel	4	6	5	8	7
		Dense Layers	0	2	1	0	0
		Dense Neurons	-	658	528	-	-

This may be linked to hidden layer neurons, which also had similar patterns in folds 2, 3, and 4, associated with better models when set between 200 and 300, with this pattern not occurring in folds 1 and 5. Another clear difference between the search space results was that folds 1, 2, and 5 generally had better results with models with few dense layers (3 and below), whereas folds 3 and 4 had better models with many dense layers (between 9 and 12). This inconsistency is likely due to the small amount of patients in the TUH (Absence) dataset, meaning training and validation samples varied across folds, with more patients potentially leading to more consistent hyperparameter search spaces between folds. However, MLP models were mostly comparable across folds in their optimal configurations (see table 5.4.3), with the number of dense layers being the exception. The number of optimal dense layers varied between 1 and 11 across folds, but the number of neurons in these layers were generally more consistently between 200 to 300. All models found training for the full available 30 epochs optimal, a finding that was similar across all other DL model configurations. It is worth noting that 30 epochs is a comparatively small number of epochs for DL models, chosen in the interest of being able to train many DL model configurations; therefore certain configurations would likely benefit from an increase in epochs. Similarly, across optimal DL models, sample weighting was never applied to the loss function based on class distribution; suggesting resampling the interictal classes alone was a better method to account for the class imbalance.

During training, RNN models were the most affected by hyperparameter choice as it had the most variable validation F1-score (see figure 5.A.2a). The search space values in figure 5.A.5 reveal poor performing models generally had many recurrent neurons and layers, a small sample of interictal data, and not enough dense neurons in the final hidden layer(s). Generally, between 10 and 40 recurrent neurons, and below 5 recurrent layers, lead to better models. The number of dense neurons associated with better models was inconsistent between folds, however above 50 appears to be a general pattern. For the optimal models (see table 5.4.3), RNN had the most consistent "network general" parameters of all the optimal DL models, and generally had a similar number of recurrent layers. However, the number of recurrent neurons in layers varied, as well as the number of dense hidden layers. Indeed, two models did not have any hidden dense layers before the final output dense layer.

The two types of CNN models (1D & 2D) were different in their optimal model parameters. However, for both 1D and 2D CNN models, limited conclusions regarding optimal configuration can be drawn from the search space (see figures 5.A.6 and 5.A.7) or optimal model configurations (see table 5.4.3). Nevertheless, it seems from the optimal 1D CNN models that the chosen model type influences the best number of filters, as VGGNet models have a large number of filters, whereas Xception models have fewer. However, there is no clear area in the search space for the number of filters, or many other parameters for that matter, associated with better models. This could be due to there not being separate search spaces for a number of parameters for each different model type (VGGNet, ResNet, Xception, and WaveNet); although we had separate search spaces for the number of hidden layer dense neurons and CNN layers/blocks. There is a balance between creating too many parameters to optimise and allowing the granularity of separate search spaces for parameters which are associated with particular model configurations. Nevertheless, this may have affected CNN model performance. With an increased computational budget, this could be addressed by training each model type separately or separating these search spaces and having more optimisation iterations. Still, it appears generally that shallow 1D CNN models, compared to other applications, provide better performance for absence seizure detection - although this is likely affected by the comparatively small data size. For optimal 2D CNN models, it is clear that models benefited from the input data being formatted as $EEGChannel \times Samples \times Freq$, rather than $Freq \times Samples \times EEGChannel$, as well as having a larger kernel than 1D CNN models. Optimal 2D models also were generally deeper than 1D CNN models, as well as other DL models, and favoured model types that had skip connections (Xception & ResNet), which was also the case for 1D CNN models (Xception & WaveNet).

Test Scores

Similar to the validation scores, where LightGBM had the best average maximum F1-score and lowest training time, LightGBM generally had the best average performance across the test records on all metrics except from sensitivity and AUC (see table 5.4.4). As with the validation scores, 2D CNNs generally performed the weakest across performance metrics.

Table 5.4.4: Average (and standard deviation) test scores, across folds, for TUH (Absence) seizure models.

Classifier	Accuracy		Sensitivity		Specificity		Precision		F1-score		AUC		FP/h		Prediction Time	
	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*
LightGBM	**98.52**	(1.36)	79.8	(9.52)	**99.39**	(1.31)	**89.42**	(15.01)	**83.07**	**(8.12)**	89.6	(4.55)	**20.66**	(44.36)	**0.04**	**(0.03)**
MLP	98.36	(1.42)	80.07	**(8.53)**	99.25	(1.35)	84.74	**(14.41)**	81.42	(8.58)	89.66	**(4.12)**	25.68	(45.6)	0.17	(0.11)
RNN	98.41	**(1.32)**	**81.55**	(9.15)	99.22	**(1.26)**	84.97	(16.18)	82.04	(9.26)	**90.39**	(4.47)	26.55	**(42.83)**	1.51	(0.58)
CNN1D	98.21	(1.47)	76.29	(9.15)	99.24	(1.34)	84.35	(16.41)	78.94	(9.65)	87.76	(4.53)	26.16	(45.4)	0.74	(0.95)
CNN2D	98.01	(1.52)	80.59	(12.3)	98.87	(1.6)	79.54	(19.99)	77.78	(10.96)	89.73	(5.86)	38.8	(54.48)	1.36	(0.78)

Note. The best average score for each metric, across classifiers, are in bold.

Table 5.4.5: Average (and standard deviation) post-processed test scores, across folds, for TUH (Absence) seizure models.

Classifier	Accuracy		Sensitivity		Specificity		Precision		F1-score		AUC		FP/h		Prediction Time	
	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*
LightGBM	98.66	(1.22)	77.07	**(9.76)**	**99.68**	(1.04)	**96.21**	(11.54)	84.65	**(6.99)**	88.37	**(4.68)**	11.02	(35.26)	**0.04**	**(0.03)**
MLP	98.65	(1.13)	77.07	(10.52)	**99.68**	(0.91)	95.57	**(10.57)**	84.43	**(6.99)**	88.37	(5.11)	10.93	(30.78)	0.17	(0.11)
RNN	**98.72**	**(1.05)**	**80.16**	(9.98)	99.6	(0.87)	92.66	(11.25)	**85.15**	(7.17)	**89.88**	(4.89)	13.63	**(29.64)**	1.51	(0.58)
CNN1D	98.42	(1.26)	71.19	(12.4)	**99.68**	(0.99)	96.03	(11.91)	80.65	(9.04)	85.44	(6.1)	**10.77**	(33.56)	0.74	(0.95)
CNN2D	98.4	(1.29)	78.58	(15.34)	99.38	(1.15)	89.5	(14.65)	81.7	(9.56)	88.98	(7.47)	21.02	(39.02)	1.36	(0.78)

Note. The best average score for each metric, across classifiers, are in bold.

Although the sensitivity of CNN2D models was comparatively good, this was at the expense of the highest number of false positives. Unlike the validation scores, the second poorest model was CNN1D, which was similar to MLP in most aspects apart from a worse sensitivity. The second best model was RNN, which had the best sensitivity and AUC, but had poorer performance on precision and FP/h compared to LightGBM. Similar to models in chapter 4, the main limitation across models was not fully marking the full duration of seizures.

In chapter 4, we found there was generally an F1-score increase in the validation set compared to the test set. We suggest in that chapter this could be due to stratified cross-validation splitting training and validation sets whilst preserving the percentage of samples for each class, rather than separating patient records so different segments from the same patient do not appear in both sets. However, in the current chapter we find there is also generally better performance in validation scores compared to the test set (see figures 5.4.1 and 5.A.8), despite having whole patient records completely separate between training and validation sets. This suggests this finding is likely to be due to some over-fitting of the

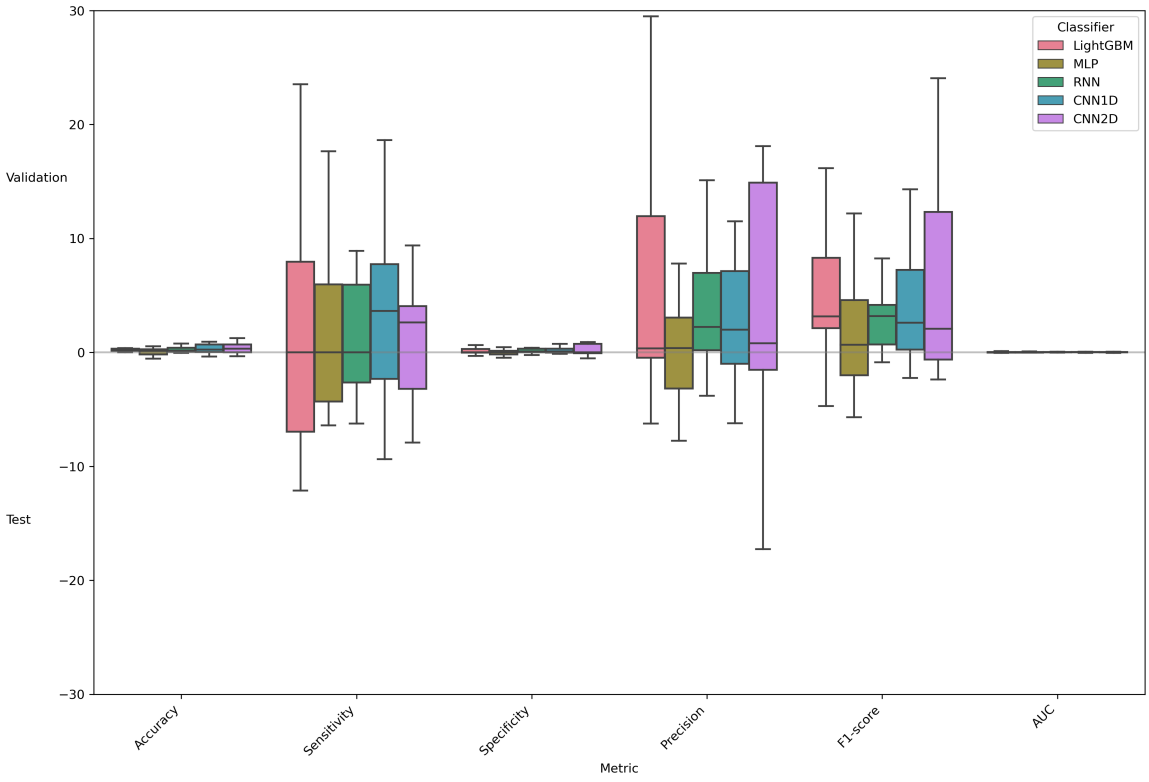


Figure 5.4.1: Difference between the best absence models validation and test score metrics.

models to the validation set during optimisation.

Post-Processing

As discussed in chapters 3 and 4, the majority of performance metrics can be improved with post-processing of the predictions. Performance is improved as short predictions are removed based on an appropriate threshold for the length of a seizure prediction. As in the previous chapters, the best window sizes for this threshold varied between 3 and 4 seconds across folds and classifiers for absence seizure detection (see figure 5.4.2). Post-processing is most beneficial for improving the precision of models, at the expense of sensitivity. After post-processing, LightGBM was no longer the best performing model, instead it is surpassed by RNN (see table 5.4.5). This is due to the increase in precision and reduction of FP/h for this model, whilst still maintaining the best sensitivity. LightGBM still performs as the second best model, however its performance is more similar to MLP. The greatest model improvement is arguably for CNN1D, which although has the worst sensitivity, now has the best specificity and lowest false positives. The improvements seen in RNN, CNN1D, and CNN2D models are due to them being more prone than LightGBM or MLP to predict short false positives which are filtered out by the post-processing.

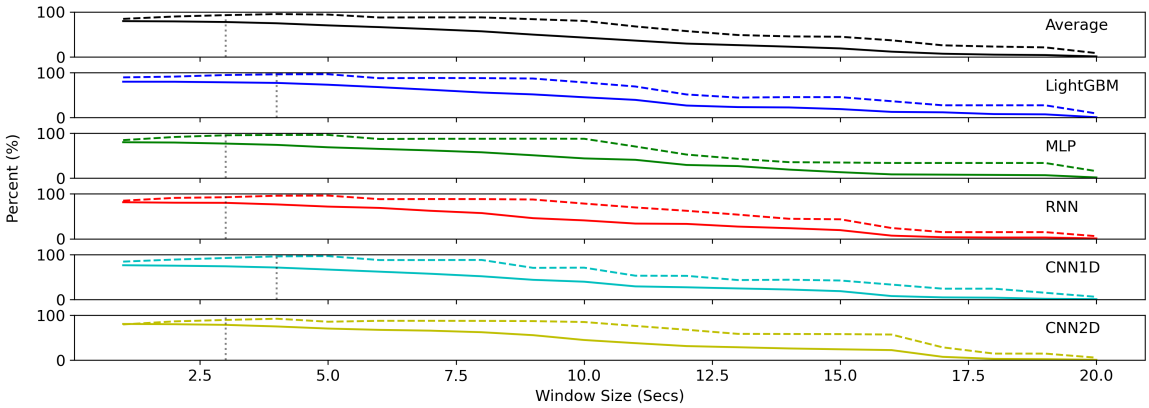


Figure 5.4.2: Effects of post-processing window size on TUH (Absence) test set performance metrics.
Note. Thick line is sensitivity and the dashed line is precision.

5.4.2 Generalised Model Training

We trained LightGBM, MLP, RNN, and CNN1D models using a similar method as above for the TUH (Generalized) dataset. However, due to the increased computational cost resulting from larger feature sets ("handcrafted" feature set, 1.2 Gb; filtered EEG set, 7.4 Gb), we only trained, validated, and tested these models using one fold instead of five - resulting in a single training/validation/test split. We did not train CNN2D models on the TUH (Generalized) dataset due to the considerable increase in training time these models required compared to other models on the TUH (Absence) dataset (see table 5.4.2) and increase in input data size for the generalized UDWT feature set (47.2 Gb). Furthermore, on the TUH (Absence) data, average 2D CNN model performance was worse than CNN1D models (see tables 5.4.4 and 5.4.5), other than for sensitivity metrics.

Validation Scores

As with the models trained on the TUH (Absence) feature sets, LightGBM models had the highest maximum F1-score on the validation data during training optimisation, as well as being the fastest to train (see table 5.4.6). Similarly, the time to train each model followed the same ordering as the absence models, except 5-20 times longer per model iteration depending on model type. Again, this reflects not only the typical model complexity, but also the size of the input data. However, in contrast to the absence analysis, RNN models were now the poorest performing model, followed by CNN1D, MLP, and LightGBM remaining as the top performing model. A limitation to the model training was that the number of unique configurations and runs varied across models due to larger models (RNN & CNN1D) requiring larger computational resources, so were more costly to train.

Model Parameters

Similar to the TUH (absence) models, changing model hyperparameters either resulted in small changes in the best F1-score the models could achieve, or models performed very poorly. MLP models were the least sensitive to hyperparameter changes, with LightGBM, RNN, and CNN1D having lots of variation between iterations (see figure 5.A.9). Further-

Table 5.4.6: Average training F1-scores and total training times across all TUH (Generalized) folds.

Model	Max F1-score	Training Time (secs) Mean	*(SD)*	Unique Configurations	Runs	Total Time (hrs)
LightGBM	57.15	33.01	(32.8)	504	673	6.17
MLP	55.34	522.47	(329.57)	600	868	125.97
RNN	41.75	5768.76	(5149.56)	259	374	599.31
CNN1D	49.34	8861.07	(12227.52)	107	154	379.06

Note. Average and standard deviation training time represents time in seconds to train 1 iteration of the Bayesian optimisation across all folds. Total time is the total GPU hours needed to train all optimisation runs across all folds.

more, the overfitting of LightGBM models to the training data is much less prominent in this dataset, raising the question whether this is due to the increased size or related to the properties of the different seizure classes. Instead, MLP models clearly were overfitting to the training data, as well as most RNN and CNN1D models.

As shown in figure 5.A.10, it is difficult to identify particular values that are optimal for LightGBM hyperparameters except a low maximum number of tree leaves (between 1-10), some weighting for seizure labels being useful, and high L1-regularization generally resulting in better performing models. However, the optimal model for LightGBM (see table 5.4.7) only used a low amount of L1-regularization, therefore high L1-regularization may not be necessary, just more likely to lead to a better model. The optimal generalized model was also relatively similar to the optimal model configurations observed on the absence data, although with lower L1- and L2-regularization; likely as with more intra-patient and inter-patient variability for seizures, there is less chance for the model to overfit to the validation data. However it is still worth keeping in mind the variation in potential model configurations which would have produced similar scores for this particular model.

For all models, except MLP, a low learning rate generally led to better validation F1-scores during training. Similar to MLP models trained on absence data, generalized MLP models also had more clear optimal areas of the search space than other models (see figure 5.A.10). Loss function weighting was generally preferable, along with a small sample of interictal data for training. Models generally consisted of 1 or 2 dense layers with between 1 and 100 neurons in the first layer and 1 and 150 neurons in the hidden layers. The

Table 5.4.7: Optimal TUH (Generalized) seizure model hyperparameters

Classifier	Type	Hyperparameter	Value
LightGBM	Boosting Specific	Boosting Type	Goss
		Early Stopping Rounds	88
		L1 Regularization	0.16
		L2 Regularization	0.03
		Learning Rate	0.17
		Maximum Tree Depth	27
		Maximum Tree Leaves	7
		Minimal Sum Hessian in a Leaf	3.51
		Minimum Samples at a Node	15
		Number of Estimators	1000
		Sample of Features	0.44
		Sample of Interictal Data	0.53
		Weight of Seizure Labels	3.59
MLP	Network General	Kernel Regularizer	-
		Learning Rate	0.18
		Loss Function Weighting	False
		Number of Epochs	30
		Sample of Interictal Data	0.13
	Network Specific	Dense Layers	1
		First Layer Neurons	272
		Hidden Layer Neurons	-
RNN	Network General	Kernel Regularizer	L2
		Learning Rate	0.02
		Loss Function Weighting	False
		Number of Epochs	7
		Sample of Interictal Data	0.1
	Network Specific	Convolutional Layer	True
		Filters	57
		Kernel	5
		Recurrent Layers	1
		Recurrent Neurons	54
		Dense Layers	2
		Dense Neurons	68
CNN1D	Network General	Kernel Regularizer	-
		Learning Rate	0.01
		Loss Function Weighting	False
		Number of Epochs	30
		Sample of Interictal Data	0.11
	Network Specific	Convolutional Model	Wavenet
		Convolutional Layers/Blocks	4
		Filters	99
		Kernel	4
		Dense Layers	0
		Dense Neurons	-

optimal MLP model for the generalized seizures (see table 5.4.7) was different to the absence optimal models in that it had a lower number of dense layers (just 1), with a smaller sample of interictal data, and higher learning rate. Therefore it appears the features input into the model require less transformations within the model to represent generalized seizures, however this is relative to the potential model performance on generalized seizures alone as overall absence seizure performance was better.

As there are fewer iterations of RNN and CNN1D models for generalized models it is more difficult to identify clear patterns in the search space. Nevertheless, 2 hidden dense layers at the end of the RNN models seem to be better than only 1 or none, and again only a small sample of interictal data was generally needed for training (see figure 5.A.10). This small sample of interictal data was also in the optimal model and is different from the generally larger amount of interictal data used in the optimal absence models. Furthermore, the optimal models for the absence folds generally had no or one dense layer, whereas here it proved beneficial to have 2. For CNN1D models, more training would be required to identify any optimal areas in the search space. For the optimal model, the sample of interictal data is again comparatively low, and it chose a deep WaveNet model configuration; a shallower WaveNet only optimal in one of the folds in the absence seizure data.

Test Scores

Similar to the absence seizures and generalized validation scores, where LightGBM had the best maximum F1-score and lowest training time, LightGBM generally had the best average performance across the test records on all metrics apart from sensitivity (see table 5.4.8). Also similar to the generalized validation scores, RNN models generally performed the weakest across performance metrics. Although this models sensitivity alone was comparatively very good, this was at the expense of a very high number of false positives which would make this algorithm realistically unusable. Similar to the generalized validation scores, CNN1D was the second worst model on the test set due to its high false positive rate and much slower prediction time than other models. However due to its comparatively high sensitivity it had the second best F1-score, although all models F1-scores were generally poor. This could however be related to this model having the least amount of unique configurations sampled

Table 5.4.8: Average (and standard deviation) test scores, across folds, for TUH (Generalized) seizure models.

Classifier	Accuracy		Sensitivity		Specificity		Precision		F1-score		AUC		FP/h		Prediction Time	
	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)
LightGBM	**84.33**	(15.39)	47.69	(40.62)	**92.19**	(10.46)	**36.69**	(29.86)	**36.38**	(29.58)	**69.94**	(20.13)	**200.24**	***(196.11)***	**0.04**	***(0.05)***
MLP	68.21	***(14.66)***	66.26	(34.0)	70.79	(19.95)	25.78	(22.36)	28.7	(23.09)	68.53	(11.67)	882.43	(614.12)	0.42	(0.58)
RNN	20.83	(16.18)	**96.6**	***(2.08)***	6.63	***(3.31)***	16.11	***(18.04)***	24.27	***(22.22)***	51.62	***(1.88)***	2826.44	(614.01)	1.92	(2.73)
CNN1D	62.66	(16.88)	70.76	(20.71)	60.14	(20.66)	24.13	(22.25)	31.29	(22.48)	65.45	(15.21)	1159.59	(617.11)	29.89	(43.37)

Note. The best average score for each metric, across classifiers, are in bold.

Table 5.4.9: Average (and standard deviation) post-processed test scores, across folds, for TUH (Generalized) seizure models.

Classifier	Accuracy		Sensitivity		Specificity		Precision		F1-score		AUC		FP/h		Prediction Time	
	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*	Mean	*(SD)*
LightGBM	**86.04**	(18.25)	39.37	(40.63)	**97.45**	***(5.82)***	46.52	(38.29)	38.37	(35.36)	**68.41**	(20.57)	**54.78**	***(94.59)***	**0.04**	***(0.05)***
MLP	85.63	***(17.03)***	46.91	(39.95)	95.25	(8.82)	58.31	(31.29)	39.8	(28.65)	71.08	(20.1)	112.25	(152.29)	0.42	(0.58)
RNN	49.56	(17.24)	**80.28**	***(16.95)***	44.66	(19.96)	20.78	***(19.35)***	28.27	***(21.0)***	62.47	***(11.08)***	1660.71	(698.47)	1.92	(2.73)
CNN1D	85.65	(15.35)	40.29	(33.07)	95.22	(9.16)	**63.56**	(38.27)	**42.29**	(30.34)	67.75	(17.3)	116.46	(230.42)	29.89	(43.37)

Note. The best average score for each metric, across classifiers, are in bold.

during optimisation (see table 5.4.6). MLP was the second best model, and was similar to CNN1D except it had a slightly worse sensitivity but with a better specificity.

Across test patient records, MLP and CNN1D models predicted lots of false positives of short/medium length, with this occouring less frequently in LightGBM models. Across most patients, RNN models predicted the majority of the records as ictal, explaining its high false positive rate but good sensitivity. Particularly poor performance for all models was on record P26, which has a large proportion of data containing focal non-specific (FN) seizures. Although not exclusively, TUHS records are generally segmented around events of interest, which is why over half of this record consists of FN seizures. FNs, as labelled in TUHS, are similar to GN seizures, in that they are a category which covers a broad range of seizure etiologies and are "delineated from GN seizures only by the number and location of the channels on which they occur" (Ochal et al., 2020). Although similar to GN seizures, FN seizures occur 4 times less in the TUH (Generalized) dataset, so poor performance on this patients records is likely due to reduced training of models on FN seizures (see table 5.2.1). Similarly, models had poor performance on P50 records; particularly LightGBM models which missed all the FN and GN seizures present in this record. Although proportionally FN seizures covered less of the patients records than P26, they still covered more than GN seizures, also likely contributing to the poor performance of models on this patient. Nevertheless, performance on P64 was also poor for all models, due to a lot of false negative classifications, and these records only contained GN seizures.

Post-Processing

A much larger post-processing window size was best for the generalized seizures compared to models trained on the absence seizures (see figure 5.4.3) Nevertheless, the best window sizes for these models (12 - 26 seconds) should be able to capture a majority of generalized seizures, as the median length of seizures have been shown to range from 18 seconds to 130 seconds depending on type (Jenssen et al., 2006). This longer window size suggests that where GN seizures were predicted, they generally covered a number of sequential windows. Indeed, post-processing did have some affect on sensitivity, but not as much as the absence seizures. LightGBM had the smallest best windowsize, followed by MLP, CNN1D, and RNN

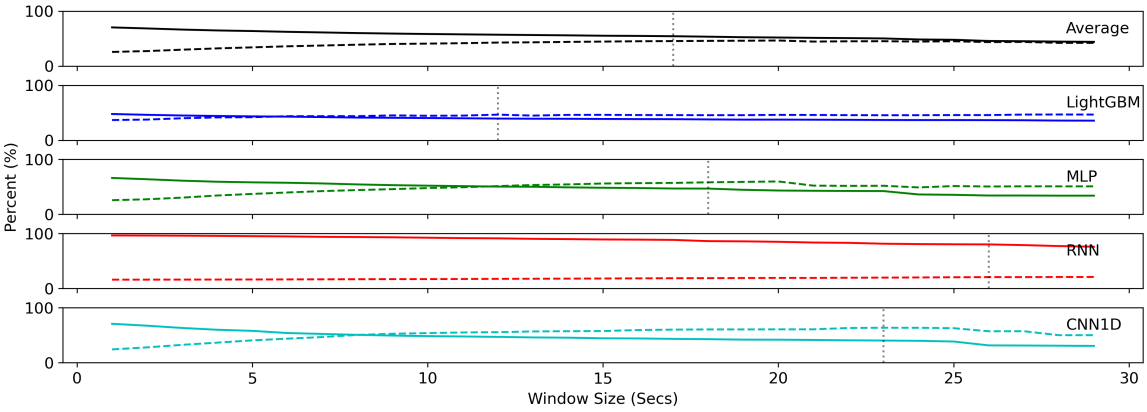


Figure 5.4.3: Effects of post-processing window size on TUH (Generalized) test set performance metrics.
Note. Thick line is sensitivity and the dashed line is precision.

with the largest. The specificity benefits were the largest in MLP and CNN1D models, due to these models predicting longer continuous false positives comparative to absence seizure records. With a larger post-processing window size, this reduced these false positives significantly.

After post-processing, LightGBM is still generally the best model (see table 5.4.9), however the other DL models have a noticeable improvement. Indeed CNN1D models now have the best precision and F1-score. However all models had relatively poor sensitivity to begin with, so a further decrease in sensitivity noticeably impacts their performance to fully classify ictal data segments. However, the FP/h is considerably reduced for the DL models using post-processing, demonstrating a clear trade-off.

5.5 Discussion

We assessed five ensemble classifiers (gradient boosting and DL) for the automatic detection of absence and GN seizures. Optimal hyperparameters and model configuration were investigated using a combination of Bayesian and Hyperband optimisation (BOHB). We found better performance across all models in their application to detecting absence seizures compared to GN seizures. This is likely due to GN seizures having large intra-patient and inter-patient variability compared to generalized absence seizures, which have little intra-patient and inter-patient variability. Across both seizure types, LightGBM appeared to

provide the best overall performance whilst being dramatically faster to train.

Performance in this chapter was comparable to, but weaker than, the balanced ensemble classifiers in chapter 4 using the same absence dataset. This difference could be accounted for by the different training/test paradigm, this chapter using five-fold-cross validation to group patients and chapter 4 using leave-one-patient-out cross-validation. Therefore, models were trained on more data in chapter 4 than in this chapter; with this different approach taken due to the computational cost associated with training DL models. There were also a number of differences between the hyperparameters in the optimal models in this chapter compared to chapter 4; LightGBM models in this chapter using Goss sampling more often, with deeper trees, less interictal data sampled per tree, and more estimators - although models in this chapter also had early stopping to try prevent overfitting. Models in this chapter also had a broader range of optimal parameters across folds, with these differences potentially due to a number of reasons, such as the different optimisation method used, less data per fold, and a more restricted choice for the number of estimators in the ensemble; as in this chapter this was used for the budget rather than having a separate search space.

In comparison to other published papers on the TUHS dataset, few have used gradient boosting to specifically detect absence or GN seizures for hyperparameters and model metric comparison. Vanabelle et al. (2020) trained another gradient boosting model (XGBoost) on a range of various seizure types in the TUHS dataset. Although they manually set hyperparameters, they used less estimators and had a smaller tree depth than the optimal LightGBM models found in this chapter (see table 5.5.1). Compared to this chapter, Vanabelle et al. (2020) reports XGBoost models as having a better average sensitivity (59.50%), using an leave-one-patient-out cross-validation, on a broader group of generalized seizures than used in this chapter. However this paper uses the Any Overlap (OVLP; Ziyabari et al., 2017)

Table 5.5.1: The most common categorical or average (and standard deviation) hyperparameter values across folds for LightGBM models compared to published research.

		Depth	**Estimators**	**Methods**	**Type**
Vanabelle et al. (2020)		3	400	GPU Hist	XGBoost
This Chapter	(Absence)	27.4 (6.95)	822.2 (355.6)	Goss	LightGBM
	(Generalized)	27	1000	Goss	LightGBM

Note. For all the optimal hyperparameters for each fold in this chapter, see tables 5.4.3 and 5.4.7.

method for calculating the metric, which inflates the sensitivity compared to the method used here. Roy et al. (2019a) also trained an XGBoost model to identify seizure type across a range of seizures in the TUHS dataset. They found the XGBoost model had the second best F1-score compared to KNN, and was better than SGD, AdaBoost, and CNN models. This paper used `HyperOpt` (Bergstra et al., 2015) to find the best hyperparameters for all models apart from the CNN model, but do not report the search space or final values for comparison. Zhang et al. (2018) also used a general gradient boosting model for seizure detection across multiple seizure types, finding poor sensitivity but good specificity; similar to the results found for GNs in this chapter. This paper also found gradient boosting to be better than naive Bayes, KNN, and random forest models, but worse than a logistic regression or support vector machine.

Although DL models are more common than gradient boosting in the seizure detection literature, comparisons between this chapter and other published research for DL model parameters is also limited by poor reporting practices (for review see Roy et al., 2019b). Nevertheless, although on a different dataset (CHB-MIT; see section 2.7), authors such as Pramod et al. (2014) and Wang and Ke (2018) have found great performance of MLP for paediatric seizure detection. Both authors also use "hand-crafted" features and balance the ictal/interictal classes before training, although only Pramod et al. (2014) test the models on full records using leave-one-patient-out-cross-validation on 1-second segments. This is likely why Wang and Ke (2018) report much better accuracy, sensitivity, and specificity metrics than found in Pramod et al. (2014) or this chapter. For comparison, our absence models (which are also trained with records from paediatric patients) have worse sensitivity, similar specificity, but better precision and F1-score than Pramod et al. (2014); but as these are on different datasets, this comparison should be made with caution. Although using random sampling for hyperparameter optimisation on a smaller search space than this chapter, Pramod et al. (2014) do not provide the best hyperparameters found for comparison. Conversely, Wang and Ke (2018) manually set all hyperparameters. Furthermore, both papers do not have a control model for comparison using the same methods.

Different types of RNN models have also been applied to records in both the CHB-MIT and TUHS datasets. For example, Vidyaratne et al. (2016) trained bidirectional RNNs on

5 subjects from the CHB-MIT dataset, as opposed to the unidirectional models we used in this chapter (see table 5.5.3). Using a patient specific leave-one-record-out validation scheme they found very high sensitivity with a low number of false detections. However, as part of the output, experimentally obtained patient-specific threshold values are used for prediction post-processing, which would be unrealistic to implement in practice beyond patients in long-term monitoring. Also on the CHB-MIT records, Yao et al. (2019) trained a much deeper independent RNN model than those trained in this chapter. They found RNNs to have better accuracy, sensitivity, specificity, F1-score, and precision than a CNN model (see table 5.5.4). However, compared our absence RNN models, their models had lower accuracy, specificity, F1-score, and precision, but better sensitivity. Again this comparison should be made with caution due to the different dataset and the fact they randomly undersampled the seizure and non-seizure segments to be used for both training and testing models in a ten-fold cross-validation. More similar to this chapter is Liu et al. (2020), who compared standard and bilinear RNN and CNN models on the TUHS dataset. They found RNN models overall generally had better performance, with this improved further by employing both in a hybrid model. Their RNN model had less recurrent layers, with fewer recurrent neurons, than the absence RNN models in this chapter, but more layers than the GN RNN models. They had a similar number of hidden dense layers at the end of the model, but each with more dense neurons. As shown in table 5.5.5, our optimal models had better accuracy for absence seizures, but were poorer for GN seizures. This could be due to their models training on a variety of generalized seizures, which may have boosted performance for the GN seizures. Indeed it is common that models are trained on multiple seizure types, however few authors report these models performance for each seizure type separately, as most just report general performance (see table 2.A.3). Their improved performance on GN seizures could also be due to their models training on more epochs, or the different input data; as they used an STFT ($Freq \times Samples \times EEGChannel$; 32, 9, 19), as opposed to a UDWT ($Freq \times Samples \times EEGChannel$; 5, 512, 19). However, this difference could merely be down to their use of stratified five-fold cross-validation for training and evaluation. Due to this cross-validation procedure, data from the same patient could have been mixed between training and test sets, although these would not be the exact same data segments.

Table 5.5.2: The most common categorical or average (and standard deviation) hyperparameter values across folds for MLP models compared to published research.

		Activation Function	Batch Size	Dense (Layers)	Dense (Neurons)	Dropout Rate	Epochs	Regularization	Learning Rate	Optimizer	Type
Pramod et al. (2014)		ReLU	??	[2-5]	[10-100]	[0.1-0.5]	200	L1 = 0.01	[0.01-0.1]	??	MLP
Wang and Ke (2018)		ReLU	50	4	hl1 = 529 hl2 = 80 hl3 = 20	hl1 = 0.1 hl2+ = 0	300	L2 = 2e-4	0.01	Gradient Descent	Cross-Layer MLP
This Chaper	(Absence)	ELU	32	6.4 (3.88)	hl1 = 237.8 (48.53) hl2+ = 263 (25.73)	0.5	30	Max Norm = 1.0	0.06 (0.04)	Adam	MLP
	(Generalized)	ELU	32	1	272	0.5	30	-	0.18	Adam	MLP

Note. For all the optimal hyperparameters for each fold in this chapter, see tables 5.4.3 and 5.4.7. Some hyperparameters have different values for each hidden layer (hl).

Table 5.5.3: The most common categorical or average (and standard deviation) hyperparameter values across folds for RNN models compared to published research.

		Activation Function	Batch Size	Epochs	Dense (Layers)	Dense (Neurons)	Regularization	Recurrent (Layers)	Recurrent (Neurons)	Learning Rate	Optimizer	Type
Vidyaratne et al. (2016)		??	??	??	0	-	-	1 (18 cells)	5	??	Unscented Kalman Filer	DRNN
Yao et al. (2019)		ReLU	30	30	1	100	-	12	120	0.0007	RMSprop	IndRNN
Liu et al. (2020)		Tanh	32	200	2	hl1 = 768 hl2 = 256	-	2	5	??	Adam	Conv LSTM
This Chaper	(Absence)	Tanh & Sigmoid	32	30	0.8 (0.75)	73.7 (16.36)	Max Norm = 1.0	3.6 (0.49)	56 (37.38)	0.012 (0.004)	Adam	GRU
	(Generalized)	Tanh & Sigmoid	32	30	2	68	L2 = 0.01	1	54	0.02	Adam	GRU

Note. For all the optimal hyperparameters for each fold in this chapter, see tables 5.4.3 and 5.4.7. Some hyperparameters have different values for each hidden layer (hl).

Table 5.5.4: The most common categorical or average (and standard deviation) hyperparameter values across folds for CNN models compared to published research.

		Activation Function	Batch Size	Convolutional (Layers/Blocks)	Convolutional (Filters)	Epochs	Dense (Layers)	Dense (Neurons)	Regularization	Learning Rate	Optimizer	Type
Iešmantas and Alzbutas (2020)		ReLU	??	2	10	??	1	1000	-	0.001	Adam	CNN2D
Yao et al. (2019)		LeakyReLU	30	5	hl1 & 2 = 100 hl3 & 4 = 200 hl5 = 260	50	2	h1 = 100 h2 = 50	-	0.01	Adam	CNN2D
Liu et al. (2020)		ReLU	32	3	hl1 = 128 hl2 = 64	200	2	h1 = 768 h2 = 256	L2 = 0.0001	0.0001	Adam	CNN2D (Bilinear)
Zhang et al. (2020)		ReLU	??	4	hl1 = 16 hl2 = 32 hl3 = 64 hl4 = 128	250	2	hl1 = 300 hl2 = 22	L2 = 0.0001	0.0001	Adam	CNN2D (Adversarial)
This Chaper	(Absence)	ReLU	32	2 (1)	87 (10)	30	1 (0)	217.5 (153.5)	-	0.095 (0.025)	Adam	CNN1D (VGGNet)
		ReLU	32	5.5 (3.5)	2 (0)	30	1 (1)	1495	L2 = 0.01	0.02 (0)	Adam	CNN1D (Xception)
		Tanh & Sigmoid	32	2	106	30	1	128	L2 = 0.01	0.02	Adam	CNN1D (WaveNet)
		ReLU	32	6	34	30	0	-	Max Norm = 1.0	0.12	Adam	CNN2D (ResNet)
		ReLU	32	1	11	30	0	-	-	0.01	Adam	CNN2D (VGGNet)
	(Generalized)	ReLU	32	6.3 (2.89)	40.67 (40.84)	30	1 (0.81)	593 (65)	L2 = 0.01	0.07 (0.08)	Adam	CNN2D (Xception)
		Tanh & Sigmoid	32	4	99	30	0	-	-	0.01	Adam	CNN1D (WaveNet)

Note. For all the optimal hyperparameters for each fold in this chapter, see tables 5.4.3 and 5.4.7. Some hyperparameters have different values for each hidden layer (hl).

Nevertheless, currently for absence seizures, due to the availability of absence records, it seems clear that they are best trained separately until they are at least more represented in larger datasets.

In this chapter, we found RNN models performed better than CNN models for absence seizures but much poorer on the GN seizure feature sets. Indeed, this latter finding is consistent with a recent systematic evaluation of architectures which concluded that currently CNNs tend to perform better than RNNs on a number of sequence modelling tasks (Bai et al., 2018). Indeed, CNNs popularity for a number of pattern recognition tasks is reflected by them being the most commonly applied DL model for EEG classification (Roy et al., 2019b). For seizure detection, CNN models tend to be the deepest DL models; indeed deeper CNNs have been found to give improved performance for neonatal seizure detection compared to shallower CNNs and a support vector machine (O'Shea et al., 2017). However, other published research has found that comparatively shallow CNN models can also be better than classical ML models (e.g. Iešmantas and Alzbutas, 2020; Zhang et al., 2020). Generally the number of electrodes and number of seizures in the data, as well as the chosen window size of each batch, limit the depth of CNN network architectures which use pooling layers (e.g. VGGNet). There are however benefits of a shallow network, such as simpler training which could aid online clinical diagnosis of epileptic signals (Zhou et al., 2018; Yan et al., 2018), and less chance of overfitting. Indeed, similar to this chapter and other published research (e.g. Schirrmeister et al., 2017), shallow networks have been shown to perform as good or better than deep networks with more complex structures (such as residual connections) for EEG classification. Nevertheless, the flexibility of DL models mean that deep complex models, such as ResNet50 (He et al., 2016), can be adapted to EEG classification by only re-training the final layer(s) (e.g. Roy et al., 2019a). Both the 1D and 2D CNN models in this chapter had worse sensitivity for absence and GN seizures than Iešmantas and Alzbutas (2020), but much better specificity and therefore better overall AUC (see table 5.5.5). Still, as these models were generally worse than other faster models, its hard to recommend these current CNN models over LightGBM, which provides substantially faster training and prediction times.

DL models generally take a long time to train due to their high computational complexity

Table 5.5.5: Comparison between average model performance reported in this chapter (after post-processing), to other published research using the TUHS dataset.

Reference	Evaluation Method	Subjects	Seizure Types	Classifier	ACC	SEN	SPEC	AUC
Iešmantas and Alzbutas (2020)	25% Holdout	246	Absence (0.68%)	CNN	-	80	50	62
			Generalized Non-Specific (16%)	CNN	-	62	58	63
Liu et al. (2020)	Stratified 5-Fold Cross-Validation	314	Absence (0.56%)	B-CNN	58.6	-	-	-
				B-RNN	66.19	-	-	-
				Hybrid	67.7	-	-	-
			Generalized Non-Specific (23%)	B-CNN	96.45	-	-	-
				B-RNN	96.46	-	-	-
				Hybrid	**96.68**	-	-	-
This chapter	5-Fold Cross-Validation	11	Absence (100%)	LightGBM	98.66	77.07	**99.68**	88.37
				MLP	98.65	77.07	**99.68**	88.37
				RNN	**98.72**	**80.16**	99.6	**89.88**
				CNN1D	98.42	71.19	**99.68**	85.44
				CNN2D	98.4	78.58	99.38	88.98
	20% Holdout	65	Generalized Non-Specific (100%)	LightGBM	86.04	39.37	**97.45**	68.41
				MLP	85.63	46.91	95.25	**71.08**
				RNN	49.56	**80.28**	44.66	62.47
				CNN1D	85.65	40.29	95.22	67.75

Note. The best average score for each metric, across classifiers, are in bold. For the full post-processed scores see table 5.4.9. Only papers which report seizure type performance are included.

(Coleman et al., 2017), with this limitation often influencing model selection strategies. In literature reviews focused on DL models for physiological signal classification, very few (Roy et al., 2019b) or even no (Faust et al., 2018) research papers were found to declare their model selection strategy or used statistical methods (e.g. cross-validation). This means most published research in this domain selected models and hyperparameters based on a single run and training the network only once, potentially resulting in a sample selection bias (Zadrozny, 2004; Huang et al., 2007; Faust et al., 2018). Computational power alone has been shown to account for 43% of the variance in image classification accuracy on the ImageNet benchmark, showing it is an inherent property of flexible and accurate DL models (Thompson et al., 2020). These computational costs also come with large financial costs; with the best performing models in other applications costing between $7,000 to $12 million to train (see Synced, 2019). Furthermore, after building and training costs, most organizations continue to commit 25-75% of the resources required to develop and deploy machine learning solutions to maintain the project (Dimensional Research, 2019). Furthermore, these costs can be more than purely financial, as large common DL models can emit five times the lifetime emissions of the average American car (Strubell et al., 2019). For an applied domain, such as seizure detection, these costs should be weighed against the current non-automated costs and performance gains available. There is clear potential monetary, performance, and productivity savings from introducing automation into healthcare (further discussed in chapter 6), but currently there will likely be a preference for ML techniques that are more computationally-efficient than current DL models, even if there is a slight improvement in performance in some cases (Thompson et al., 2020). As such, along with generally having better overall performance for both the detection of absence and GN seizures, future research should focus on training gradient boosted classifiers as well as the popular deep learning models for seizure detection.

There are also many other recommendations for future research to further the work presented in this chapter. Firstly, future research should investigate if/how including different seizure types improves generalized seizure detection for each distinct seizure category. This would contextualise the worse sensitivity for GN seizures found in this chapter compared to other research on generalized seizures. Furthermore, CNN models should be further in-

vestigated to see if separate search spaces for hyperparameters which are associated with particular model configurations improve model performance. Often research focuses on one particular CNN model configuration, but as there are now many potential model structures available in the DL literature, these should be compared using hyperparameter optimisation. Furthermore, this investigation could include other CNN structures which have promising applications, such as temporal convolutional networks (Bai et al., 2018).

A limitation for the application of DL for seizure detection is the lack of research on their application to highly imbalanced datasets (see Johnson and Khoshgoftaar, 2019). Indeed most published DL research for seizure detection input random undersamples of data into the networks for training, however balanced ensembles (see chapter 4) have a more robust way of accounting for imbalance; each tree of the forest having a balanced bootstrap sample rather than only one undersample for the whole model. Although LightGBM has hyperparameters specific to resampling input data, further investigation should be undertaken to look into methods to further regularize the model to prevent overfitting to the balanced training data, as was seen for the absence seizures.

There was a general inconsistency in hyperparameter spaces between folds which could be due to the small amount of patients in the absence dataset, with more patients potentially leading to more consistent hyperparameter search spaces. Therefore, an investigation of the methodology outlined in this chapter with a larger absence dataset and with additional folds for generalized datasets would be beneficial; although there is a large computational cost associated with such work. This chapter is limited by its high-dimensionality, as there are many hyperparameters, so it was difficult to identify the interactions between hyperparameters in models. As the search space is large due to limited previous research for guidance, subsequent research could do additional optimisation focusing on specific areas around the optimal parameters found in the results presented here. Additionally, although we looked at the features used for seizure classification for LightGBM models in chapter 4, it would be beneficial to look into the features used in optimal DL models for both absence and generalized seizures. Indeed this would be required to meet legislative requirements which require insight into automated decision making processes (European Parliament, 2016).

5.6 Conclusion

A gradient boosted tree algorithm (LightGBM) and several DL models (MLP, RNN, CNN) were compared for offline detection of absence and GN seizures within EEG recordings. Models trained to detect absence seizures, a seizure type with little intra-patient and inter-patient variability, had better performance across all investigated metrics compared to GN seizures; a seizure type with large intra-patient and inter-patient variability. This is however consistent with human raters, where visual review of ICU EEG records for seizures is also more prone to error. LightGBM models provided the best overall performance and were much faster to train, providing a computationally-efficient model which would be easier to implement into current NHS practice than costly deep learning models. Further investigation is required to assess performance across multiple seizure types, LightGBM and DL models decision making processes, and the cost/benefits of implementing different automated seizure detection models into NHS practice. Nevertheless, such an implementation has promise for monetary, performance, and productivity savings for the diagnosis of epilepsy and the monitoring of patients with seizures.

5.A Appendix D

Table 5.A.1: Channel occurrence across TUH (Generalised) records.

Used Channels	EEG PZ-REF	EEG T4-REF	EEG C3-REF	EEG FP1-REF	EEG C4-REF	EEG T5-REF	EEG F3-REF	EEG F8-REF	EEG T6-REF	EEG CZ-REF
(Count = 557)	EEG O1-REF	EEG FZ-REF	EEG FP2-REF	EEG T3-REF	EEG O2-REF	EEG F7-REF	EEG P3-REF	EEG F4-REF	EEG P4-REF	
Dropped Channels	EEG T2-REF	EEG T1-REF	SUPPR	EEG EKG1-REF	IBI	BURSTS	EEG A2-REF	EEG A1-REF	EEG 31-REF	EEG 32-REF
Counts	509	509	487	487	487	487	421	421	368	368
Dropped Channels	EEG C4P-REF	EEG C3P-REF	EEG SP1-REF	EEG SP2-REF	EMG-REF	EEG 29-REF	EEG 30-REF	PHOTIC-REF	EEG 26-REF	EEG 28-REF
Counts	316	316	248	220	209	164	164	115	85	85
Dropped Channels	EEG 27-REF	EEG LOC-REF	EEG ROC-REF	EEG 21-REF	EEG 25-REF	EEG 23-REF	EEG 20-REF	EEG 24-REF	EEG 22-REF	RESP ABDOMEN-REF
Counts	85	79	79	48	48	48	48	48	48	
Dropped Channels	EEG EKG-REF	EEG LUC-REF	EEG RESP1-REF	EEG RLC-REF	EEG RESP2-REF					
Counts	22	22	22	22	22					

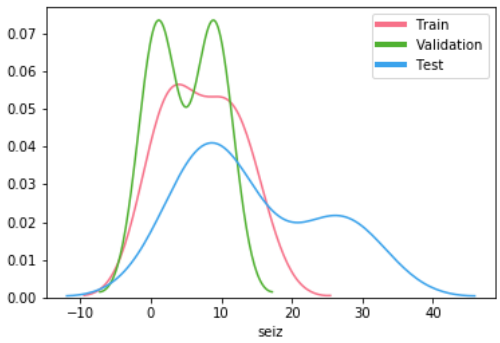
Table 5.A.2: Number of filters in the convolutional layers at each block.

Type	**Layer**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
VGGNet/ResNet	**1**	1	-	-	-	-	-	-	-	-	-
	2	1	2	-	-	-	-	-	-	-	-
	3	1	2	4	-	-	-	-	-	-	-
	4	1	2	4	8	-	-	-	-	-	-
	5	1	2	4	8	8	-	-	-	-	-
	6	1	2	4	4	8	8	-	-	-	-
	7	1	2	4	4	8	8	8	-	-	-
	8	1	2	4	4	4	8	8	8	-	-
	9	1	2	2	4	4	4	8	8	8	-
	10	1	2	2	4	4	4	8	8	8	8
Xception	**1**	1—2	-	-	-	-	-	-	-	-	-
	2	1—2	4	-	-	-	-	-	-	-	-
	3	1—2	4	8	-	-	-	-	-	-	-
	4	1—2	4	8	16	-	-	-	-	-	-
	5	1—2	4	8	16	16	-	-	-	-	-
	6	1—2	4	8	16	16	16	-	-	-	-
	7	1—2	4	8	16	16	16—20	22—24	-	-	-
	8	1—2	4	8	16	16	16	16—20	22—24	-	-
	9	1—2	4	8	16	16	16	16	16—20	22—24	-
	10	1—2	4	8	16	16	16	16	16	16—20	22—24

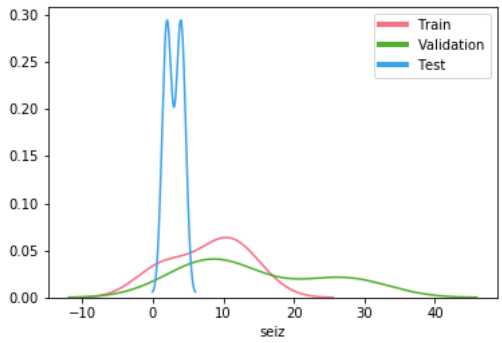
Note. Values based on if the search space were to select 1 filter for the first layer. — if change occours within the same layer. These are based on the increase found in their respective original papers.

Table 5.A.3: Medical history of patients in the TUH (Generalised) dataset according to patient notes.

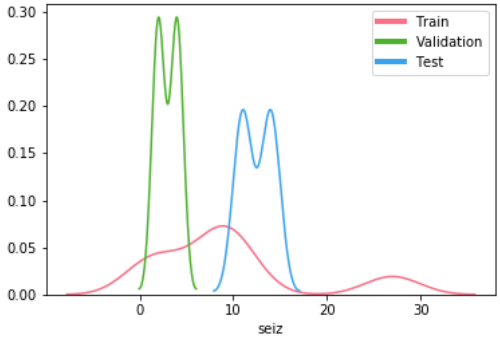
Patient ID	Age (Gender)	Patient History								
P1 (00000492)	54 (M)	Epilepsy	Head Trauma							
P2 (00000975)	19 (F)	Aspiration Pneumonia	Shaken Baby Syndrome							
P3 (00002380)	- (M)	Neurosurgical History								
P4 (00002521)	27 (F)	Anoxic Brain Injury								
P5 (00002868)	64 (F)	Cryptococcal Meningitis								
P6 (00002991)	63 (M)	Epilepsy	HIV Disease							
P7 (00003210)	29 (F)	Epilepsy	Head Trauma							
P8 (00004087)	58 (F)	Hypertension	Hyperlipidemia	Elevated Glucose	Refractory Seizures					
P9 (00004456)	43, 47 (F)	Refractory Epilepsy								
P10 (00004671)	22 (M)	Cardiomyopathy	MELAS syndrome	Refractory Epilepsy						
P11 (00005101)	82 (F)	Cranial Surgery	Refractory Statue Epilepticus	Tumor Resection						
P12 (00005265)	22 (M)	Mitochondrial Disease	Refractory Epilepsy							
P13 (00006107)	89 (M)	Post-Craniectomy	Subdural Hematoma							
P14 (00006230)	29, 32 (M)	Refractory Epilepsy								
P15 (00006440)	47 (M)	Renal Disease								
P16 (00006520)	20 (F)	Refractory Epilepsy								
P17 (00006546)	38, 40, 41, 42 (M)	Rasmussen's Encephalitis	Refractory Epilepsy							
P18 (00006563)	55 (F)	Anoxic Brain Injury								
P19 (00007032)	68 (F)	Stroke								
P20 (00007170)	80 (F)	Atherosclerosis	Dementia	Post-Traumatic Epilepsy						
P21 (00007828)	60 (M)	Anoxic Brain Injury								
P22 (00007936)	55 (F)	Diabetes	Hypertension	Obesity						
P23 (00007937)	38 (M)	Limbic Encephalitis								
P24 (00008174)	91 (M)	Dementia	Generalized Compulsive Seizures	Hypertension						
P25 (00008204)	60 (F)	Head Trauma?								
P26 (00008295)	22 (F)	Central Neurocytoma	Post-Craniectomy							
P27 (00008303)	55 (F)	Epiglottitis								
P28 (00008453)	47, 48, 49 (M)	Alcohol Dependence	Hypertension	Temporal Lobe Epilepsy						
P29 (00008479)	59 (M)	Cardiac Arrest								
P30 (00008480)	43 (M)	Childhood Epilepsy	Stroke							
P31 (00008512)	60, 61 (M)	Aphasia	Atrial Fibrillation	Hypertension	Stroke					
P32 (00008760)	42 (F)	Asthma	Migraines	Schizophrenia						
P33 (00009104)	60, 62 (M)	Epilepsy								
P34 (00009158)	82 (F)	Cancer	Craniotomy	Glioma/Encephalitis	Stroke					
P35 (00009162)	58 (F)	Bipolar	Depression	Diabetes	Hypertension	Craniectomy				
P36 (00009231)	- (F)	Dementia	Epilepsy	Schizophrenia						
P37 (00009232)	49 (M)	HIV	Hypertension							
P38 (00009370)	61 (F)	-								
P39 (00009540)	50 (F)	Thyroid Cancer								
P40 (00009623)	61 (M)	Acute Hyponatremia	Alcohol Abuse							
P41 (00009839)	64, 65 (M)	Alcoholic Cirrhosis	Cerebrovascular Accident	Hepatitis	Hypertension	Stroke				
P42 (00009852)	39 (M)	HIV	Post-Traumatic Epilepsy							
P43 (00009932)	53 (F)	Anxiety	Epilepsy	Hypertension						
P44 (00009934)	24 (M)	-								
P45 (00009994)	29 (M)	Lennox-Gastaut Syndrome								
P46 (00010020)	70 (F)	Atrial Fibrillation	Cardiac Arrest	Congestive Heart Failure	Diabetes	Hypertension				
P47 (00010062)	39 (F)	Lennox-Gastaut Syndrome								
P48 (00010106)	53 (F)	Breast Cancer	Depression	Diabetes	Hypertension					
P49 (00010158)	59 (M)	Chronic Obstructive Pulmonary Disease	Coronary Artery Disease	Schizophrenia						
P50 (00010418)	66 (F)	Cancer								
P51 (00010421)	42 (F)	Hepatitis C	History of Seizures	Drug Abuse						
P52 (00010455)	55 (F)	Asthma	Chronic Obstructive Pulmonary Disease	Diabetes	Drug Abuse	Emphysema	Heart Disease	Hepatitis B	HIV	Hypertension
P53 (00010639)	61, 62 (F)	Congestive Heart Failure	Coronary Artery Disease	Mitral Regurgitation	Stroke					
P54 (00010760)	61 (F)	Glioblastoma								
P55 (00010843)	64, 65 (F)	History of Seizures								
P56 (00010861)	57 (F)	Breast Cancer	HIV							
P57 (00011272)	58 (M)	-								
P58 (00011580)	- (F)	-								
P59 (00011870)	88 (F)	Craniotomy	Stroke							
P60 (00011972)	66 (F)	-								
P61 (00011999)	40 (M)	AIDS	CNS Lymphoma							
P62 (00012046)	21 (F)	Atypical Absence Epilepsy	Learning Disability							
P63 (00012707)	28 (F)	Bipolar								
P64 (00012940)	20 (M)	Anoxic Brain Injury								
P65 (00012941)	- (F)	Cardiac Arrest	Drug Abuse							



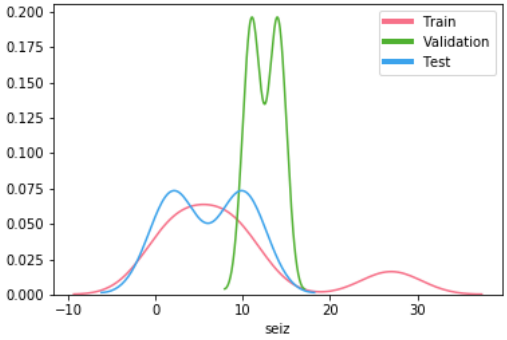
(a) TUH (Absence) - Fold 1



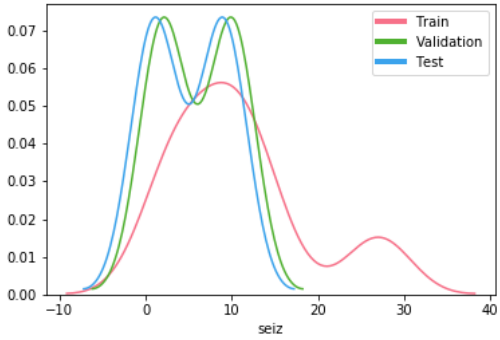
(b) TUH (Absence) - Fold 2



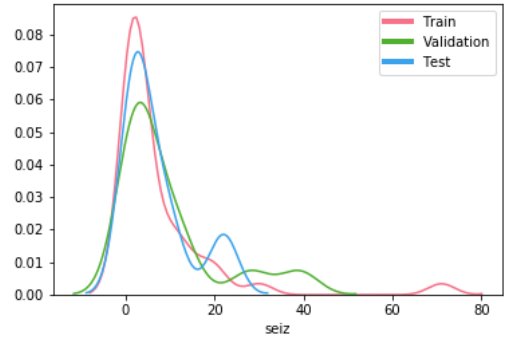
(c) TUH (Absence) - Fold 3



(d) TUH (Absence) - Fold 4

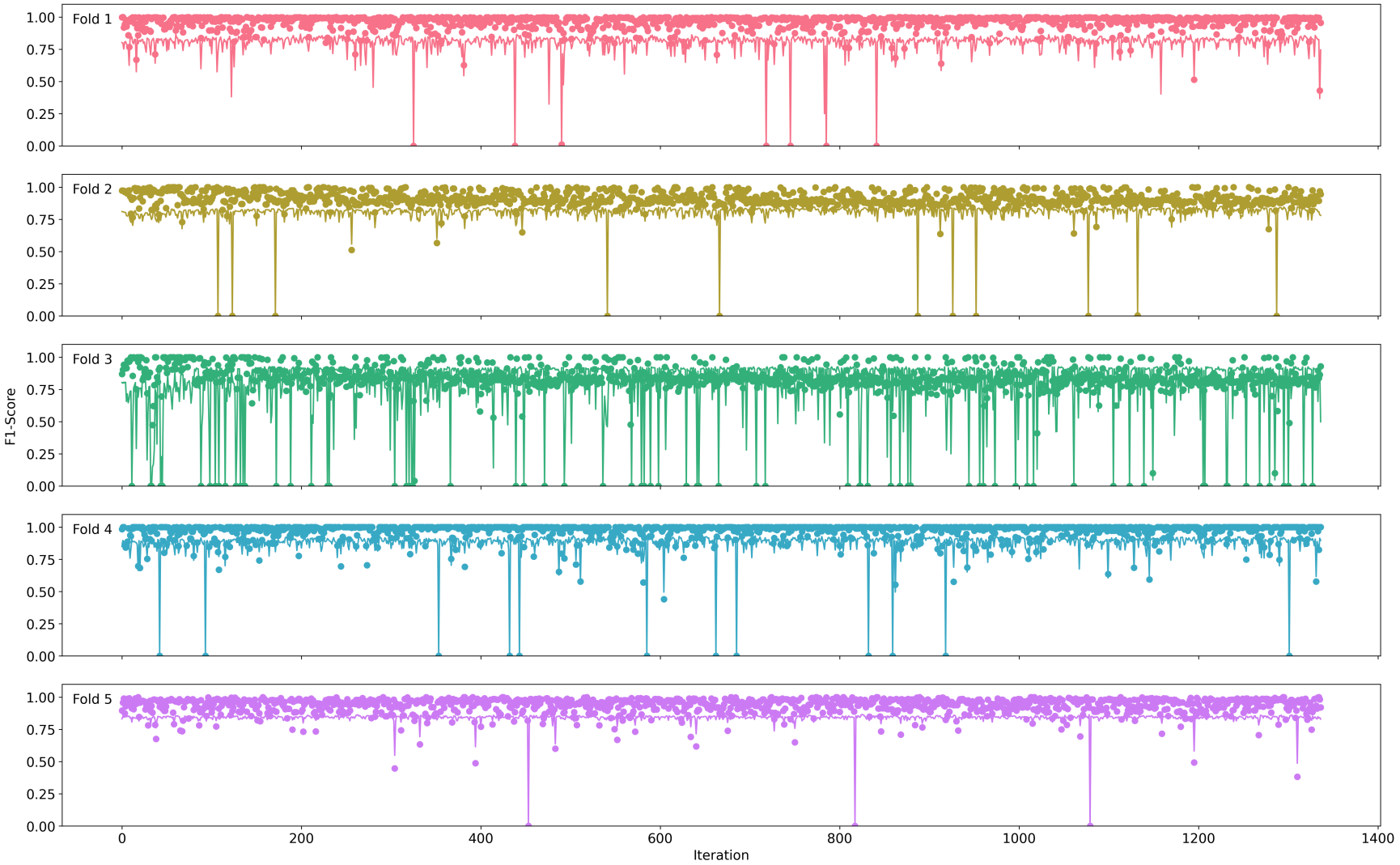


(e) TUH (Absence) - Fold 5

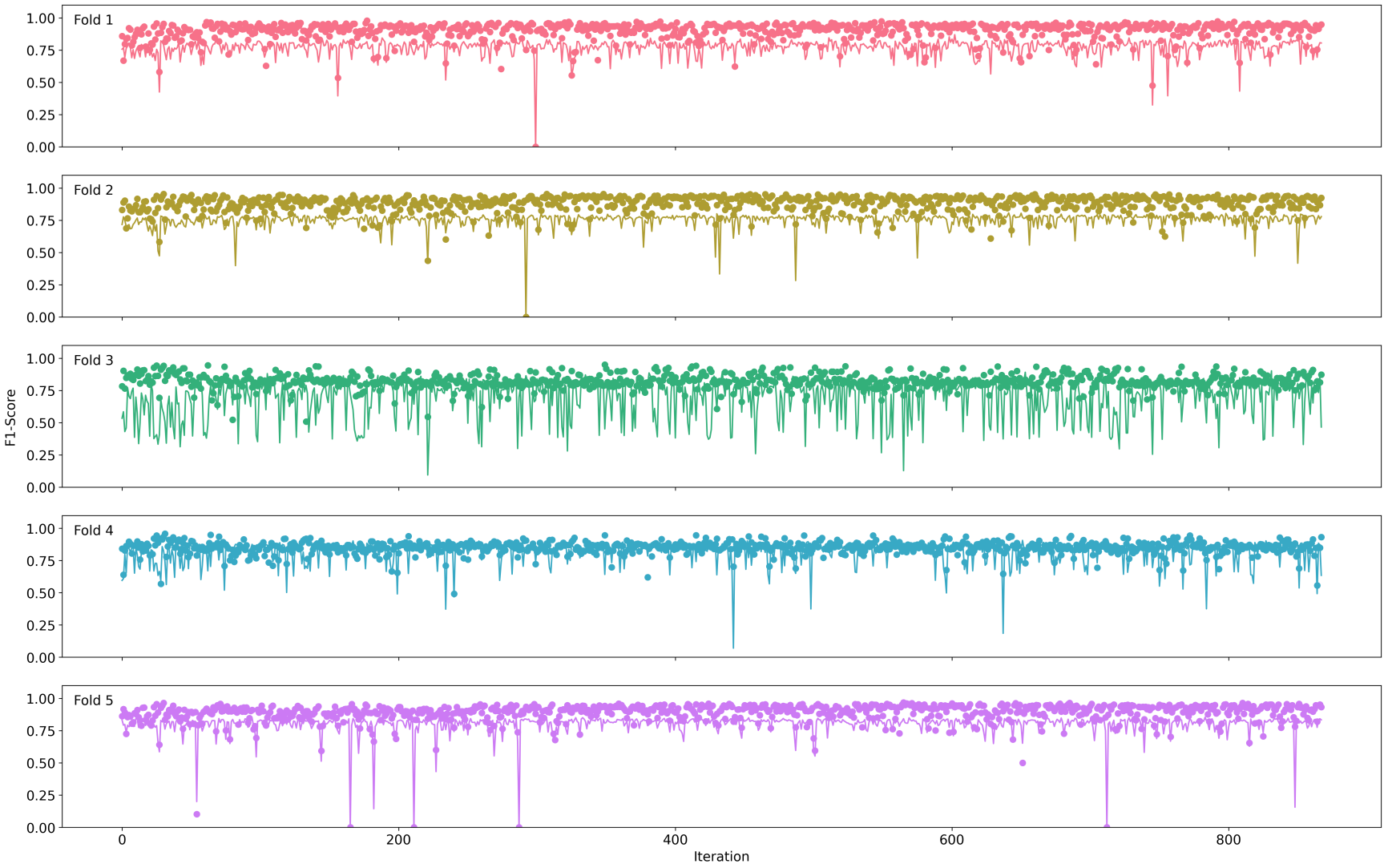


(f) TUH (Generalised)

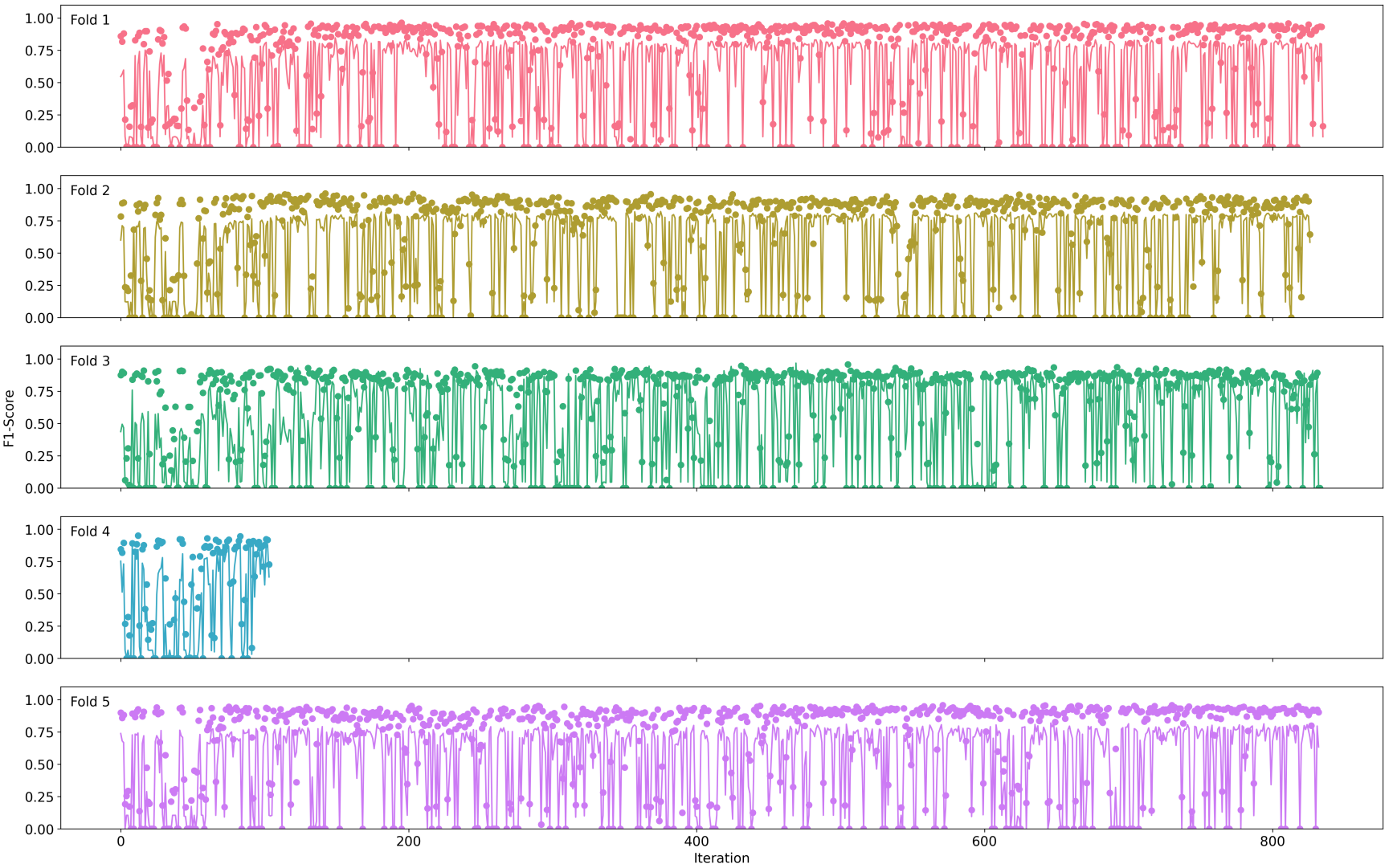
Figure 5.A.1: Gaussian kernel density estimates for the number of seizures in each fold


Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
F1-Score
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
1000
1200
1400
Iteration

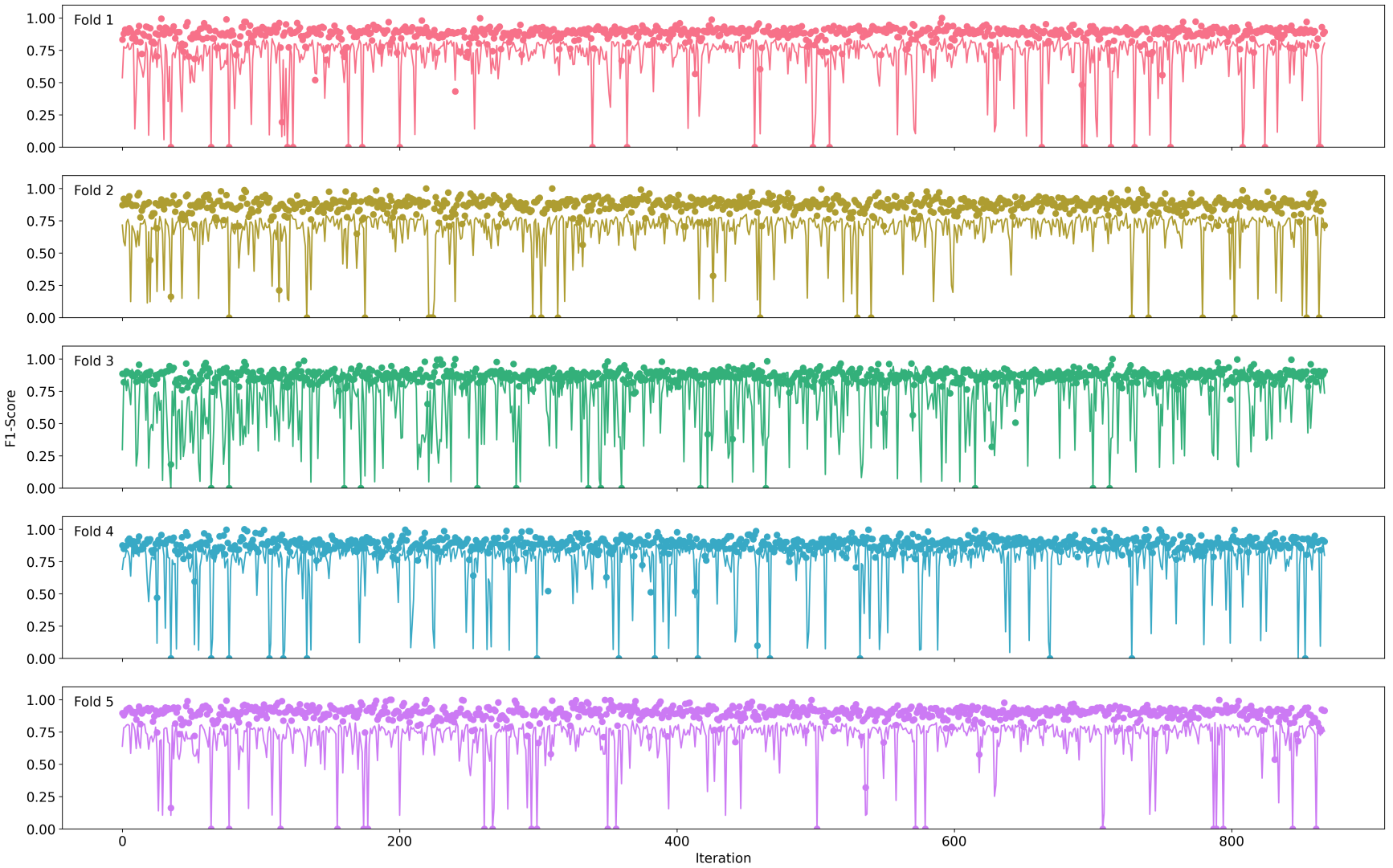
(a) LightGBM


Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
F1-Score
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
Iteration

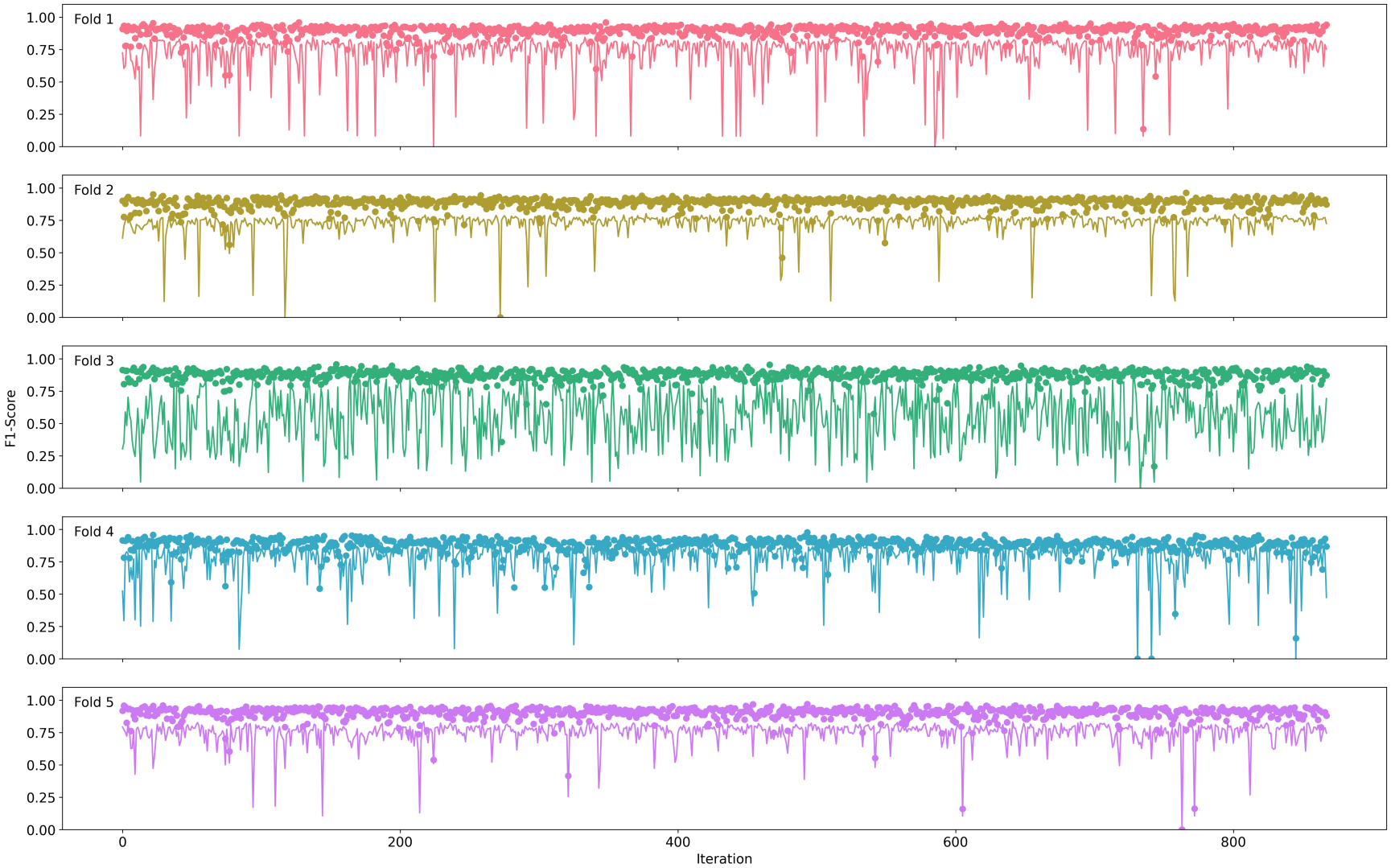
(b) MLP


Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
F1-Score
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
Iteration

(c) RNN

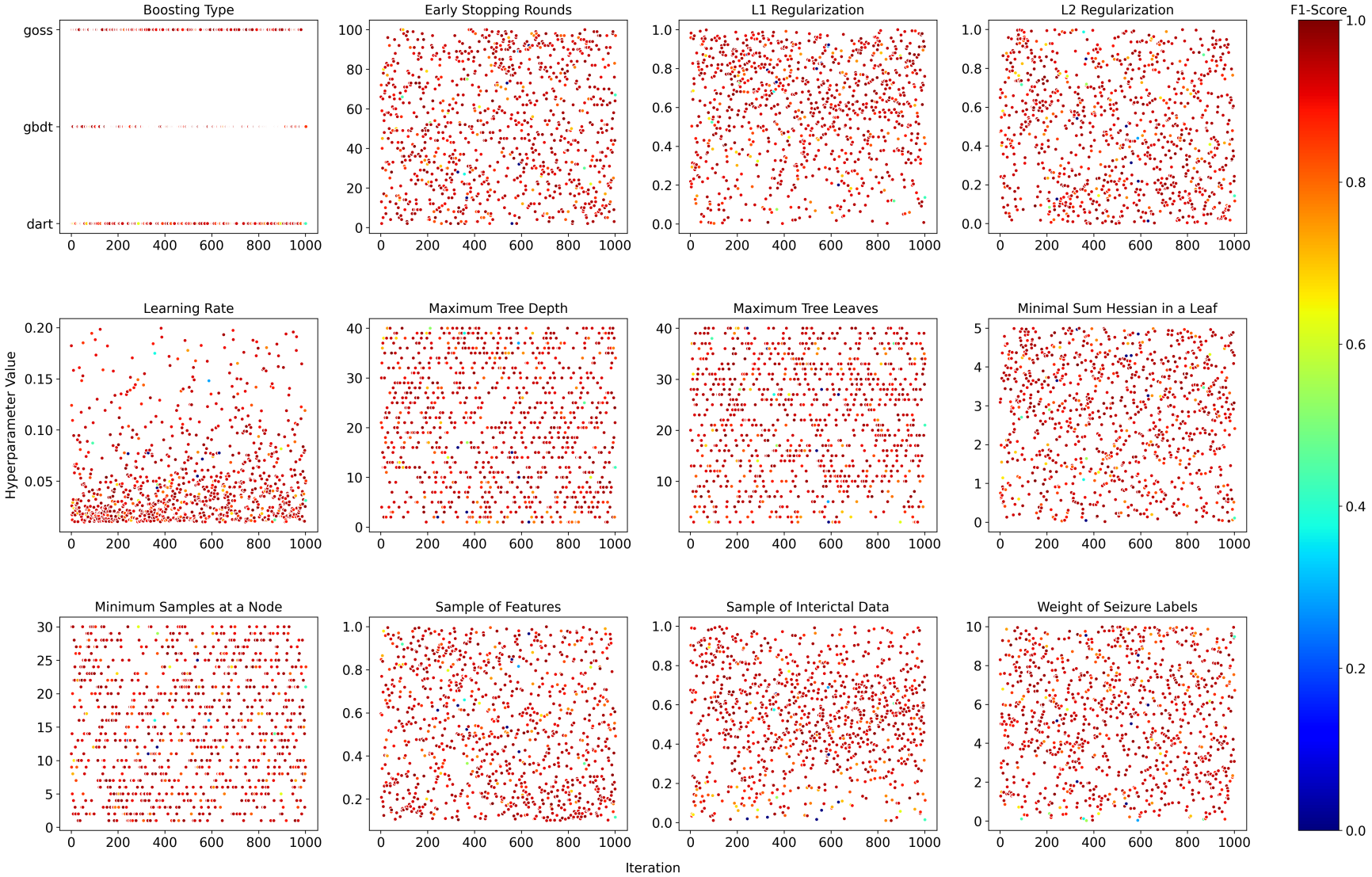

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
F1-Score
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
Iteration

(d) CNN1D

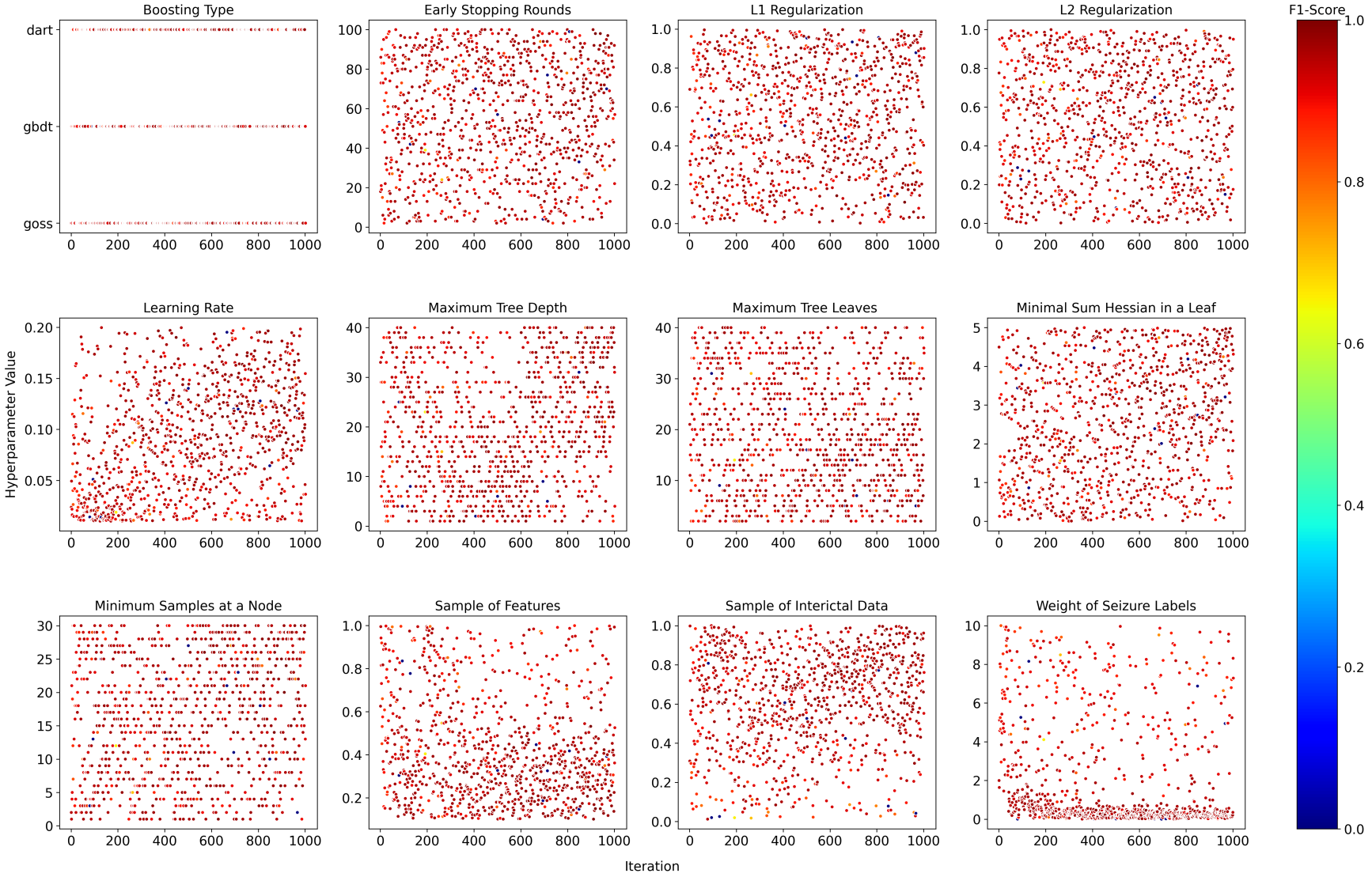


(e) CNN2D

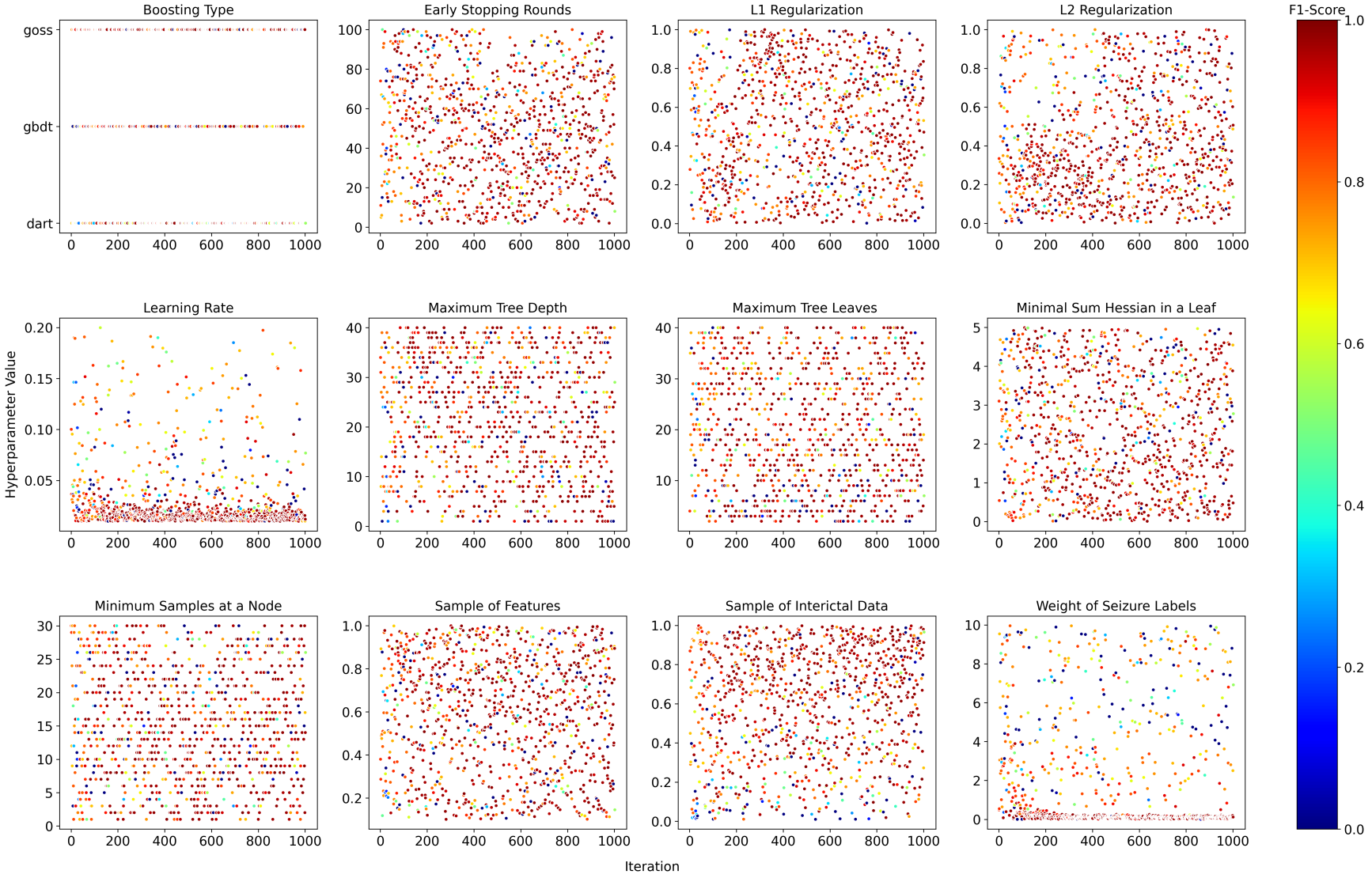
Figure 5.A.2: F1-scores during BOHB optimisation for models trained on TUH (Absence) records.
Note. Dots: Training scores; Line: Validation scores


Boosting Type
goss
gbdt
dart
Early Stopping Rounds
L1 Regularization
L2 Regularization
F1-Score
Learning Rate
Maximum Tree Depth
Maximum Tree Leaves
Minimal Sum Hessian in a Leaf
Hyperparameter Value
Minimum Samples at a Node
Sample of Features
Sample of Interictal Data
Weight of Seizure Labels
Iteration

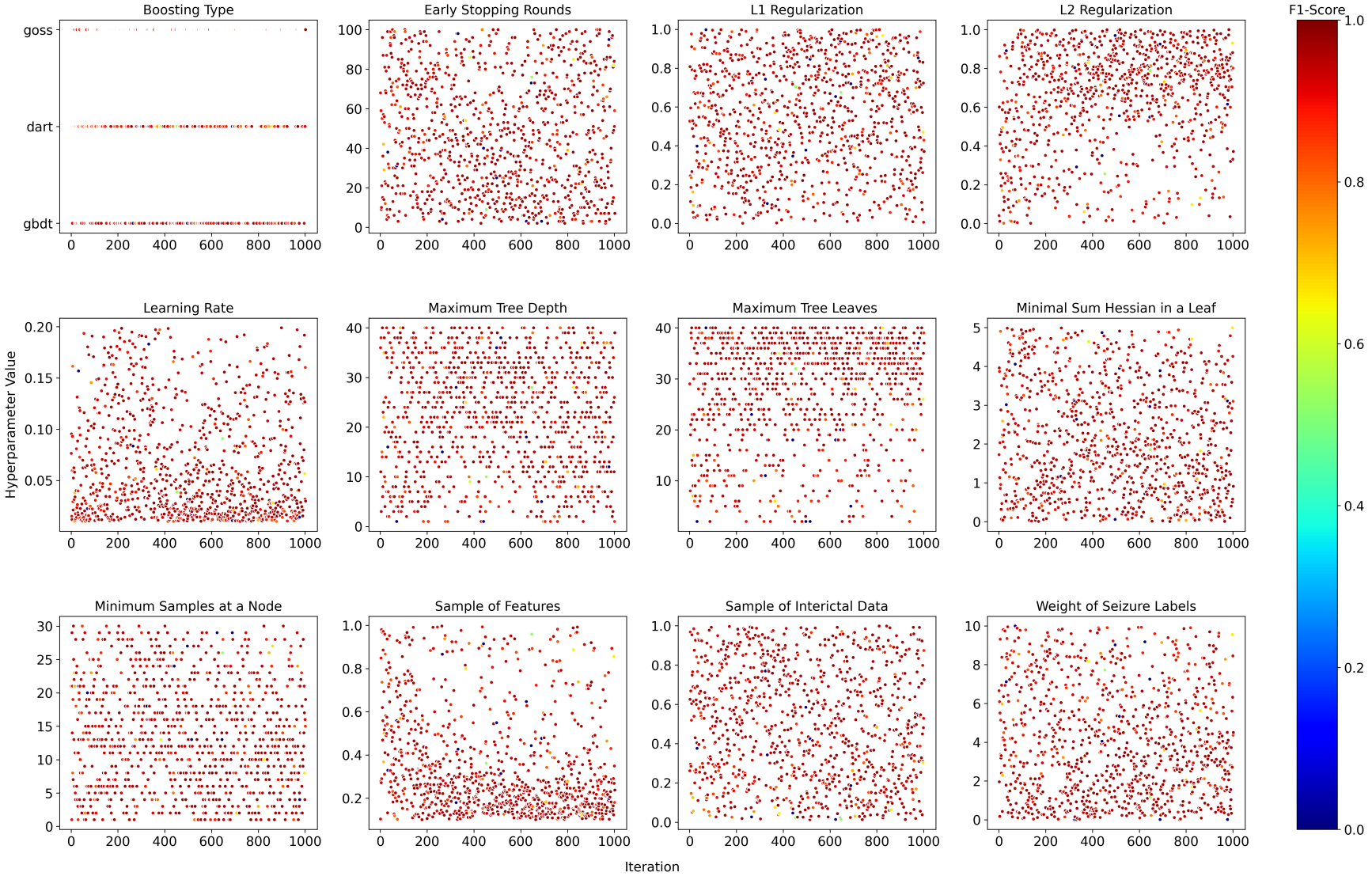
(a) Fold 1


Boosting Type
dart
gbdt
goss
Early Stopping Rounds
L1 Regularization
L2 Regularization
F1-Score
Learning Rate
Maximum Tree Depth
Maximum Tree Leaves
Minimal Sum Hessian in a Leaf
Hyperparameter Value
Minimum Samples at a Node
Sample of Features
Sample of Interictal Data
Weight of Seizure Labels
Iteration

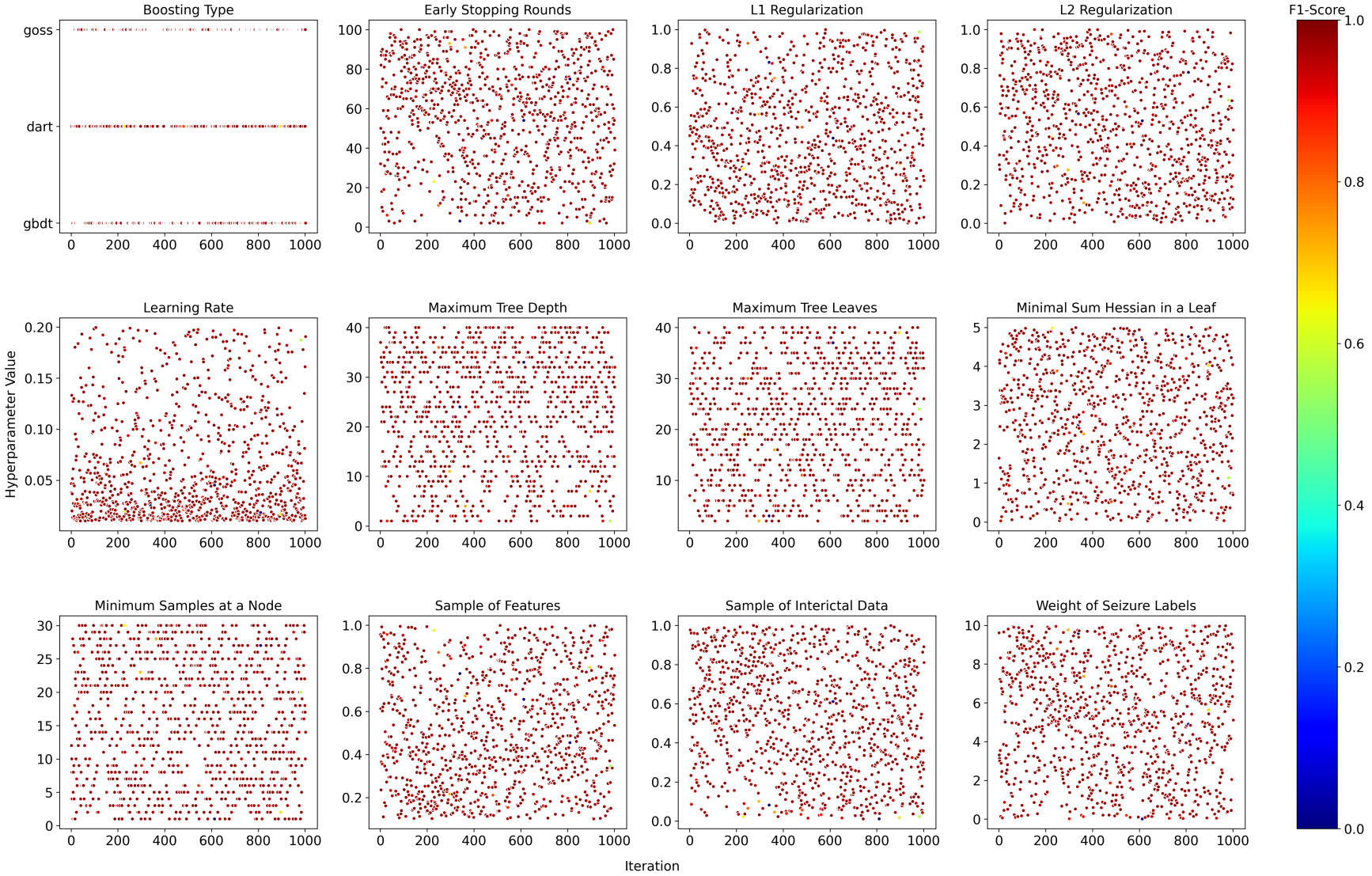
(b) Fold 2


Boosting Type
goss
gbdt
dart
Early Stopping Rounds
L1 Regularization
L2 Regularization
F1-Score
Learning Rate
Maximum Tree Depth
Maximum Tree Leaves
Minimal Sum Hessian in a Leaf
Hyperparameter Value
Minimum Samples at a Node
Sample of Features
Sample of Interictal Data
Weight of Seizure Labels
Iteration

(c) Fold 3

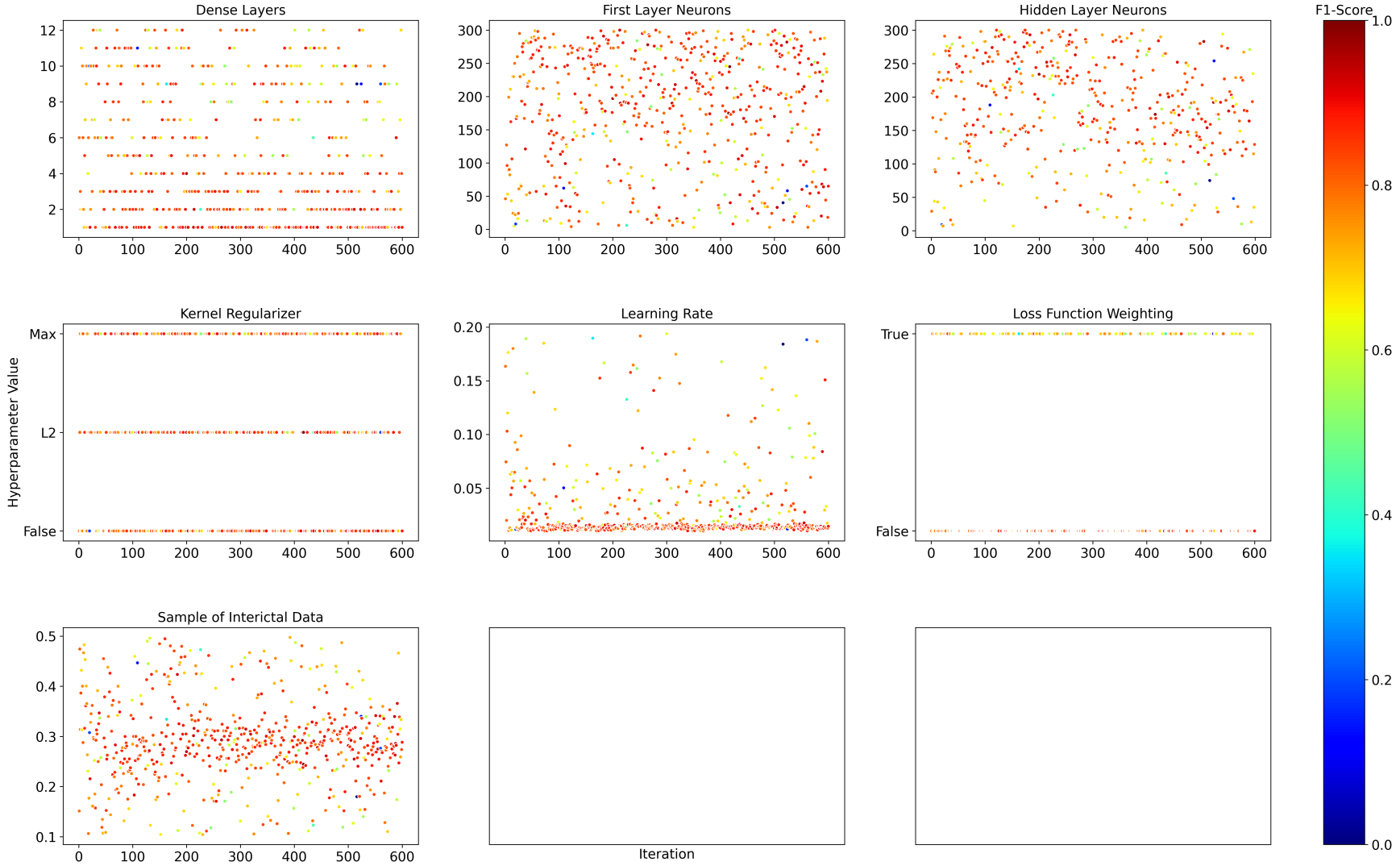

Boosting Type
Early Stopping Rounds
L1 Regularization
L2 Regularization
F1-Score
goss
dart
gbdt
Learning Rate
Maximum Tree Depth
Maximum Tree Leaves
Minimal Sum Hessian in a Leaf
Hyperparameter Value
Minimum Samples at a Node
Sample of Features
Sample of Interictal Data
Weight of Seizure Labels
Iteration

(d) Fold 4

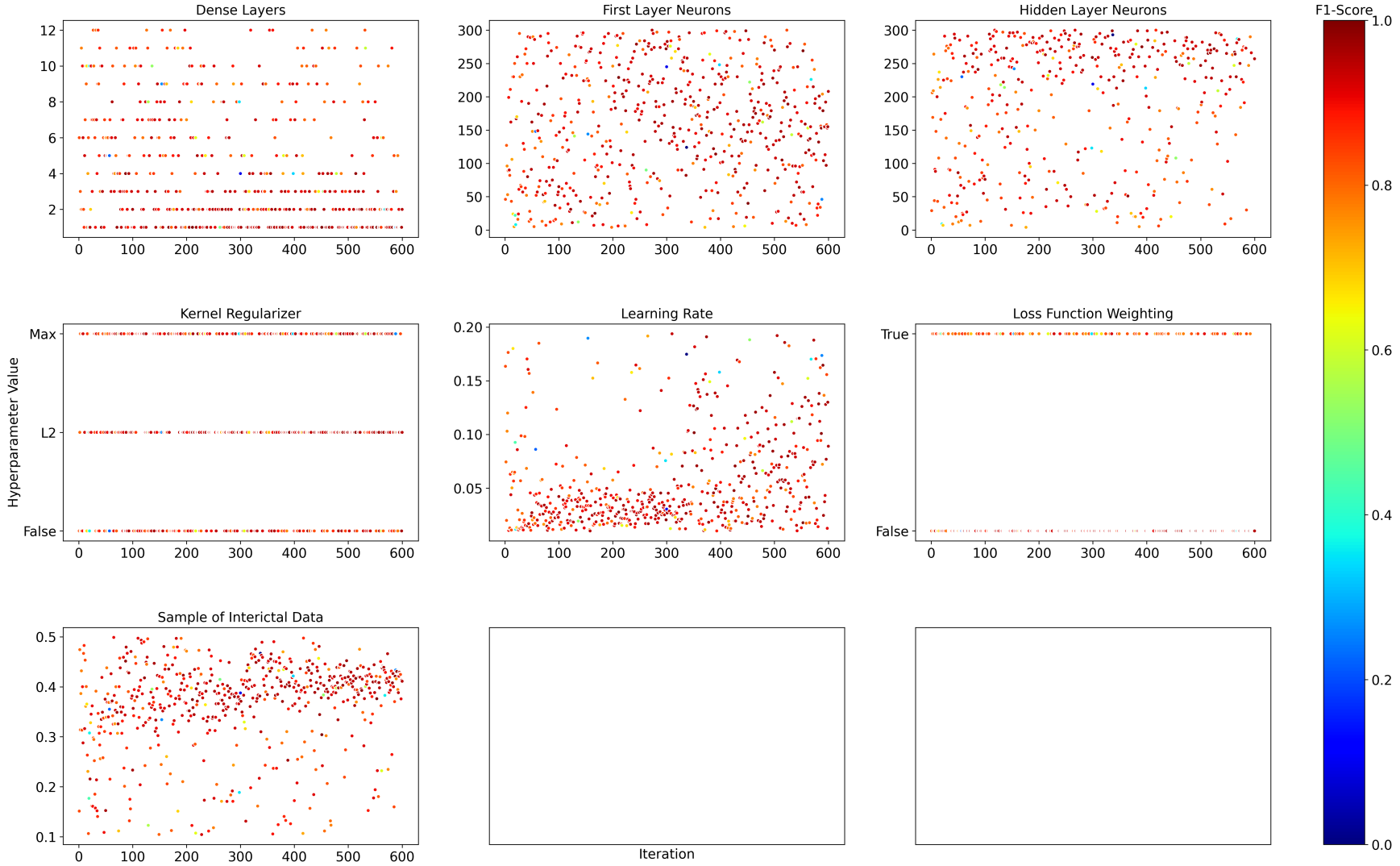


(e) Fold 5

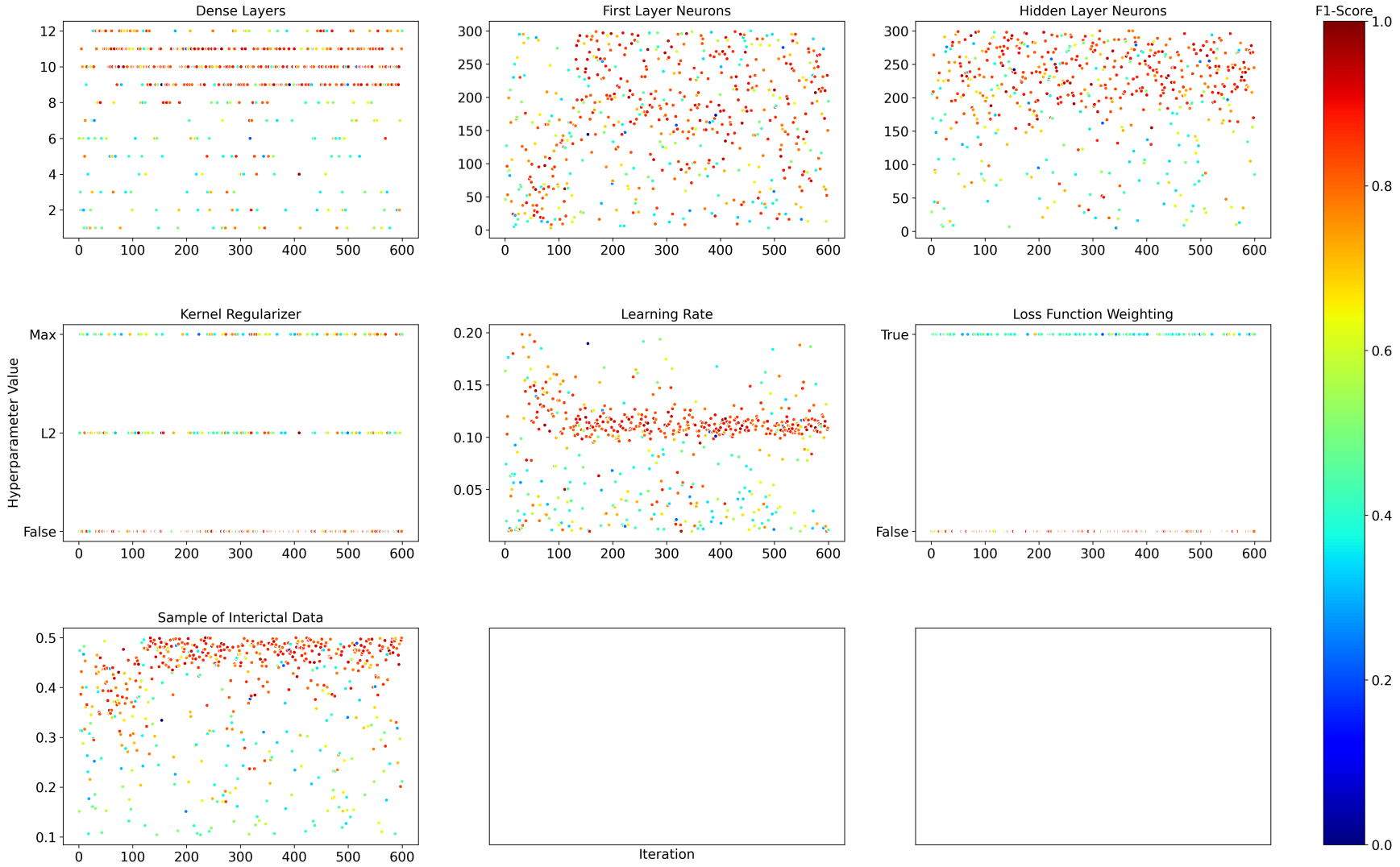
Figure 5.A.3: LightGBM hyperparameter values, and F1-scores on the validation set, during model training on TUH (Absence) records.


Dense Layers
First Layer Neurons
Hidden Layer Neurons
F1-Score
Kernel Regularizer
Learning Rate
Loss Function Weighting
Hyperparameter Value
Max
L2
False
True
Sample of Interictal Data
Iteration

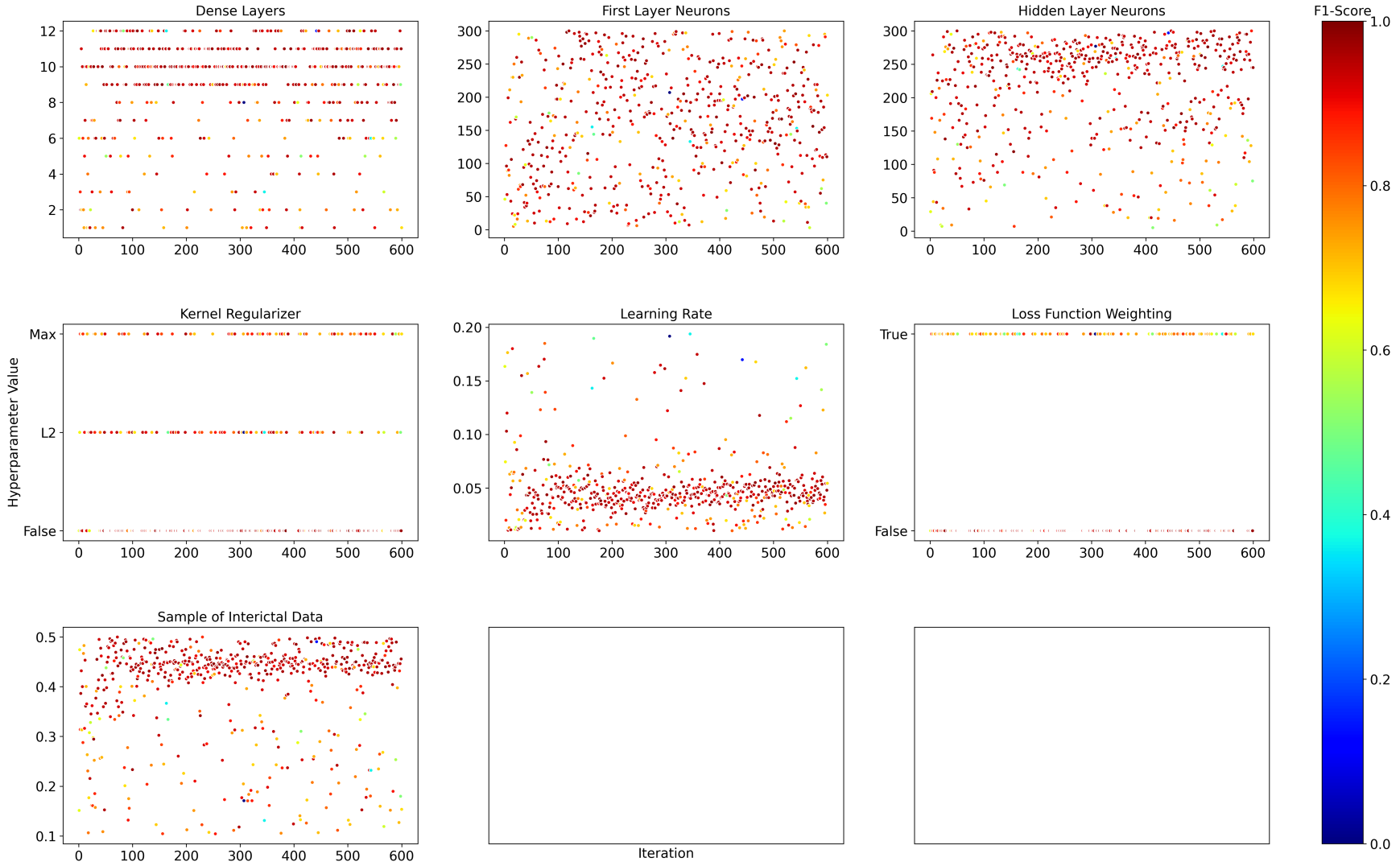
(a) Fold 1


Dense Layers
First Layer Neurons
Hidden Layer Neurons
F1-Score
Kernel Regularizer
Learning Rate
Loss Function Weighting
Hyperparameter Value
Max
L2
False
True
Sample of Interictal Data
Iteration

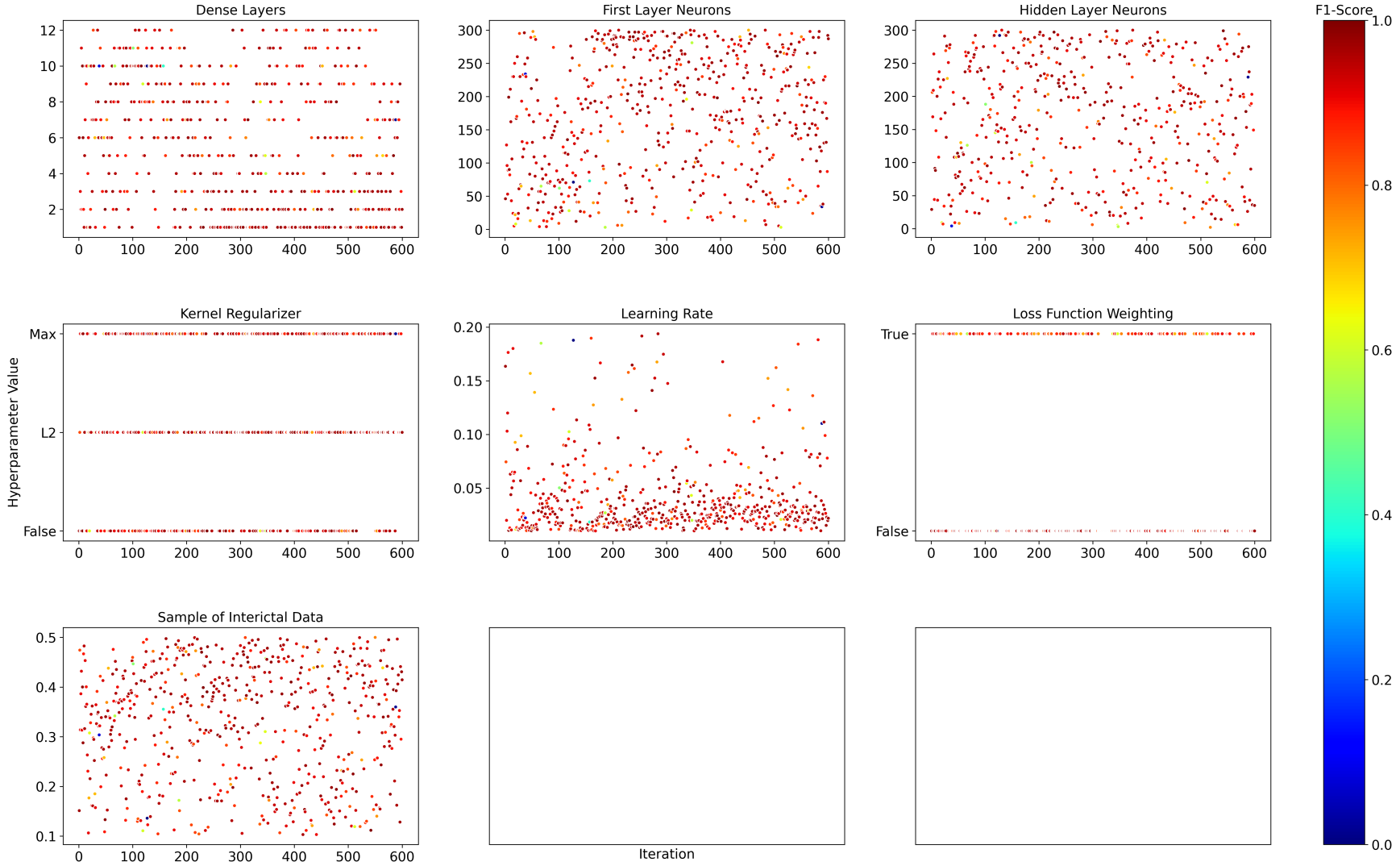
(b) Fold 2


Dense Layers
First Layer Neurons
Hidden Layer Neurons
F1-Score
Kernel Regularizer
Learning Rate
Loss Function Weighting
Hyperparameter Value
Max
L2
False
True
Sample of Interictal Data
Iteration

(c) Fold 3

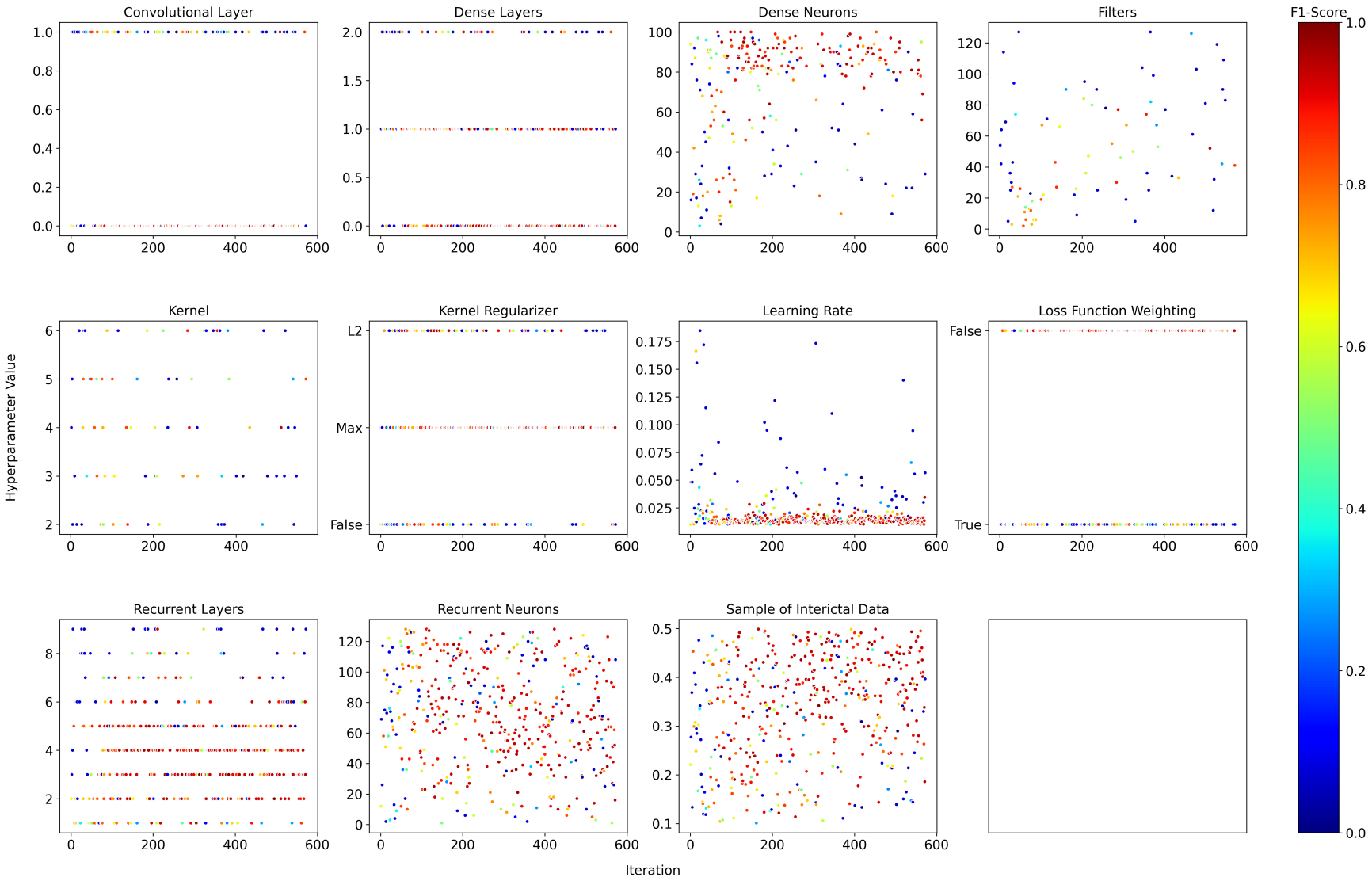

Dense Layers
First Layer Neurons
Hidden Layer Neurons
F1-Score
Kernel Regularizer
Learning Rate
Loss Function Weighting
Hyperparameter Value
Max
L2
False
True
Sample of Interictal Data
Iteration

(d) Fold 4

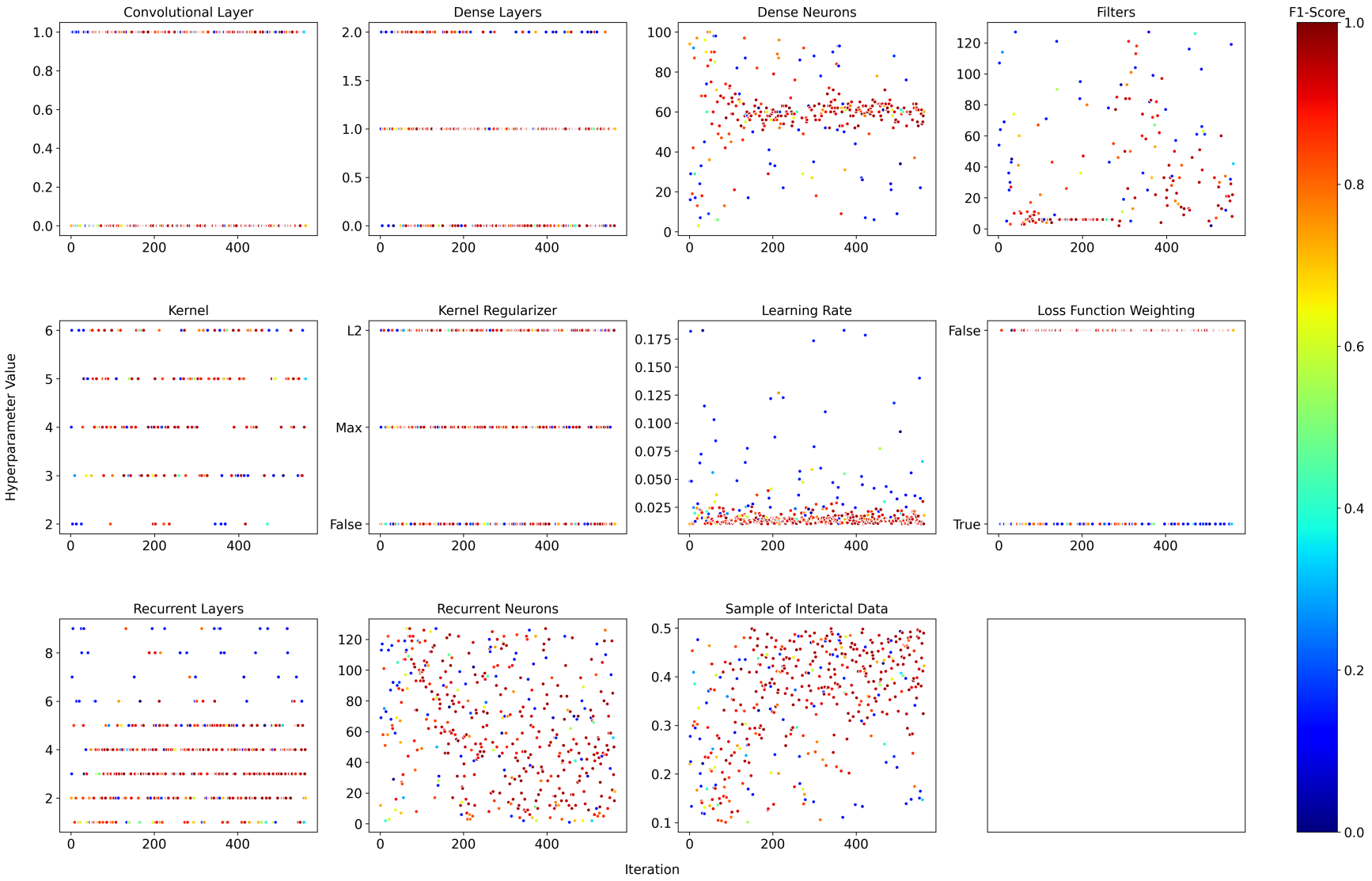


(e) Fold 5

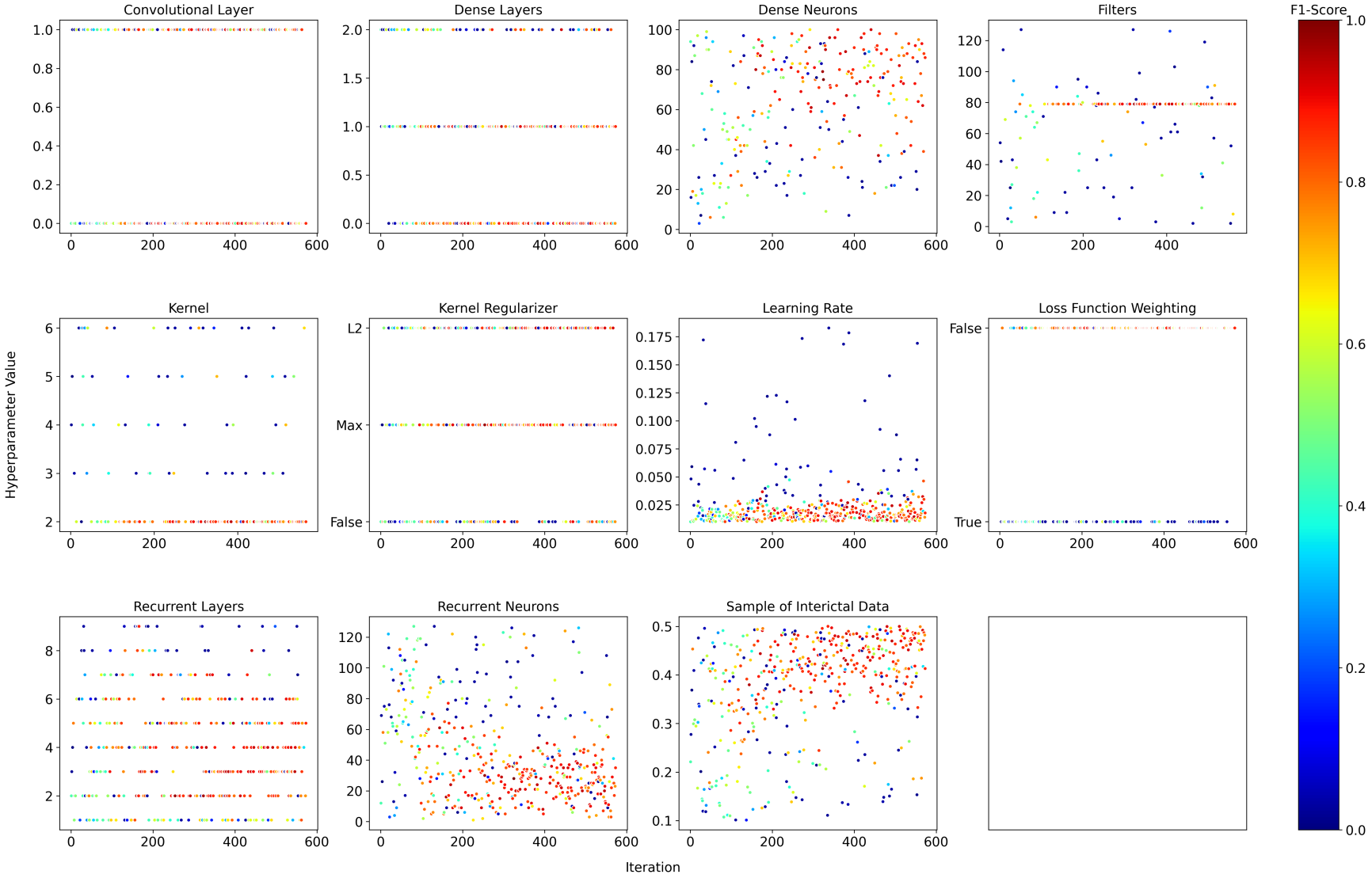
Figure 5.A.4: MLP hyperparameter values, and F1-scores on the validation set, during model training on TUH (Absence) records.


Convolutional Layer
Dense Layers
Dense Neurons
Filters
F1-Score
Kernel
Kernel Regularizer
Learning Rate
Loss Function Weighting
Recurrent Layers
Recurrent Neurons
Sample of Interictal Data
Hyperparameter Value
Iteration
L2
Max
False
True

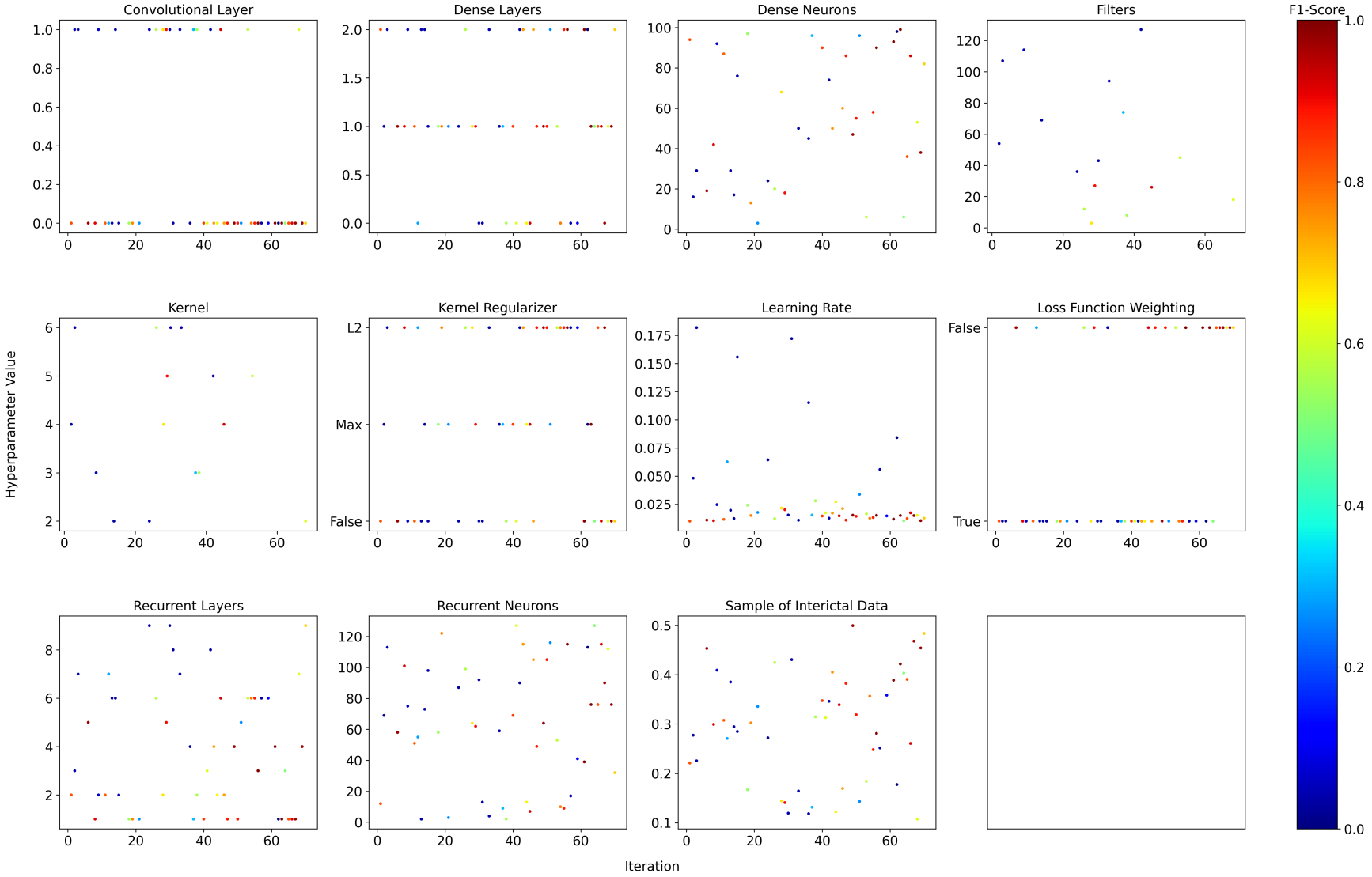
(a) Fold 1


Convolutional Layer
Dense Layers
Dense Neurons
Filters
F1-Score
Kernel
Kernel Regularizer
Learning Rate
Loss Function Weighting
Recurrent Layers
Recurrent Neurons
Sample of Interictal Data
Hyperparameter Value
Iteration
L2
Max
False
True

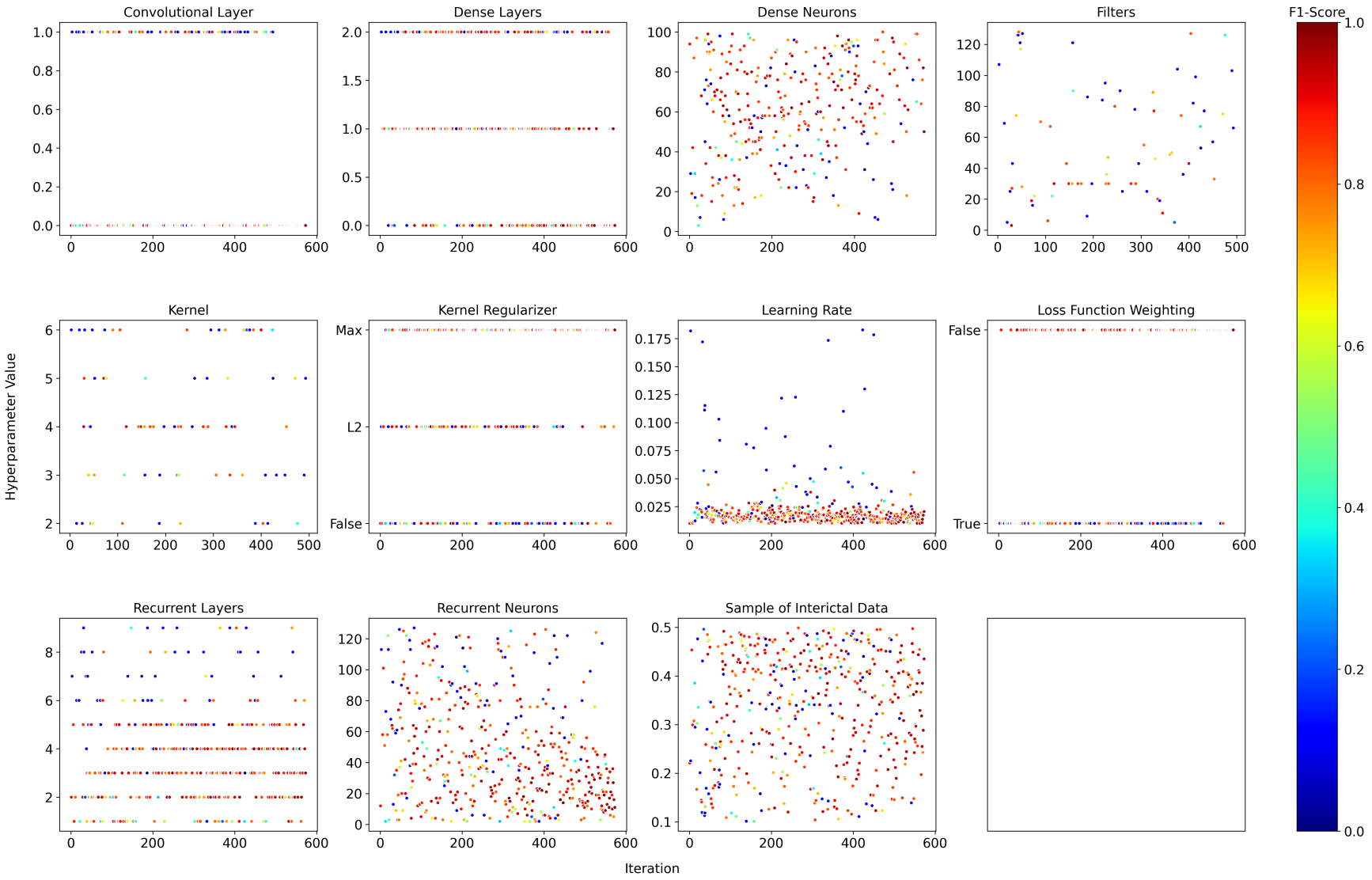
(b) Fold 2


Convolutional Layer
Dense Layers
Dense Neurons
Filters
F1-Score
Kernel
Kernel Regularizer
Learning Rate
Loss Function Weighting
Recurrent Layers
Recurrent Neurons
Sample of Interictal Data
Hyperparameter Value
Iteration
L2
Max
False
True

(c) Fold 3

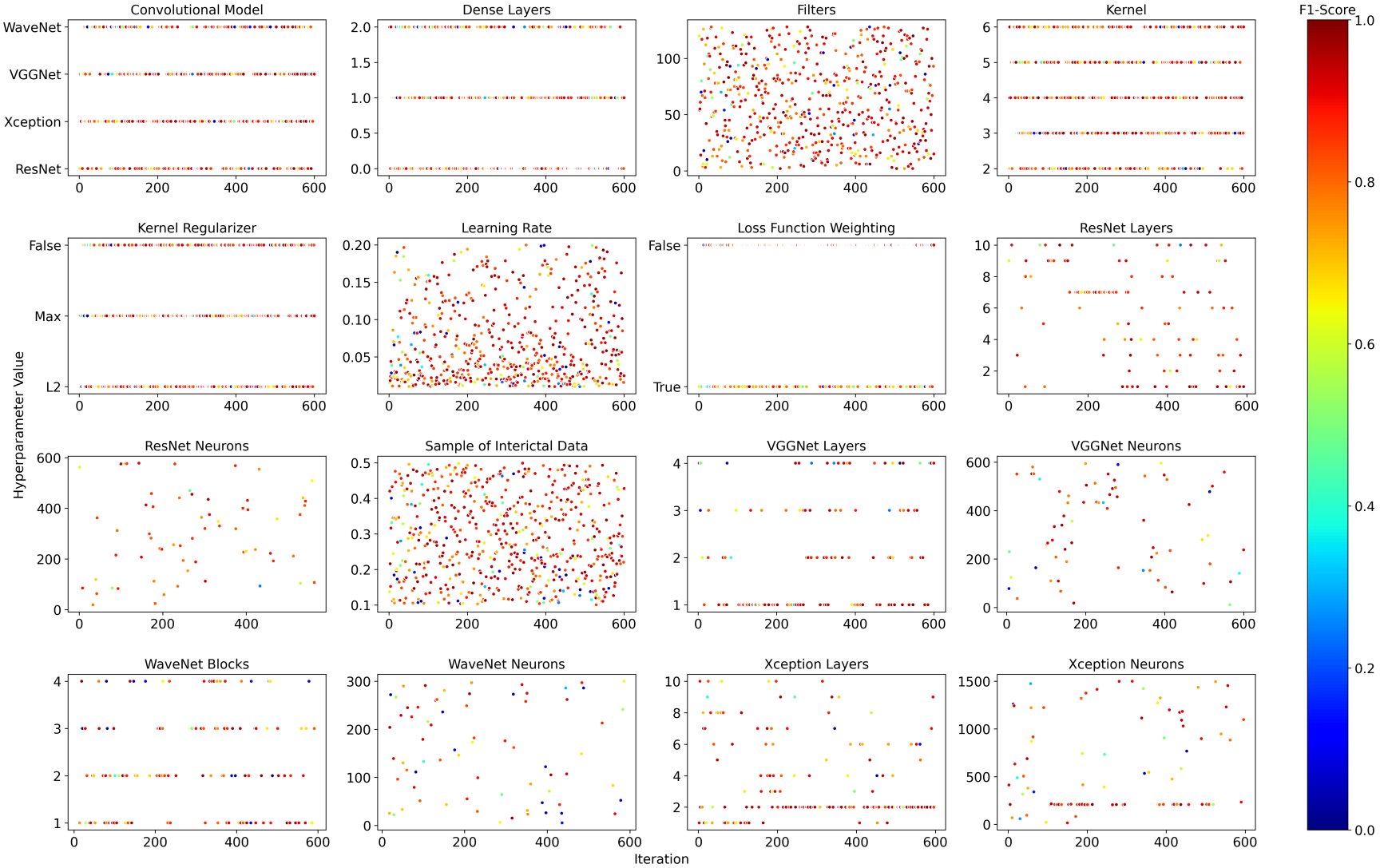

Convolutional Layer
Dense Layers
Dense Neurons
Filters
F1-Score
Kernel
Kernel Regularizer
Learning Rate
Loss Function Weighting
Recurrent Layers
Recurrent Neurons
Sample of Interictal Data
Hyperparameter Value
Iteration
L2
Max
False
True

(d) Fold 4

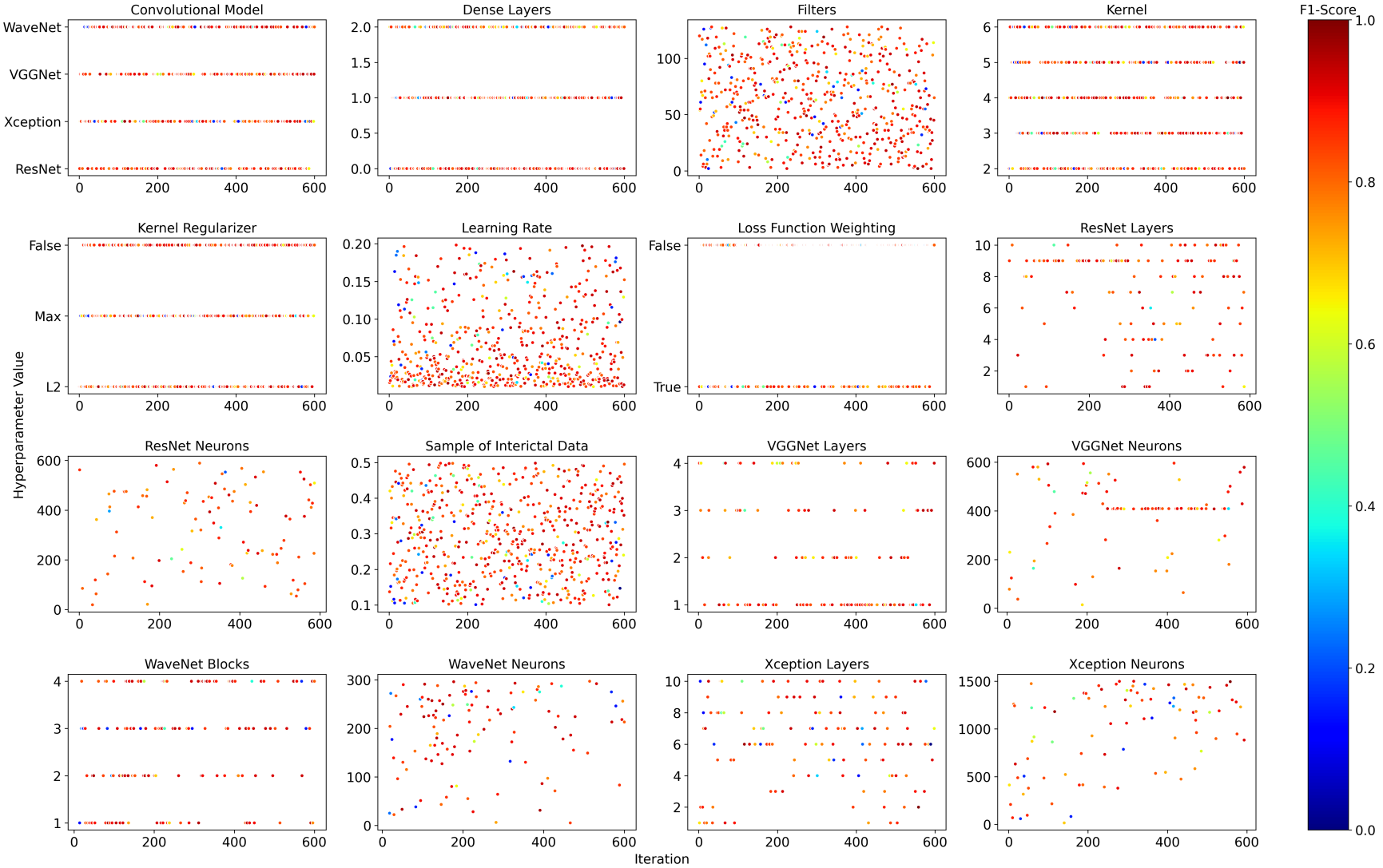


(e) Fold 5

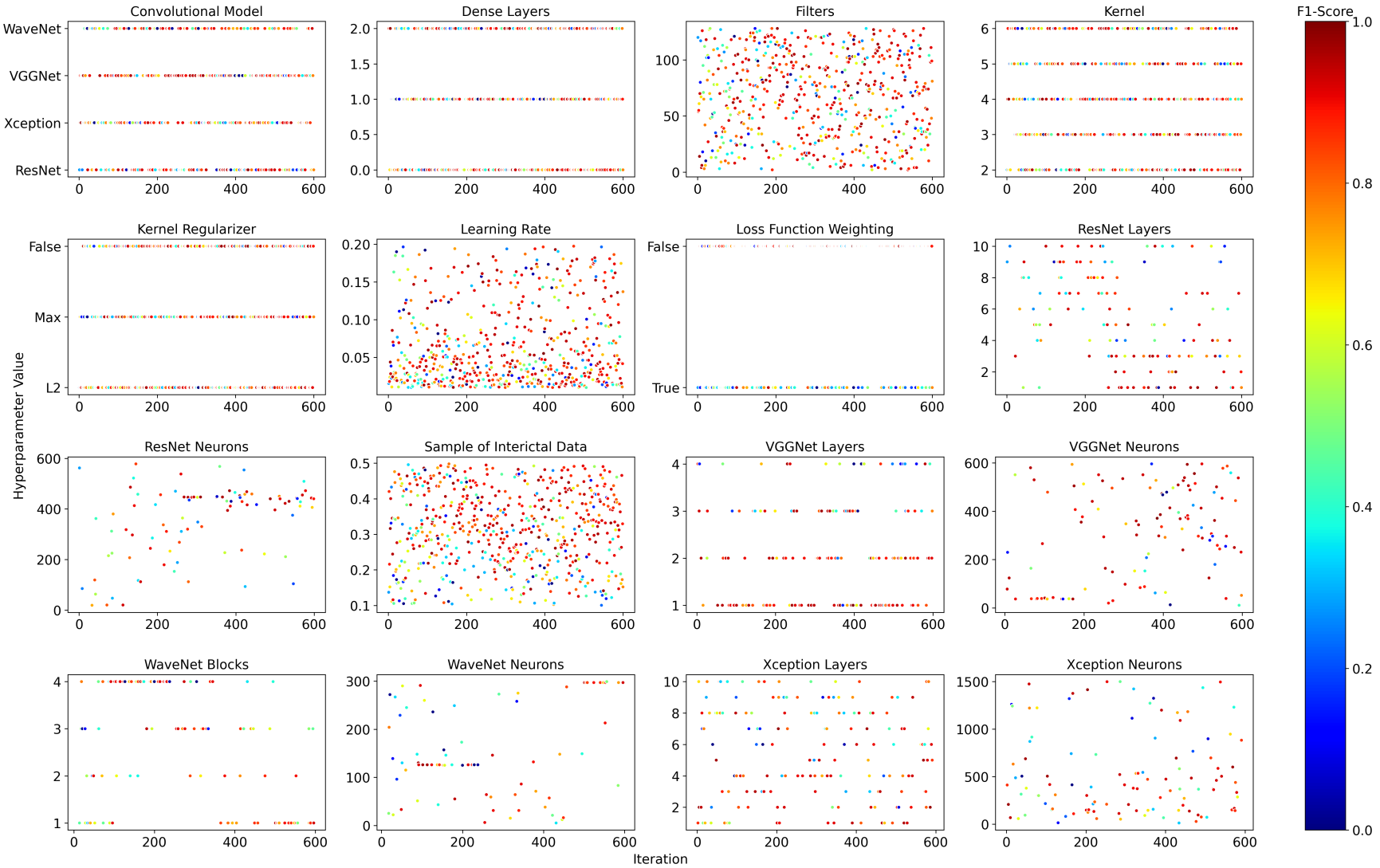
Figure 5.A.5: RNN hyperparameter values, and F1-scores on the validation set, during model training on TUH (Absence) records.


Convolutional Model
Dense Layers
Filters
Kernel
F1-Score
WaveNet
VGGNet
Xception
ResNet
Kernel Regularizer
Learning Rate
Loss Function Weighting
ResNet Layers
False
Max
L2
True
ResNet Neurons
Sample of Interictal Data
VGGNet Layers
VGGNet Neurons
WaveNet Blocks
WaveNet Neurons
Xception Layers
Xception Neurons
Hyperparameter Value
Iteration

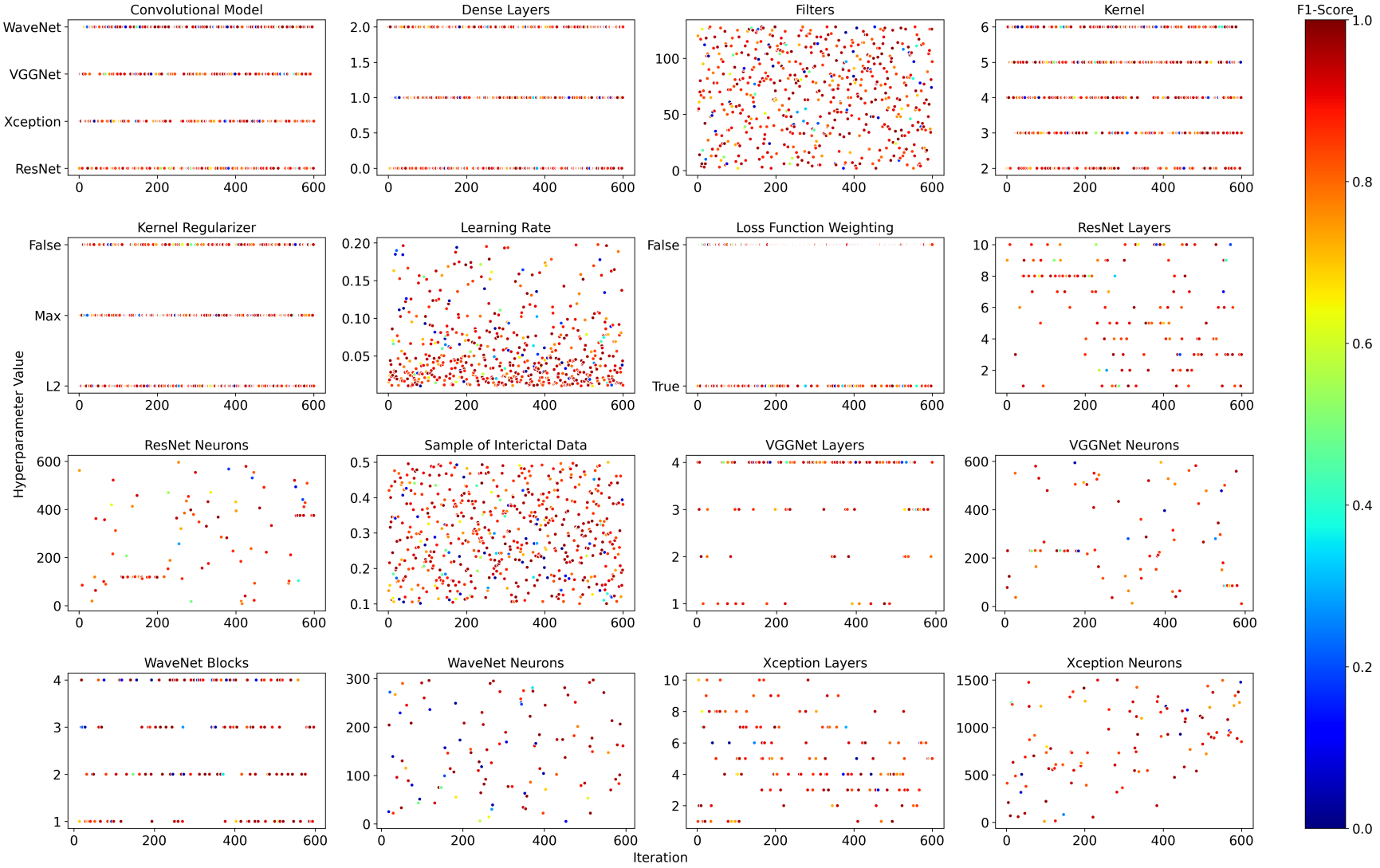
(a) Fold 1


Convolutional Model
Dense Layers
Filters
Kernel
F1-Score
WaveNet
VGGNet
Xception
ResNet
Kernel Regularizer
Learning Rate
Loss Function Weighting
ResNet Layers
False
Max
L2
True
ResNet Neurons
Sample of Interictal Data
VGGNet Layers
VGGNet Neurons
WaveNet Blocks
WaveNet Neurons
Xception Layers
Xception Neurons
Hyperparameter Value
Iteration

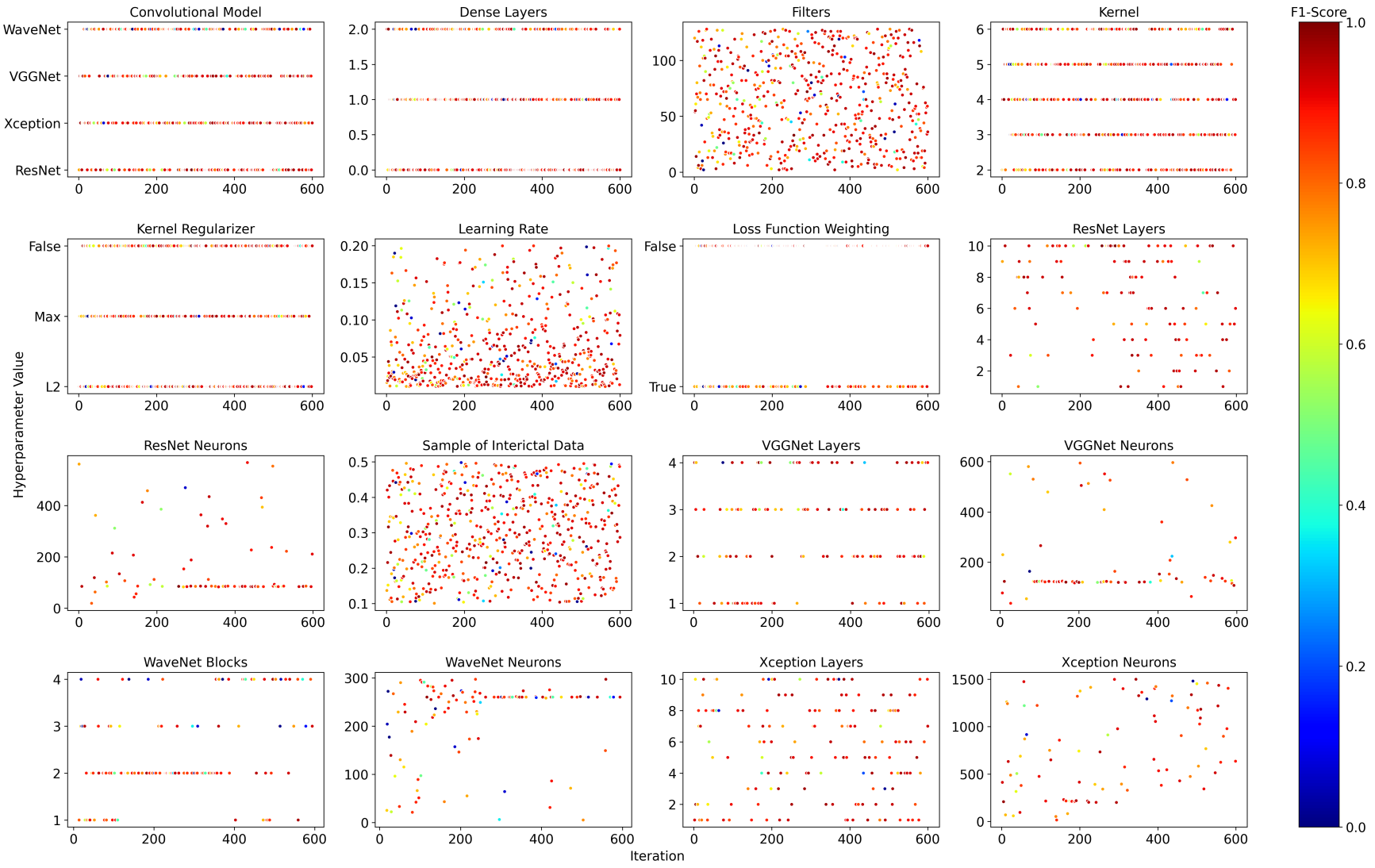
(b) Fold 2


Convolutional Model
Dense Layers
Filters
Kernel
F1-Score
WaveNet
VGGNet
Xception
ResNet
Kernel Regularizer
Learning Rate
Loss Function Weighting
ResNet Layers
False
Max
L2
True
Hyperparameter Value
ResNet Neurons
Sample of Interictal Data
VGGNet Layers
VGGNet Neurons
WaveNet Blocks
WaveNet Neurons
Xception Layers
Xception Neurons
Iteration

(c) Fold 3

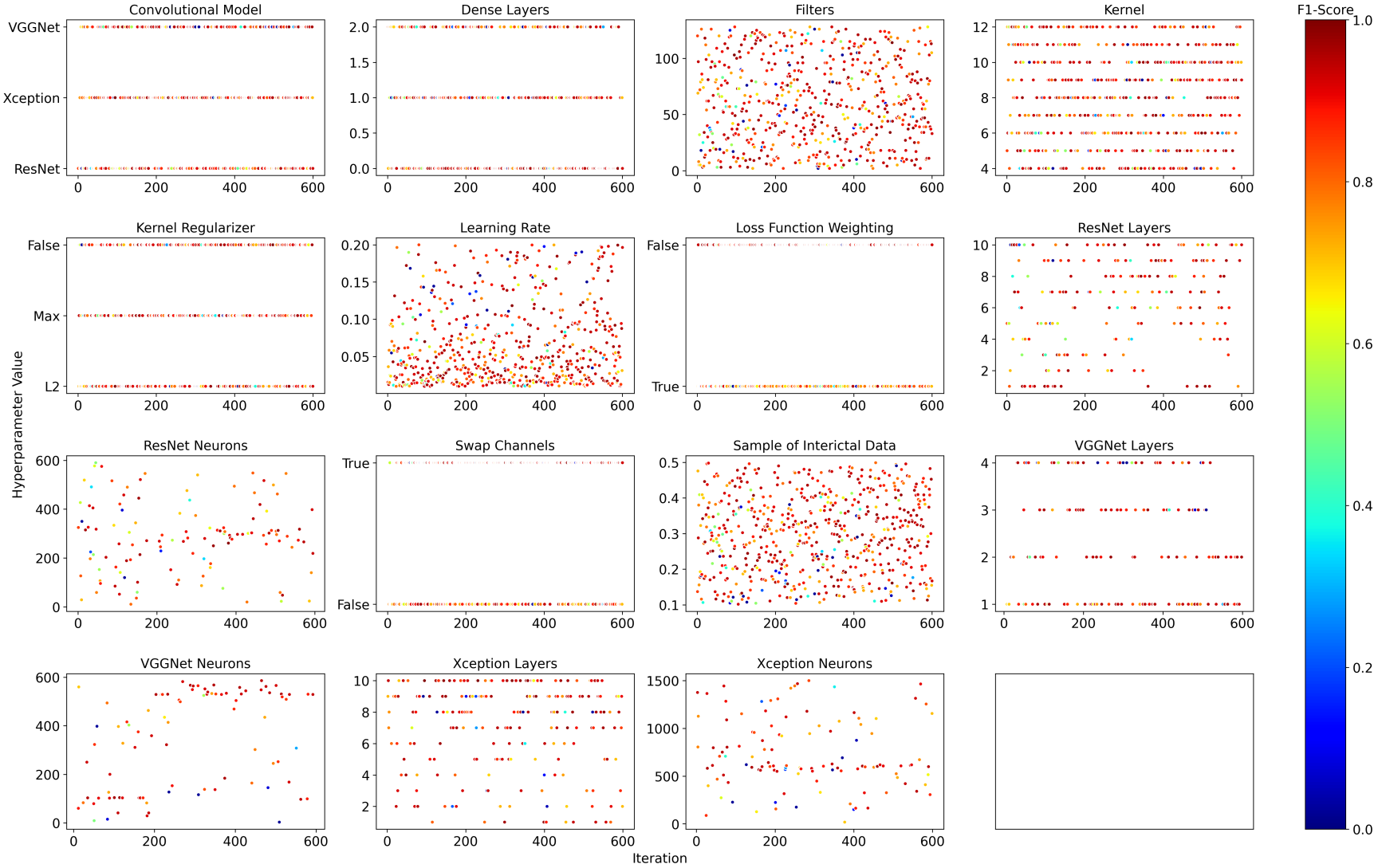

Convolutional Model
Dense Layers
Filters
Kernel
F1-Score
WaveNet
VGGNet
Xception
ResNet
Kernel Regularizer
Learning Rate
Loss Function Weighting
ResNet Layers
False
Max
L2
True
ResNet Neurons
Sample of Interictal Data
VGGNet Layers
VGGNet Neurons
WaveNet Blocks
WaveNet Neurons
Xception Layers
Xception Neurons
Hyperparameter Value
Iteration

(d) Fold 4

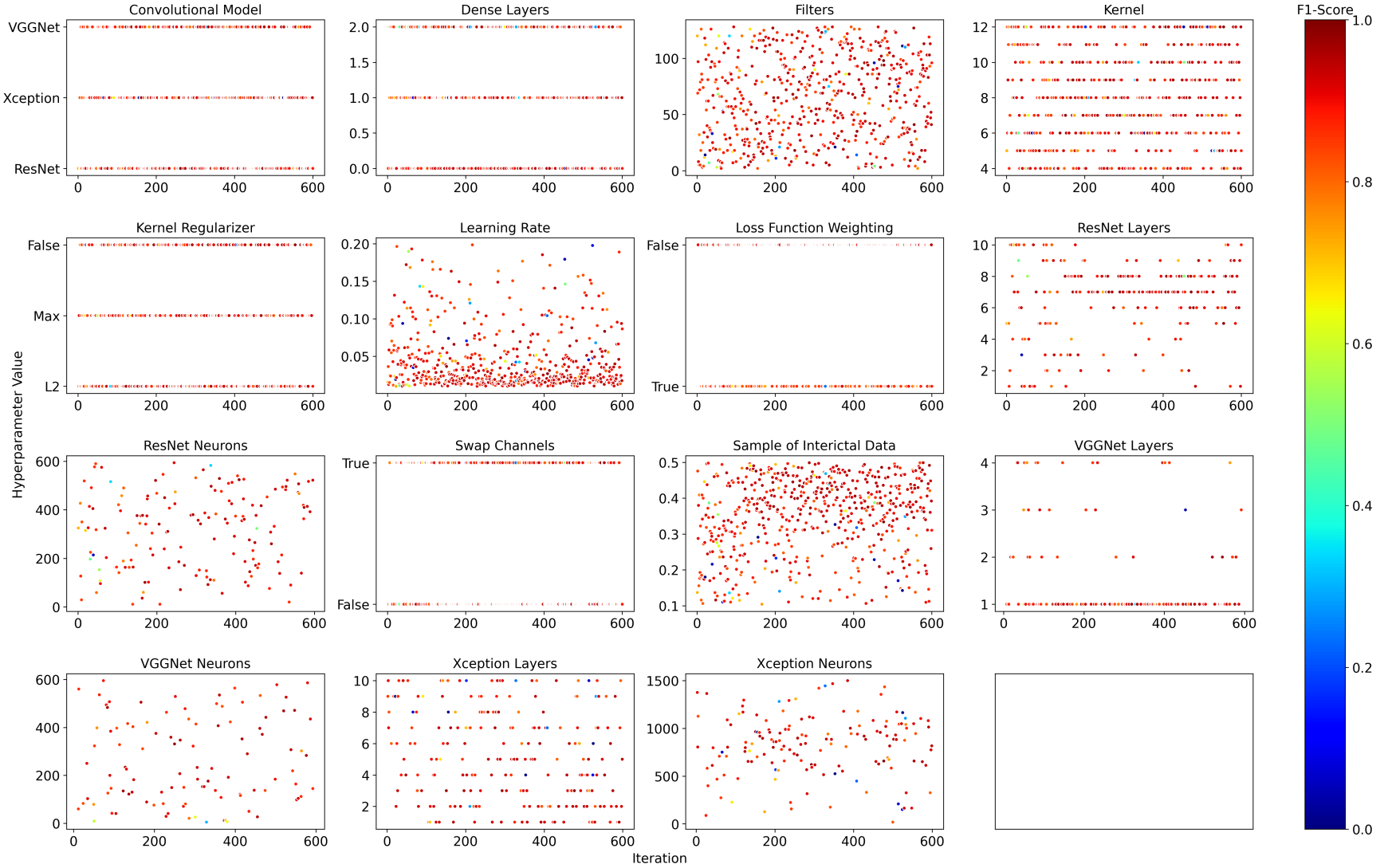


(e) Fold 5

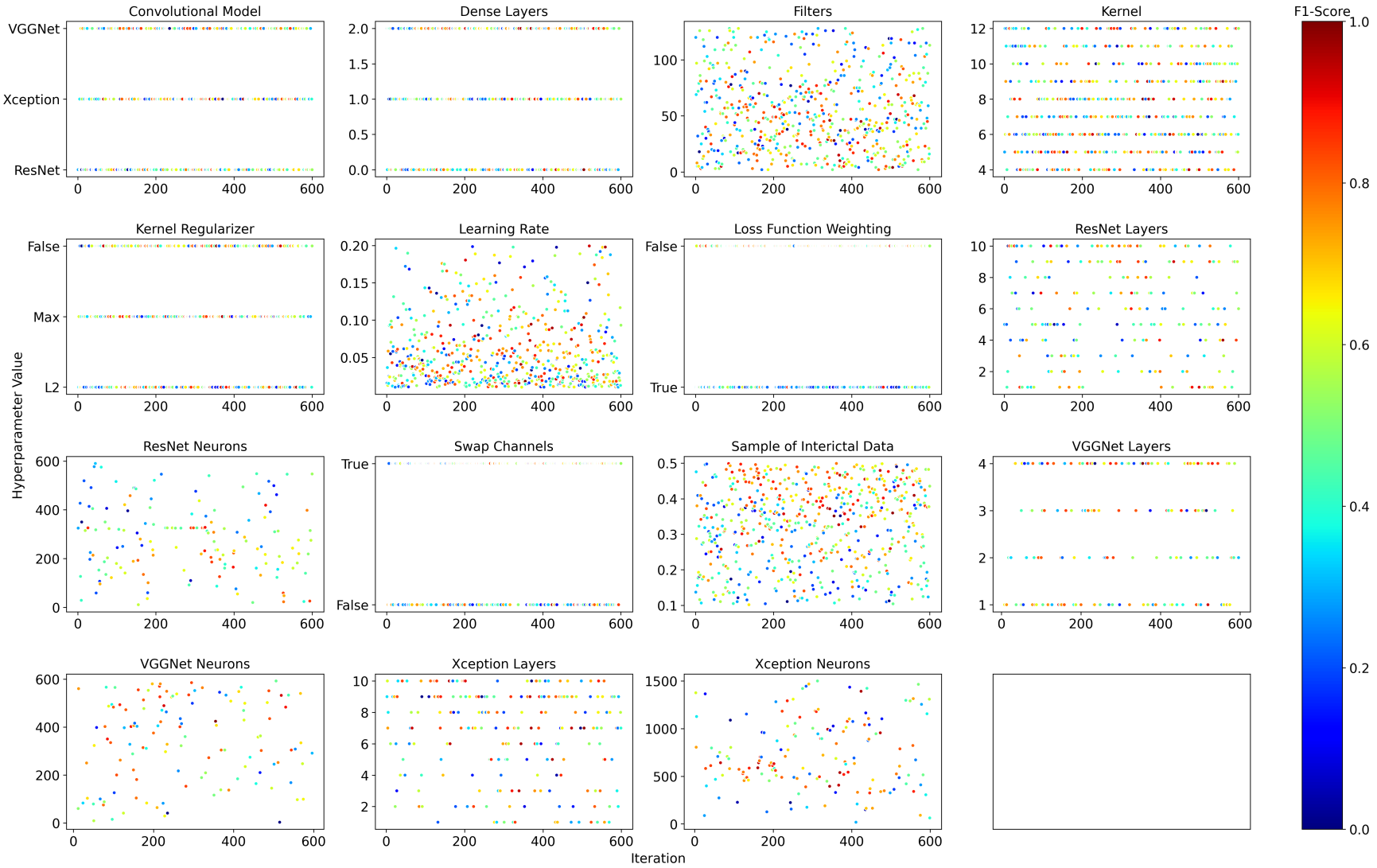
Figure 5.A.6: CNN1D hyperparameter values, and F1-scores on the validation set, during model training on TUH (Absence) records.


Convolutional Model
Dense Layers
Filters
Kernel
F1-Score
VGGNet
Xception
ResNet
Kernel Regularizer
Learning Rate
Loss Function Weighting
ResNet Layers
False
Max
L2
True
ResNet Neurons
Swap Channels
Sample of Interictal Data
VGGNet Layers
VGGNet Neurons
Xception Layers
Xception Neurons
Hyperparameter Value
Iteration

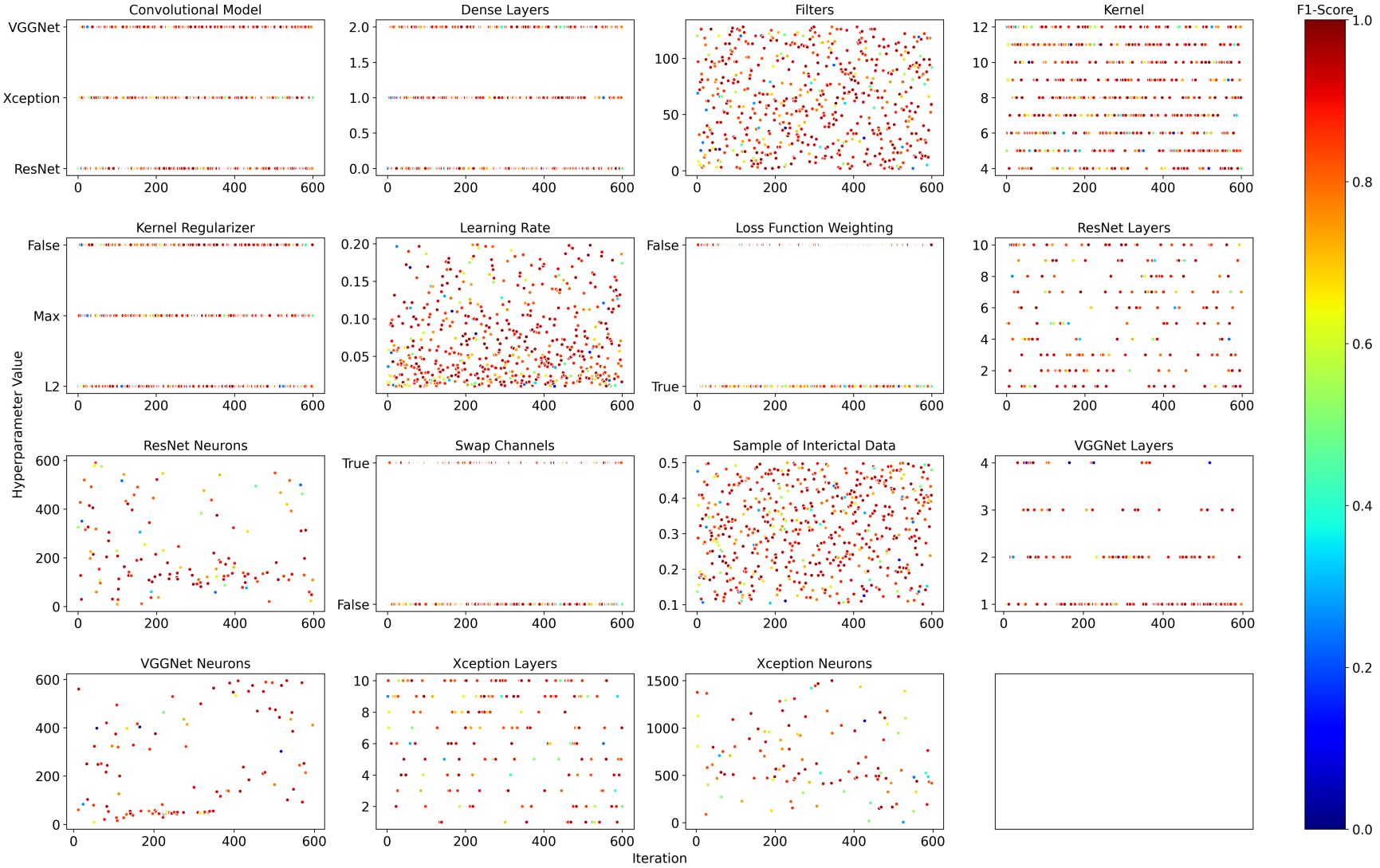
(a) Fold 1


Convolutional Model
Dense Layers
Filters
Kernel
F1-Score
VGGNet
Xception
ResNet
Kernel Regularizer
Learning Rate
Loss Function Weighting
ResNet Layers
False
Max
L2
True
ResNet Neurons
Swap Channels
Sample of Interictal Data
VGGNet Layers
VGGNet Neurons
Xception Layers
Xception Neurons
Hyperparameter Value
Iteration

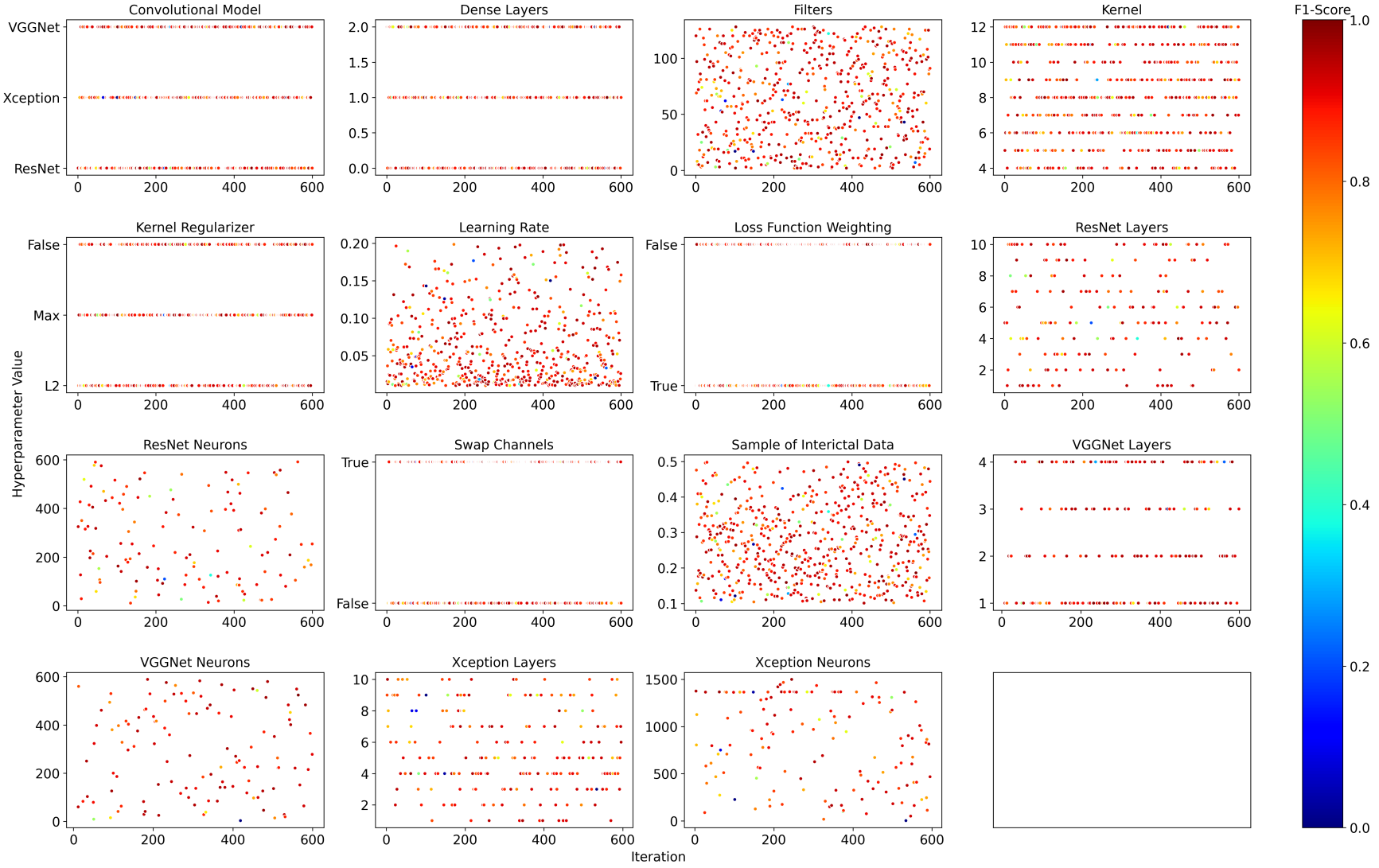
(b) Fold 2


Convolutional Model
Dense Layers
Filters
Kernel
F1-Score
VGGNet
Xception
ResNet
Kernel Regularizer
Learning Rate
Loss Function Weighting
ResNet Layers
False
Max
L2
True
ResNet Neurons
Swap Channels
Sample of Interictal Data
VGGNet Layers
VGGNet Neurons
Xception Layers
Xception Neurons
Hyperparameter Value
Iteration

(c) Fold 3


Convolutional Model
Dense Layers
Filters
Kernel
F1-Score
Kernel Regularizer
Learning Rate
Loss Function Weighting
ResNet Layers
ResNet Neurons
Swap Channels
Sample of Interictal Data
VGGNet Layers
VGGNet Neurons
Xception Layers
Xception Neurons
Hyperparameter Value
Iteration
VGGNet
Xception
ResNet
False
Max
L2
True

(d) Fold 4



(e) Fold 5

Figure 5.A.7: CNN2D hyperparameter values, and F1-scores on the validation set, during model training on TUH (Absence) records.

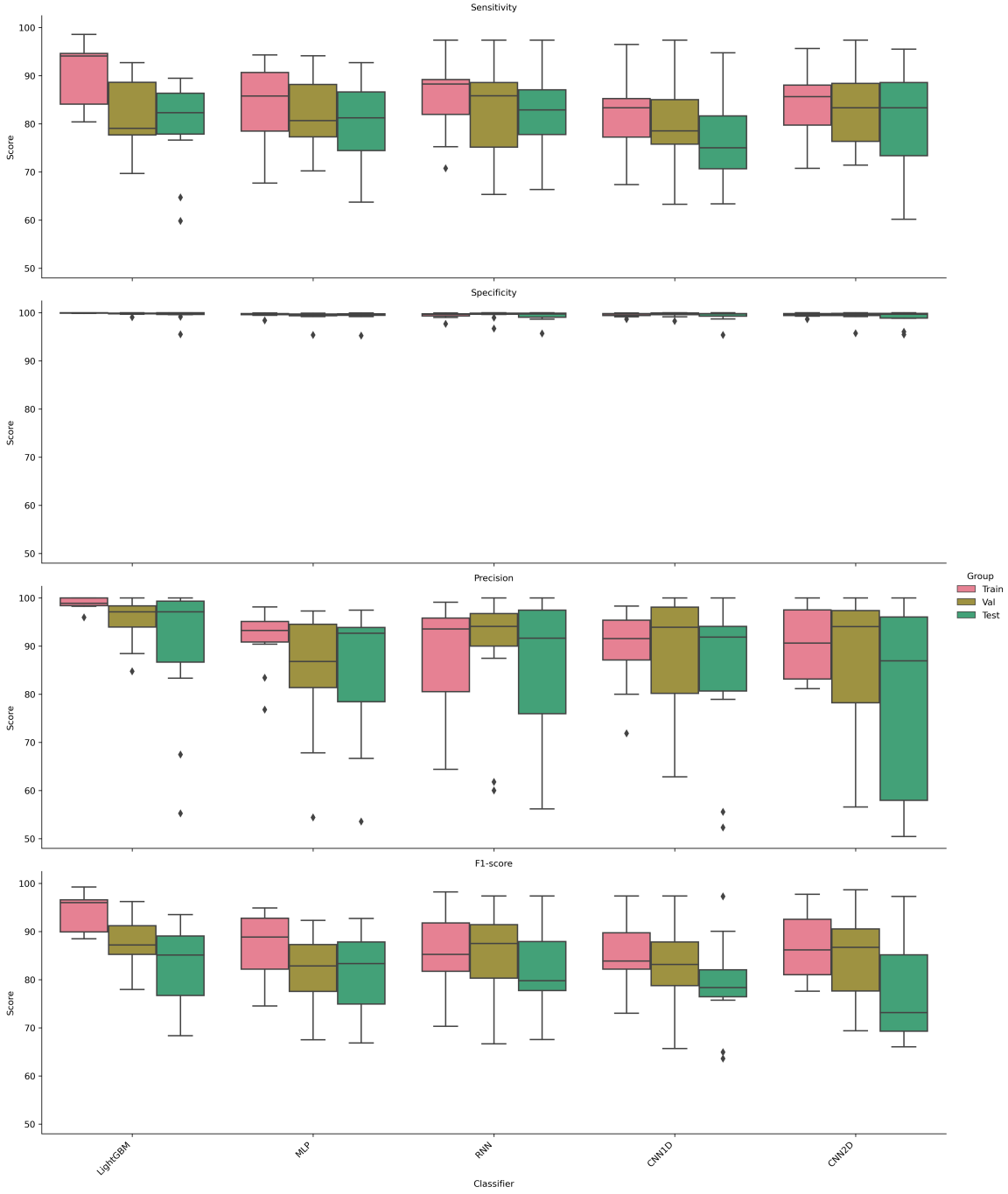


Figure 5.A.8: Boxplots to show the performance of optimal TUH (Absence) model configurations across all folds on each data split type.

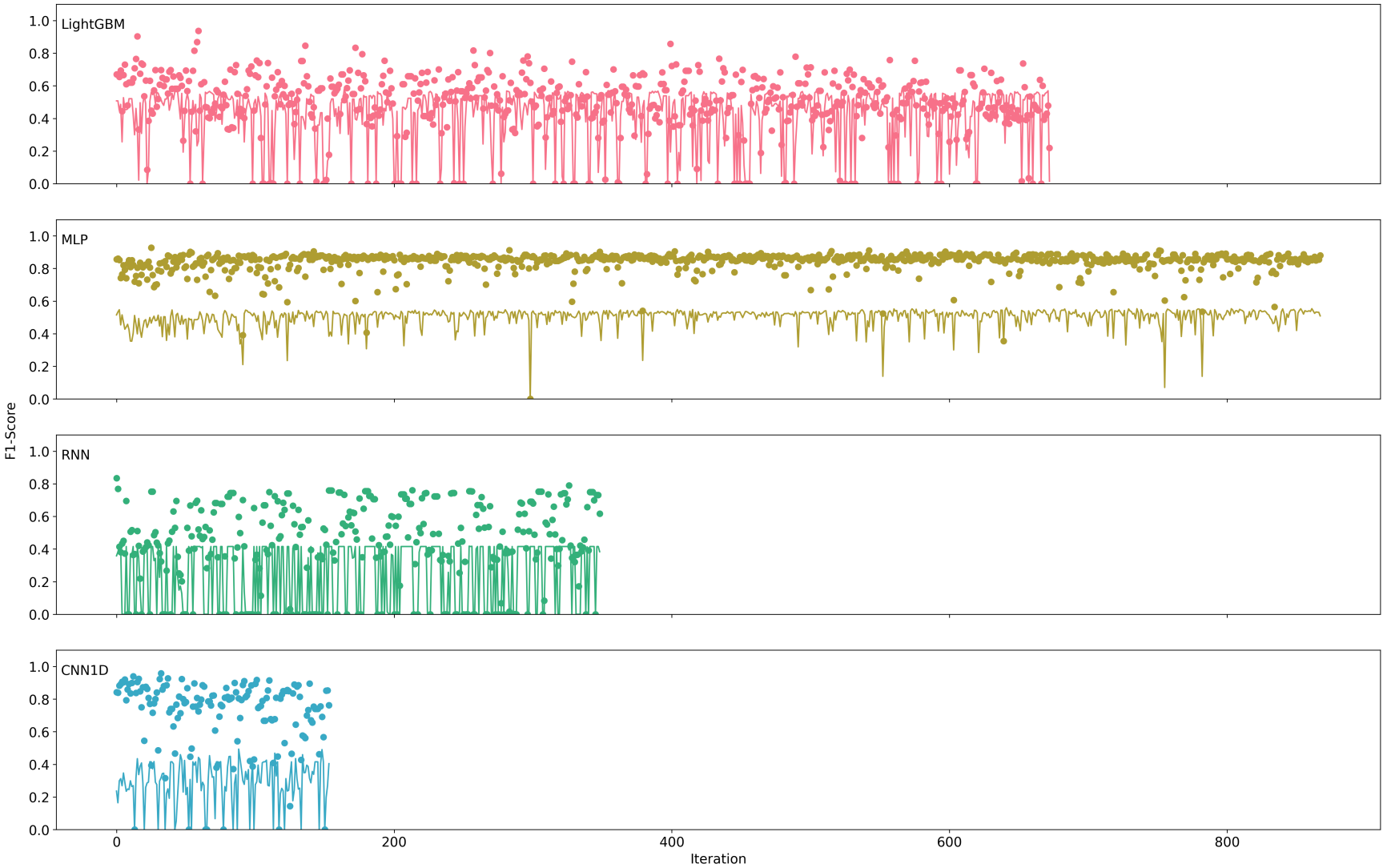
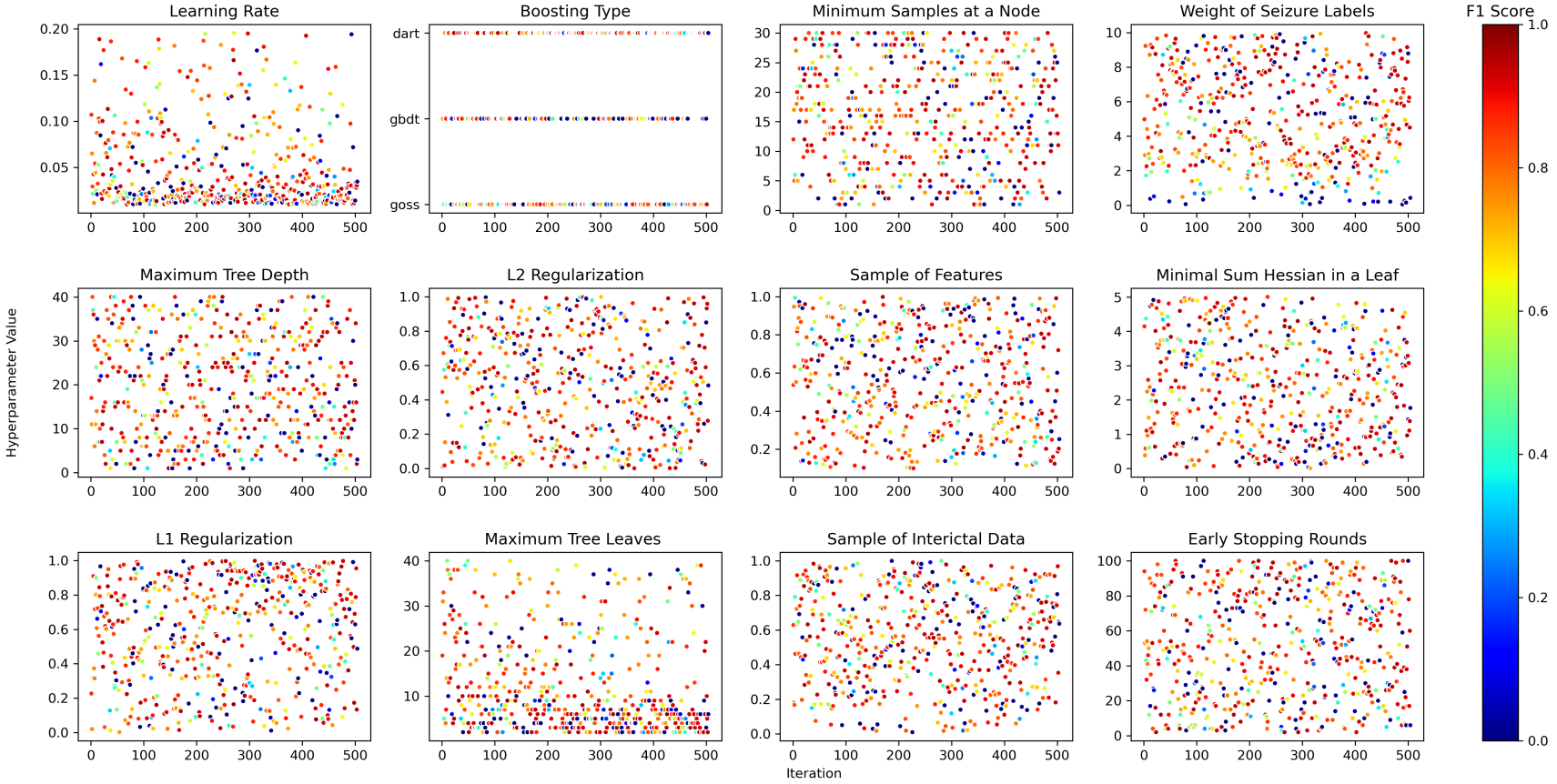
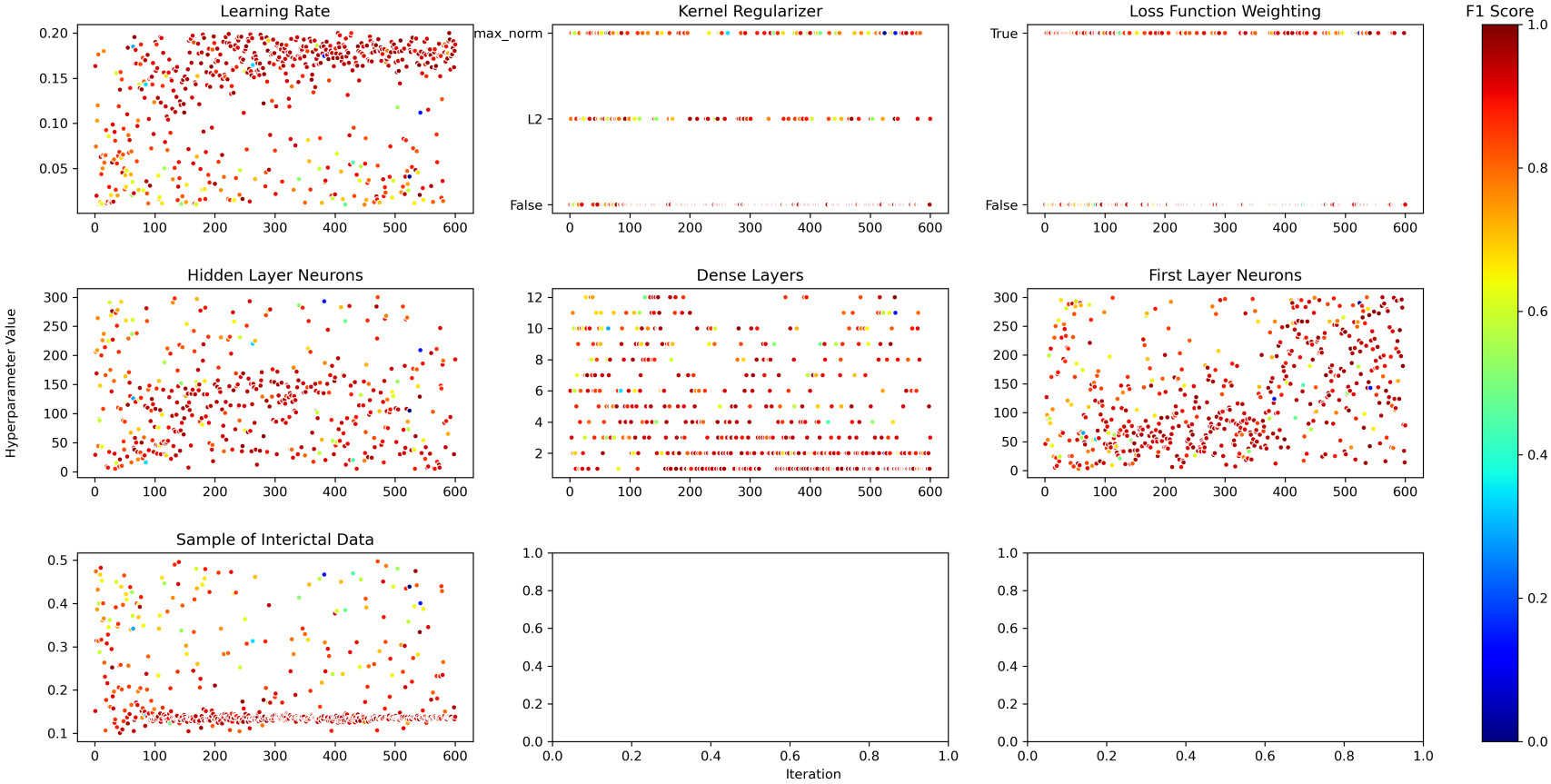


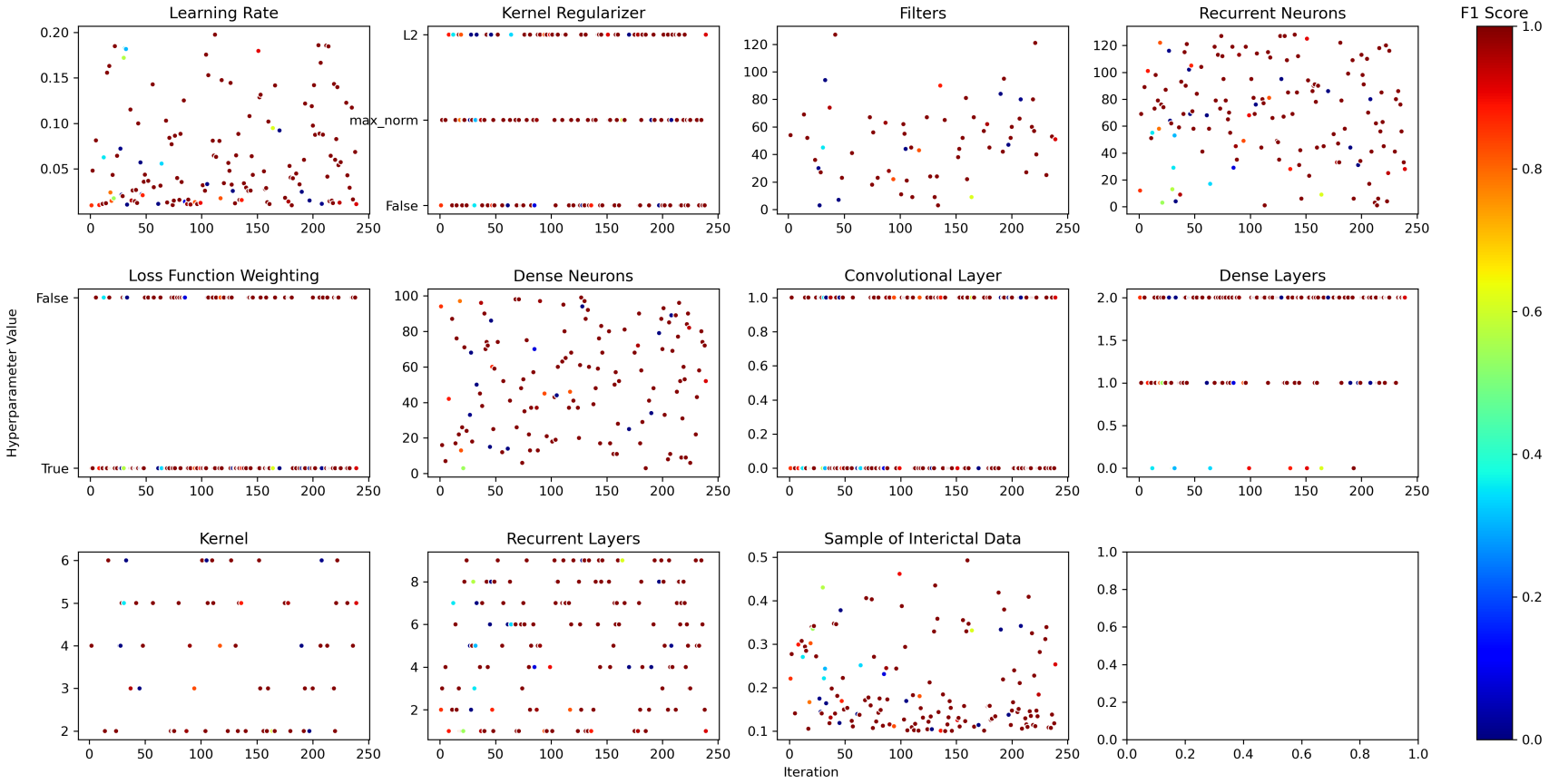
Figure 5.A.9: F1-scores during BOHB optimisation for models trained on TUH (Generalised) records.
Note. Dots: Training scores; Line: Validation scores


Learning Rate
Boosting Type
Minimum Samples at a Node
Weight of Seizure Labels
F1 Score
dart
gbdt
goss
Maximum Tree Depth
L2 Regularization
Sample of Features
Minimal Sum Hessian in a Leaf
L1 Regularization
Maximum Tree Leaves
Sample of Interictal Data
Early Stopping Rounds
Hyperparameter Value
Iteration

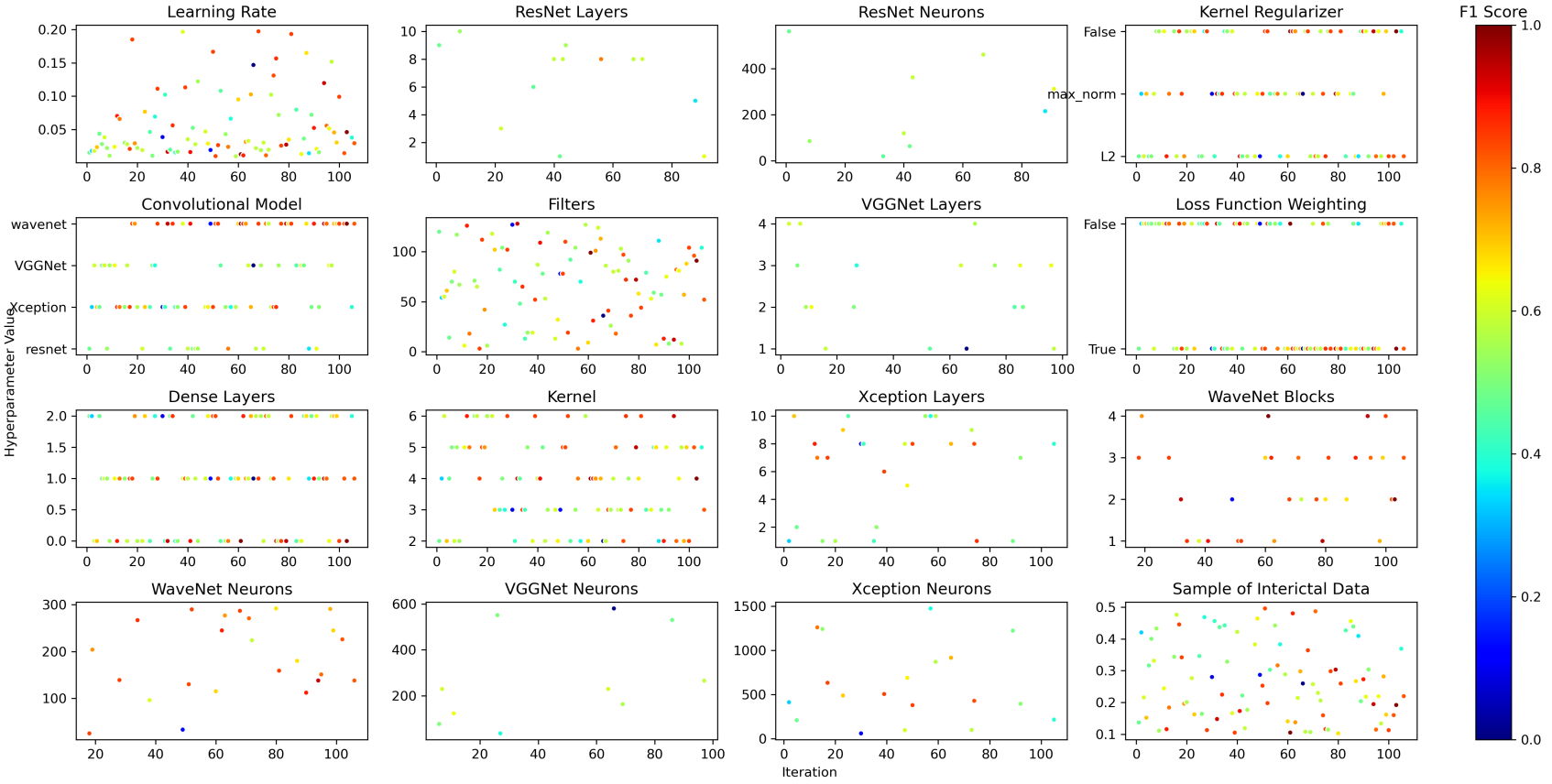
(a) LightGBM


Learning Rate
Kernel Regularizer
Loss Function Weighting
F1 Score
Hidden Layer Neurons
Dense Layers
First Layer Neurons
Sample of Interictal Data
Hyperparameter Value
Iteration
max_norm
L2
False
True

(b) MLP


Learning Rate
Kernel Regularizer
Filters
Recurrent Neurons
F1 Score
Loss Function Weighting
Dense Neurons
Convolutional Layer
Dense Layers
Kernel
Recurrent Layers
Sample of Interictal Data
Hyperparameter Value
Iteration
L2
max_norm
False
True

(c) RNN



(d) CNN1D

Figure 5.A.10: Hyperparameter values, and F1-scores on the validation set, during model training on TUH (Generalised) records.

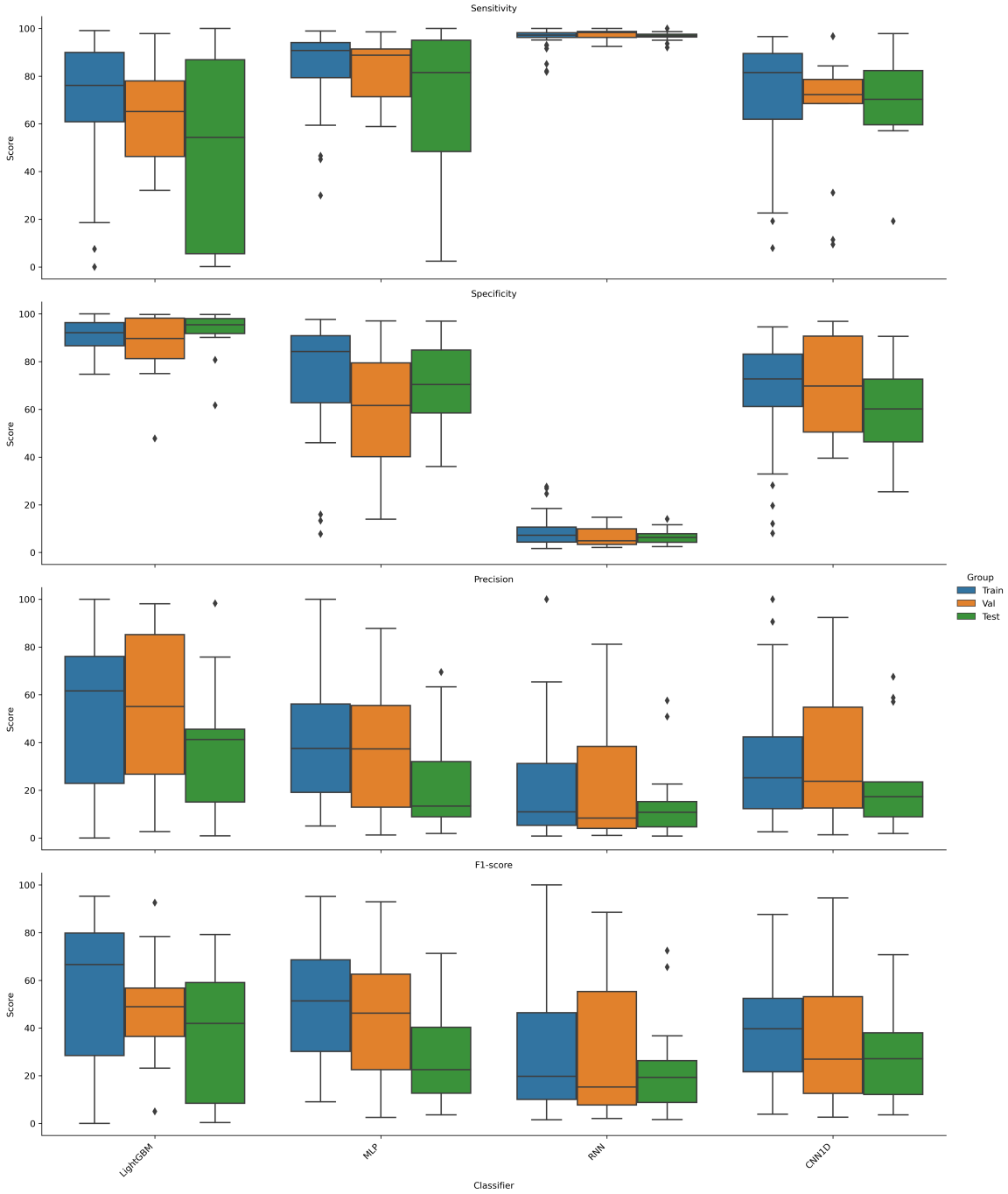


Figure 5.A.11: Boxplots to show the the performance of optimal TUH (Generalised) model configurations across all folds on each data split type.

Chapter 6

Recommendations and Conclusions

This thesis has examined a range of signal features and classification pipelines for the optimal detection of generalized seizures, focusing particularly on generalized absence seizures. Chapter 2 provided a broad overview of the possible components of a classification pipeline, demonstrating the large search space over which classification pipeline components and hyperparameters could be selected from. This also emphasised the current lack of consensus in the literature over optimal configurations, therefore chapters 3, 4, and 5 used a systematic approach to compare hundreds/thousands of potential model configurations, at a scale not seen in the current seizure detection literature. This provides a more complete representation of the potential search space for model design, and will aid in comparing results from published papers. Specifically, chapter 3 focused on the use of Bayesian optimisation to select optimal pipeline components and hyperparameters for the detection of absence seizures in a novel clinical dataset. This chapter is the first research to use Bayesian optimisation for absence epilepsy detection and to use NHS EEG patient records for automated epilepsy detection, and we find some of the best pipelines for detecting this type of seizure are comparable to the current literature. This work demonstrated the best pipeline stacked a Random Forest (RF) for feature selection followed by a k-Nearest Neighbours (KNN) algorithm for classification. The features that were selected by the RF reflected the prominent properties of absence seizures familiar to physiologists and clinicians, these being average alpha frequency amplitude in the frontal/central electrodes. This is important as future clinical adoption of an algorithm depends on its processing pipeline being explainable and justifi-

able (Vollmer et al., 2020) in order to meet recent legislative changes (e.g. the EU General Data Protection Regulation; European Parliament, 2016). In chapter 4, we built upon this work by assessing more complex balanced ensemble models on a broader range of absence patient data. Indeed, by training the models on more patient data from different NHS sites, we were able to improve model performance. The models chosen in this chapter also had improved training speeds, important as data size increases, and a lower false positive rate than those found in chapter 3. However, the observed higher specificity and precision was at the expense of sensitivity; likely due to each balanced classifier in the ensemble being trained on different sampled interictal data. Furthermore, we also replicated our previous finding that optimal random forest models selected slow frequency frontal channel features for training, which could reflect models identifying these as optimal features for identifying absence seizures. Chapter 5 then expanded on the approaches of the previous chapters, applying a combination of Bayesian and Hyperband optimisation to deep learning model structures and hyperparameters. Comparisons between this approach for absence seizure detection and a broader range of generalized non-specific seizures demonstrated absence seizures are easier for current models to detect accurately; highlighting differences in performance depending on intra-patient and inter-patient variability. Nevertheless, the speed and applicability of LightGBM models to large patient databases provide promising applications for future healthcare needs.

6.1 Recommendations

There are various avenues of future research which can further the work in this thesis. Chapter 3 investigated a range of features for classical machine learning models, however optimal feature selection was not investigated exclusively. Hundreds of features have been used in the literature for seizure detection (with some discussed in chapter 2), however there is still little consensus regarding the best choices. Very little research is focused specifically on feature engineering despite it being one of the most important aspects of a detection pipeline. More reviews and research specific to the best feature engineering methods (e.g. Greene et al., 2008; Logesparan et al., 2012) need to be conducted in order to give confidence

in the differences found between components downstream in a detection pipeline. Similarly, future investigation should explicitly consider the correlation structure between potential features and investigations of optimal group selection of these features for dimensionality reduction. As features in EEG encapsulate both temporal and spatial information, this should include the investigation of optimal limited channel locations for seizure detection (e.g. Chang et al., 2012; Duun-Henriksen et al., 2012a). However, caution should be taken to ensure that these channels are selected due to relevance to a seizure type rather than due to being less prone to noise/artefacts. There are many methods for handling artefacts which could be investigated in classification pipelines in the future; from setting thresholds on the number of zero crossings (e.g. Duun-Henriksen et al., 2012b), to semi-automatic ICA approaches (e.g. Himberg and Hyvxrinen, 2003). An in-depth exploration of these potential pipeline steps was beyond the scope of this research, however in chapter 3 we did investigate multi-class labelling for classifying both seizures and artefacts, but did not find a benefit of such an approach compared to binary classification for "classical" machine learning models. Nevertheless, feature reduction/extraction methods could play a role in long term patient monitoring as they could enable clinicians to find the best few EEG channels for seizure detection, allowing for flexible and personalised electrode positioning. Indeed, such flexible electrode systems are being developed currently (e.g. Epilog; Epitel Inc., 2019), with specialised long term monitoring systems also recently available (e.g. Epihunter, 2020).

As demonstrated in chapter 5, different seizure types provide different challenges, therefore methodologies specific to each seizure type should be investigated separately to find optimal pipelines for each seizure type. Having said this, ultimately these approaches will need to be brought together into a unified approach. This is not to suggest ceasing the investigation of multi-class pipelines, instead it supports a global appreciation of the different challenges between different seizure types. More general systems should be considered for seizure diagnosis beyond one single multi-class model, which likely would be inferior to a holistic approach. This would need to encapsulate different diagnostic protocols for technicians and clinicians as well as the output of various algorithms overseen and interpreted by trained users. For example, classical and ensemble methods in chapters 3 and 4 describe the data in different, but complimentary, ways; as the more specific ensemble algorithms

give better information on the number of seizures, and the more sensitive classical methods on seizure duration. It could be that future implementations would combine markings from both, with seizures predicted by a classical model removed if not partly overlapping with predictions from an ensemble model. There could also be control over the post-processing set by the end-user, with sensible defaults for minimum detection length provided based on research; for example, an investigation into absence seizures could have the default at 4 seconds and generalized seizures set at 17 seconds.

There is often a lack of multi-institution datasets used or compared in research for machine learning in healthcare applications (McDermott et al., 2019), as discussed in chapter 4. This is of course partly due to health data being privacy sensitive (discussed further in section 6.2), nevertheless potential data providers such as hospitals and clinical research centres produce vast quantities of valuable data which could further ensure future models could replicate and generalise over care practices. Currently researchers have limited access to data resources from which to develop new algorithms. Therefore, like other authors (e.g. McDermott et al., 2019), we recommend the creation of more large data trusts where medical institutions can anonymously pool data; such as MIMIC (Johnson et al., 2016), the U.K. and Japan Biobanks (Sudlow et al., 2015; Nagai et al., 2017), eICU (Pollard et al., 2018), the Temple University Hospital EEG Corpus (Harati et al., 2014), and Physionet (Goldberger et al., 2000). This is particularly important due to the number of corporate entities investing in datasets (Rajkomar et al., 2018; Wood, 2019) and patents (Google, 2019), which keep medical data and models in the non-public/non-academic domain, thus reducing research reproducibility and replicability.

6.2 Limitations and Challenges

This thesis focused on the technical limitations of general EEG classification algorithms, which was highlighted in chapter 1 as a reason why semi-automated EEG scoring is not more widely adopted in healthcare. However, there are many other limitations that need to be addressed before future clinical adoption. For instance, it should be taken into account that currently markings of clinical EEG records typically fluctuate between neurologists,

due to the difficulties of following EEG scoring rules, leading to high inter- and intra-scorer variability (Wilson et al., 2003; Younes et al., 2018). There are published criteria of EEG signal characteristics for recognizing electrographic seizures and periodic discharges (e.g. Arif et al., 2013), but inter-rater agreement among EEG experts can be poor, especially where there are complex and abnormal background activities (Ronner et al., 2009). Typically an EEG is marked by experienced clinicians who only have time to skim though records annotating interesting events. However, these transcriptions lack detailed information required for classification training and often miss subtle or brief events. Indeed, periodic discharges have been shown to have a worse inter-rater agreement than seizures (Halford et al., 2015). Brief events are often the most contentious for seizure labelling and there is no official length of time required to define a seizure, with the consequence that the minimum time sufficient to define a seizure is often variable (D'Ambrosio and Miller, 2010). However, discharges accompanied by clinical seizures qualify as electrographic seizures regardless of duration; although whether a person is shown to have clinical signs (e.g. limited responsiveness) depends on how carefully they are observed (Fisher et al., 2014). Furthermore, there is often not always one definitive label for a seizures onset and offset time as the borders of ictal, interictal, and postictal are often indistinct (Fisher et al., 2014). This is unlike other clinical data where labels are more concrete e.g. time of patient death. This uncertainty is often not represented during model training, where onset and offset labels are used as concrete labels rather than based on a certainty metric. Accounting for this uncertainty, with a wider range of seizure onset/offset labelling or certainty weighting, could improve the false positive rate of models which identify onsets sooner than currently labelled.

Another main limitation is that many current datasets, including those used in this research, consist of retrospective data that is collected as part of routine care and later used for research purposes. This method of collection can cause potential privacy risks and contain confounding variables if not handled carefully. These can be somewhat, although not completely, reduced from prospective data collection from consenting participants. Such a regime poses logistical challenges, indeed we struggled to gain prospective data from NHS trusts over the course of this research, but is possible (e.g. All of Us Research Program Investigators, 2019; Arges et al., 2020). There are further security and privacy issues as we

move past training dataset collection and to implementation. For example, some current automated EEG sleep scoring software requires the uploading of recordings to the cloud (e.g. Tay et al., 2017) or external servers (e.g. Younes et al., 2015), which can conflict with data protection policies of healthcare providers (Ali et al., 2018). Indeed healthcare information is highly personal, therefore any transfer of information between parties involves risks, both actual and perceived (Fichman et al., 2011). Its also worth appreciating that different models have different hardware requirements, as shown across this thesis. The models in chapters 3 and 4 for example can run locally on basic CPU hardware, whereas chapter 5 used models that require more expensive GPU hardware and are ideal for running on large computing clusters, in this case we used Google's Cloud Platform. Therefore due to the varying requirements of different algorithms, there is a variation in the degree of data transfer that would be required to deploy them at scale. However, careful societal consideration needs to be taken when implementing any cloud machine learning technology in a healthcare system as such technologies lend themselves to natural monopolies, so regulation will be required to ensure this technology does not get exploited (e.g. increased healthcare costs).

There are other important limitations, which are beyond the scope of this research. These include; the lack of friendly user interfaces (Marcilly et al., 2016), and a general aversion to new technologies in the healthcare sector (Fichman et al., 2011).

6.3 Future Impact

The models discussed in this work are ultimately intended to be further extended and incorporated into assessment tools for EEG diagnosis. These would reduce the time taken for EEG technicians to assess an EEG record, allowing for a greater involvement in patient focused treatment plans (see Topol, 2019) and a broader range of assessment options less constrained by the mark-up time of the EEG record. Seizures are unlikely to be marked as a binary decision, requiring systems that enable a combination of expert and automated workflows to highlight differences between different seizure types and benign features. Such systems should be clear regarding the decision process, such as highlighting where potential seizures are in records, the certainty regarding the diagnostic label, and what specific features

of the highlighted segment were important for labelling. Such a user friendly system (e.g. Selvakumari et al., 2019), would no doubt aid clinical decision making and could also lead to future discoveries in brain functioning (Roy et al., 2019a).

Investments into automation presents a significant opportunity to improve both the efficiency and the quality of care in the NHS. Freed up time for care of patients by introducing automation varies, but estimates range from 11% to 57% for different job roles, at a total estimated value of £12.5 billion a year for the NHS and £6 billion for social care (Darzi, 2018). There are many barriers to increase the pace of adoption of automation, many of which have been covered in this thesis. However, this can be accomplished provided there is appropriate investments in infrastructure and staff training to re-design care pathways. Scientific and health service research is powered by quality datasets and there is a powerful moral imperative to improve care for others through research; which does not require personally identifiable data, but does require integrated datasets shared securely with researchers (Darzi, 2018). Indeed, the NHS has a distinct advantage than other health systems for developing and integrating automated systems due to the "single payer" system which provides a complete, deep, and broad dataset for the whole population; as well as world-leading big data and artificial intelligence research in the UK. With the appropriate investments and continued research interest, automated/semi-automated diagnostic imaging promises to improve care practices in the NHS, with the UK becoming world-leaders in its implementation into healthcare practice.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., and Google Brain (2016). TensorFlow: A System for Large-Scale Machine Learning. *Proc. 12th USENIX Symp. Oper. Syst. Des. Implement. (OSDI '16)*, pages 265–283.

Abbasi, B. and Goldenholz, D. M. (2019). Machine learning applications in epilepsy. *Epilepsia*, 60(10):2037–2047.

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., and Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Comput. Biol. Med.*, 100:270–278.

Acunzo, D. J., MacKenzie, G., and van Rossum, M. C. (2012). Systematic biases in early ERP and ERF components as a result of high-pass filtering. *J. Neurosci. Methods*, 209(1):212–218.

Adeli, H., Zhou, Z., and Dadmehr, N. (2003). Analysis of EEG records in an epileptic patient using wavelet transform. *J. Neurosci. Methods*, 123(1):69–87.

Aeyels, D. (1981). Generic Observability of Differentiable Systems. *SIAM J. Control Optim.*, 19(5):595–603.

Ahammad, N., Fathima, T., and Joseph, P. (2014). Detection of Epileptic Seizure Event and Onset Using EEG. *Biomed Res. Int.*, 2014:1–7.

Akhtar, M. T., Mitsuhashi, W., and James, C. J. (2012). Employing spatially constrained ICA and wavelet denoising, for automatic removal of artifacts from multichannel EEG data. *Signal Processing*, 92(2):401–416.

Akin, M. and Kiymik, M. K. (2000). Application of periodogram and AR spectral analysis to EEG signals. *J. Med. Syst.*, 24(4):247–256.

Akkar, H. A. and Ali Jasim, F. (2017). Optimal Mother Wavelet Function for EEG Signal Analyze Based on Packet Wavelet Transform. *Int. J. Sci. Eng. Res.*, 8(2):1222–1227.

Al-Qazzaz, N. K., Mohd Ali, S. H. B., Ahmad, S. A., Islam, M. S., and Escudero, J. (2015). Selection of mother wavelet functions for multi-channel EEG signal analysis during a working memory task. *Sensors (Switzerland)*, 15(11):29015–29035.

Alakus, T. B. and Turkoglu, I. (2018). Detection of pre-epileptic seizure by using wavelet packet decomposition and artifical neural networks. In *2017 10th Int. Conf. Electr. Electron. Eng.*, pages 511–515.

Albera, L., Kachenoura, A., Comon, P., Karfoul, A., Wendling, F., Senhadji, L., and Merlet, I. (2012). ICA-based EEG denoising: A comparative analysis of fifteen methods. *Bull. Polish Acad. Sci. Tech. Sci.*, 60(3):407–418.

Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti S. Hämäläinen (2014). MNE software for processing MEG and EEG data. *Neuroimage*, 86:446–460.

Alhussein, M., Muhammad, G., Hossain, M. S., and Amin, S. U. (2018). Cognitive IoT-Cloud Integration for Smart Healthcare: Case Study for Epileptic Seizure Detection and Monitoring. *Mob. Networks Appl.*, 23(6):1624–1635.

Ali, O., Shrestha, A., Soar, J., and Wamba, S. F. (2018). Cloud computing-enabled healthcare opportunities, issues, and applications: A systematic review. *Int. J. Inf. Manage.*, 43:146–158.

Alickovic, E., Kevric, J., and Subasi, A. (2018). Performance evaluation of empirical mode decomposition, discrete wavelet transform, and wavelet packed decomposition for au-

tomated epileptic seizure detection and prediction. *Biomed. Signal Process. Control*, 39:94–102.

Alkan, A., Koklukaya, E., and Subasi, A. (2005). Automatic seizure detection in EEG using logistic regression and artificial neural network. *J. Neurosci. Methods*, 148(2):167–176.

Alkanhal, I., Kumar, B. V., and Savvides, M. (2018). Automatic Seizure Detection via an Optimized Image-Based Deep Feature Learning. In *2018 17th IEEE Int. Conf. Mach. Learn. Appl.*, pages 536–540. IEEE.

All of Us Research Program Investigators (2019). The "All of Us" Research Program. *N. Engl. J. Med.*, 381(7):668–676.

Almogbel, M. A., Dang, A. H., and Kameyama, W. (2019). Cognitive Workload Detection from Raw EEG-Signals of Vehicle Driver using Deep Learning. In *2019 21st Int. Conf. Adv. Commun. Technol.*, pages 1167–1172. Global IT Research Institute (GiRI).

Alotaiby, T. N., El-Samie, F. E., Alshebeili, S. A., Aljibreen, K. H., and Alkhanen, E. (2015). Seizure detection with common spatial pattern and Support Vector Machines. In *2015 Int. Conf. Inf. Commun. Technol. Res. ICTRC 2015*, pages 152–155. IEEE.

Amin, S. and Kamboh, A. M. (2016). A robust approach towards epileptic seizure detection. In *2016 IEEE 26th Int. Work. Mach. Learn. Signal Process.*, pages 1–6. IEEE.

Ammar, S. and Senouci, M. (2017). Seizure detection with single-channel EEG using Extreme Learning Machine. In *2016 17th Int. Conf. Sci. Tech. Autom. Control Comput. Eng.*, pages 776–779. IEEE.

Ammar, S., Trigui, O., and Senouci, M. (2018). Genetic and practical swarm optimisation algorithms for patient-specific seizure detection systems. In *2018 4th Int. Conf. Adv. Technol. Signal Image Process.*, pages 1–4. IEEE.

Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., and Elger, C. E. (2001). Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state. *Phys. Rev. E*, 64(6):061907.

Arges, K., Assimes, T., Bajaj, V., Balu, S., Bashir, M. R., Beskow, L., Blanco, R., Califf, R., Campbell, P., Carin, L., Christian, V., Cousins, S., Das, M., Dockery, M., Douglas, P. S., Dunham, A., Eckstrand, J., Fleischmann, D., Ford, E., Fraulo, E., French, J., Gambhir, S. S., Ginsburg, G. S., Green, R. C., Haddad, F., Hernandez, A., Hernandez, J., Huang, E. S., Jaffe, G., King, D., Koweek, L. H., Langlotz, C., Liao, Y. J., Mahaffey, K. W., Marcom, K., Marks, W. J., Maron, D., McCabe, R., McCall, S., McCue, R., Mega, J., Miller, D., Muhlbaier, L. H., Munshi, R., Newby, L. K., Pak-Harvey, E., Patrick-Lake, B., Pencina, M., Peterson, E. D., Rodriguez, F., Shore, S., Shah, S., Shipes, S., Sledge, G., Spielman, S., Spitler, R., Schaack, T., Swamy, G., Willemink, M. J., and Wong, C. A. (2020). The Project Baseline Health Study: a step towards a broader mission to map human health. *npj Digit. Med.*, 3(1):1–10.

Arif, H., Hirsch, L., LaRoche, S., Gaspard, N., Gerard, E., Svoronos, A., Herman, S., Mani, R., Jetté, N., Minazad, Y., Kerrigan, J., Vespa, P., Hantus, S., Claassen, J., Young, G., So, E., Kaplan, P., Nuwer, M., Fountain, N., and Drislane, F. (2013). American Clinical Neurophysiology Society's standardized critical care EEG terminology: Interrater reliability and 2012 version. *J. Neurol. Sci.*, 333(1):e15–e16.

Artoni, F., Delorme, A., and Makeig, S. (2018). Applying dimension reduction to EEG data by Principal Component Analysis reduces the quality of its subsequent Independent Component decomposition. *Neuroimage*, 175:176–187.

Asif, U., Roy, S., Tang, J., and Harrer, S. (2019). SeizureNet: A Deep Convolutional Neural Network for Accurate Seizure Type Classification and Seizure Detection. arXiv:1903.03232.

Awan, S. E., Khawaja, S. G., Khan, M. A., and Usman Akram, M. (2016). A surrogate channel based analysis of EEG signals for detection of epileptic seizure. In *2016 IEEE Int. Conf. Imaging Syst. Tech.*, pages 384–388. IEEE.

Aznan, N. K., Bonner, S., Connolly, J., Al Moubayed, N., and Breckon, T. (2018). On the Classification of SSVEP-Based Dry-EEG Signals via Convolutional Neural Networks. In *2018 IEEE Int. Conf. Syst. Man, Cybern.*, pages 3726–3731. IEEE.

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:1803.01271.

Baier, G., Hermann, T., Sahle, S., and Stephani, U. (2006). Sonified epilectic rhythms. In *Proc. Int. Conf. Audit. Disp.*, pages 148–151.

Baldassano, S., Wulsin, D., Ung, H., Blevins, T., Brown, M., Fox, E., and Litt, B. (2016). A novel seizure detection algorithm informed by hidden Markov model event states. *J. Neural Eng.*, 13(3):036011.

Baldassano, S. N., Brinkmann, B. H., Ung, H., Blevins, T., Conrad, E. C., Leyde, K., Cook, M. J., Khambhati, A. N., Wagenaar, J. B., Worrell, G. A., and Litt, B. (2017). Crowdsourcing seizure detection: Algorithm development and validation on human implanted device recordings. *Brain*, 140(6):1680–1691.

Bartlett, A. M. S. and Medhi, J. (1955). On the Efficiency of Procedures for Smoothing Periodograms from Time Series with Continuous Spectra. *Biometrika*, 42(1):143–150.

Behnam, M. and Pourghassem, H. (2015). Periodogram pattern feature-based seizure detection algorithm using optimized hybrid model of MLP and ant colony. In *2015 23rd Iran. Conf. Electr. Eng.*, pages 32–37.

Belouchrani, A., Abed-Meraim, K., Cardoso, J., and Moulines, E. (1993). Second-order blind separation of temporally correlated sources. In *Proc. Int. Conf. Digit. Signal Process.*, pages 346–51.

Benedetto, J. J., Heil, C., and Walnut, D. F. (1994). Differentiation and the Balian-Low Theorem. *J. Fourier Anal. Appl.*, 1(4):355–402.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.

Berg, P. and Scherg, M. (1991). Dipole modelling of eye activity and its application to the removal of eye artefacts from the eeg and meg. *Clin. Phys. Physiol. Meas.*, 12:49–54.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., and Cox, D. D. (2015). Hyperopt: A Python library for model selection and hyperparameter optimization. *Comput. Sci. Discov.*, 8(1):0–24.

Bergstra, J., Pinto, N., and Cox, D. (2012). Machine learning for predictive auto-tuning with boosted regression trees. In *2012 Innov. Parallel Comput.*, pages 1–9. IEEE.

Bergstra, J., Yamins, D. L. K., and Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proc. 30th Int. Conf. Mach. Learn.*, pages 115–123.

Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. In *Adv. Neural Inf. Process. Syst.*, pages 2546–2554.

Bhat, S., Acharya, U. R., Adeli, H., Bairy, G. M., and Adeli, A. (2014). Automated diagnosis of autism: In search of a mathematical marker. *Rev. Neurosci.*, 25(6):851–861.

Bhat, S., Acharya, U. R., Hagiwara, Y., Dadmehr, N., and Adeli, H. (2018). Parkinson's disease: Cause factors, measurable indicators, and early diagnosis. *Comput. Biol. Med.*, 102:234–241.

Bhattacharyya, A. and Pachori, R. B. (2017). A multivariate approach for patient specific EEG seizure detection using empirical wavelet transform. *IEEE Trans. Biomed. Eng.*, 64(9):2003–2015.

Bidwell, J., Khuwatsamrit, T., Askew, B., Ehrenberg, J. A., and Helmers, S. (2015). Seizure reporting technologies for epilepsy treatment: A review of clinical information needs and supporting technologies. *Seizure*, 32:109–117.

Bigdely-Shamlo, N., Mullen, T., Kothe, C., Su, K.-M., and Robbins, K. A. (2015). The PREP pipeline: standardized preprocessing for large-scale EEG analysis. *Front. Neuroinform.*, 9:1–20.

Birjandtalab, J., Baran Pouyan, M., Cogan, D., Nourani, M., and Harvey, J. (2017). Automated seizure detection using limited-channel EEG and non-linear dimension reduction. *Comput. Biol. Med.*, 82:49–58.

Bolagh, S. N. G. and Clifford, G. D. (2017). Subject Selection on a Riemannian Manifold for Unsupervised Cross-subject Seizure Detection. arXiv:1712.00465.

Boostani, R., Karimzadeh, F., and Nami, M. (2017). A comparative review on sleep stage classification methods in patients and healthy individuals. *Comput. Methods Programs Biomed.*, 140:77–91.

Bottou, L. and Lin, C.-J. (2007). Support Vector Machine Solvers. In *Large-scale kernel Mach.*, pages 1–29. MIT press.

Bouckaert, R. (2005). Low replicability of machine learning experiments is not a small data set phenomenon. In *Proc. ICML - 2005 Work. Meta-learning*.

Bradbury, J., Merity, S., Xiong, C., and Socher, R. (2017). Quasi-recurrent neural networks. In *5th Int. Conf. Learn. Represent. {ICLR} 2017*.

Breiman, L. (1996). Bagging predictors. *Mach. Learn.*, 140:123–140.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. (1984). *Classification and Regression Trees*. Taylor & Francis.

Breiman, L., Wolpert, D., Chan, P., and Stolfo, S. (1999). Pasting Small Votes for Classification in Large Databases and On-Line. *Mach. Learn.*, 36:85–103.

Breiman, L. E. O. (2001). Random Forests. *Mach. Learn.*, 45(1):5–32.

Brinkmann, B. H., Bower, M., Stengel, K. A., Worrell, G. A., and Stead, M. (2009). Large-scale Electrophysiology: Acquisition, Compression, Encryption, and Storage of Big Data. *J Neurosci Methods*, 180(1):185–192.

Britton, J., Frey, L., Hopp, J., Korb, P., Koubeissi, M., Lievens, W., Pestana-Knight, E., and St. Louis, E. (2016). *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants*. American Epilepsy Society, Chicago.

Browne, T. R., Penry, J. K., Proter, R. J., and Dreifuss, F. E. (1974). Responsiveness before, during, and after spike-wave paroxysms. *Neurology*, 24(7):659–65.

Bugeja, S., Garg, L., and Audu, E. E. (2016). A novel method of EEG data acquisition, feature extraction and feature space creation for early detection of epileptic seizures. In *2016 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 837–840. IEEE.

Burkov, A. (2019). *The Hundred - Page Machine Learning Book*. Andriy Burkov.

Buzsáki, G. (2009). *Rhythms of the Brain*. Oxford University Press.

Cao, Y., Guo, Y., Yu, H., and Yu, X. (2017). Epileptic Seizure Auto-detection Using Deep Learning Method. In *2017 4th Int. Conf. Syst. Informatics*, pages 1076–1081.

Caplan, J. B., Madsen, J. R., Raghavachari, S., and Kahana, M. J. (2001). Distinct Patterns of Brain Oscillations Underlie Two Basic Parameters of Human Maze Learning. *J. Neurophysiol.*, 86(1):368–380.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. data Min.*, pages 1721–1730.

Casarotto, S., Bianchi, A. M., Cerutti, S., and Chiarenza, G. A. (2004). Principal component analysis for reduction of ocular artefacts in event-related potentials of normal and dyslexic children. *Clin. Neurophysiol.*, 115(3):609–619.

Casdagli, M. C., Iasemidis, L. D., Savit, R. S., Gilmore, R. L., Roper, S. N., and Sackellares, J. C. (1997). Non-linearity in invasive EEG recordings from patients with temporal lobe epilepsy. *Electroencephalogr. Clin. Neurophysiol.*, 102(2):98–105.

Chandel, G., Farooq, O., Khan, Y., and Chawla, M. (2016). Seizure Onset Detection by Analyzing Long-Duration EEG Signals. In *Proc. Second Int. Conf. Comput. Commun. Technol.*, pages 215–224, New Delhi. Springer.

Chandel, G., Farooq, O., U. Khan, Y., and Shanir, P. M. (2017). Seizure Onset and Offset Detection by using Wavelet Based Features. In *2017 4th Int. Conf. "Computing Sustain. Glob. Dev.*, pages 5801–5807.

Chandel, G., Upadhyaya, P., Farooq, O., and Khan, Y. U. (2019). Detection of Seizure Event and Its Onset/Offset Using Orthonormal Triadic Wavelet Based Features. *IRBM*, 40(2):103–112.

Chang, N. F., Chen, T. C., Chiang, C. Y., and Chen, L. G. (2012). Channel selection for epilepsy seizure prediction method based on machine learning. In *2012 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 5162–5165. IEEE.

Chang, S., Zhang, Y., Han, W., Yu, M., Guo, X., Tan, W., Cui, X., Witbrock, M., Hasegawa-Johnson, M., and Huang, T. (2017). Dilated Recurrent Neural Networks. In *Adv. Neural Inf. Process. Syst.*, pages 77–87.

Chavakula, V., Sánchez Fernández, I., Peters, J. M., Popli, G., Bosl, W., Rakhade, S., Rotenberg, A., and Loddenkemper, T. (2013). Automated quantification of spikes. *Epilepsy Behav.*, 26(2):143–152.

Chen, C., Liaw, A., and Breiman, L. (2004). Using random forest to learn imbalanced data. *Berkeley Dep. Stat. Tech Reports*, (666).

Chen, L. L., Zhang, J., Zou, J. Z., Zhao, C. J., and Wang, G. S. (2014). A framework on wavelet-based nonlinear features and extreme learning machine for epileptic seizure detection. *Biomed. Signal Process. Control*, 10(1):1–10.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proc. 22nd acm sigkdd Int. Conf. Knowl. Discov. data Min.*, pages 785–794.

Chen, X., Ji, J., Ji, T., and Li, P. (2018). Cost-Sensitive Deep Active Learning for Epileptic Seizure Detection. In *Proc. 2018 ACM Int. Conf. Bioinformatics, Comput. Biol. Heal. Informatics*, pages 226–235.

Choi, G., Park, C., Kim, J., Cho, K., Kim, T. J., Bae, H., Min, K., Jung, K.-Y., and Chong, J. (2019). A Novel Multi-scale 3D CNN with Deep Neural Network for Epileptic Seizure Detection. In *2019 IEEE Int. Conf. Consum. Electron.*, pages 1–2. IEEE.

Choi, S., Cichocki, A., Park, H.-M., and Lee, S.-Y. (2005). Blind source separation and independent component analysis: A Review. *Neural Inf. Process. Rev.*, 6(1):1–57.

Chollet, F. (2017a). *Deep learning with python*. Manning Publications.

Chollet, F. (2017b). Xception: Deep learning with depthwise separable convolutions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1251–1258.

Cohen, M. X. (2014). *Analyzing Neural Time Series Data: Theory and Practice*. MIT Press.

Coifman, R. R. and Wickerhauser, M. V. (1992). Entropy-based algorithms for best basis selection. *IEEE Trans. Inf. Theory*, 38(2):713–718.

Cole, S. R. (2016). *Empirical Mode Decomposition (EMD) tutorial*. Retrieved from https://github.com/srcole/binder_emd.

Cole, S. R. and Voytek, B. (2019). Cycle-by-cycle analysis of neural oscillations. *J. Neurophysiol.*, 122(2):849–861.

Coleman, C., Narayanan, D., Kang, D., Zhao, T., Zhang, J., Nardi, L., Bailis, P., Olukotun, K., Ré, C., and Zaharia, M. (2017). DAWNBench: An End-to-End Deep Learning Benchmark and Competition. In *Thirty-first Annu. Conf. Neural Inf. Process. Syst.*

Coles, L. D., Patterson, E. E., Sheffield, W. D., Mavooric, J., Higgins, J., Bland, M., Leyde, K., Cloyd, J. C., Litt, B., Vite, C., and Worrell, G. (2013). Feasibility Study of a Caregiver Seizure Alert System in Canine Epilepsy. *Epilepsy Res*, 106(3):456–460.

Connor, C. W. (2019). Artificial Intelligence and Machine Learning in Anesthesiology. *Anesthesiology*, 131(6):1346–1359.

Costa, J. C. G., Da-Silva, P. J. G., Almeida, R. M. V., and Infantosi, A. F. C. (2014). Validation in principal components analysis applied to EEG data. *Comput. Math. Methods Med.*, 2014:10.

Cover, T. M. (1965). Geometric and statistical properties of systems of linear in-equalities with applications in pattern recognition. *IEEE Trans. Electron. Comput.*, 14(3):326.

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. Wiley, New York.

Cranstoun, S. D., Ombao, H. C., Von Sachs, R., Guo, W., and Litt, B. (2002). Time-frequency spectral estimation of multichannel EEG using the auto-SLEX method. *IEEE Trans. Biomed. Eng.*, 49(9):988–996.

Croft, R. and Barry, R. (2000). Removal of ocular artifact from the EEG: a review. *Neurophysiol. Clin. Neurophysiol.*, 30(1):5–19.

Çınar, S. and Acır, N. (2017). A novel system for automatic removal of ocular artefacts in EEG by using outlier detection methods and independent component analysis. *Expert Syst. Appl.*, 68:36–44.

D'Ambrosio, R. and Miller, J. W. (2010). What is an epileptic seizure? Unifying definitions in clinical practice and animal research to develop novel treatments. *Epilepsy Curr.*, 10(3):61–66.

Darzi, A. (2018). *Better Health And Care For All: A 10-Point Plan For The 2020s.* Institute for Public Policy Research. Retrieved from www.ippr.org.

Das, A. B., Pantho, M. J. H., and Bhuiyan, M. I. H. (2016). Discrimination of scalp EEG signals in wavelet transform domain and channel selection for the patient-invariant seizure detection. In *2015 Int. Conf. Electr. Electron. Eng.*, pages 77–80. IEEE.

de la Cal, E., Villar, J. R., Vergara, P., Sedano, J., and Herrero, Á. (2018). A SMOTE extension for balancing multivariate epilepsy-related time series datasets. *Adv. Intell. Syst. Comput.*, 649:439–448.

De Vos, M., Deburchgraeve, W., Cherian, P. J., Matic, V., Swarte, R. M., Govaert, P., Visser, G. H., and Van Huffel, S. (2011). Automated artifact removal as preprocessing refines neonatal seizure detection. *Clin. Neurophysiol.*, 122(12):2345–2354.

Deiss, O., Biswal, S., Jin, J., Sun, H., Westover, M. B., and Sun, J. (2018). HAMLET: Interpretable Human And Machine co-LEarning Technique. arXiv:1803.09702.

Delorme, A. and Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J. Neurosci. Methods*, 134(1):9–21.

Delorme, A., Palmer, J., Onton, J., Oostenveld, R., and Makeig, S. (2012). Independent EEG sources are dipolar. *PLoS One*, 7(2):1–14.

Deng, Z., Xu, P., Xie, L., Choi, K. S., and Wang, S. (2018). Transductive Joint-Knowledge-Transfer TSK FS for Recognition of Epileptic EEG Signals. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 26(8):1481–1494.

Diks, C. (1999). *Nonlinear Time Series Analysis: Methods and Applications*. World Scientific Press.

Dimensional Research (2019). Artificial Intelligence and Machine Learning Projects Are Obstructed by Data Issues Global Survey of Data Scientists, AI Experts and Stakeholders. Technical Report May, Dimensional Research.

Dong, H., Supratak, A., Pan, W., Wu, C., Matthews, P. M., and Guo, Y. (2018). Mixed Neural Network Approach for Temporal Sleep Stage Classification. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 26(2):324–333.

Duann, J. R., Jung, T. P., Kuo, W. J., Yeh, T. C., Makeig, S., Hsieh, J. C., and Sejnowski, T. J. (2001). Measuring the Variability of Event-Related Bold Signal. *Proc., 3rd Int. Conf. Indep. Compon. Anal. Blind Signal Sep.*, pages 528–533.

Duann, J.-R., Jung, T.-P., Makeig, S., and Sejnowski, T. J. (2003). Consistency of Infomax ICA Decomposition of Functional Brain Imaging Data. In *Proc. 4th Int. Symp. Indep. Compon. Anal. Blind Signal Sep. (ICA 2003)*, pages 289–294.

Durá Travé, T. and Yoldi Petri, M. (2006). Ausencias típicas: características epidemiológicas, clínicas y evolutivas. *An. Pediatría*, 64(1):28–33.

Duun-Henriksen, J., Kjaer, T. W., Madsen, R. E., Remvig, L. S., Thomsen, C. E., and Sorensen, H. B. D. (2012a). Channel selection for automatic seizure detection. *Clin. Neurophysiol.*, 123(1):84–92.

Duun-Henriksen, J., Madsen, R. E., Remvig, L. S., Thomsen, C. E., Sorensen, H. B. D., and Kjaer, T. W. (2012b). Automatic detection of childhood absence epilepsy seizures: Toward a monitoring device. *Pediatr. Neurol.*, 46(5):287–292.

Dwork, C. and Ullman, J. (2018). The fienberg problem: How to allow human interactive data analysis in the age of differential privacy. *J. Priv. Confidentiality*, 8(1):1–10.

Elger, C. E., Mormann, F., Kreuz, T., Andrzejak, R. G., Rieke, C., Sowa, R., Florin, S., David, P., and Lehnertz, K. (2002). Characterizing the spatio-temporal dynamics of the epileptogenic process with nonlinear EEG analyses. In *Proc. 2002 7th IEEE Int. Work. Cell. Neural Networks Their Appl.*, pages 228–242.

Elmahdy, A. E., Yahya, N., Kamel, N. S., and Shahid, A. (2015). Epileptic seizure detection using singular values and classical features of EEG signals. In *2015 Int. Conf. BioSignal Anal. Process. Syst.*, pages 162–167. IEEE.

Engel Jr, J. (2013). *Seizures and epilepsy*. Oxford University Press.

Epihunter (2020). *Epihunter*. Retrieved from https://www.epihunter.com.

Epitel Inc. (2019). *Epilog*. Retrieved from https://www.epitel.com.

Esposito, F., Formisano, E., Seifritz, E., Goebel, R., Morrone, R., Tedeschi, G., and Di Salle, F. (2002). Spatial independent component analysis of functional MRI time-series: To what extent do results depend on the algorithm used? *Hum. Brain Mapp.*, 16(3):146–157.

European Parliament (2016). Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data , and repealing Directive 95/46/EC (General Data Protection Regulation – GDPR). Technical report.

Falkner, S., Klein, A., and Hutter, F. (2018). BOHB: Robust and Efficient Hyperparameter Optimization at Scale. *35th Int. Conf. Mach. Learn. ICML 2018*, 4:2323–2341.

Faust, O., Hagiwara, Y., Hong, T. J., Lih, O. S., and Acharya, U. R. (2018). Deep learning for healthcare applications based on physiological signals: A review. *Comput. Methods Programs Biomed.*, 161:1–13.

Fearnhead, P., Maidstone, R., and Letchford, A. (2019). Detecting Changes in Slope With an L0 Penalty. *J. Comput. Graph. Stat.*, 28(2):265–275.

Fergus, P., Hussain, A., Hignett, D., Al-Jumeily, D., Abdel-Aziz, K., and Hamdan, H. (2015). Automatic Epileptic Seizure Detection Using Scalp EEG and Advanced Artificial Intelligence Techniques. *Biomed Res. Int.*

Fergus, P., Hussain, A., Hignett, D., Al-Jumeily, D., Abdel-Aziz, K., and Hamdan, H. (2016). A machine learning system for automated whole-brain seizure detection. *Appl. Comput. Informatics*, 12(1):70–89.

Fichman, R. G., Kohli, R., and Krishnan, R. (2011). The role of information systems in healthcare: Current research and future trends. *Inf. Syst. Res.*, 22(3):419–428.

Fiorillo, L., Puiatti, A., Papandrea, M., Ratti, P. L., Favaro, P., Roth, C., Bargiotas, P., Bassetti, C. L., and Faraci, F. D. (2019). Automated sleep scoring: A review of the latest approaches. *Sleep Med. Rev.*, 48:101204.

Fisher, R. S. (2017). The New Classification of Seizures by the International League Against Epilepsy 2017. *Curr. Neurol. Neurosci. Rep.*, 17(6):1–6.

Fisher, R. S., Scharfman, H. E., and DeCurtis, M. (2014). How Can We Identify Ictal and Interictal Abnormal Activity? *Adv. Exp. Med. Biol.*, 813:3–23.

Fitzgibbon, S. P. and Powers, D. M. W. (2007). Removal of EEG Noise and Artifact Using Blind Source Separation. *J. Clin. Neurophysiol.*, 24(3):1–13.

Fourier, J. B. J. (1878). *The Analytical Theory of Heat.* The University Press.

Freund, Y. and Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.*, 55(1):119–139.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Trans. Math. Softw.*, 3(3):209–226.

Gabor, A. J. (1998). Seizure detection using a self-organizing neural network: Validation and comparison with other detection strategies. *Electroencephalogr. Clin. Neurophysiol.*, 107(1):27–32.

Gabor, A. J., Leach, R. R., and Dowla, F. U. (1996). Automated seizure detection using a self-organizing neural network. *Electroencephalogr. Clin. Neurophysiol.*, 99(3):257–266.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Int. Conf. Mach. Learn.*, pages 1050–1059.

Gandhi, T., Panigrahi, B. K., and Anand, S. (2011). A comparative study of wavelet families for EEG signal classification. *Neurocomputing*, 74(17):3051–3057.

George, S. T., Subathra, M. S., Sairamya, N. J., Susmitha, L., and Joel Premkumar, M. (2020). Classification of epileptic EEG signals using PSO based artificial neural network and tunable-Q wavelet transform. *Biocybern. Biomed. Eng.*, 40(2):709–728.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2nd edition.

Ghorbanian, P., Devilbiss, D. M., Verma, A., Bernstein, A., Hess, T., Simon, A. J., and Ashrafiuon, H. (2013). Identification of resting and active state EEG features of alzheimer's disease using discrete wavelet transform. *Ann. Biomed. Eng.*, 41(6):1243–1257.

Ghosh, A., dal Maso, F., Roig, M., Mitsis, G. D., and Boudrias, M.-H. (2018). Deep Semantic Architecture with discriminative feature visualization for neuroimage analysis. arXiv:1805.11704.

Giger, M. L. (2018). Machine Learning in Medical Imaging. *J. Am. Coll. Radiol.*, 15(3):512–520.

Giourou, E., Stavropoulou-Deli, A., Giannakopoulou, A., Kostopoulos, G. K., and Koutroumanidis, M. (2015). Introduction to Epilepsy and Related Brain Disorders. In Voros, N. S. and Antonopoulos, C. P., editors, *Cyberphysical Syst. Epilepsy Relat.*

Brain Disord. Multi-Parametric Monit. Anal. Diagnosis Optim. Dis. Manag., chapter 2, pages 11–38. Springer International Publishing.

Glauser, T. A., Cnaan, A., Shinnar, S., Hirtz, D. G., Dlugos, D., Masur, D., Clark, P. O., and Adamson, P. C. (2013). Ethosuximide, valproic acid, and lamotrigine in childhood absence epilepsy: Initial monotherapy outcomes at 12 months. *Epilepsia*, 54(1):141–155.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-k., and Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, 101(23):e215–e220.

Golmohammadi, M., Torbati, A. H. H. N., de Diego, S. L., Obeid, I., and Picone, J. (2019). Automatic Analysis of EEGs Using Big Data and Hybrid Deep Learning Architectures. *Front. Hum. Neurosci.*, 13(76):1–14.

Golmohammadi, M., Ziyabari, S., Shah, V., Obeid, I., and Picone, J. (2018). Deep Architectures for Spatio-Temporal Modeling: Automated Seizure Detection in Scalp EEGs. In *2018 17th IEEE Int. Conf. Mach. Learn. Appl.*, pages 745–750. IEEE.

Golmohammadi, M., Ziyabari, S., Shah, V., Von Weltin, E., Campbell, C., Obeid, I., and Picone, J. (2017). Gated recurrent networks for seizure detection. In *2017 IEEE Signal Process. Med. Biol. Symp.*, pages 1–5.

Goncharova, I. I., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R. (2003). EMG contamination of EEG: Spectral and topographical characteristics. *Clin. Neurophysiol.*, 114(9):1580–1593.

Gong, J. J., Naumann, T., Szolovits, P., and Guttag, J. V. (2017). Predicting clinical outcomes across changing electronic health record systems. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pages 1497–1506.

Gonzalez-Vellon, B., Sanei, S., and Chambers, J. A. (2003). Support vector machines for

seizure detection. In *Proc. 3rd IEEE Int. Symp. Signal Process. Inf. Technol. (IEEE Cat. No. 03EX795)*, pages 126–129.

Goode, D. J., Penry, J. K., and Dreifuss, F. E. (1970). Effects of Paroxysmal Spike-Wave on Continuous Visual-Motor Performance. *Epilepsia*, 11(3):241–254.

Google (2019). System and Method for Predicting and Summarizing Medical Events from EHRs. *U.S. Pat. No. 15/690,721*, Washington.

Gotman, J. (1982). Automatic recognition of epileptic seizures in the EEG. *Electroencephalogr. Clin. Neurophysiol.*, 54(5):530–540.

Gotman, J. (1990). Automatic seizure detection: improvements and evaluation. *Electroencephalogr. Clin. Neurophysiol.*, 76(4):317–324.

Gotman, J. (2013). High frequency oscillations: The new EEG frontier? *Epilepsia*, 51(Suppl 1):63–65.

Gotman, J., Flanagan, D., Zhang, J., and Rosenblatt, B. (1997). Automatic seizure detection in the newborn: Methods and initial evaluation. *Electroencephalogr. Clin. Neurophysiol.*, 103(3):356–362.

Gotman, J., Ives, J., and Gloor, P. (1981). Frequency content of EEG and EMG at seizure onset: Possibility of removal of EMG artefact by digital filtering. *Electroencephalogr. Clin. Neurophysiol.*, 52(6):626–639.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., and Hämäläinen, M. (2013). MEG and EEG data analysis with MNE-Python. *Front. Neurosci.*, 7:1–13.

Greene, B. R., Faul, S., Marnane, W. P., Lightbody, G., Korotchikova, I., and Boylan, G. B. (2008). A comparison of quantitative EEG features for neonatal seizure detection. *Clin. Neurophysiol.*, 119(6):1248–1261.

Gu, Y., Cleeren, E., Dan, J., Claes, K., Van Paesschen, W., Van Huffel, S., and Hunyadi, B. (2018). Comparison between scalp EEG and behind-the-ear EEG for development of

a wearable seizure detection system for patients with focal epilepsy. *Sensors (Switzerland)*, 18(1):1–17.

Gyaourova, A., Kamath, C., and Fodor, I. (2002). Undecimated wavelet transforms for image de-noising. Technical report, CA.

Halford, J. J., Shiau, D., Desrochers, J. A., Kolls, B. J., Dean, B. C., Waters, C. G., Azar, N. J., Haas, K. F., Kutluay, E., Martz, G. U., Sinha, S. R., Kern, R. T., Kelly, K. M., Sackellares, J. C., and LaRoche, S. M. (2015). Inter-rater agreement on identification of electrographic seizures and periodic discharges in ICU EEG recordings. *Clin. Neurophysiol.*, 126(9):1661–1669.

Hamdan, H., Hignett, D., Al-Jumeily, D., Abdel-Aziz, K., Hussain, A., and Fergus, P. (2015). A machine learning system for automated whole-brain seizure detection. *Appl. Comput. Informatics*, 12(1):70–89.

Harati, A., Lopez, S., Obeid, I., Picone, J., Jacobson, M. P., and Tobochnik, S. (2014). The TUH EEG CORPUS: A big data resource for automated EEG interpretation. In *2014 IEEE Signal Process. Med. Biol. Symp.*, pages 1–5. IEEE.

Harpale, V. and Bairagi, V. (2018). An adaptive method for feature selection and extraction for classification of epileptic EEG signal in significant states. *J. King Saud Univ. - Comput. Inf. Sci.*

Haynes, K., Eckley, I. A., and Fearnhead, P. (2017). Computationally Efficient Changepoint Detection for a Range of Penalties. *J. Comput. Graph. Stat.*, 26(1):134–143.

Hazarika, N., Chen, J. Z., Tsoi, A. C., and Sergejew, A. (1997). Classification of EEG signals using the wavelet transform. *Signal Processing*, 59(1):61–72.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, pages 770–8.

Henriksen, J., Remvig, L. S., Madsen, R. E., Conradsen, I., Kjaer, T. W., Thomsen, C. E., and Sorensen, H. B. D. (2010). Automatic seizure detection: Going from sEEG to iEEG. In *2010 Annu. Int. Conf. IEEE Eng. Med. Biol.*, pages 2431–2434.

Hesse, C. W. and James, C. J. (2006). On semi-blind source separation using spatial constraints with applications in EEG analysis. *IEEE Trans. Biomed. Eng.*, 53(12):2525–2534.

Hido, S., Kashima, H., and Takahashi, Y. (2009). Roughly Balanced Bagging for Imbalanced Data. *Stat. Anal. Data Min. ASA Data Sci. J.*, 2(5-6):412–426.

Himberg, J. and Hyvxrinen, A. (2003). Icasso: software for investigating the reliability of ICA estimates by clustering and visualization. In *2003 IEEE XIII Work. Neural Networks Signal Process.*, pages 259–268.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580.

Holmes, M. D., Brown, M., and Tucker, D. M. (2004). Are "generalized" seizures truly generalized? Evidence of localized mesial frontal and frontopolar discharges in absence. *Epilepsia*, 45(12):1568–1579.

Holschneider, M., Kronland-Martinet, R., Morlet, J., and Tchamitchian, P. (1989). A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets, Time-Frequency Methods Phase Sp.*, pages 289–297. Springer-Verlag.

Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., and Schölkopf, B. (2007). Correcting sample selection bias by unlabeled data. *Adv. Neural Inf. Process. Syst.*, pages 601–608.

Huang, K. and Aviyente, S. (2006). Information-theoretic wavelet packet subband selection for texture classification. *Signal Processing*, 86(7):1410–1420.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Yen, N.-c., Tung, C. C., and Liu, H. H. (1996). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *R. Soc. London Proc. Ser. A*, 454(1):903–995.

Huber, P. (1985). Projection Pursuit. *Ann. Stat.*, 13:435–475.

Hughes, J. R. (2009). Absence seizures: A review of recent reports with new concepts. *Epilepsy Behav.*, 15(4):404–412.

Hulse, J. V. and Khoshgoftaar, T. M. (2007). Experimental Perspectives on Learning from Imbalanced Data. In *Proc. 24th Int. Conf. Mach. Learn.*, pages 935–942.

Hussain, S. J. (2018). Epileptic Seizure Detection Using Wavelets and EMD. In *2018 Fourth Int. Conf. Biosignals, Images Instrum.*, pages 206–212. IEEE.

Hyvarinen, A. (2008). Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks, IEEE Trans.*, 10(3):2008.

Hyvarinen, A. and Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Comput.*, 9(7):1483–1492.

Hyvarinen, A. and Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430.

Ibrahim, S. W. and Majzoub, S. (2017). EEG-based epileptic seizures detection with adaptive learning capability. *Int. J. Electr. Eng. Informatics*, 9(4):813–824.

Iešmantas, T. and Alzbutas, R. (2020). Convolutional neural network for detection and classification of seizures in clinical data. *Med. Biol. Eng. Comput.*, 58:1919–1932.

Ihle, M., Feldwisch-Drentrup, H., Teixeira, C. A., Witon, A., Schelter, B., Timmer, J., and Schulze-Bonhage, A. (2012). EPILEPSIAE - A European epilepsy database. *Comput. Methods Programs Biomed.*, 106(3):127–138.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd Int. Conf. Mach. Learn. ICML 2015*, pages 448–456.

Iskandaryan, D., Ramos, F., and Trilles, S. (2020). Air quality prediction in smart cities using machine learning technologies based on sensor data: A review. *Appl. Sci.*, 10(7):2401.

Jaber, A. M., Ismail, M. T., and Altaher, A. M. (2014). Application of empirical mode decomposition with local linear quantile regression in financial time series forecasting. *Sci. World J.*, 2014.

James, C. J. and Gibson, O. J. (2003). Temporally constrained ICA: An application to artifact rejection in electromagnetic brain signal analysis. *IEEE Trans. Biomed. Eng.*, 50(9):1108–1116.

James, C. J. and Hesse, C. W. (2005). Independent component analysis for biomedical signals. *Physiol. Meas.*, 26(1):15–39.

Jana, G. C., Sabath, A., and Agrawal, A. (2019). Performance Analysis of Supervised Machine Learning Algorithms for Epileptic Seizure Detection with high variability EEG datasets: A Comparative Study. In *2019 Int. Conf. Electr. Electron. Comput. Eng.*, pages 1–6.

Jang, H. J. and Cho, K. O. (2019). Applications of deep learning for the analysis of medical data. *Arch. Pharm. Res.*, 42(6):492–504.

Jasper, H. (1958). Report of the committee on methods of clinical examination in electroencephalography. *Electroencephalogr. Clin. Neurophysiol.*, 10(2):370–375.

Javaid, N., Jamil, M., Omer Gillani, S., Ayaz, Y., Ahmad, M. A., Majeed, W., Rasheed, M. B., Imran, M., and Khan, N. A. (2015). Comparative Analysis of Classifiers for Developing an Adaptive Computer-Assisted EEG Analysis System for Diagnosing Epilepsy. *Biomed Res. Int.*, 2015:1–14.

Jenke, R., Peer, A., and Buss, M. (2014). Feature extraction and selection for emotion recognition from EEG. *IEEE Trans. Affect. Comput.*, 5(3):327–339.

Jensen, O., Bonnefond, M., and VanRullen, R. (2012). An oscillatory mechanism for prioritizing salient unattended stimuli. *Trends Cogn. Sci.*, 16(4):200–206.

Jenssen, S., Gracely, E. J., and Sperling, M. R. (2006). How long do most seizures last? A systematic comparison of seizures recorded in the epilepsy monitoring unit. *Epilepsia*, 47(9):1499–1503.

Jiang, X. and Bian, G.-b. (2019). Removal of Artifacts from EEG Signals: A Review. *Sensors*, 19(5):1–18.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. W. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Sci. Data*, 3(1):1–9.

Johnson, J. M. and Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *J. Big Data*, 6(1).

Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer, New York.

Joyce, C. A., Gorodnitsky, I. F., and Kutas, M. (2004). Automatic removal of eye movement and blink artifacts from EEG data using blink component separation. *Psychophysiology*, 41(2):313–325.

Jung, T., Makeig, S., Humphries, C., Lee, T., McKeown, M., Iragui, V., and Sejnowski, T. (2000). Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37:163–178.

Kaleem, M., Guergachi, A., and Krishnan, S. (2018). Patient-specific seizure detection in long-term EEG using wavelet decomposition. *Biomed. Signal Process. Control*, 46:157–165.

Kantz, H. and Schreiber, T. (2004). *Nonlinear time series analysis*. Cambridge University Press.

Kaplan, A. Y. and Shishkin, S. L. (2000). Application of the change-point analysis to investigation of the brain electrical activity. In Brodsky, E. and Darkhovsky, B. S., editors, *Non-Parametric Stat. Diagnosis Probl. Methods*, chapter 7, pages 333–388. Springer Science & Business Media.

Ke, G., Meng, Q., Wang, T., Chen, W., Ma, W., and Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Adv. Neural Inf. Process. Syst.*, pages 3148–3156.

Ke, H., Chen, D., Li, X., Tang, Y., Shah, T., and Ranjan, R. (2018). Towards Brain Big Data Classification: Epileptic EEG Identification with a Lightweight VGGNet on Global MIC. *IEEE Access*, 6:14722–14733.

Kelly, K. M., Shiau, D. S., Kern, R. T., Chien, J. H., Yang, M. C., Yandora, K. A., Valeriano, J. P., Halford, J. J., and Sackellares, J. C. (2010). Assessment of a scalp EEG-based automated seizure detection system. *Clin. Neurophysiol.*, 121(11):1832–1843.

Keskar, N. S., Nocedal, J., Tang, P. T. P., Mudigere, D., and Smelyanskiy, M. (2019). On large-batch training for deep learning: Generalization gap and sharp minima. In *5th Int. Conf. Learn. Represent. ICLR 2017*, pages 1–16.

Khan, A. T. and Khan, Y. U. (2017). Time Domain based Seizure Onset Analysis of Brain Signatures in Pediatric EEG. In *4th Int. Conf. "Computing Sustain. Glob. Dev.*, pages 5813–5818.

Khan, Y. U., Rafiuddin, N., and Farooq, O. (2012). Automated seizure detection in scalp EEG using multiple wavelet scales. In *2012 IEEE Int. Conf. signal Process. Comput. Control*, pages 1–5. IEEE.

Khanmohammadi, S. and Chou, C. A. (2018). Adaptive Seizure Onset Detection Framework Using a Hybrid PCA-CSP Approach. *IEEE J. Biomed. Heal. Informatics*, 22(1):154–160.

Kharbouch, A., Shoeb, A., Guttag, J., and Cash, S. S. (2011). An algorithm for seizure onset detection using intracranial EEG. *Epilepsy Behav.*, 22(Suppl. 1):S29–S35.

Khatun, S., Mahajan, R., and Morshed, B. I. (2016). Comparative Study of Wavelet-Based Unsupervised Ocular Artifact Removal Techniques for Single-Channel EEG Data. *IEEE J. Transl. Eng. Heal. Med.*, 4:1–8.

King, G. and Zeng, L. (2001). Society for Political Methodology Logistic Regression in Rare Events Data. *Polit. Anal.*, 9(2):137–163.

King, M. A., Newton, M. R., Jackson, G. D., Fitt, G. J., Mitchell, L. A., Silvapulle, M. J., and Berkovic, S. F. (1998). Epileptology of the first-seizure presentation: A clinical,

electroencephalographic, and magnetic resonance imaging study of 300 consecutive patients. *Lancet*, 352(9133):1007–1011.

Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd Int. Conf. Learn. Represent. ICLR 2015*, pages 1–15.

Kiranyaz, S., Ince, T., Zabihi, M., and Ince, D. (2014). Automated patient-specific classification of long-term Electroencephalography. *J. Biomed. Inform.*, 49:16–31.

Kiymik, M. K., Güler, I., Dizibüyük, A., and Akin, M. (2005). Comparison of STFT and wavelet transform methods in determining epileptic seizure activity in EEG signals for real-time application. *Comput. Biol. Med.*, 35(7):603–616.

Kjaer, T. W., Sorensen, H. B. D., Groenborg, S., Pedersen, C. R., and Duun-Henriksen, J. (2017). Detection of Paroxysms in Long-Term, Single Channel EEG-Monitoring of Patients with Typical Absence Seizure. *IEEE J. Transl. Eng. Heal. Med.*, 5:1–8.

Klados, M. A., Papadelis, C. L., and Bamidis, P. D. (2009). REG-ICA: A new hybrid method for EOG artifact rejection. In *2009 9th Int. Conf. Inf. Technol. Appl. Biomed.*, pages 5–7.

Klatt, J., Feldwisch-Drentrup, H., Ihle, M., Navarro, V., Neufang, M., Teixeira, C., Adam, C., Valderrama, M., Alvarado-Rojas, C., Witon, A., Le Van Quyen, M., Sales, F., Dourado, A., Timmer, J., Schulze-Bonhage, A., and Schelter, B. (2012). The EPILEPSIAE database: An extensive electroencephalography database of epilepsy patients. *Epilepsia*, 53(9):1669–1676.

Klimesch, W., Sauseng, P., and Hanslmayr, S. (2007). EEG alpha oscillations: The inhibition-timing hypothesis. *Brain Res. Rev.*, 53(1):63–88.

Krumholz, A., Wiebe, S., Gronseth, G., Shinnar, S., Levisohn, P., Ting, T., Hopp, J., Shafer, P., Morris, H., Seiden, L., Barkley, G., and French, J. (2007). Evaluating an Apparent Unprovoked First Seizure in Adults (An Evidence-Based Review). *Neurology*, 69(21):1996–2007.

Larsen, S. N., Conradsen, I., Beniczky, S., and Sorensen, H. B. (2014). Detection of tonic epileptic seizures based on surface electromyography. In *2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 942–945.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., and Lance, B. J. (2018). EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces. *J. Neural Eng.*, 15(5):1–30.

Lebart, L., Morineau, A., and Warwick, K. M. (1984). *Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related techniques for Large Matrices*. John Wiley & Sons, New York.

Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lee, G. R., Gommers, R., Wasilewski, F., Wohlfahrt, K., and O'Leary, A. (2019). PyWavelets: A Python package for wavelet analysis. *J. Open Source Softw.*, 4(36):1237.

Lee, T. (1998). Independent component analysis. In *Indep. Compon. Anal. Theory Appl.*, pages 27–66. Springer, New York, NY.

Lemaitre, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res.*, 18(17):1–5.

Li, D., Zhou, W., Drury, I. V. O., and Savit, R. (2003). Linear and Nonlinear Measures and Seizure Anticipation. *J. Comput. Neurosci.*, 15(3):335–345.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.*, 18:1–52.

Li, Y., Liu, Y., Cui, W. G., Guo, Y. Z., Huang, H., and Hu, Z. Y. (2020). Epileptic Seizure Detection in EEG Signals Using a Unified Temporal-Spectral Squeeze-and-Excitation Network. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 28(4):782–794.

Liang, S.-f., Chang, W.-l., and Chiueh, H. (2010). EEG-based Absence Seizure Detection Methods. In *2010 Int. Jt. Conf. Neural Networks*, pages 1–4.

Liang, X. (2019). Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. *Comput. Civ. Infrastruct. Eng.*, 34(5):415–430.

Liao, C. Y., Chen, R. C., and Tai, S. K. (2018). Emotion stress detection using EEG signal and deep learning technologies. In *2018 IEEE Int. Conf. Appl. Syst. Invent.*, pages 90–93. IEEE.

Light, G. A., Williams, L. E., Minow, F., Sprock, J., Rissling, A., Sharp, R., Swerdlow, N. R., and Braff, D. L. (2010). Electroencephalography (EEG) and event-related potentials (ERPs) with human participants. *Curr. Protoc. Neurosci.*, 52(1):6–25.

Lindauer, M., Eggensperger, K., Feurer, M., Biedenkapp, A., Marben, J., Müller, P., and Hutter, F. (2019). BOAH: A Tool Suite for Multi-Fidelity Bayesian Optimization & Analysis of Hyperparameters. arXiv:1908.06756.

Liu, H., Xi, L., Zhao, Y., and Li, Z. (2019). Using Deep Learning and Machine Learning to Detect Epileptic Seizure with Electroencephalography (EEG) Data. *Mach. Learn. Res.*, 4(3):39.

Liu, H. S., Zhang, T., and Yang, F. S. (2002). A multistage, multimethod approach for automatic detection and classification of epileptiform EEG. *IEEE Trans. Biomed. Eng.*, 49(12):1557–1566.

Liu, T., Truong, N. D., Nikpour, A., Zhou, L., and Kavehei, O. (2020). Epileptic Seizure Classification with Symmetric and Hybrid Bilinear Models. *IEEE J. Biomed. Heal. Informatics*, pages 1–1.

Liu, X.-y., Wu, J., and Zhou, Z.-h. (2008). Exploratory Undersampling for Class-Imbalance Learning. *IEEE Trans. Syst. Man, Cybern. Part B*, 39(2):539–550.

Logesparan, L., Casson, A. J., and Rodriguez-Villegas, E. (2012). Optimal features for online seizure detection. *Med. Biol. Eng. Comput.*, 50(7):659–669.

Lopes Da Silva, F. H. (1978). Analysis of EEG non-stationarities. *Electroencephalogr. Clin. Neurophysiol. Suppl.*, (34):163–79.

Lu, W. and Rajapakse, J. C. (2000). Constrained independent component analysis. *Adv. neural Inf. Process. Syst.*, 10:570–576.

Lu, W. and Rajapakse, J. C. (2005). Approach and applications of constrained ICA. *IEEE Trans. Neural Networks*, 16(1):203–212.

Lu, W. and Rajapakse, J. C. (2006). ICA with Reference. *Neurocomputing*, 69(16-18):2244–2257.

Luck, S. J. (2014a). A Closer Look at Averaging: Convolution, Latency Variability, and Overlap. In *An Introd. to Event-Related Potential Tech.*, chapter 11. Online edition.

Luck, S. J. (2014b). Basics of Fourier Analysis and Filtering. In *An Introd. to event-related potential Tech.*, chapter 7, pages 219–248. 2nd edition.

Luck, S. J. (2014c). Time and Frequency: A Closer Look at Filtering and Time-Frequency Analysis. In *An Introd. to Event-Related Potential Tech.*, chapter 12. Online edition.

Mahajan, R. and Morshed, B. I. (2015). Unsupervised eye blink artifact denoising of EEG data with modified multiscale sample entropy, kurtosis, and wavelet-ICA. *IEEE J. Biomed. Heal. Informatics*, 19(1):158–165.

Mammone, N. and Morabito, F. C. (2014). Enhanced automatic wavelet independent component analysis for electroencephalographic artifact removal. *Entropy*, 16(12):6553–6572.

Mamun, M., Al-Kadi, M., and Marufuzzaman, M. (2013). Effectiveness of wavelet denoising on electroencephalogram signals. *J. Appl. Res. Technol.*, 11(1):156–160.

Manoranjan, P. and Parvez, M. Z. (2015). Epileptic seizure detection by exploiting temporal correlation of electroencephalogram signals. *IET Signal Process.*, 9(6):467–475.

Manzouri, F., Heller, S., Dümpelmann, M., Woias, P., and Schulze-Bonhage, A. (2018). A Comparison of Machine Learning Classifiers for Energy-Efficient Implementation of Seizure Detection. *Front. Syst. Neurosci.*, 12:43.

Marcilly, R., Peute, L., and Beuscart-Zéphir, M.-C. (2016). From usability engineering to evidence-based usability in health IT. In Ammenwerth, E. and Rigby, M., editors, *Evidence-Based Heal. Informatics Promot. Saf. Effic. Through Sci. Methods Ethical Policy Ed.*, pages 126–38. IOS Press.

Mariani, E., Rossi, L. N., and Vajani, S. (2011). Interictal paroxysmal EEG abnormalities in childhood absence epilepsy. *Seizure*, 20(4):299–304.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5(4):115–133.

McDermott, M. B., Wang, S., Marinsek, N., Ranganath, R., Ghassemi, M., and Foschini, L. (2019). Reproducibility in machine learning for health. In *2019 Reprod. Mach. Learn. RML@ICLR 2019 Work.*

McInnes, L. and Healy, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *J. Open Source Softw.*, 3(29):2–3.

McInnes, L., Healy, J., Saul, N., and Grossberger, L. (2018). *UMAP*. Retrieved from https://github.com/lmcinnes/umap.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In *Proc. 9th Python Sci. Conf.*, pages 51–56.

Mincholé, A., Camps, J., Lyon, A., and Rodríguez, B. (2019). Machine learning in the electrocardiogram. *J. Electrocardiol.*, 57:S61–S64.

Mincholé, A. and Rodriguez, B. (2019). Artificial intelligence for the electrocardiogram. *Nat. Med.*, 25(1):22–23.

Minsky, M. and Papert, S. (1969). *Perceptrons*. M.I.T. Press.

Mirarchi, D., Vizza, P., Cinaglia, P., Tradigo, G., and Veltri, P. (2017). MEEG: A system for electroencephalogram data management and analysis. In *2017 IEEE Int. Conf. Bioinforma. Biomed.*, pages 1642–1646.

Mitha, M., Shiju, S. S., and Viswanadhan, M. (2014). Automated epileptic seizure detection using relevant features in support vector machines. *2014 Int. Conf. Control. Instrumentation, Commun. Comput. Technol. ICCICCT 2014*, pages 1000–1004.

Mitra, P. P. and Pesaran, B. (1999). Analysis of dynamic brain imaging data. *Biophys. J.*, 76(2):691–708.

Mohammadpoory, Z., Nasrolahzadeh, M., and Haddadnia, J. (2017). Epileptic seizure detection in EEGs signals based on the weighted visibility graph entropy. *Seizure*, 50:202–208.

Mohanraj, R., Norrie, J., Stephen, L. J., Kelly, K., Hitiris, N., and Brodie, M. J. (2006). Mortality in adults with newly diagnosed and chronic epilepsy: a retrospective comparative study. *Lancet Neurol.*, 5(6):481–487.

Mohr, D. C., Zhang, M., and Schueller, S. M. (2017). Personal Sensing: Understanding Mental Health Using Ubiquitous Sensors and Machine Learning. *Annu. Rev. Clin. Psychol.*, 13(1):23–47.

Mojsilovic, A., Markovic, S., and Popovic, M. (1997). Texture analysis and classification with the nonseparable wavelet transform. *IEEE Int. Conf. Image Process.*, 3(4):182–185.

Montez, T., Poil, S.-S., Jones, B. F., Manshanden, I., Verbunt, J. P. A., van Dijk, B. W., Brussaard, A. B., van Ooyen, A., Stam, C. J., Scheltens, P., and Linkenkaer-Hansen, K. (2009). Altered temporal correlations in parietal alpha and prefrontal theta oscillations in early-stage Alzheimer disease. *Proc. Natl. Acad. Sci.*, 106(5):1614–1619.

Muhammad, G., Masud, M., Amin, S. U., Alrobaea, R., and Alhamid, M. F. (2018). Automatic seizure detection in a mobile multimedia framework. *IEEE Access*, 6:45372–45383.

Nagai, A., Hirata, M., Kamatani, Y., Muto, K., and Matsuda, K. (2017). Overview of the BioBank Japan Project: Study design and profile. *J. Epidemiol.*, 27:S2–S8.

Nandy, A., Alahe, M. A., Nasim Uddin, S. M., Alam, S., Nahid, A. A., and Awal, M. A. (2019). Feature extraction and classification of EEG signals for seizure detection. In *2019 Int. Conf. Robot. Electr. Signal Process. Tech.*, pages 480–485. IEEE.

Nanobashvili, Z. I., Chachua, T. R., Bilanishvili, I. G., Khizanishvili, N. A., Nebieridze, N. G., and Koreli, A. G. (2011). Peculiarities of the effects of stimulation of emotiogenic central structures under conditions of a kindling model of epilepsy. *Neurophysiology*, 43(4):292–298.

Nasehi, S. and Pourghassem, H. (2012). Seizure detection algorithms based on analysis of EEG and ECG signals: A survey. *Neurophysiology*, 44(2):174–186.

Nasehi, S. and Pourghassem, H. (2013). Patient-specific epileptic seizure onset detection algorithm based on spectral features and IPSONN classifier. In *2013 Int. Conf. Commun. Syst. Netw. Technol.*, pages 186–190. IEEE.

National Academies of Sciences Engineering and Medicine (2019). *Reproducibility and Replicability in Science*. The National Academies Press, Washington, DC.

Nestor, B., McDermott, M. B. A., Chauhan, G., Naumann, T., Hughes, M. C., Goldenberg, A., and Ghassemi, M. (2018). Rethinking clinical prediction: Why machine learning must consider year of care and feature aggregation. In *Mach. Learn. Heal. Work. NeurIPS 2018*, pages 1–7.

NeuPy (2019). *Hyperparameter optimization for Neural Networks*. Retrieved from http://neupy.com/2016/12/17/hyperparameter_optimization_for_neural_networks.html#tree-structured-parzen-estimators-tpe.

Nguyen, V., Schulze, S., and Osborne, M. A. (2019). Bayesian Optimization for Iterative Learning. arXiv:1909.09593.

NICE Clinical Guidelines and Evidence Review for the Epilepsies (2004). Appendix G The costs of epilepsy misdiagnosis.

Niedermeyer, E. and Da Silva, F. (1999). *Electroencephalography: Basic principles, clinical applications, and related fields*. Williams and Wilkins, 4th edition.

NVIDIA, Vingelmann, P., and Fitzek, F. H. (2019). CUDA, release: 10.1.

Obeid, I. and Picone, J. (2016). The Temple University Hospital EEG Data Corpus. *Front. Neurosci.*, 10:196.

Ocak, H. (2009). Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy. *Expert Syst. Appl.*, 36(2):2027–2036.

Ochal, D., Rahman, S., Ferrell, S., Elseify, T., Obeid, I., and Picone, J. (2020). The Temple University Hospital EEG Corpus: Annotation Guidelines.

Olund, T., Duun-Henriksen, J., Kjaer, T. W., and Sorensen, H. B. D. (2014). Automatic detection and classification of artifacts in single-channel EEG. In *2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 922–925.

Omerhodzic, I., Avdakovic, S., Nuhanovic, A., and Dizdarevic, K. (2013). Energy Distribution of EEG Signals: EEG Signal Wavelet-Neural Network Classifier. arXiv:1307.7897.

Opp, J., Wenzel, D., and Brandl, U. (1992). Visuomotor Coordination During Focal and Generalized EEG Discharges. *Epilepsia*, 33(5):836–840.

Orellana, M. P. and Cerqueira, F. (2016). Personalized epilepsy seizure detection using random forest classification over one-dimension transformed EEG data. *bioRxiv*, page 070300.

Orosco, L., Correa, A. G., Diez, P., and Laciar, E. (2016). Patient non-specific algorithm for seizures detection in scalp EEG. *Comput. Biol. Med.*, 71:128–134.

O'Shea, A., Lightbody, G., Boylan, G., and Temko, A. (2017). Neonatal seizure detection using convolutional neural networks. In *2017 IEEE 27th Int. Work. Mach. Learn. Signal Process.*, pages 1–6.

Osorio, I., Frei, F. G., and Wilkinson, S. B. (1998). Real-time automated detection and quantitative analysis of seizures and short-term prediction of clinical onset. *Epilepsia*, 39(6):615–627.

Oweis, R. J. and Abdulhay, E. W. (2011). Seizure classification in EEG signals utilizing Hilbert-Huang transform. *Biomed. Eng. Online*, 10:1–15.

Page, A., Shea, C., and Mohsenin, T. (2016). Wearable seizure detection using convolutional neural networks with transfer learning. In *2016 IEEE Int. Symp. Circuits Syst.*, pages 1086–1089. IEEE.

Päivinen, N., Lammi, S., Pitkänen, A., Nissinen, J., Penttonen, M., and Grönfors, T. (2005). Epileptic seizure detection: A nonlinear viewpoint. *Comput. Methods Programs Biomed.*, 79(2):151–159.

Papoulis, A. (1991). *Probability, random variables, and stochastic processes*. McGraw-Hill, New York, 3rd edition.

Park, C., Choi, G., Kim, J., Kim, S., Kim, T.-j., Min, K., Jung, K., and Chong, J. (2018). Epileptic Seizure Detection for Multi-channel EEG with Deep Convolutional Neural Network. In *2018 Int. Conf. Electron. Information, Commun.*, pages 1–5.

Parker, T. S. and Chua, L. (2012). *Practical numerical algorithms for chaotic systems*. Springer Science & Business Media.

Parks, T. and Burrus, C. (1987). *Digital Filter Design*. Wiley-Interscience, New York.

Parvez, M. Z. and Paul, M. (2016). Epileptic seizure prediction by exploiting spatiotemporal relationship of EEG signals using phase correlation. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 24(1):158–168.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Conf. Rec. Asilomar Conf. Signals, Syst. Comput.*, 1:40–44.

Paulose, A. and Bedeeuzzaman, M. (2014). Seizure detection using median based feature. In *2014 First Int. Conf. Comput. Syst. Commun.*, pages 328–332. IEEE.

Pauri, F., Pierelli, F., Chatrian, G.-E., and Erdly, W. W. (1992). Long-term EEG-video-audio monitoring: computer detection of focal EEG seizure patterns. *Electroencephalogr. Clin. Neurophysiol.*, 82(1):1–9.

Paz, J. T. and Huguenard, J. R. (2014). Optogenetics and epilepsy: Past, present and future. *Epilepsy Curr.*, 15(1):34–38.

Pediaditis, M., Tsiknakis, M., Koumakis, L., Karachaliou, M., Voutoufianakis, S., and Vorgia, P. (2012). Vision-Based Absence Seizure Detection. *2012 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 65–68.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830.

Perera, N. D., Madarasingha, C., and De Silva, A. C. (2017). Spatial Feature Reduction in Long-term EEG for Patient-specific Epileptic Seizure Event Detection. In *Proc. 9th Int. Conf. Signal Process. Syst.*, pages 230–234.

Petersen, E. B., Duun-Henriksen, J., Mazzaretto, A., Kjar, T. W., Thomsen, C. E., and Sorensen, H. B. D. (2011). Generic single-channel detection of absence seizures. In *2011 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 4820–4823.

Picone, J. and Obeid, I. (2016). *Temple University Hospital EEG Corpus*. Neural Engineering Data Consortium. Retrieved from www.nedcdata.org.

Polat, H. and Ozerdem, M. S. (2016). Epileptic seizure detection from EEG signals by using wavelet and Hilbert transform. In *2016 XII Int. Conf. Perspect. Technol. Methods MEMS Des.*, pages 66–69. Lviv Polytechnic National University.

Polat, K. and Güneş, S. (2007). Classification of epileptiform EEG using a hybrid system

based on decision tree classifier and fast Fourier transform. *Appl. Math. Comput.*, 187(2):1017–1026.

Pollard, T. J., Johnson, A. E., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. (2018). The eICU collaborative research database, a freely available multi-center database for critical care research. *Sci. Data*, 5:1–13.

Pontifex, M. B., Gwizdala, K. L., Parks, A. C., Billinger, M., and Brunner, C. (2017). Variability of ICA decomposition may impact EEG signals when used to remove eyeblink artifacts. *Psychophysiology*, 54(3):386–398.

Pramod, S., Page, A., Mohsenin, T., and Oates, T. (2014). Detecting Epileptic Seizures from EEG Data using Neural Networks. arXiv:1412.6502.

Proakis, J. G. and Manolakis, D. G. (2006). *Digital Signal Processing*. Pearson, 4 edition.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. In *Adv. Neural Inf. Process. Syst.*, pages 6638–6648.

Quinlan, J. (2014). *C4.5: programs for machine learning*. Elsevier.

Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1(1):81–106.

Rafiuddin, N., Khan, Y. U., and Farooq, O. (2011). Feature extraction and classification of EEG for automatic seizure detection. In *2011 Int. Conf. Multimedia, Signal Process. Commun. Technol.*, pages 184–187. IEEE.

Raghu, S., Sriraam, N., and Kumar, G. P. (2017). Classification of epileptic seizures using wavelet packet log energy and norm entropies with recurrent Elman neural network classifier. *Cogn. Neurodyn.*, 11(1):51–66.

Raja, C. and Gangatharan, N. (2015). Appropriate sub-band selection in wavelet packet decomposition for automated glaucoma diagnoses. *Int. J. Autom. Comput.*, 12(4):393–401.

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., Sun, M., Sundberg, P., Yee, H., Zhang, K., Zhang, Y., Flores, G., Duggan, G. E., Irvine, J., Le, Q., Litsch, K., Mossin, A., Tansuwan, J., Wang, D., Wexler, J., Wilson, J., Ludwig, D., Volchenboum, S. L., Chou, K., Pearson, M., Madabushi, S., Shah, N. H., Butte, A. J., Howell, M. D., Cui, C., Corrado, G. S., and Dean, J. (2018). Scalable and accurate deep learning with electronic health records. *npj Digit. Med.*, 1(1):18.

Ramadhani, I., Saputro, D., Maryati, N. D., and Solihati, S. R. (2019). Seizure Type Classification on EEG Signal using Support Vector Machine Seizure Type Classification on EEG Signal using Support Vector Machine. *J. Phys. Conf. Ser.*, 1201(1):012065.

Ramakrishnan, S. and Muthanantha Murugavel, A. S. (2018). Epileptic seizure detection using fuzzy-rules-based sub-band specific features and layered multi-class SVM. *Pattern Anal. Appl.*, 22(3):1161–1176.

Rao, R. and Bopardikar, A. (1998). *Wavelet transforms: Introduction to theory and applications*. Addison-Wesley Longman, Inc.

Raschka, S. and Mirjalili, V. (2019). *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd, third edition.

Richman, J. S. and Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy maturity in premature infants Physiological time-series analysis using approximate entropy and sample entropy. *Am. J. Physiol. Hear. Circ. Physiol.*, 278(6):H2039–H2049.

Romero, S., Mañanas, M. A., and Barbanoj, M. J. (2008). A comparative study of automatic techniques for ocular artifact reduction in spontaneous EEG signals based on clinical target variables: A simulation case. *Comput. Biol. Med.*, 38(3):348–360.

Ronner, H. E., Ponten, S. C., Stam, C. J., and Uitdehaag, B. M. (2009). Inter-observer

variability of the EEG diagnosis of seizures in comatose patients. *Seizure*, 18(4):257–263.

Rosso, O. A., Figliola, A., Creso, J., and Serrano, E. (2004). Analysis of wavelet-filtered tonic-clonic electroencephalogram recordings. *Med. Biol. Eng. Comput.*, 42(4):516–523.

Rosso, O. A., Martin, M. T., Figliola, A., Keller, K., and Plastino, A. (2006). EEG analysis using wavelet-based information tools. *J. Neurosci. Methods*, 153(2):163–182.

Rothman, S. and Yang, X.-F. (2003). Local Cooling: A Therapy for Intractable Neocortical Epilepsy. *Epilepsy Curr.*, 3(5):153–156.

Roy, S., Asif, U., Tang, J., and Harrer, S. (2019a). Machine Learning for Seizure Type Classification: Setting the benchmark. arXiv:1902.01012.

Roy, Y., Hubert, B., Isabela, A., Alexandre, G., and Jocelyn, F. (2019b). Deep learning-based electroencephalography analysis: a systematic review. arXiv:1901.05498v2.

Ruffini, G., Ibañez, D., Castellano, M., Dubreuil, L., Gagnon, J.-F., Montplaisir, J., and Soria-Frisch, A. (2019). Deep learning with EEG spectrograms in rapid eye movement behavior disorder. *Front. Neurol.*, 10:806.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.*, 115(3):211–252.

Safieddine, D., Kachenoura, A., Albera, L., Birot, G., Karfoul, A., Pasnicu, A., Biraben, A., Wendling, F., Senhadji, L., and Merlet, I. (2012). Removal of muscle artifact from EEG data: Comparison between stochastic (ICA and CCA) and deterministic (EMD and wavelet-based) approaches. *EURASIP J. Adv. Signal Process.*, 127.

Sahu, R., Dash, S. R., Cacha, L. A., Poznanski, R. R., and Parida, S. (2020). Epileptic seizure detection: A comparative study between deep and traditional machine learning techniques. *J. Integr. Neurosci.*, 19(1):1–9.

Sakkalis, V., Cassar, T., Zervakis, M., Camilleri, K. P., Fabri, S. G., Bigan, C., Karakonstantaki, E., and Micheloyannis, S. (2008). Parametric and nonparametric EEG analysis for the evaluation of EEG activity in young children with controlled epilepsy. *Comput. Intell. Neurosci.*, 2008.

Sakkalis, V., Giannakakis, G., Farmaki, C., Mousas, A., Pediaditis, M., Vorgia, P., and Tsiknakis, M. (2013). Absence Seizure Epilepsy Detection using linear and nonlinear EEG analysis methods. In *2013 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 6333–6336.

Sakkalis, V., Zervakis, M., and Micheloyannis, S. (2006). Significant EEG features involved in mathematical reasoning: Evidence from wavelet analysis. *Brain Topogr.*, 19(1-2):53–60.

Samiee, K., Kovács, P., and Gabbouj, M. (2017). Epileptic seizure detection in long-term EEG records using sparse rational decomposition and local Gabor binary patterns feature extraction. *Knowledge-Based Syst.*, 118:228–240.

Sankari, Z., Adeli, H., and Adeli, A. (2012). Wavelet coherence model for diagnosis of Alzheimer disease. *Clin. EEG Neurosci.*, 43(4):268–78.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? *Adv. Neural Inf. Process. Syst.*, pages 2483–2493.

Sarvas, J. (1987). Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem. *Phys. Med. Biol.*, 32(1):11.

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., and Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Hum. Brain Mapp.*, 38(11):5391–5420.

Schulze-Bonhage, A., Sales, F., Wagner, K., Teotonio, R., Carius, A., Schelle, A., and Ihle, M. (2010). Views of patients with epilepsy on seizure prediction devices. *Epilepsy Behav.*, 18(4):388–396.

Schwabedal, J. T. C., Snyder, J. C., Cakmak, A., Nemati, S., and Clifford, G. D. (2018). Addressing Class Imbalance in Classification Problems of Noisy Signals by using Fourier Transform Surrogates. arXiv:1806.08675.

Scikit-learn (2019). *Ensemble*. Retrieved from https://scikit-learn.org/stable/modules/ensemble.html.

Seiffert, C., Khoshgoftaar, T. M., Hulse, J. V., and Napolitano, A. (2009). RUSBoost : A Hybrid Approach to Alleviating Class Imbalance. In *IEEE Trans. Syst. Man, Cybern. A Syst. Humans*, volume 40, pages 185–197.

Seiffert, C., Khoshgoftaar, T. M., and Raton, B. (2008). RUSBoost: Improving Classification Performance when Training Data is Skewed. In *19th Int. Conf. Pattern Recognit.*, pages 8–11.

Sejnowski, T. J., Lee, T.-W., and Girolami, M. (1999). Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural Comput.*, 11(2):417–441.

Selvakumari, R. S., Mahalakshmi, M., and Prashalee, P. (2019). Patient-Specific Seizure Detection Method using Hybrid Classifier with Optimized Electrodes. *J. Med. Syst.*, 43(5):121.

Selvathi, D. and Meera, V. K. (2018). Realization of epileptic seizure detection in EEG signal using wavelet transform and SVM classifier. In *2017 Int. Conf. Signal Process. Commun.*, pages 18–22.

Sendi, M. S., Heydarzadeh, M., and Mahmoudi, B. (2018). A Spark-based Analytic Pipeline for Seizure Detection in EEG Big Data Streams. *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, 2018-July:4003–4006.

Settles, B. (2010). Active Learning Literature Survey. Technical Report 55-66.

Shah, V., Golmohammadi, M., Ziyabari, S., Von Weltin, E., Obeid, I., and Picone, J. (2017). Optimizing channel selection for seizure detection. In *2017 IEEE Signal Process. Med. Biol. Symp.*, pages 1–5.

Shah, V., von Weltin, E., Lopez, S., McHugh, J. R., Veloso, L., Golmohammadi, M., Obeid, I., and Picone, J. (2018). The Temple University Hospital Seizure Detection Corpus. *Front. Neuroinform.*, 12:1–6.

Shanir, P. M., U. Khan, Y., and Farooq, O. (2015). Time Domain Analysis of EEG for Automatic Seizure Detection. In *Natl. Conf. Emerg. Trends Electr. Electron. Eng.*, pages 1–5.

Shanir, P. P., Khan, K. A., Khan, Y. U., Farooq, O., and Adeli, H. (2018). Automatic Seizure Detection Based on Morphological Features Using One-Dimensional Local Binary Pattern on Long-Term EEG. *Clin. EEG Neurosci.*, 49(5):351–362.

Shi, L. C., Duan, R. N., and Lu, B. L. (2013). A robust principal component analysis algorithm for EEG-based vigilance estimation. In *2013 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 6623–6626. IEEE.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Adv. Neural Inf. Process. Syst.*, pages 802–810.

Shoeb, A., Edwards, H., Connolly, J., Bourgeois, B., Ted Treves, S., and Guttag, J. (2004). Patient-specific seizure onset detection. *Epilepsy Behav.*, 5(4):483–498.

Shoeb, A. and Guttag, J. (2010). Application of machine learning to epileptic seizure detection. In *Proc. 27th Int. Conf. Mach. Learn.*, pages 975–982.

Shumway, R. H. and Stoffer, D. S. (2017). *Time Series Analysis and Its Applications*. Springer.

Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Solaija, M. S. J., Saleem, S., Khurshid, K., Hassan, S. A., and Kamboh, A. M. (2018). Dynamic mode decomposition based epileptic seizure detection from scalp EEG. *IEEE Access*, 6:38683–38692.

Sopic, D., Aminifar, A., and Atienza, D. (2018). E-Glass: A Wearable System for Real-Time Detection of Epileptic Seizures. In *2018 IEEE Int. Symp. Circuits Syst.*, pages 1–5. IEEE.

Sörnmo, L. and Laguna, P. (2005). *Bioelectrical signal processing in cardiac and neurological applications*. Academic Press.

Sors, A., Bonnet, S., Mirek, S., Vercueil, L., and Payen, J. F. (2018). A convolutional neural network for sleep stage scoring from raw single-channel EEG. *Biomed. Signal Process. Control*, 42:107–114.

Spriggs, W. (2009). *Essentials of polysomnography*. Jones & Bartlett Publishers.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfittin. *J. Mach. Learn. Res.*, 15:1929–1958.

Stead, M., Bower, M., Brinkmann, B. H., Lee, K., Marsh, W. R., Meyer, F. B., Litt, B., Van Gompel, J., and Worrell, G. A. (2010). Microseizures and the spatiotemporal scales of human partial epilepsy. *Brain*, 133(9):2789–2797.

Steriade, M. (2006). Grouping of brain rhythms in corticothalamic systems. *Neuroscience*, 137(4):1087–1106.

Stober, S., Cameron, D. J., and Grahn, J. a. (2014). Using Convolutional Neural Networks to Recognize Rhythm Stimuli from Electroencephalography Recordings. *Neural Inf. Process. Syst. 2014*, pages 1–9.

Stober, S., Sternin, A., Owen, A. M., and Grahn, J. A. (2015). Deep Feature Learning for EEG Recordings. arXiv:1511.04306.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proc. 57th Annu. Meet. Assoc. Comput. Linguist.*, pages 3645–3650. Association for Computational Linguistics.

Subasi, A. (2007a). Application of adaptive neuro-fuzzy inference system for epileptic seizure detection using wavelet feature extraction. *Comput. Biol. Med.*, 37(2):227–244.

Subasi, A. (2007b). EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Syst. Appl.*, 32(4):1084–1093.

Subhash Chandran, K. S., Mishra, A., Shirhatti, V., and Ray, S. (2016). Comparison of matching pursuit algorithm with other signal processing techniques for computation of the time-frequency power spectrum of brain signals. *J. Neurosci.*, 36(12):3399–3408.

Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., Liu, B., Matthews, P., Ong, G., Pell, J., Silman, A., Young, A., Sprosen, T., Peakman, T., and Collins, R. (2015). UK Biobank: An Open Access Resource for Identifying the Causes of a Wide Range of Complex Diseases of Middle and Old Age. *PLoS Med.*, 12(3):1–10.

Sun, M., Scher, M. S., Dahl, R. E., Ryan, N. D., Iyengar, S., Kosanovic, B., and Sclabassi, R. J. (1993). Analysis of aliasing and quantization problems in EEG data acquisition. In *1993 Proc. Twelfth South. Biomed. Eng. Conf.*, pages 280–282.

Supratak, A., Li, L., and Guo, Y. (2014). Feature extraction with stacked autoencoders for epileptic seizure detection. In *2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 4184–4187. IEEE.

Swartz Center for Computational Neuroscience (2018). *Firfilt FAQ*. Retrieved from https://sccn.ucsd.edu/wiki/Firfilt_FAQ.

Sweeney, K., Ward, T., and McLoone, S. (2012). Artifact Removal in Physiological Signals - Practices and Possibilities. *IEEE Trans. Inf. Technol. Biomed.*, 16(3):488–500.

Synced (2019). *The Staggering Cost of Training SOTA AI Models.* Medium. Retrieved from https://medium.com/syncedreview/the-staggering-cost-of-training-sota-ai-models-e329e80fa82.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper with Convolutions. In *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, pages 1–9.

Takens, F. (1981). Detecting strange attractors in turbulence. In Rand, D. and Young, L.-S., editors, *Dyn. Syst. Turbul. Warwick 1980*, pages 366–381. Springer, Berlin.

Tanaka, M., Olsen, R. W., Medina, M. T., Schwartz, E., Alonso, M. E., Duron, R. M., Castro-Ortega, R., Martinez-Juarez, I. E., Pascual-Castroviejo, I., Machado-Salas, J., Silva, R., Bailey, J. N., Bai, D., Ochoa, A., Jara-Prado, A., Pineda, G., Macdonald, R. L., and Delgado-Escueta, A. V. (2008). Hyperglycosylation and Reduced GABA Currents of Mutated GABRB3 Polypeptide in Remitting Childhood Absence Epilepsy. *Am. J. Hum. Genet.*, 82(6):1249–1261.

Tatum, W. O., Ho, S., and Benbadis, S. (2010). Polyspike Ictal Onset Absence Seizures. *J. Clin. Neurophysiol.*, 27(2):93–99.

Tay, J., Toh, S., Leow, L., and Senin, S. (2017). Assessing competency of Z3Score automated sleep stage scoring system with manual sleep stage scoring by multiple scorers. *Sleep Med.*, 40(2017):e326.

Teixeira, A. R., Tomé, A. M., Stadlthanner, K., and Lang, E. W. (2008). KPCA denoising and the pre-image problem revisited. *Digit. Signal Process. A Rev. J.*, 18(4):568–580.

Teo, J., Hou, C. L., and Mountstephens, J. (2018). Preference classification using Electroencephalography (EEG) and deep learning. *J. Telecommun. Electron. Comput. Eng.*, 10(1-11):87–91.

Teplan, M. (2002). Fundamentals of EEG measurement. *Meas. Sci. Rev.*, 2(2):1–11.

The MathWorks Inc. (2020). *wavefun.* Retrieved from https://www.mathworks.com/help/wavelet/ref/wavefun.html.

Theodore, W. H. and Fisher, R. (2004). Brain stimulation for epilepsy. *Lancet Neurol.*, 3(2):111–118.

Thodoroff, P., Pineau, J., and Lim, A. (2016). Learning Robust Features using Deep Learning for Automatic Seizure Detection. In *Proc. Mach. Learn. Healthc. 2016*, pages 178–190.

Thomas Sheffield, L. (1987). Computer-aided electrocardiography. *J. Am. Coll. Cardiol.*, 10(2):448–455.

Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. (2020). The Computational Limits of Deep Learning. arXiv:2007.05558.

Thomsen, K., Iversen, L., Titlestad, T. L., and Winther, O. (2020). Systematic review of machine learning for diagnosis and prognosis in dermatology. *J. Dermatolog. Treat.*, 31(5):496–510.

Thomson, D. J. (1982). Spectrum estimation and harmonic analysis. *Proc. IEEE*, 70(9):1055–1096.

Tjepkema-Cloostermans, M. C., de Carvalho, R. C., and van Putten, M. J. (2018). Deep learning for detection of focal epileptiform discharges from scalp EEG recordings. *Clin. Neurophysiol.*, 129(10):2191–2196.

Topol, E. (2019). *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again.* Basic Books, New York.

Tovia, E., Goldberg-Stern, H., Shahar, E., and Kramer, U. (2006). Outcome of children with juvenile absence epilepsy. *J Child Neurol.*, 21(9):766–768.

Truong, N. D., Nguyen, A. D., Kuhlmann, L., Bonyadi, M. R., Yang, J., Ippolito, S., and Kavehei, O. (2018). Integer convolutional neural network for seizure detection. *IEEE J. Emerg. Sel. Top. Circuits Syst.*, 8(4):849–857.

Tsiouris, K. M., Pezoulas, V. C., Koutsouris, D. D., Zervakis, M., and Fotiadis, D. I. (2017).

Discrimination of Preictal and Interictal Brain States from Long-Term EEG Data. In *IEEE 30th Int. Symp. Comput. Med. Syst.*, pages 318–323.

Tyralis, H. and Papacharalampous, G. (2020). Boosting algorithms in energy research: A systematic review. arXiv:2004.07049.

Tzallas, A. T., Tsipouras, M. G., and Fotiadis, D. I. (2009). Epileptic Seizure Detection in EEGs Using Time-Frequency Analysis. {*IEEE*} *Trans. Inf. Technol. Biomed.*, 13(5):703–710.

Tzallas, A. T., Tsipouras, M. G., Tsalikakis, D. G., Karvounis, E. C., Astrakas, L., Konitsiotis, S., and Tzaphlidou, M. (2012). Automated Epileptic Seizure Detection Methods: A Review Study. In Stevanovic, D., editor, *Epilepsy-histological, Electroencephalogr. Psychol. Asp.*, pages 75–99.

Ulate-Campos, A., Coughlin, F., Gaínza-Lein, M., Fernández, I. S., Pearl, P. L., and Loddenkemper, T. (2016). Automated seizure detection systems and their effectiveness for each type of seizure. *Seizure*, 40:88–101.

Unser, M. and Aldroubi, A. (1996). A review of wavelets in biomedical applications. *Proc. IEEE*, 84(4):626–638.

Urigüen, J. A. and Garcia-Zapirain, B. (2015). EEG artifact removal - State-of-the-art and guidelines. *J. Neural Eng.*, 12(3):031001.

Uyulan, C. and Erguzel, T. (2016). Comparison of Wavelet Families for Mental Task Classification. *J. Neurobehav. Sci.*, 3(2):59.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. arXiv:1609.03499.

Van Der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng.*, 13(2):22–30.

Van Esbroeck, A., Smith, L., Syed, Z., Singh, S., and Karam, Z. (2016). Multi-task seizure detection: addressing intra-patient variation in seizure morphologies. *Mach. Learn.*, 102(3):309–321.

van Veen, F. and Leijnen, S. (2019). *The Neural Network Zoo.* Retrieved from https://www.asimovinstitute.org/neural-network-zoo.

van Vugt, M. K., Sederberg, P. B., and Kahana, M. J. (2007). Comparison of spectral analysis methods for characterizing brain oscillations. *J. Neurosci. Methods*, 162(1-2):49–63.

Vanabelle, P., Handschutter, P. D., Tahry, R. E., Benjelloun, M., and Boukhebouze, M. (2020). Epileptic Seizure Detection Using EEG Signals and Extreme Gradient Boosting. *J. Biomed. Res.*, 34(3):226–237.

VanRullen, R. (2011). Four common conceptual fallacies in mapping the time course of recognition. *Front. Psychol.*, 2:1–6.

Varsavsky, A., Mareels, I., and Cook, M. (2011a). *Epileptic Seizures and the EEG: Measurement, Models, Detection and Prediction.* CRC Press.

Varsavsky, A., Mareels, I., and Cook, M. (2011b). Signal Processing in EEG Analysis. In *Epileptic Seizures EEG Meas. Model. Detect. Predict.*, chapter 3, page 337. CRC Press.

Vidyaratne, L., Glandon, A., Alam, M., and Iftekharuddin, K. M. (2016). Deep recurrent neural network for seizure detection. In *Proc. Int. Jt. Conf. Neural Networks*, pages 1202–1207. IEEE.

Vidyaratne, L. S. and Iftekharuddin, K. M. (2017). Real-Time Epileptic Seizure Detection Using EEG. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 25(11):2146–2156.

Vigário, R. and Oja, E. (2008). BSS and ICA in Neuroinformatics: From Current Practices to Open Challenges. *IEEE Rev. Biomed. Eng.*, 1:50–61.

Vigário, R. N. (1997). Extraction of ocular artefacts from EEG using independent component analysis. *Electroencephalogr. Clin. Neurophysiol.*, 103(3):395–404.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., Vijaykumar, A., Bardelli, A. P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C. N., Fulton, C., Masson, C., Haggstrom, C., Fitzgerald, C., Nicholson, D. A., Hagen, D. R., Pasechnik, D. V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G. A., Ingold, G. L., Allen, G. E., Lee, G. R., Audren, H., Probst, I., Dietrich, J. P., Silterra, J., Webber, J. T., Slavic, J., Nothman, J., Buchner, J., Kulick, J., Schonberger, J. L., de Miranda Cardoso, J. V., Reimer, J., Harrington, J., Rodriguez, J. L. C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kummerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N. J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P. A., Lee, P., McGibbon, R. T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T. J., Robitaille, T. P., Spura, T., Jones, T. R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y. O., and Vazquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*, 17(3):261–272.

Volker, M., Schirrmeister, R. T., Fiederer, L. D., Burgard, W., and Ball, T. (2018). Deep transfer learning for error decoding from non-invasive EEG. In *2018 6th Int. Conf. Brain-Computer Interface*, pages 1–6.

Vollmer, S., Mateen, B. A., Bohner, G., Király, F. J., Ghani, R., Jonsson, P., Cumbers, S., Jonas, A., McAllister, K. S., Myles, P., Granger, D., Birse, M., Branson, R., Moons, K. G., Collins, G. S., Ioannidis, J. P., Holmes, C., and Hemingway, H. (2020). Machine learning and artificial intelligence research for patient benefit: 20 critical questions on transparency, replicability, ethics, and effectiveness. *BMJ*, 368:1–12.

von Stein, A. and Sarnthein, J. (2000). Different frequencies for different scales of cortical

integration: from local gamma to long range alpha/theta synchronization. *Int. J. Psychophysiol.*, 38:301–313.

Vos, D. M., Ri??s, S., Vanderperren, K., Vanrumste, B., Alario, F. X., Huffel, V. S., and Burle, B. (2010). Removal of muscle artifacts from EEG recordings of spoken language production. *Neuroinformatics*, 8(2):135–150.

Wagenaar, J. B., Brinkmann, B. H., Ives, Z., Worrell, G. A., and Litt, B. (2013). A multimodal platform for cloud-based collaborative research. In *2013 6th Int. IEEE/EMBS Conf. neural Eng.*, pages 1386–1389. IEEE.

Wagenaar, J. B., Worrell, G. A., Ives, Z., Dümpelmann, M., Litt, B., and Schulze-Bonhage, A. (2015). Collaborating and sharing data in Epilepsy Research. *J. Clin. Neurophysiol. Off. Publ. Am. Electroencephalogr. Soc.*, 32(3):235.

Wallace, B. C., Small, K., Brodley, C. E., and Trikalinos, T. A. (2011). Class imbalance, redux. In *2011 IEEE 11th Int. Conf. data Min.*, pages 754–763. IEEE.

Wang, F. and Ke, H. (2018). Global Epileptic Seizure Identification With Affinity Propagation Clustering Partition Mutual Information Using Cross-Layer Fully Connected Neural Network. *Front. Hum. Neurosci.*, 12:396.

Wang, H., Shi, W., and Choy, C. S. (2017). Integrating channel selection and feature selection in a real time epileptic seizure detection system. In *2017 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 3206–3211. IEEE.

Wang, J., Xu, J., and Wang, X. (2018). Combination of Hyperband and Bayesian Optimization for Hyperparameter Optimization in Deep Learning. arXiv:1801.01596.

Welch, P. D. (1967). The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms. *IEEE Trans. Audio Electroacoust.*, 15(2):70–73.

Wickerhauser, M. V. (1996). *Adapted Wavelet Analysis from Theory to Software*. CRC Press.

Widmann, A. and Schröger, E. (2012). Filter effects and filter artifacts in the analysis of electrophysiological data. *Front. Psychol.*, 3:1–5.

Widmann, A., Schröger, E., and Maess, B. (2015). Digital filter design for electrophysiological data - a practical approach. *J. Neurosci. Methods*, 250:34–46.

Williams, C. and Rasmussen, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press. Retrieved from http://www.newton.ac.uk/files/seminar/20070809140015001-150844.pdf.

Wilson, S. B. (2005). A neural network method for automatic and incremental learning applied to patient-dependent seizure detection. *Clin. Neurophysiol.*, 116(8):1785–1795.

Wilson, S. B. (2006). Algorithm architectures for patient dependent seizure detection. *Clin. Neurophysiol.*, 117(6):1204–1216.

Wilson, S. B., Scheuer, M. L., Emerson, R. G., and Gabor, A. J. (2004). Seizure detection: Evaluation of the Reveal algorithm. *Clin. Neurophysiol.*, 115(10):2280–2291.

Wilson, S. B., Scheuer, M. L., Plummer, C., Young, B., and Pacia, S. (2003). Seizure detection: Correlation of human experts. *Clin. Neurophysiol.*, 114(11):2156–2164.

Wood, M. (2019). *Improving Patient Care with Machine Learning At Beth Israel Deaconess Medical Center*. AWS Machine Learning Blog. Retrieved from https://aws.amazon.com/blogs/machine-learning/improving-patient-care-with-machine-learning-at-beth-israel-deaconess-medical-center/.

Wu, J., Zhou, T., and Li, T. (2020). Detecting epileptic seizures in EEG signals with complementary ensemble empirical mode decomposition and extreme gradient boosting. *Entropy*, 22(2).

Xiang, J., Li, C., Li, H., Cao, R., Wang, B., Han, X., and Chen, J. (2015). The detection of epileptic seizure signals based on fuzzy entropy. *J. Neurosci. Methods*, 243:18–25.

Xun, G., Jia, X., and Zhang, A. (2015). Context-learning based electroencephalogram

analysis for epileptic seizure detection. In *2015 IEEE Int. Conf. Bioinforma. Biomed.*, pages 325–330. IEEE.

Xun, G., Jia, X., and Zhang, A. (2016). Detecting epileptic seizures with electroencephalogram via a context-learning model. *BMC Med. Inform. Decis. Mak.*, 16(2):97–109.

Yan, T., Wang, W., Yang, L., Chen, K., Chen, R., and Han, Y. (2018). Rich club disturbances of the human connectome from subjective cognitive decline to Alzheimer's disease. *Theranostics*, 8(12):3237.

Yang, B.-h. (2015). Fast removal of ocular artifacts from electroencephalogram signals using spatial constraint independent component analysis based recursive least squares in brain-computer interface. *Front. Inf. Technol. Electron. Eng.*, 16(6):486–496.

Yang, B. H., Yan, G. Z., Yan, R. G., and Wu, T. (2006). Feature extraction for EEG-based brain-computer interfaces by wavelet packet best basis decomposition. *J. Neural Eng.*, 3(4):251.

Yao, X., Cheng, Q., and Zhang, G.-Q. (2019). A Novel Independent RNN Approach to Classification of Seizures against Non-seizures. arXiv:1903.09326.

Yao, X., Li, X., Ye, Q., Huang, Y., Cheng, Q., and Zhang, G. (2018). A Robust Deep Learning Approach for Automatic Seizure Detection. arXiv:1812.06562.

Yash, P. (2018). Various epileptic seizure detection techniques using biomedical signals: a review. *Brain Informatics*, 5(2):6.

Yoon, J., Lee, J., and Whang, M. (2018). Spatial and Time Domain Feature of ERP Speller System Extracted via Convolutional Neural Network. *Comput. Intell. Neurosci.*, 2018.

Younes, M., Kuna, S. T., Pack, A. I., Walsh, J. K., Kushida, C. A., Staley, B., and Pien, G. W. (2018). Reliability of the American Academy of Sleep Medicine Rules for Assessing Sleep Depth in Clinical Practice. *J. Clin. Sleep Med.*, 14(2):205–213.

Younes, M., Thompson, W., Leslie, C., Egan, T., and Giannouli, E. (2015). Utility of

technologist editing of polysomnography scoring performed by a validated automatic system. *Ann. Am. Thorac. Soc.*, 12(8):1206–1218.

Yuan, Q., Zhou, W., Liu, Y., and Wang, J. (2012). Epileptic seizure detection with linear and nonlinear features. *Epilepsy Behav.*, 24(4):415–421.

Yuan, Q., Zhou, W., Zhang, L., Zhang, F., Xu, F., Leng, Y., Wei, D., and Chen, M. (2017a). Epileptic seizure detection based on imbalanced classification and wavelet packet transform. *Seizure*, 50:99–108.

Yuan, Y., Xun, G., Jia, K., and Zhang, A. (2017b). A novel wavelet-based model for EEG epileptic seizure detection using multi-context learning. In *2017 IEEE Int. Conf. Bioinforma. Biomed.*, pages 694–699.

Yuan, Y., Xun, G., Jia, K., and Zhang, A. (2018a). A multi-context learning approach for EEG epileptic seizure detection. *BMC Syst. Biol.*, 12(6):47–57.

Yuan, Y., Xun, G., Jia, K., and Zhang, A. (2019a). A multi-view deep learning framework for EEG seizure detection. *IEEE J. Biomed. Heal. Informatics*, 23(1):83–94.

Yuan, Y., Xun, G., Ma, F., Suo, Q., Xue, H., Jia, K., and Zhang, A. (2018b). A novel channel-aware attention framework for multi-channel EEG seizure detection via multi-view deep learning. In *2018 IEEE EMBS Int. Conf. Biomed. Heal. Informatics*, pages 206–209.

Yuan, Y., Xun, G., Suo, Q., Jia, K., and Zhang, A. (2017c). Wave2Vec: Learning deep representations for biosignals. In *2017 IEEE Int. Conf. Data Min.*, pages 1159–1164.

Yuan, Y., Xun, G., Suo, Q., Jia, K., and Zhang, A. (2019b). Wave2Vec: Deep representation learning for clinical temporal data. *Neurocomputing*, 324:31–42.

Yuvaraj, R., Thomas, J., Kluge, T., and Dauwels, J. (2018). A deep Learning Scheme for Automatic Seizure Detection from Long-Term Scalp EEG. In *2018 52nd Asilomar Conf. Signals, Syst. Comput.*, pages 368–372. IEEE.

Zabihi, M., Kiranyaz, S., Ince, T., and Gabbouj, M. (2013). Patient-specific epileptic seizure detection in long-term EEG recording in paediatric patients with intractable seizures. In *IET Intell. Signal Process. Conf. 2013 (ISP 2013)*, pages 1–7.

Zabihi, M., Kiranyaz, S., Jantti, V., Lipping, T., and Gabbouj, M. (2019). Patient-Specific Seizure Detection Using Nonlinear Dynamics and Nullclines. *IEEE J. Biomed. Heal. Informatics*, 24(2):543–555.

Zabihi, M., Kiranyaz, S., Rad, A. B., Katsaggelos, A. K., Gabbouj, M., and Ince, T. (2016). Analysis of High-Dimensional Phase Space via Poincaré Section for Patient-Specific Seizure Detection. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 24(3):386–398.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proc. twenty-first Int. Conf. Mach. Learn.*, pages 114–122.

Zeng, K., Yan, J., Wang, Y., Sik, A., Ouyang, G., and Li, X. (2016). Automatic detection of absence seizures with compressive sensing EEG. *Neurocomputing*, 171:497–502.

Zhang, S.-L., Zhang, B., Su, Y.-L., and Song, J.-L. (2019). A novel EEG-complexity-based feature and its application on the epileptic seizure detection. *Int. J. Mach. Learn. Cybern.*, 10(12):3339–3348.

Zhang, X., Yao, L., Dong, M., Liu, Z., Zhang, Y., and Li, Y. (2020). Adversarial Representation Learning for Robust Patient-Independent Epileptic Seizure Detection. *IEEE J. Biomed. Heal. Informatics*, pages 1–1.

Zhang, Y., Dong, Z., Wang, S., Ji, G., and Yang, J. (2015). Preclinical diagnosis of magnetic resonance (MR) brain images via discrete wavelet packet transform with tsallis entropy and generalized eigenvalue proximal support vector machine (GEPSVM). *Entropy*, 17(4):1795–1813.

Zhang, Y., Yang, S., Liu, Y., Zhang, Y., Han, B., and Zhou, F. (2018). Integration of 24 feature types to accurately detect and predict seizures using scalp EEG signals. *Sensors (Switzerland)*, 18(5):1372.

Zheng, A. and Casari, A. (2018). *Feature Engineering for Machine Learning principles and techniques for data scientists*. O'Reilly Media, Inc.

Zhou, M., Tian, C., Cao, R., Wang, B., Niu, Y., Hu, T., Guo, H., and Xiang, J. (2018). Epileptic Seizure Detection Based on EEG Signals and CNN. *Front. Neuroinform.*, 12:95.

Zhou, W. and Gotman, J. (2009). Automatic removal of eye movement artifacts from the EEG using ICA and the dipole model. *Prog. Nat. Sci.*, 19(9):1165–1170.

Zhu, X., Xu, H., Zhao, J., and Tian, J. (2017). Automated Epileptic Seizure Detection in Scalp EEG Based on Spatial-Temporal Complexity. *Complexity*, 2017:1–8.

Ziyabari, S., Shah, V., Golmohammadi, M., Obeid, I., and Picone, J. (2017). Objective evaluation metrics for automatic classification of EEG events. arXiv:1712.10107.

Zou, L., Liu, X., Jiang, A., and Zhousp, X. (2018). Epileptic Seizure Detection Using Deep Convolutional Network. In *2018 IEEE 23rd Int. Conf. Digit. Signal Process.*, pages 1–4.