

A Study of Decision Deadlocks using Electroencephalography and Machine Learning

Katarzyna Szymaniak

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Master of Research



Swansea University
Prifysgol Abertawe

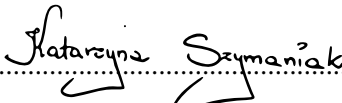
Department of Computer Science

Swansea University

December 30, 2019

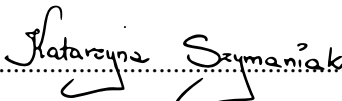
Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed  (candidate)
Date 30.12.2019

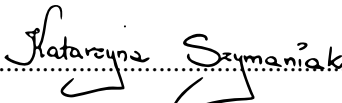
Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed  (candidate)
Date 30.12.2019

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed  (candidate)
Date 30.12.2019

Abstract

Decision making is an intrinsic part of human life. While some decisions are trivial and simple, some of them are not. The paralysis we experience in such situations can be defined as decision making deadlock. As a result of a collaboration with ccBrain Lab at CUBRIC an experiment relating to a decision making deadlock was conducted. Each subject had to choose between two equally rewarding shapes i.e. the probability of getting a reward was the same for both shapes. Three cases have been examined: 100%, 80% and 20%. The reaction time and brain activity (EEG) has been measured during the experiment. The tendency among subjects showed the inverse proportion to the probability of getting the reward. This gave the basis for the hypothesis of different human cognition processes being discriminative.

Using frame-wise and sequential approaches this problem was explored. Applying k-NN, Random Forest and LDA in frame-wise approach on both, subject generic scenario with Leave One Group Out cross-validation and subject specific scenario using k-folds validation did not result in satisfactory classification outcomes. The attempt to enhance models performance using dimensionality reduction (PCA) didn't show any significant improvement. However, using Gini Impurity to investigate feature importance gave an insight into significant cognition period in brain activity across the patients proving singularity of the signal for each case.

Due to the time dependencies, memory mechanism in sequential approach was examined using RNN, LSTM and GRU. Alternative input formulation was presented using hand-crafted features which are based on Discrete Wavelet Transform coefficients. While the problem remained challenging to generalise, interesting observation of the possible relationship between the channels was presented creating cross-channel connectivity theory which could be applied over time.

Acknowledgements

First of all, I would love to thank my dearest supervisor, Dr Jingjing Deng, who managed to cope with me throughout the entire project. It must have been hard work. You are a true inspiration where your enthusiasm for the field, dedication and hardworking spirit makes you an expert in the domain. Also, I would like to thank Prof. Xianghua Xie who helped me find my passion in machine learning and let me explore an interest of mine: neuroscience by establishing collaboration with ccBrain Lab at Cardiff University Brain Research Imaging Centre. Special thanks to Jiexiang Zhang from ccBrain Lab whose experiment the research of mine was based on. A Sweet and dearing thank you to my lovely and supportive mentor, Prof. John Tucker who believed in me and shared his wisdom in all areas of life. I hope to keep on learning and growing thanks to you all. I am grateful for the support, and inspirational conversations to all friends in Computer Vision Group and second floor PhD lab.

I would like to thank in no particular order to Fluff who was always there for me with his Michelin 3-star meals, support on every step and positive energy; Irfan with his amazing smile that brighten even the darkest days; Elif for countless hugs; dates with Omnijah; Jad for making PhD lab more lively and others.

I would also like to make personal message to my family:

Chciałabym szczególnie podziękować Kasi, rodzicom i Arkowi, którzy zawsze przy mnie byli i zarażali mnie śmiechem i pozytywną energią. Dzięki Wam jestem w stanie sięgać gwiazd i nadal być sercem i duszą w mojej małej stromieckiej ojczyźnie.

Who knows what I want to do? Who knows what anyone wants to do? How can you be sure about something like that? Isn't it all a question of brain chemistry, signals going back and forth electrical energy in the cortex? How do you know whether something is really what you want to do or just some kind of nerve impulse in the brain? Some minor little activity takes place somewhere in this unimportant place in one of the brain hemispheres and suddenly I want to go to Montana or I don't want to go to Montana.

– Don DeLillo, *White Noise*

Contents

Abstract	iii
Acknowledgements	iv
List of Figures	1
Abbreviations	3
1 Introduction	6
1 Motivation	6
2 Contribution	8
3 Outline	9
2 Background	11
1 Classic Machine Learning	11
1.1 Random Forest	12
1.2 k-Nearest Neighbor	15
1.3 Principle Component Analysis	17
1.4 Linear Discriminant Analysis	19
2 Deep Learning	20
2.1 Recurrent Neural Network	22
2.2 Long-Short Term Memory	25
2.3 Gated Recurrent Unit	27
3 Evaluation Metrics	28
4 EEG Data	29
5 Machine Learning in Neuroscience	30

6	Signal Decomposition	35
7	Summary	37
3	Dataset	39
1	Experiment	39
2	Data Preprocessing	42
2.1	Data formatting and normalization	44
2.2	Feature Extraction	45
3	Summary	47
4	Frame-wise Approach	48
1	Methodology	49
1.1	Subject Generic	49
1.2	Subject Specific	50
2	Results and Discussion	51
2.1	k-Nearest Neighbors	51
2.2	Random Forest	52
2.3	Linear Discriminant Analysis	55
2.4	Dimensionality Reduction Model Enhancement	57
3	Result Overview	60
3.1	Significant Cognition Period	61
4	Summary	63
5	Sequential Approach	64
1	Methodology	64
1.1	Self Learnt Features with Neural Networks	64
1.2	Hand-Crafted Features	65
2	Results and Discussion	68
2.1	Self Learnt Features with Neural Networks	68
2.1.1	Recurrent Neural Network	68
2.1.2	Long-Short Term Memory	69
2.1.3	Gated Recurrent Unit	71
2.2	Hand-Crafted with Wavelets	72
2.2.1	Random Forest	72

2.2.2	Fully Connected Neural Network	75
3	Summary	76
6	Conclusions	78
1	Contributions	79
2	Future Work	80
	Bibliography	81

List of Figures

2.1	Top machine learning algorithms in medical literature.	12
2.2	Decision Tree.	13
2.3	k-Nearest Neighbor.	15
2.4	Singular Value Decomposition.	18
2.5	LDA: distance between means of two classes.	20
2.6	LDA: maximising separation between the classes and minimise within.	21
2.7	Single-layer perceptron.	22
2.8	Memory mechanism.	23
2.9	Backpropagation through time.	25
2.10	Relationship styles of architectures.	25
2.11	LSTM unit.	26
2.12	GRU unit.	27
2.13	Confusion matrix.	28
2.14	Dataset prediction representation: precision and recall.	29
2.15	Pipeline for Motor imagery EEG classificaiton.	31
2.16	Normalisation techniques.	32
2.17	16 different classifiers result.	33
2.18	EEG survey: input formulation by task.	34
2.19	Choice of architecture based on the type of the task.	35
2.20	Families of Wavelets.	36
2.21	Filtering process in Discrete Wavelet Transform.	37
3.1	Experiment shapes.	40
3.2	Experiment procedure.	41
3.3	Observation of experiment from ccBrain Lab.	42

List of Figures

3.4	Channel localisation on EEG cap model.	43
3.5	Data representation.	44
4.1	Sample signal representation from 32 channels.	48
4.2	Majority Voting.	49
4.3	Leave One Group Out.	50
4.4	k-NN results in the subject generic and the subject specific scenario.	52
4.5	Random Forest results in a subject generic and subject specific scenario.	53
4.6	Feature importance for all features.	54
4.7	Random Forest feature importance results in a subject specific scenario.	55
4.8	LDA results in a subject generic and subject specific scenario.	56
4.9	Average results from Random Forest, k-NN and LDA in subject generic and subject specific scenario.	57
4.10	k-NN with PCA results in subject generic and subject specific scenario.	58
4.11	RF with PCA results in subject generic and subject specific scenario.	59
4.12	LDA with PCA results in a subject generic and subject specific scenario.	59
4.13	Average results for RF, k-NN and LDA with PCA.	60
4.14	Sequential cognition period: Case1vs2.	61
4.15	Sequential cognition period: Case1vs3.	62
4.16	Sequential cognition period: Case2vs3.	62
5.1	Wavelet Decomposition Vector.	65
5.2	Wavelet Decomposition on signal from a single trial.	66
5.3	Detailed decomposition coefficients across all the sub-bands.	67
5.4	Vanilla RNN results.	69
5.5	LSTM results.	70
5.6	GRU results.	71
5.7	Average results: RNN, LSTM and GRU.	72
5.8	Random Forest and Random Forest with feature importance results.	73
5.9	Cross-channel feature importance: Case1vs2.	74
5.10	Cross-channel feature importance: Case1vs3.	74
5.11	Cross-channel feature importance: Case2vs3.	74
5.12	Fully Connected Neural Network results.	75

Abbreviations

AdaBoost Adaptive Boosting	33
AE Autoencoder	30
AI Artificial Intelligence	12
ANN Artificial Neural Network	21
BCI Brain-Computer Interface	7
CART Classification and Regression Trees	12
ccBrain Lab Cognition and Computational Brain Lab	7
CNN Convolutional Neural Network	33
CSP Common Spatial Pattern	34
CUBRIC Cardiff University Brain Research Imaging Centre	7
CVT Complex Value Transformation	34
CWT Continuous Wavelet Transform	36
DBN Deep Belief Network	34
DE Dynamic Energy	34
DT Decision Tree	33
DWT Discrete Wavelet Transform	9

Abbreviations

EEG Electroencephalography	7
ERP Event Related Potential	8
FFT fast Fourier Transform	34
FT Fourier Transform	35
GMDH Group Method of Data Handling	22
GRU Gated Recurrent Unit	9
ICA Independent Component Analysis	33
k-NN k-Nearest Neighbors	8
LDA Linear Discriminant Analysis	8
LFDA Local Fishers Discriminant Analysis	33
LOG Leave One Group Out	49
LSTM Long Short-Term Memory	9
MAD Mean Absolute Difference	34
MCI Mild Cognitive Impairment	30
MRA Multiresolution Analysis	35
MRI Magnetic Resonance Imaging	30
PCA Principal Component Analysis	8
PSD Power Spectral Density	33
RF Random Forest	8
RNN Recurrent Neural Network	9
S.G. Subject Generic	51

Abbreviations

S.S. Subject Specific	51
SAE Stacked Autoencoder	32
STFT Short Time Fourier Transform	34
SVD Singular Value Decomposition	34
SVM Support Vector Machine	33
SWD Swarm Decomposition	34
WT Wavelet Transform	9
XGBoost ExtremeGradient Boosting	30

Chapter 1

Introduction

For the past few decades, machine learning has been going through its renaissance where not only traditional machine learning techniques but also new architectures are being used. Many of emerging applications of machine learning are applied to disciplines such as healthcare, education, transport and logistics, public services, finance, pharmaceuticals, energy legal sector and more [1]. Those widespread techniques are applied to a variety of domains to learn underlying information based on the gathered data (also known as learning by example). Medicine is an extensive field, consisting of many sub-domains where machine learning has the potential to change its trajectory. Sub-domains of medicine: cognitive science, psychology and neuroscience are intersecting areas of expertise where exploring and comprehending the processes within the human brain is the main principle [2].

1 Motivation

In psychology, decision-making is defined as a cognitive process of selection over several alternative possibilities. This process is composed of three parts: multiple options, expectations of the future events related to each option and consequences associated with possible outcome [3]. While it seems to be a complicated process, according to a German psychologist and neuroscientist, Ernest Pöppel, "We make about 20,000 decisions every day, most of them at lightning speed." *As soon as you wake up, you decide whether to get out of the bed or not. Then, it is time to choose what's for breakfast, outfit for the day and so forth. Human being has to make choices at every single step, whether it is trivial or not. Some answers come quicker than others. When you decide on the life path you take, career or the right partner for you, the

1. Introduction

decision becomes more problematic. The paralysis we experience in such a situation can be defined as decision making deadlock [4].

As the result of collaboration with ccBrain Lab (Cognition and Computational Brain Lab) at Cardiff University Brain Research Imaging Centre (CUBRIC) which specialise in “computational mechanisms of decision-making, learning and action”[†] the cognition process of decision making has been investigated.

Due to the experiment conducted by ccBrain Lab, a question related to a decision making deadlock has been raised [4]. By choosing between two independent shapes the subject had to make a decision while their reaction time and brain activity (EEG) were recorded. The tendency among subjects showed the inverse proportion to the probability of getting the reward a.k.a. the higher the probability of getting the reward was, the quicker the subject made a decision. This gave the basis for the hypothesis of three cases (scenarios) being discriminative.

Comprehending the process of decision making is a huge step towards the exploration of human cognition in a field of psychology and neuroscience where applications of this research can be powerful and endless. As decision making is an intrinsic part of our nature, increasing comfort of daily activities could potentially influence the mass of people. We live in the era of consumerism where the interest of customers is the highest priority. Analysing how to improve products, advertising or other marketing elements by measuring the brain’s responses to marketing stimuli is denoted as Neuromarketing. Learning why consumers make decisions they do and what part of the brain is responsible for this action, is the main purpose behind neuromarketing [5]. Frequently we are paralysed by the number of options we have not knowing what to choose. Performing deep analysis of the data and understanding the process of decision-making deadlocks (paralyse situations) could help businesses understand customers and their needs to make the choice clear and dynamic.

Since the second half of the XX century we have entered a digital revolution which continues till now. As many domains of our lives have been digitised the idea of communication between a human and a machine was born. Interface between human (brain) and a machine (computer) was defined as Brain-Computer Interface (BCI). Basic pipeline for BCI is denoted as signal acquisition; preprocessing and feature extraction; feature classification and tasks/commands. This concept can be applied from electronics, automatic cars to embedded

*ErnestPoppelDecisionMaking

[†]<https://ccbrain.org/>

systems (smart houses), medicine and so forth [6] [7]. One of the common techniques for EEG BCI is using P300 (P3). P3 signal is a component of Event Related Potential (ERP) elicited in the process of decision making. It is categorised as an endogenous potential since its occurrence is based on a person's reaction to a stimulus. The ability to discriminate between different deadlock decision-making cases could enhance the classification process for the actions we desire to proceed (ability to control things using our minds using BCI systems).

2 Contribution

The purpose of the research is to investigate and discriminate decision making between equal choices using machine learning techniques. Previous research shares some insights into equal choice decisions where benefit of 'rushing to decisions' [8] and lack of choice randomness in consumer [9] and lab-based settings [10] have been explored. Yet, it is unclear how reward certainty and shape preference (bias) influences sub-components of the whole process of decision-making. Furthermore, observing macroscopic brain activities during equal decision choice in time could explain the preference bias and response time dependencies between varied reward certainty options. Interpretation and classification of the EEG data has the potential to give more insights into the process of decision making for equal choices to unravel above presented unresolved issues.

Two main approaches are investigated: frame-wise and sequential. The main concept behind the frame-wise approach is based on dividing the sequence space of the signals (from 32 channels) into independent timestep frames. No intercorrelations between frames are assumed. The prediction is made per each frame where using majority voting the final outcome of each sequence is calculated. Two methodologies are used in this approach: subject generic and subject specific. In the subject generic scenario, Leave One Group Out cross-validation is applied while subject specific implements k-fold cross-validation. This approach was examined using classical machine learning techniques: k-NN, Random Forest and LDA.

The results of both methodologies were limited. To enhance models performance dimensionality reduction using PCA was applied yet with no significant improvement. However, extensive analysis of the data enabled finding interesting observation on significant cognition period. The singularity of brain activity (for each case) demonstrated certain prediction pattern within a sequence, common across all the subjects, proving cognition to be time-dependent.

The second approach focuses on investigating time dependencies and prediction of the sequence as one. The first approach uses self learnt features with neural networks where memory

mechanism present in RNN, LSTM and GRU was applied. The models were fed with frames one by one to pass the information within and output the final result of the sequence in many to one manner (for the whole sequences). The second approach used hand-crafted features based on signal decomposition where Discrete Wavelet Transform was implemented. Transformed and modified input from each channel was then flatten into one observation (within a single sequence) and fed to Random Forest and Fully Connected Neural Network.

Extending the idea of prediction based on the whole sequence (trial) remained unsatisfactory. However, using hand-crafted features based on Discrete Wavelet Transform enabled us to make further observations about cross-channel connectivity which will be investigated in the future.

Extensive data analysis provided problem formulation, initial benchmarking and link to the hypothesis as the pillars for the future work.

3 Outline

The remainder of this thesis is outlined as follows:

Section 2 Background:

The background section introduces current research related to the problem domain. Furthermore, techniques used to analyse and explore the data have been presented in-depth where classical machine learning and deep learning were used. The section includes additional information on signal decomposition using Wavelet Transform which was applied to the data. Common techniques for evaluation of the models' performance were demonstrated.

Section 3 Dataset:

Dataset section presents an experiment conducted by ccBrain Lab in CUBRIC. The initial observations made by the neuroscientists are used to formulate the hypothesis of this work. Data formatting and normalisation used as part of the pre-processing step are presented and applied. Techniques used for feature extraction are defined.

Section 4 Frame-wise Approach:

This section presents an initial approach where frame-wise prediction is implemented. Two methodologies used in this approach: subject generic and subject specific are explained. The results of applied classical machine learning architectures: k-NN, Random

1. Introduction

Forest, LDA and PCA are discussed. Based on the findings of frame-wise approach, observation of significant cognition period is introduced.

Section 5 Sequential Approach:

Sequential approach section presents two methodologies of the input data formulation: self learnt features with neural networks and hand-crafted features. Deep learning techniques such as RNN, LSTM and GRU are implemented and investigated. Signal decomposition for hand-crafted features is applied using Wavelets to Random Forest and Fully Connected Neural Network. Discussed findings give observations based on the sequential approach to classification of the signal data presenting cross-channel connectivity theory.

Section 6 Conclusions:

Conclusions section gives an outline of findings from the conducted experiments. It presents contributions to the research and future work to apply.

Chapter 2

Background

1 Classic Machine Learning

As previously mentioned, machine learning has become a new trend in science, using statistical models and algorithms computers learn how to perform specific tasks based on gathered data. Machine learning can be divided into subcategories depending on their learning process where supervised, unsupervised and reinforcement learning are distinguishable approaches.

Out of all, the ability to create function through the learning process by matching input and output data is possible due to the supervised learning. The mathematical model by learning through experience (input-output pairs) is then able to solve a particular task on unseen data (test data). Depending on the problem that has to be tackled, predicting future outcome can be approached in varied ways. The fundamental division into classification and regression algorithms enables finding an appropriate plan of attack with respect to the end result. Regression algorithms are used for estimation of continuous outputs, for instance, temperature, price, length, while classification is used for discrete response variables [11].

An algorithm that implements classification, known as classifier, identifies within the set of categories to which the new sample (observation) belongs based on previously trained examples. The simplistic case involves binary choice $Y : \{0, 1\}$ called binary classification. This technique is often used in medical testing to determine whether a patient has a certain disease or not. In the multi-class classification problem, the number of categories is equal to three or more. Using algorithm adaptation techniques extension from binary classification can be used in multi-class classification problems. Architectures such as Support Vector Machines, k-Nearest Neighbours, Decision Trees, Naive Bayes or Neural networks are adaptable estima-

2. Background

tors for a higher number of outputs [12]. Many of these algorithms are used in medicine, but support vector machine and neural networks dominated AI in healthcare. [13] *

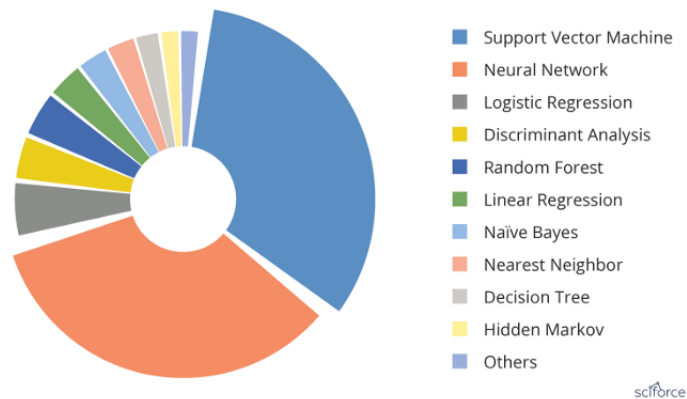


Figure 2.1: Top machine learning algorithms in medical literature. Data has been generated based on searching results of machine learning algorithms within healthcare on PubMed thanks to SciForce[†].

1.1 Random Forest

In the late 1970s and early 1980s, a decision tree algorithm known as ID3 (Iterative Dichotomiser) has been developed by J.Ross Quinlan who later presented its successor C4.5. In 1984, Leo Breiman et al. published book *Classification and Regression Trees* [14] which presented a similar approach of decision trees learning from the training tuples. All previously mentioned, ID3, C4.5 and CART, have been constructed in a “top-down recursive divide-and-conquer manner”.

As the prediction follows multiple branches of “if ... then ...” decision splits we create somehow a structure looking like a tree. Each branch split is based on the feature threshold that divides remaining samples in the most efficient way. To define the “best split” Gini impurity (CART), information gain (ID3) or gain ratio (C4.5) are used. As ID3 does not deal with continuous data and C4.5 is susceptible to outliers, CART is frequently used.

CART is a binary partitioning recursively performed on continuous and nominal attributes (for both predictors and targets) [15] [12]. It handles data in raw form where in continuous

[†]<https://medium.com/sciforce/top-ai-algorithms-for-healthcare-aa5007ffa330>

2. Background

input no binning is required. Each root/parent node holds an attribute (a single input variable) x_i which divides the input space.

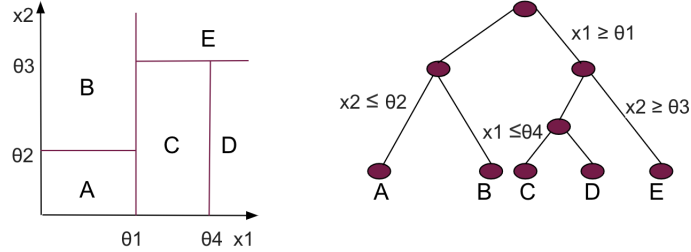


Figure 2.2: Binary decision tree partition of the feature space. The presented tree has threshold values θ_i which lead to leaf values (A,B,C,..).

This process is done by using the greedy approach in *recursive binary splitting* where the best split (lowest value of a cost function) is selected. Previously mentioned Gini Index also known as Gini coefficient or Gini impurity is used to measure the impurity of D , a training dataset or a data partition, by subtracting the sum of the squared probabilities from all the classes from one:

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2 \quad (2.1)$$

where p_i represents the probability of a tuple belonging to a class C_i . The sum is an accumulation of squared probabilities p_i across all the m classes. For each attribute all possible scenarios of a binary split are tested by computing a weighted sum of the impurity of every single partition to always find the best split:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2), \quad (2.2)$$

where A is an attribute and D_1, D_2 represent possible split values. While for an attribute with a discrete value finding the minimum of Gini Index indicates the splitting subset, for continuous-valued attributes all possible split points have to be taken into consideration. In such a case the split point is defined by localising the midpoint between each pair of sorted adjacent values which becomes a split-point if the subset gives the minimum Gini Index.

The impurity reduction in a binary split on a discrete-/continuous-valued attribute A is computed as follows:

$$\Delta Gini(A) = Gini(D) - Gini_A(D), \quad (2.3)$$

2. Background

Maximising the reduction in impurity (equivalent to reaching the minimum Gini Index) indicates the best splitting attribute. This attribute using either splitting subset (discrete value of a splitting attribute) or split-point (in continuous value of a splitting attribute) creates the splitting criterion.

To decide when the tree is already built the stopping criterion needs to be satisfied. Without any early termination, the tree stops splitting if all instances have identical attribute values or belong to the same class. While decision tree classifiers are trained to differentiate data samples based on attributes, many times the model grows a tree to maximum size with each leaf for a single class data (causes overfitting) rather than the overall population which leads to bad performance on the test dataset. Decision trees often can be affected by data anomalies in training dataset by noise and outliers.

Tree pruning is an approach to face such obstacles where pre- and post-pruning methods are used. Pruning trees results in easy to comprehend, less complex and faster structure which boosts model generalization for unseen data. In pre-pruning the further split of the node or partition the subset is halt based on the certain criterion, for instance, minimum sample leaf or measuring Gini Index which results in a node transforming into a leaf. The method of *trimming* (removing subtrees) from a *fully grown* tree where the removed subtree is replaced by the most frequent class (among the subtree) is known as post-pruning.

An example of a post-pruning algorithm is the cost complexity pruning used in CART which considers the error rate of the tree (percentage of misclassified tuples) and the number of leaves. Starting from the root, it computes the cost complexity of the subtree at node N and the cost complexity at the same node if it would be pruned. Those values are compared and if pruning the subtree would result in a smaller cost complexity, the subtree is pruned. To estimate the cost complexity a pruning set of tuples with class labels is used. It is independent of both training and test data and the algorithm generates a set gradually expanding pruned trees.

To enhance the robustness of the model the ensemble technique whereby combining predictions of several base estimators the performance can be improved. Random Forest groups together multiple individual decision trees and combines their output using bagging, also known as Bootstrap aggregation. Using bagging, randomly sampled subsets of the dataset are used to train individual decision trees where sampling is done with replacement. This methodology

2. Background

decreases the variance of the model as individual trees are sensitive to noise and prone to over-fitting. By randomly subsampling the data for each and every tree the correlation risk is low and the bagging makes the process more robust without increasing the bias.

The random forest provides also feature bagging which at each split considers only a random subset of features which reduces correlation among trees even more effectively. As every decision tree is independent the model can be run in parallel to average all the predictions or use a majority vote (mode) at the final stage. The final result of our model is calculated by averaging over all predictions from these sampled trees or by majority vote.

1.2 k-Nearest Neighbor

k-Nearest Neighbor is a non-parametric algorithm used in classification and regression problems. It is instance-based learning i.e. instead of performing explicit generalization the function approximation is done locally and computations are postponed until classification [15] [16]. The idea behind k-NN concentrates on feature space of all the training samples where for a new test observation k nearest objects from train dataset are found. Based on those a particular class is assigned to a new sample with respect to the predominance of that class in the neighbourhood.

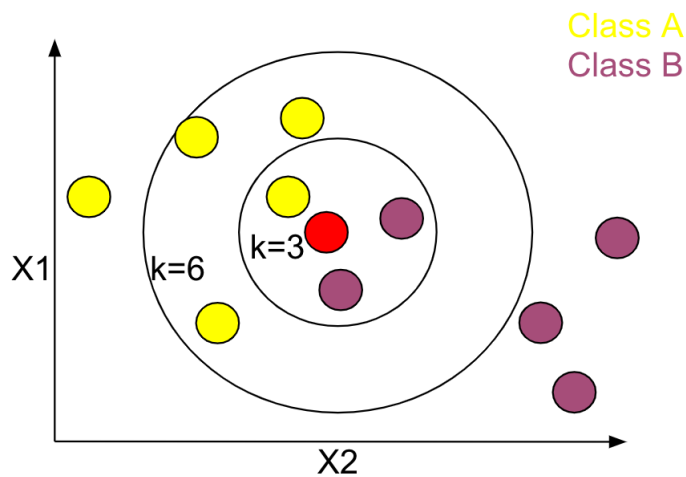


Figure 2.3: k-Nearest Neighbor algorithm presented in the feature space $X1$ and $X2$ using different values of k (3 and 6) classifies new data point (red) based on the majority of samples in the neighborhood.

2. Background

Three key aspects have to be considered while using this approach. The data has to be labelled (supervised learning), metric for computing distance (or similarity) between samples has to be chosen and the number of neighbours that influence the final output, hyperparameter k needs to be selected.

Given a new test sample $z = (x', y')$ and training dataset D , the algorithm computes the distance between z and every observation in dataset $(x, y) \in D$ to define D_z (all nearest neighbour samples), where $D_z \subseteq D$. Then based on D_z , test object is classified using majority voting:

$$\text{Majority Voting : } y' = \operatorname{argmax}_c \sum_{(x_i, y_i) \in D_z} I(c = y_i), \quad (2.4)$$

where c represents class label, y_i is a label of the i th neighbor and $I()$ is an indicator function which if true returns 1 and 0 otherwise.

Several aspects have a significant influence on the performance of k -NN. Choosing *the right* number of neighbours, k , can result in either a model sensitive to noise (if k is too small) or holding a higher number of samples from other classes (k being too big), as shown in Figure 2.3.

Finding the most efficient way of measuring the distance (or similarity) between two points aims to create a relationship where the smaller the distance, the greater the likelihood of belonging to the same class. The naive approach includes linear search where the distance is being measured between every single point in the training dataset and the new sample. The running time of $O(dN)$, where N represents cardinality (no. of observations) of D and d is the dimensionality of the feature space. In higher dimensional feature space, the naive search usually outperforms space partitioning [17].

Space partitioning using the branch and bound methodology has been applied to the nearest search neighbour problem to decrease the complexity of the algorithm. Using a k -d tree, the search space is bisected into two regions (half-spaces) recursively creating hyperplanes that result in a binary tree where every leaf node stores a k -dimensional point. Using traversal of the tree query point can be found (starting from the root to a leaf) due to evaluation of the query point at each split. Depending on k sometimes neighbouring branches has to be considered. The average complexity (“for constant dimension query time”) is $O(\log N)$. Another type of space partitioning algorithm divides the dataset into a nested set of *balls* (hyperspheres) hence the name: ball tree. For every internal node, the dataset is partitioned into two disjoint subsets which are assigned to different balls. Although the balls may intersect, the data in the subsets

2. Background

are explicit for each ball according to their distance from the ball's centre point. Leaf nodes are represented as balls with all the data points within. Thus, for an unseen test point t , the distance between t and point in a subset (inside a ball B) is either equal or greater than the distance between t and the ball which can be presented in a mathematical formula as follows:

$$D^B(t) = \begin{cases} \max(|t - B.pivot| - B.radius, D^{B.parent}), & \text{if } B \neq Root \\ \max(|t - B.pivot| - B.radius, 0), & \text{if } B = Root \end{cases} \quad (2.5)$$

Where $D^B(t)$ represents the minimum distance between test point t any point in the ball B .

The last major issue is choosing the right approach to combine all $y_i, y_i \in D_z$ (class labels of the nearest neighbours points). Although using majority voting is the common approach, in a case where there is high variation in the distance (among nearest neighbours) and closer located data points indicate more accurately the class of the test sample, another technique needs to be applied. By weighting all samples in D_z by their distance to the object the prediction becomes more accurate and usually less sensitive to hyperparameter k .

$$\text{Distance - Weighted Voting : } y' = \operatorname{argmax}_c \sum_{(x_i, y_i) \in D_z} w_i \times I(c = y_i), \quad (2.6)$$

where w_i represents weight that is reciprocal of the distance (squared): $w_i = \frac{1}{d(x', x_i)^2}$.

1.3 Principle Component Analysis

One of the popular techniques used for dimensionality reduction is Principal Component Analysis (PCA) [18]. This unsupervised technique projects data onto linearly uncorrelated orthogonal axes (principal components, PCs) in m -dimensional space. In order to find components that describe the data *the best*, the variance in the data projected onto the PCs needs to be maximised. The first principal component captures variance of the data in the best possible way, then others come in descending order.

Principal components can be found by calculating covariance matrix as defined:

$$A = \operatorname{cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])], \quad (2.7)$$

where $\mathbb{E}(\cdot)$ denotes an expected value (i.e. mean) and X and Y and two (random) variables. As it is a square matrix, the eigenvalues and eigenvectors are calculated respectively:

$$\det(\lambda I - A) = 0 \quad (2.8)$$

$$(\lambda I - A)v = 0, \quad (2.9)$$

2. Background

where \det is the determinant of the matrix, λ represents eigenvalue, v is an eigenvector, I denotes identity matrix and A is previously calculated matrix. The eigenvector corresponding to the highest eigenvalue is the principal component that projects the data onto itself with the highest variance.

Another technique frequently used to achieve principal components is Singular Value Decomposition (SVD) [19]. It is a factorization technique that decomposes a matrix M ($m \times n$ matrix) from the data in the field K into the following:

$$M = U\Sigma V^* \tag{2.10}$$

where U is a unitary matrix ($U^*U = UU^* = I$) over K (if $K = \mathbb{R}$, unitary matrix is as well an orthogonal matrix), Σ is a diagonal $m \times n$ matrix which contains non-negative real numbers and V is another unitary matrix of a size $n \times n$ over K , and V^* is the conjugate transpose of V . The entries on the diagonal in a Σ matrix are known as the singular values of M which usually are listed in a descending order to rank eigenvalues (together with their eigenvectors) by the measure of spreadability of the data.

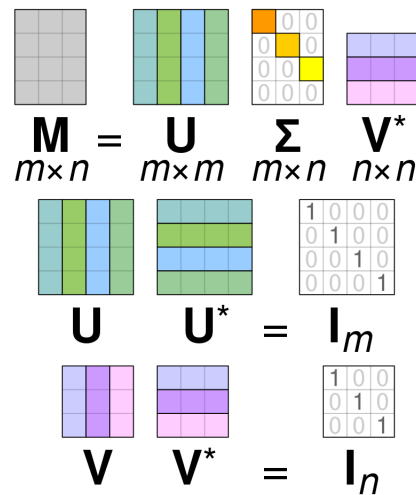


Figure 2.4: A diagram of a Singular Value Decomposition and matrix U and V as unitary matrix[‡].

[‡]https://en.wikipedia.org/wiki/Singular_value_decomposition

1.4 Linear Discriminant Analysis

Linear Discriminant Analysis is another technique used for dimensionality reduction and pattern-classification [20]. The principle idea behind it is to project a n -dimensional dataset in a lower-dimensional space k (where $k \leq n - 1$) while preserving class-separability to avoid overfitting, *curse of dimensionality*, and reduce computational costs. This linear classifier has been formulated by Ronald A. Fisher in 1936, *The Use of Multiple Measurements in Taxonomic Problems* [21]. While LDA and Principal Component Analysis approach is generally similar, PCA focuses on finding the component axis that maximises the variance of the data and LDA maximises the distance between multiple classes.

LDA finds most discriminant projection by maximising between-class distance and minimising within-class distance. Lets assume we have a set of 2-dimensional samples, the goal is to obtain a scalar y by projecting samples x onto the line (where w represents the projection):

$$y = w^T x \quad (2.11)$$

Based on all the possible lines, the one which would maximise the separability of the scalars would be chosen. The key to finding a good projection vector is defining a measure of separation between projections. This can be achieved by calculating the mean vector of each class in x (2.12) and y (2.13) feature space as shown respectively:

$$\mu_i = \frac{1}{N_i} \sum_{x \in w_i} x \quad (2.12)$$

$$\tilde{\mu}_i = \frac{1}{N_i} \sum_{x \in w_i} y = \frac{1}{N_i} \sum_{x \in w_i} w^T x = w^T \mu_i, \quad (2.13)$$

where N represents number of samples that belongs to a class w . Then the distance between the projected mean values represents separability of two different classes after projection:

$$|\tilde{\mu}_1 - \tilde{\mu}_2| = |w^T (\mu_1 - \mu_2)| \quad (2.14)$$

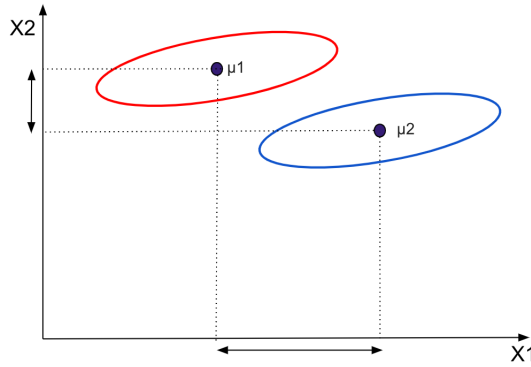


Figure 2.5: Distance between means of two classes where although in X_1 the distance is greater yet the overlap of the classes is higher showing poor class separability.

However, this is not sufficient enough, which is visualised in the Figure 2.5 as x_1 has a larger distance between the means of two classes while x_2 gives better class separability. Proposed by Fisher solution aims for maximisation of the distance between the means which is normalised by within-class scatter. The scatter of a class (equivalent to variance) is defined as:

$$\tilde{s}_i^2 = \sum_{x \in w_i} (y - \tilde{\mu}_i)^2 \quad (2.15)$$

Within-class scatter of projected samples ($\tilde{s}_1^2 + \tilde{s}_2^2$) is an essential part of criterion function J which linear function $w^T x$ aims to maximise. The Fisher's LDA criterion is described as in the equation 2.16. This results in efficient projection of the data onto the new feature space with maximised between class separation as presented in the Figure 2.6.

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \quad (2.16)$$

2 Deep Learning

In 1943, a logician, Walter Pitts, and a neuroscientist, Warren McCulloch presented the first mathematical model of neural network [22]. Using propositional logic and various applications of calculus they gave the origins to algorithms that attempt to mimic human brain functionality. In 1947, one of the greatest brains in the computer science field, Alan Turing, gave a talk in London Mathematical Society, where he said “What we want is a machine that can learn from experience”. This brought artificial intelligence to a new era of machine

2. Background

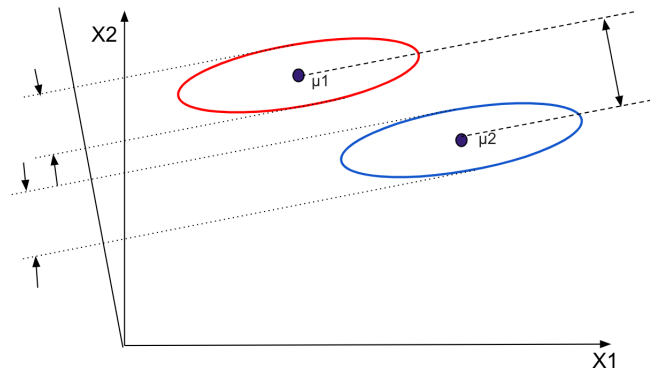


Figure 2.6: Projection of a new feature space using LDA by maximising separation between the classes and minimise within.

learning. In 1950, Turing also published a paper “Computing machinery and intelligence” [23] where he introduced a phenomenon of The Turing Test, which originally was presented as The Imitation Game. The main question was whether machines can think or not and to what level a machine is supposed to act intelligently enough to be taken as a human being.

The main concept focused on designing a model that learns from previous experiences to create accurate predictions [24]. Inspired by a human brain structure, neural nets became one of the sub-fields of machine learning. A brain is made of 86 billion interconnected neurons [25]. In a comparison of computational speed, machine outperforms neurobiological device by 100,000 times. Although they can perform these type of tasks in an extremely quick time, simple decision making, which a child is capable of, like face recognition is a challenge for them [26].

However, the inspiration driven by a human brain decided to overcome this issue. The process of learning by a child is formed on examples verified by their parents and the environment. Based on a trained dataset with labels, the neural network model predicts the output of new data previously not known. When a child at first explores the world and learns, it is not being told by parents what features look for in a cat: whiskers, shape of eyes, ears etc. It is being shown many examples of this animal that brain’s model processes and learns features from.

As a reflection of a human brain, an Artificial Neural Network (ANN) is an architecture where the basic building block is an artificial neuron (perceptron) where the sum of the dot

2. Background

product of the inputs and weights $\sum_{i=1}^m w_i x_i$ (where w_i is weight assigned to each node x_i and m represents all the inputs) is then fed to the activation function to give a final output.

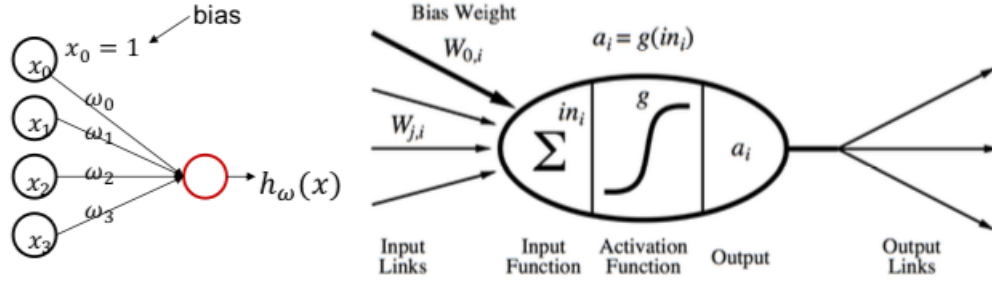


Figure 2.7: Single-layer perceptron (only one hidden layer) [27].

The other crucial design element taken from the brain is the ability to train the neurons, using weights stored on synapses of the neural network to pass through only useful information [25]. Based on the trained model ANN is enabled to make predictions.

In 1965, Alexey Ivakhnenko et al. developed the Group Method of Data Handling (GMDH) and in 1971 he demonstrated the first working deep neural network. Using a feedforward 8-layer neural network, he trained his model in a “computer identification system called Alpha”. Then Kunihiro Fukushima in 1979-80, created neocognitron, a hierarchical and multi-layered ANN which learnt how to recognize visual patterns and became an inspiration for the development of convolutional neural networks.

Considered as a godfather of deep learning, Geoffrey Hinton et al. published a paper about “Learning representations by back-propagating errors” in 1986 [28]. Use of the back-propagation algorithm solved one of the biggest problems in multi-layer neural networks called the credit assignment problem (Minsky, 1961) by calculating synaptic weight changes on each layer [29].

2.1 Recurrent Neural Network

The concept of RNN (recurrent neural network) was shortly presented in 1974 [30], but the actual notion of this model was presented by Hopfield Network in 1982 [31]. Usually, depending on the data and its characteristic the right approach is being chosen. In some cases, for

2. Background

instance, text translation, stock price prediction or sentiment classification, the data is sequential and the context is crucial (data inside the sequence are not identically distributed). Using memory context in a process of translation, the following word can be heavily influenced by its predecessors.

The memory mechanism learns from the past where at timestep $t + 1$, previous information $(x_t, x_{t-1}, x_{t-2}, \dots, x_1)$ from timestep $1, \dots, t$ are projected onto the latent space c_t . The parameters θ at the new timestep $t + 1$ are re-used:

$$c_{t+1} = h_{\theta}(x_{t+1}, c_t) \quad (2.17)$$

which can be expanded to this form:

$$c_{t+1} = h_{\theta}(x_{t+1}, h_{\theta}(x_t, h_{\theta}(x_{t-1}, h_{\theta}(x_{t-2}, \dots, h_{\theta}(x_1, c_0)))) \quad (2.18)$$

By looking at this formula we can conclude the model has a repeating block which can be presented in a loop. When the memory mechanism is modelled then input, output and memory I/O are assigned weights to form a structure presented in the Figure 2.8.

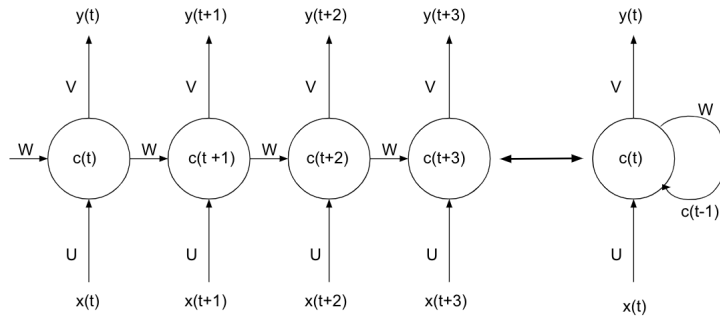


Figure 2.8: On the left side unrolled structure and its equivalent representation (using a loop) on the right side.

The model is based on a chain of the same modules with a single tanh layer (Equation 2.19) which to be scaled down to the range between (0,1) uses softmax layer as presented in Equation 2.20:

$$c_t = \tanh(U x_t + W c_{t-1}) \quad (2.19)$$

$$y_t = \text{softmax}(V c_t), \quad (2.20)$$

where standard (unit) softmax is defined as a function that after inputting a vector of M real numbers, it creates probability distribution consisting of M probabilities (which sum up to 1)

2. Background

as in the following formula:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^M e^{z_j}} \text{ for } i = 1, \dots, M \text{ and } z = (z_1, \dots, z_M) \in \mathbb{R}^M, \quad (2.21)$$

where z represents an input vector and i is i th element of this vector.

Back Propagation Through Time is an extended form of back propagation used in neural networks for optimization purposes. After unfolding an RNN the model essentially becomes a very deep in time neural architecture. In order for the model to learn, it has to be trained where by using a cost function (loss function) the output (\hat{y}) is compared to the desirable result (y). Often used logistic regression which also known as cross entropy is used to calculate the error at each timestep t (where $t = 1, \dots, T$) (2.22).

$$L_t(\hat{y}_t, y_t) = -y_t \log \hat{y}_t - (1 - y_t) \log(1 - \hat{y}_t), \quad (2.22)$$

which is used to find the overall loss of the entire sequence 2.23:

$$L(\hat{y}_t, y_t) = \sum_{t=1}^{T_y} L_t(\hat{y}_t, y_t). \quad (2.23)$$

By calculating the gradient of a loss function (with respect to the parameters of the network), the optimisation technique such as gradient descent aims to minimise the loss function by updating the weights in the model. In recurrent neural networks, backpropagation happens through the time where from L_t every single neuron that participated in this prediction at time t should have their weight updated as presented in the Figure 2.9.

Depending on the problem formulation the models' structure can have its variations (Figure 2.10). Typical Vanilla Neural Network will have fixed-size input and will give fixed-size output and can be used for instance in image classification (A). Per contra, to create a model which will output image captioning one to many relationship (image to a sequence of words) would be used (B). In a case where the input is a sequence and the output is a vector (for instance sentiment classification), many to one model structure would be created (C). When both, input and output of the architecture are sequences, depending on the type of the problem the model can have two forms as presented below. In a D sub-point the output is delayed which can be used in language translation where the form of a certain word can be heavily influenced by the others or use another structure (E) for video classification (on frame level).

2. Background

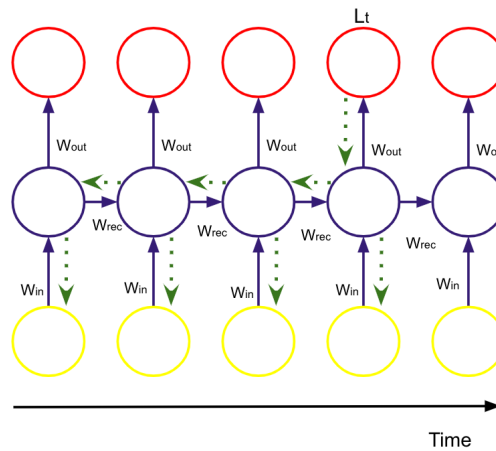


Figure 2.9: Backpropagation through time shown from the timestep t where based on L_t the weights in the previous timesteps have to be updated.

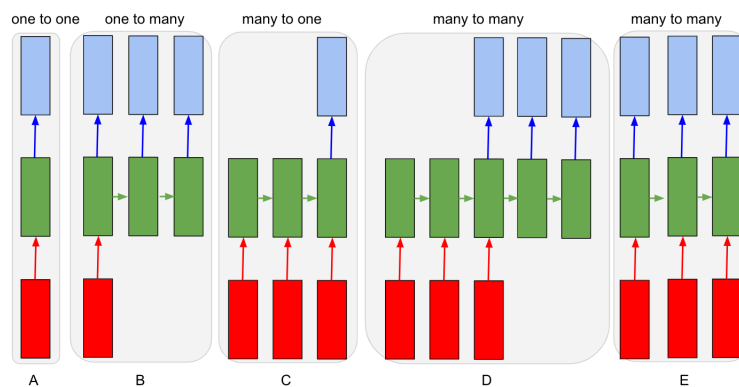


Figure 2.10: Relationship types within sequence architecture where depending on the purpose of the model the models structure varies.

2.2 Long-Short Term Memory

As previously mentioned, gradient descent is a technique used for optimizing the performance of the architecture such as RNN. While all the weights are being updated in the process of backpropagation, the common issue with long sequences is vanishing gradient descent. It is caused by the gradient of the loss function getting smaller and smaller (approaching towards zero) which makes the training process problematic. These long-term dependencies have been solved in 1997 by Sepp Hochreiter and his supervisor during his PhD Jürgen Schmidhuber. They presented Long Short-Term Memory(LSTM) [32] which structure includes 4 interacting layers as in a Figure2.11.

2. Background

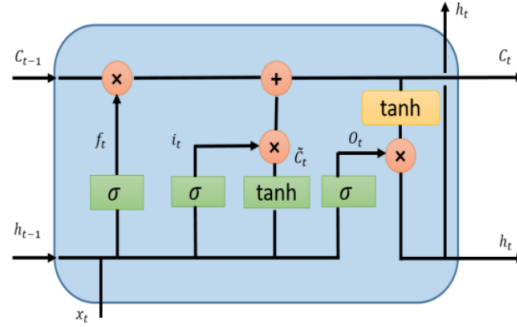


Figure 2.11: The LSTM structure with 4 interacting layers (indicated by a green colour)[§] [32].

The main idea behind this model is a use of a cell state (the horizontal line which lies at the top of the module). It works like a conveyor belt which has only 2 element-wise operations, Equation 2.27. Minor linear modifications allow to add and remove certain information from the cell state which let it stay almost unchanged. In the first gate known as *forget gate* (f_t) using sigmoid layer the model decides what information are redundant based on the output of the previous timestep h_{t-1} and a new input in the following timestep x_t (2.24).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2.24)$$

In order to decide what information shall be added, the following *input gate* (i_t) (Equation 2.25) is used to modulate the input followed by the \tilde{C} (Equation 2.26) which generates a vector of candidates using \tanh layer (Equation 2.26). Updating the cell state is the result of forgetting unnecessary information (by multiplying old cell state C_{t-1} and forget gate f_t) and adding new candidate values thanks to i_t and \tilde{C} (Equation 2.27).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2.25)$$

$$\tilde{C} = (\tanh(W_c[h_{t-1}, x_t] + b_c)) \quad (2.26)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C} \quad (2.27)$$

The last, *output gate* o_t (Equation 2.28) together with \tanh value of C_t enables finding a value to be taken by h_t (hidden layer vector), the output:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.28)$$

$$h_t = o_t \odot \tanh(C_t) \quad (2.29)$$

[§]<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

2.3 Gated Recurrent Unit

Another type of recurrent neural network presented in 2014 by Cho et al. was Gated Recurrent Unit (GRU) [33]. The model shows its simplicity which makes it a lighter version of LSTM including computational cost and its topology. The main structure of this network is presented below.

To avoid vanishing gradient problem, GRU uses update (z_t) and reset (r_t) gates. These vectors decide what information will go through to the output. To update gate we use the following formula which applies sigmoid activation function (σ):

$$z_t = \sigma(W_z[h_{t-1}, x_t]) \quad (2.30)$$

The reset gate decides what information are redundant:

$$r_t = \sigma(W_r[h_{t-1}, x_t]) \quad (2.31)$$

Then \tilde{h}_t is calculated using r_t to store related information from the previous layers. This new memory content \tilde{h}_t is then used to produce *the final memory* at the current time step h_t (2.33):

$$\tilde{h}_t = \tanh(W[r_t \odot h_{t-1}, x_t]) \quad (2.32)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (2.33)$$

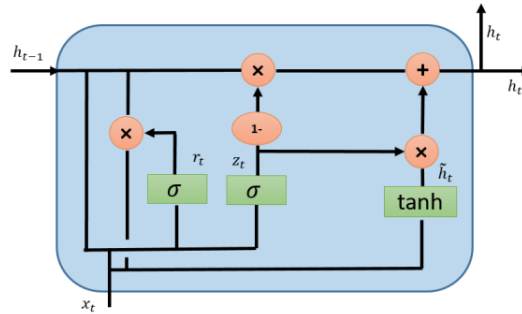


Figure 2.12: Structure of Gated Recurrent Unit ¶.

According to studies, it is unclear which RNN is *better*. While GRU is faster due to the fewer number of parameters, LSTM with sufficient computational power and enough data obtains better results [34]. Nevertheless, investigating time dependencies in spatio-temporal data using RNN will be performed.

¶<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

3 Evaluation Metrics

Evaluation is an essential part of a machine learning algorithm where the performance of the model can be verified and evaluated using various measurement techniques. Two main components that these analyses are based on are actual class labels known as well as ground truth labels (y) and predicted class values (\hat{y}). Four different categories are created based on those values. True positive (TP) values are positive samples correctly classified while false positive (FP) represent negative tuples which are not labelled correctly (as positive). Negative tuples that are correctly classified are named as true negative (TN) and false negative (FN) symbolize misclassified positive tuples as negative. These categories are used as fundamentals of metrics for evaluation accuracy.

		Predicted Class		Total
		Yes	No	
Actual Class	Yes	TP	FN	P
	No	FP	TN	N
Total		P'	N'	P + N

Figure 2.13: Confusion matrix representing combination of ground truth and predicted values resulting in four different categories: true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

Based on these values the accuracy (recognition rate), proportion of correctly predicted data samples against all data, can be calculated (2.34). The other evaluation metrics which evaluates how many selected items are relevant is based on the number of correctly predicted samples from all the positive ones is precision (2.35). The recall known as sensitivity or true positive rate calculates how many relevant items are selected as a result of a number of real positive values (TP) against all samples predicted as positive (Equation 2.36, Figure2.14). Depending on precision and recall, F1 Score is “weighted average” of them are presented in Equation 2.37.

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN} \quad (2.34)$$

$$Precision = \frac{TP}{FP + TP} \quad (2.35)$$

2. Background

$$Recall = \frac{TP}{P} \quad (2.36)$$

$$F1_Score = \frac{precision \times recall \times 2}{precision + recall} \quad (2.37)$$

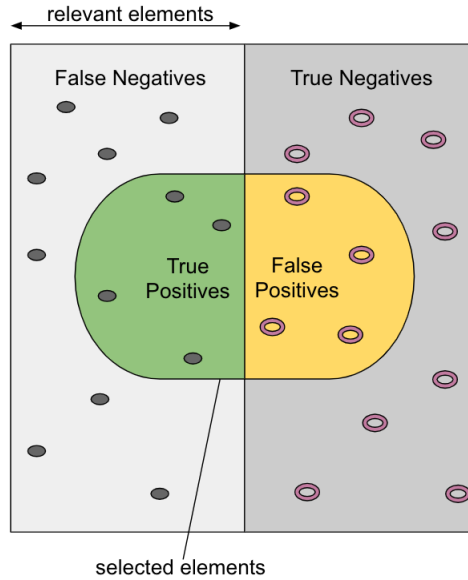


Figure 2.14: Figure presents predicted values where based on number of selected elements and relevant elements precision and recall can be calculated.

4 EEG Data

In 1875, Richard Caton as the first man, observed on the exposed brains of monkeys and rabbits EEG. Then in 1924, German psychiatrist, Hans Berger recorded the electric field of a human brain and named it electroencephalogram (EEG). Between 1929 and 1938 he published 20 scientific papers based on this discovery [35]. After describing the technique for recording the electrical brain activity from the human scalp in 1929, he faced scepticism and doubtfulness which disappeared after replicating Berger's experiment by Lord Adrian, Cambridge physiologist, in 1934 [36]. Nevertheless, his method due to its non-invasive nature is still used for diagnosis of a variety of brain diseases such as stroke, tumour and other focal brain disorders [37].

Electroencephalography is an electrophysiological monitoring technique that acquires electrical activity in a human brain using sensors (electrodes) on the surface of the head. EEG data is measured by recording voltage fluctuations from ionic current within neurons [38]. This

happens due to the synaptic excitation of dendrites inside of pyramidal neurons in the cerebral cortex where current flows are produced. *Differences of electrical potentials are caused by summed postsynaptic graded potentials from pyramidal cells that create electrical dipoles between soma (body of a neuron) and apical dendrites (neural branches)* [39].

Recording small potential changes in the EEG signal as a direct result of a thought process or perception in response to an internal or external stimulus is known as Event-Related Potentials (ERPs). It is an often-used technique for analysis of psychophysiological states within the brain. One of its biggest advantages of EEG is the temporal resolution which lets us precisely define the timing of neural activity and sequence of mental operations. However, a lack of spatial resolution leads to difficulties in localizing brain neural activity.

5 Machine Learning in Neuroscience

Deep learning has many applications in various fields where medical data analysis is one of the fields evolving rapidly using image classification, segmentation, object detection etc. [40]. Over the past few years, a great amount of research has been done towards the brain data analysis. Using Magnetic Resonance Imaging, disorder classification (Alzheimer's disease, MCI and Schizophrenia) was investigated using numerous deep learning models [41] [42].

As most of the researches based on EEG data is focused on the binary activity of a single person, multi-brain and multi-class scenario needed further investigation. Conducted by Xiang Zhang et al. research presented a "Multi-Person Brain Activity Recognition via Comprehensive EEG Signal Analysis [43]. This study approaches the multi-person and multi-class brain activity recognition solutions while dealing with massive noises in raw EEG data and the "low signal-to-noise ratio" in this data. Achieving this goal was based on employing XGBoost classifier on extracted by Autoencoder (AE) features from EEG data.

EEG data has been gathered from PhysioNet eegmmidb (EEG motor movement/imagery database), collected by BCI2000 (Brain-Computer Interface) instrumentation system. While EEG data was recorded the subject had five tasks to do:

2. Background

Task 1: The subject closes eyes and remains relaxed.

Task 2: When a target appears at the left side of the screen then subject focuses on the left hand and imagines opening and closing this hand until the target disappears.

Task 3: When a target appears at the right side of the screen then subject focuses on the right hand and imagines opening and closing this hand until the target disappears.

Task 4: When a target appears on the top of the screen then subject focuses on both hands and imagines opening and closing them until the target disappears.

Task 5: When a target appears at the bottom of the screen then subject focuses on both feet and imagines opening and closing them until the target disappears.

The experiment was conducted on 560,000 samples from 20 subjects and 5 classes. Every sample corresponded to one of the five tasks and represented a vector made of 64 channels.

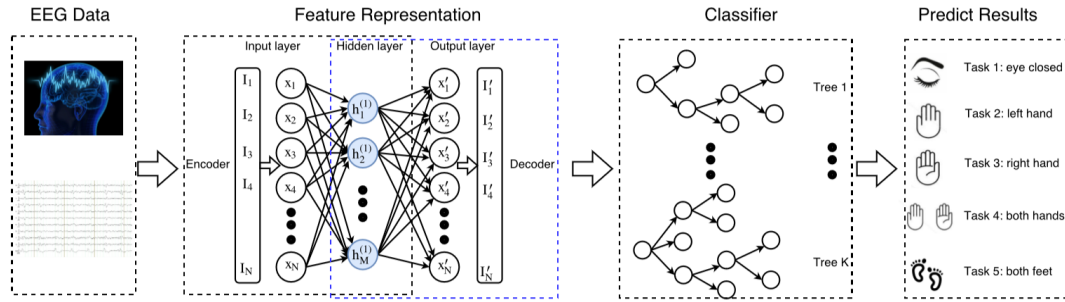


Figure 2.15: The methodology flowchart represents EEG data used as an input to Autoencoder where x_i indicates input layer, h_i hidden layer and x'_i output layer. The compact form of the data (h) is sent to an XGBoost classification model with K trees. The prediction is based on the user's brain activity and five actions (classes) they performed [43].

To discover and analyze the discrepancy between different EEG classes with robustness over different subjects they considered the effect of normalization methods, the training data size and the impact of a neuron size in hidden representation in Autoencoder (number of dimensions of extracted features).

After analyzing widespread normalization methods, a Z-score outperformed Unity and Min-Max by achieving the best test error results. This normalization method was used throughout the whole study.

2. Background

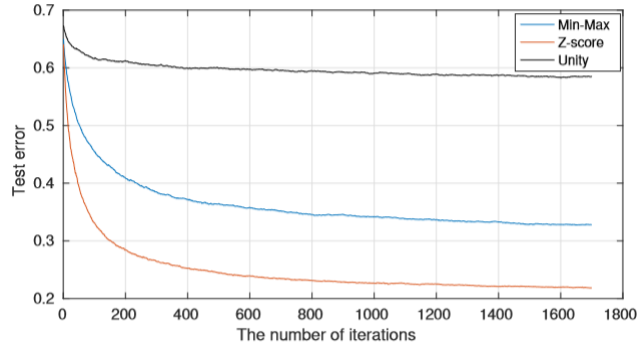


Figure 2.16: Comparison of three different normalization methods applied to 121 neurons in hidden layer which implemented into XGBoost produced test errors over the number of iterations [43].

While investigating training data size, five experiments with a proportion of 60%, 70%, 80%, 90% and 95% were considered. After performing five iterations, the test error was the lowest (0.206) for the training dataset with the proportion of 95%. This ratio, 95% training and 5% of the testing dataset, was maintained throughout the whole study.

To evaluate the performance of varied classifiers applied to EEG data, 16 different methods have been investigated. As shown in Figure 2.17, the first 9 of the classification techniques present typical data classifiers where based on the results, the highest accuracy was performed by XGBoost (0.7453). The following 7 groups present the attempt of improving this score by applying various feature extraction methods (e.g., PCA, AE and Discrete Wavelet Transform). Based on the table, the best score (0.794) was achieved by an AE applied to XGBoost method, where Basic Autoencoder with a hidden layer of 121 neurons outperformed Stacked Autoencoder with 3 hidden layers: 100, 121, 100 neurons, respectively. What's even more interesting, the Stacked Autoencoder applied to XGBoost performed worse (0.7048) than an XGBoost classifier by itself (0.7453).

No.	1	2	3	4	5	6	7	8
Classifier	SVM	RNN	LDA	RNN+SVM	CNN	DT	AdaBoost	RF
Acc	0.3333	0.6104	0.3384	0.6134	0.5729	0.3345	0.3533	0.6805
No.	9	10	11	12	13	14	15	16
Classifier	XGBoost	PCA+XGBoost	PCA+AE+XGBoost	EIG+AE+XGBoost	EIG+PCA+XGBoost	DWT+XGBoost	Stacked AE+XGBoost	AE+XGBoost
Acc	0.7453	0.7902	0.6717	0.5125	0.6937	0.7221	0.7048	0.794

2. Background

Figure 2.17: Comparison of 16 different classifications approaches that has been investigated throughout this study [43]. Classifiers presented in a table are as follows: SVM, RNN, LDA, CNN, DT, AdaBoost, RF, XGBoost. The last 7 groups uses various feature representation methods, such as: PCA (Principal component analysis), AE, EIG (eigenvector-based dimensionality reduction presented in Eigenface recognition) and DWT.

To prove the efficiency of this approach the case study was designed, where 172,800 samples were gathered and collected from 5 subjects and 6 classes. The accuracy of 74.85% outperformed state-of-the-art methods.

On the 9th April 2019, a Journal of Neural Engineering published A.Craik et al. paper about *Deep learning for electroencephalogram (EEG) classification tasks: a review* [44]. This review of the literature was based on data from the past 5 years from Web of Science and PubMed which resulted in 90 studies to analyse. The aim was to answer three critical questions common for EEG classification problems. What type of EEG classification tasks have been explored using deep learning, how the input of the data has been formulated and whether there are certain architectures suitable for a particular type of tasks.

Classification of EEG data usually follows a pipeline which includes artifact removal, feature extraction and classification. Frequently used technique for artifact removal is known as Independent Component Analysis (ICA) while Principal Component Analysis and Local Fishers Discriminant Analysis (LFDA) are effective methods for dimensionality reduction. Classical machine learning techniques such as Linear Discriminant Analysis, Support Vector Machines, and Decision Trees are commonly used in classification tasks.

Based on gathered studies, there are six different groups representing different tasks: motor imagery (22%), emotion recognition (16%), mental workload (16%), seizure detection (14%), event-related potential detection (10%) and sleep stage scoring (9%) and other studies (13%). Besides that, researchers analysed input formulation in all the studies to find three main categories of input formulation: calculated features (41%), the signal values (39%) and images (20%) as shown in the Figure 2.18. As feature extraction is dominating input formulation (41%), techniques such as statistical measures of signal, Power Spectral Density (PSD) and wavelet decomposition were found to be the most common.

2. Background

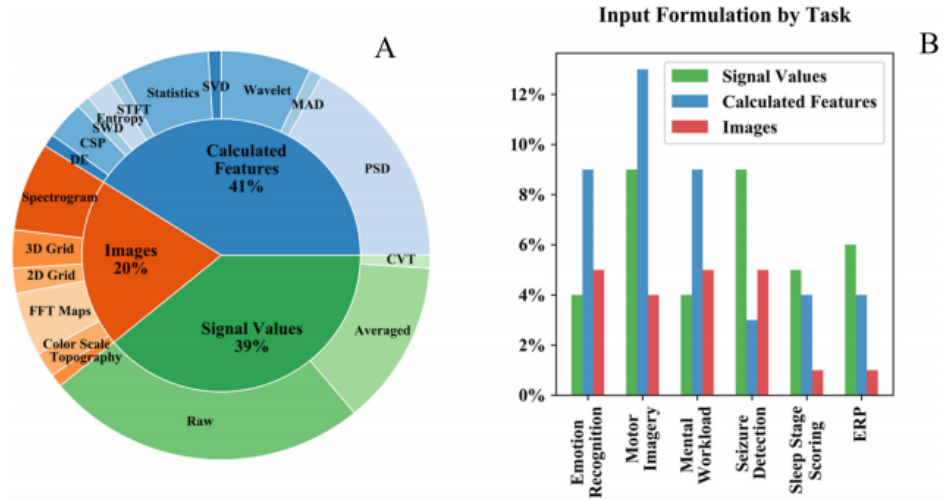


Figure 2.18: (A) Input formulations across all studies. The inner circle shows the general input formulation while the outer circle shows more specific choices. (B) General input formulation compared across different tasks. Most tasks had calculated features as inputs, with seizure detection studies instead having a much higher proportion of signal values. Key CVT: Complex Value Transformation, CSP: Common Spatial Pattern, DE: Dynamic Energy, FFT: fast Fourier Transform, MAD: Mean Absolute Difference, PSD: Power Spectral Density, STFT: Short Time Fourier Transform, SVD: Singular Value Decomposition, SWD: Swarm Decomposition [44].

Input formulations are also task-specific where emotion recognition, motor imagery and mental workload tasks use in majority calculated features whereas seizure detection, sleep stage scoring, and event-related potential analysis tend to use the signal as input values. In terms of classification techniques, Convolutional Neural Network (43%), Deep Belief Network (18%) and Recurrent Neural Network (10%) outperformed Stacked Autoencoder (8%) and Multi-Layer Perceptron Neural Networks (9%) in accuracy for classification tasks. While CNN dominates, depending on the task different architectures are also desirable as presented in the Figure 2.19.

2. Background

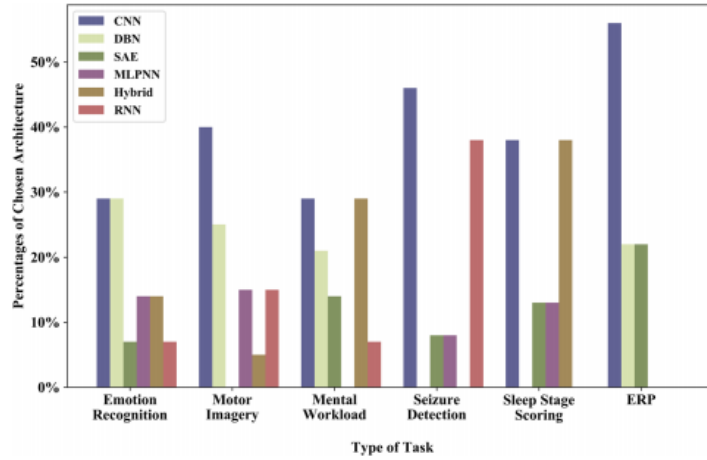


Figure 2.19: Proportions of deep learning architectures categorised by the task type. [44].

In summary, this systematic literature review presents EEG classification research undertaken within the last 5 years using machine learning techniques. As we can see the classification of EEG data can be approached in various ways depending on the data, different input formulations and classification architectures are preferable. Some tasks are still in a need for further in-depth research, yet this review gives high-quality guidance for future work.

6 Signal Decomposition

Frequently, a signal is presented as a relationship between time and amplitude. However, in many cases, the frequency of the signal is desired. A popular useful technique for analysing the frequency components of the signal is known as Fourier Transform (FT) [45]. According to Joseph Fourier's theory which is based on Euler's formula, every signal $f(t)$ can be decomposed using a series of sine waves with different frequencies ω [46](Equation 2.38). The high frequency resolution in FT enables finding peaks in the frequency spectrum which indicates the most frequently occurring frequencies in the signal. Although Fourier Transform can produce exact frequencies present in the signal, their location in time is unknown as FT has zero resolution in time-domain. An alternative type of signal decomposition which makes a trade-off between resolution in frequency and time domain is Wavelet Transform. The ability to analyse signal at different frequencies with different resolutions is known as Multiresolution Analysis (MRA).

2. Background

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (2.38)$$

Wavelet Transform uses mathematical functions, wavelets to describe the signal [47]. Wavelet is a rapidly decaying wave-like oscillation with zero mean which exists for a finite duration (Figure 2.20). There are various types of wavelets and depending on the application, the choice of the wavelet can vary.

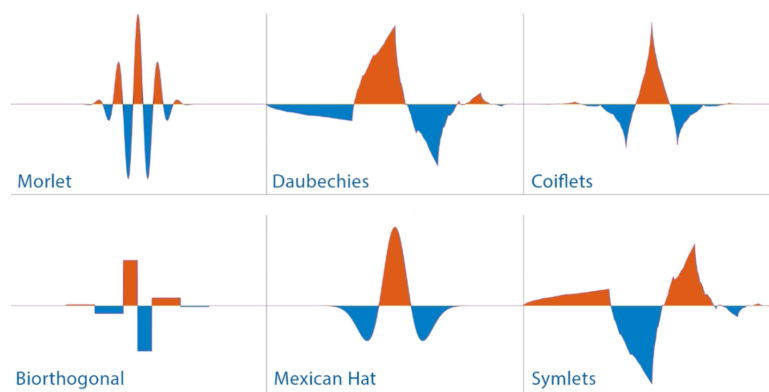


Figure 2.20: Figure presents examples of different families of wavelets (with mean 0) [¶].

Two main wavelet transform concepts are scaling and shifting. Scale refers to the signal shrinking and extending in time where the scale value is inversely proportional to the frequency. A stretched wavelet captures slowly varying changes in a signal while shrank wavelet is able to recognize abrupt changes. Hence, scales which presents interesting time-dependent features have high resolution in time-domain while scales which show interesting frequency-dependent features have high resolution in the frequency domain.

Shifting on the other hands is responsible for advancing or delaying the onset of the wavelet along the length of the signal. This helps in aligning the wavelet with the signal when we look for a feature.

There are two main types of wavelets: Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT). CWT is a sum of scaled and shifted mother wavelet function Ψ across all time of the signal which outputs coefficients of scale and time position (Equation 2.39).

[¶]WaveletFamiliesMathworks

2. Background

$$C(\text{scale}, \text{position}) = \int_{-\infty}^{\infty} \Psi(\text{scale}, \text{position}, t) dt \quad (2.39)$$

While CWT can calculate wavelet coefficients at every possible scale it causes a huge amount of data to deal with. Discretization method for the scale and the translation parameters is what differs CWT from DWT. Using so-called *dyadic* scales and positions the signal analysis can be much more efficient and still accurate. Signal analysis using wavelets is done through a filtering process where signal S is passed through low-pass and high-pass filter. Each filter produces the same amount of samples as the original signal. As the results from both filters are merged, the signal will have twice as many samples as initially. Hence, *downsampling* is performed to produce two sequences of coefficients, approximation: cA and detail: cD (each half-length of the original signal) as shown in Figure 2.21. The approximations represent high-scale and low-frequency components while details are the low-scale and high-frequency components. The signal decomposition process can be iterated, where successive approximations can be decomposed further creating wavelet decomposition tree.

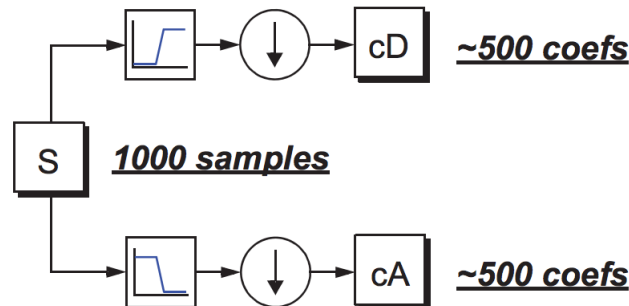


Figure 2.21: Filtering process on signal S , where the signal is passed through high-pass and low-pass filters then *downsampled* to create approximate (cA) and detail (cD) coefficients which together have approximately the same size as the original signal**.

7 Summary

In this chapter related research work has been presented where various machine learning techniques were applied to the EEG data. Presented survey of EEG classification gave us an

**[https://www.ltu.se/cms_fs/1.51590!/wavelet%20toolbox%204%20user's%20guide%20\(larger%20selection\).pdf](https://www.ltu.se/cms_fs/1.51590!/wavelet%20toolbox%204%20user's%20guide%20(larger%20selection).pdf)

2. Background

overview of task-specificity which influences the choice to input formulation and architectures used for classification. Section 4 introduced EEG data characteristics which were followed by classical machine learning techniques: Random Forest, k-NN, PCA and LDA explained in depth. An idea behind an advanced branch of machine learning, deep learning, was then introduced. Architectures which share the memory mechanism ideology with back-propagation through time learning technique (such as RNN, LSTM and GRU) were then demonstrated.

As signal can be represented in different ways, signal decomposition technique was introduced where the trade-off between frequency and time domain resolution using WT can be applied. The last part of this chapter includes various techniques used for model evaluation.

Chapter 3

Dataset

1 Experiment

Conducted by ccBrain Lab experiment used visual stimuli (varied cues) with assigned to each certainty of getting a reward to record and analyse the process of decision-making. Recruited from Cardiff University School of Psychology 23 participants (20 females) were the subjects of the experiment. The age range of participants was between 19 and 32, 22 of them were right-handed. Unfortunately due to the quality of observed data, only data of 21 participants were further used in the experiment analysis.

During experiment visual stimuli was shown on 24-inch LED monitor, situated approximately in a distance of 100cm in front of subjects. Using a response box (NATA technologies*) the decision made by a subject was recorded. Each participant had to choose between two shapes the one, with a higher probability of getting a reward. Two shapes had 100%, another two had 80% and the last ones 20% probability of payoff (Figure 3.1). All the shapes (cues) were presented on the black background and had the same colour, RGB = (246,242,92). The subject was familiar with cues and their *value* before performing a decision making task. The allocation of the reward for each shape was randomized across all the subjects and changed half-way through the experiment (for each subject).

An experiment was based on 3 conditions: equal (e.g. 100% of probability of getting a reward versus another shape with 100% of probability of payoff shape), not equal (e.g. 100% versus 20%) and single. The study was conducted in 4 blocks each having 160 trials. Participants took a break after every 40 trials and between the blocks. After block 1&2 the reward

*[www.http://www.natatech.com/](http://www.natatech.com/)

3. Dataset

probabilities were re-mapped for all participants to reduce the possibility of a bias associated with particular cue shapes. For experiment analysis block 1&2 were defined as *First Part* of the results, while block 3&4 were associated with the *Second Part*. Each block consisted of 40% of the trials in equal condition (32 trials for 100% vs. 100%, 16 for 80% vs. 80% and 16 for 20% vs. 20%); another 40% for not equal (32 trials for 80% vs. 20%, 16 for 100% vs. 20% and 16 for 100% vs. 80%) and 20% of the trials on single (16 for 100%, 8 for 80% and 8 for 20%). In equal condition the choice of a cue shape did not affect the result as the chance of getting the reward for both shapes was noted with the same probability value.

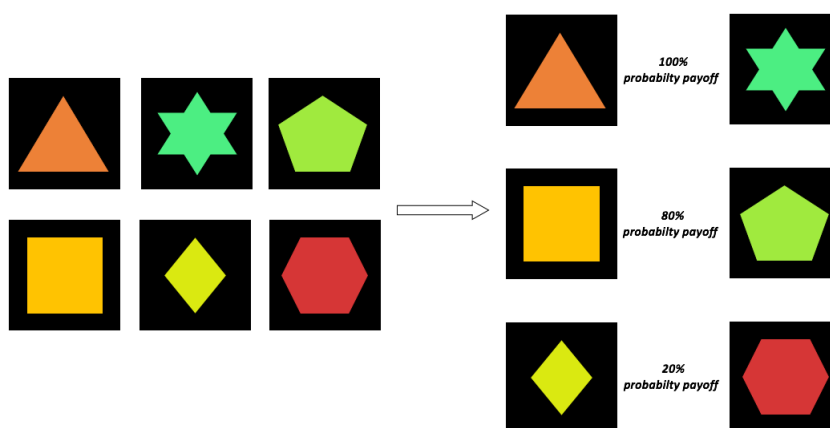


Figure 3.1: On the left side 6 different shapes were presented. The allocation of the probability of the reward to shape was randomized for all participants. The allocation of the reward probability was changed halfway through the experiment for each participant[†].

As presented in Figure 3.2 the maximum duration time of a singular procedure was 5100ms. Firstly, the subject saw a fixation point at the centre of the screen for 500ms. Then depending on the case, different variations of cues appeared on the left and right side of a fixation point in a horizontal distance of 4.34° on the screen. In equal and not equal cases 2 different cues appeared on the left and right side of the screen. The subject had to choose by pressing the left or right button using the right-hand index and middle fingers. For a single case, a shape appeared only on one side. Each subject had maximum 2000ms to make a decision. The cues disappeared after the maximum time duration was reached or the subject chose the shape. Instant feedback on the screen (whether a subject received 10 points or not) was then shown for 800ms followed by a random intertrial interval. The total game points were printed out at the bottom of the screen throughout the experiment.

3. Dataset

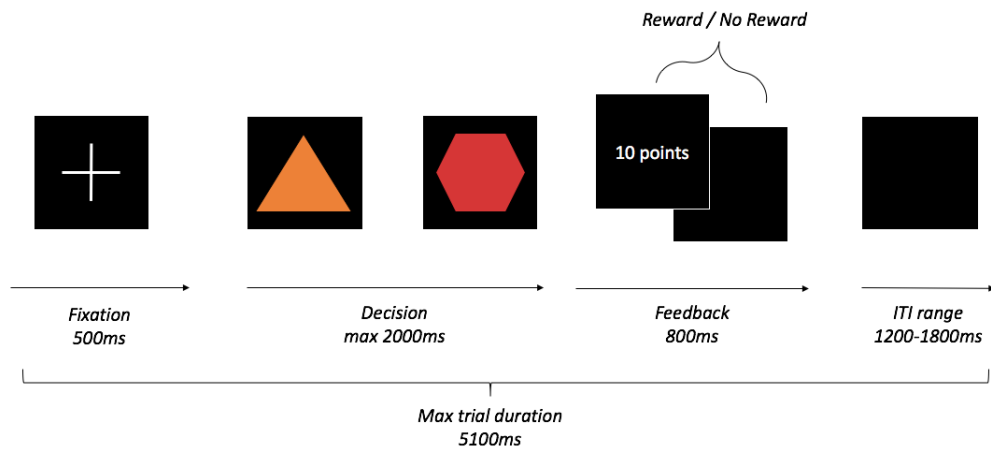


Figure 3.2: A procedure of decision making during the experiment [†].

The most interesting results of this study showed a peculiar tendency in timing decision deadlock where the higher probability of getting the rewards was, the quicker subject was able to make up their mind. Although in all three cases subjects had to choose between equally rewarding shapes, the timing was varying as shown in Figure 3.3. The reaction time (RT) of the results in all scenarios was higher for *First Part* than for *Second Part*. During the breaks, the subject saw on the screen cue reward mapping and participants were allowed to take time in the process of memorising them.

Finding the reasoning behind this decision making deadlock problem and comprehending the results of it is the main purpose of this research. Proving these 3 cases to be discriminative will confirm the initial observations gathered from ccBrain Lab.

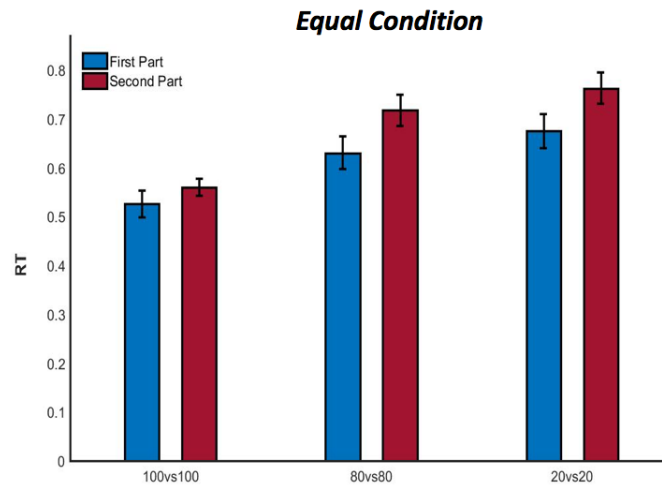


Figure 3.3: Results of equal condition where *First Part* denotes block 1&2 and *Second Part* denotes 3&4 [†].

2 Data Preprocessing

Data gathered by the Jiaxiang Zhang (principal investigator at ccBrain Lab) et al., comes from highly advanced medical equipment which they use to obtain magnetic resonance imaging, electroencephalography and Magnetoencephalography data. Using this type of data cognitive tasks like decision making, reading, remembering and paying attention can be investigated. For the purpose of this research, EEG data was gathered during the experiment [‡]. Electroencephalography is a non-invasive, electrophysiological monitoring technique that enables recording electrical activity in a brain using sensors (electrodes) on the scalp [§]. EEG data is measured by recording voltage fluctuations from ionic current within neurons [38] and it has been recorded using 32-channel Biosemi ActiveTwo device.

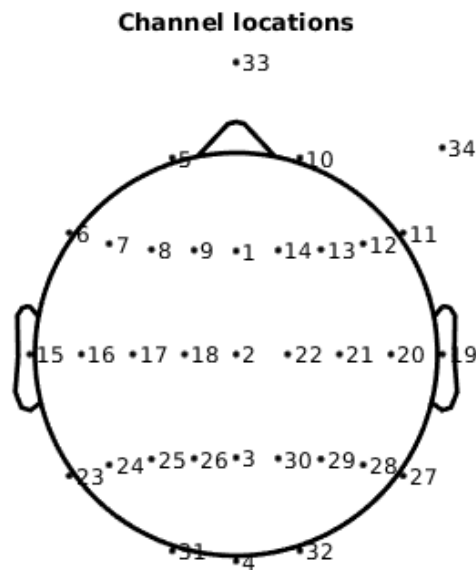
Below, Figure 3.4 presents the EEG cap with 32 channels (channel 33 and 34 are used for artifact removal, hence not relevant). Each of them is localised on the scalp where the number corresponds to the signal index that has been gathered from a particular localisation. This model by enabling us to visualise the location of channels from the data can be used for analysis in the future.

[†]<https://ccbrain.org/>

[‡]<https://www.cardiff.ac.uk/cardiff-university-brain-research-imaging-centre/facilities/electroencephalography-labs>

[§]<http://drmridha.com/services/eeg>

3. Dataset



34 of 34 electrode locations shown

Figure 3.4: Channel localisation on EEG cap model where channel 33 and 34 are irrelevant (used for removing artifacts in the preprocessing stage).

For each subject in each case, there were 32 channels and 350 time frames. In the data, there is a small non-uniformity towards the number of trials performed in different cases. While in *Equal80* and *Equal20* case the number of trials was 64, the *Equal100* case has 128 trials. This certainly has to be taken into account while modelling the classification architectures. Additionally, raw EEG data has been transformed by ccBrain Lab using several preprocessing techniques including:

- Linked ears reference subtraction
- Filtering data from 0.1 to 100 Hz + notch filter in 50 Hz; downsampled to 250 Hz
- ICA artifact rejection (decomposing signal into 50 spatial components)
- Correction of bad channels

[§]<https://ccbrain.org/>

3. Dataset

- Low-pass filtering at 40 Hz; creating epochs from -400ms to 1000ms and time-locking to the onset of the stimulus in each trial. ‘Every epoch was baseline corrected by subtracting the mean signal from -100 ms to 0 ms relative to the onset of reward cues’ [4].

2.1 Data formatting and normalization

The data provided by ccBrain Lab include three structures inside of Full_Data representing equal cases where each of them contains separate 21 subjects matrices with data. The dimensionality of the data in Equal100 is 32x350x128 and 32x350x64 in Equal80 and Equal20 cases.

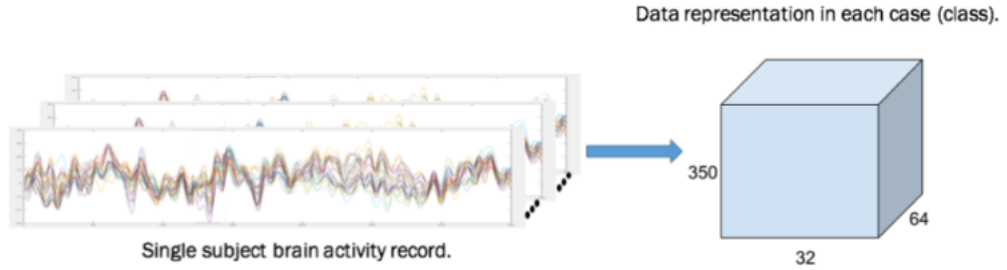


Figure 3.5: Signal data from 32 channels in single trial (350 timesteps) was transformed to be used for further analysis in a frame wise manner. Each frame represented time t with signal data from all the channels recorded in this time.

As Matlab is a powerful language to deal with numerical data, by using permutation function the data has been transformed to be read frame-wised, such that the first two dimensions of the matrix were swapped (Figure 3.5). Then, after concatenating trials along the first dimension (rows) each subject in Equal100 case had 44800 samples while Equal80 and Equal20 cases have 22400 samples (each) of data all with 32 channels.

Inspired by the Multi-Person Brain Activity Recognition via Comprehensive EEG Signal Analysis” paper [43], a normalization of the data has been implemented. To assure the input features are not dominated by the others (depending on the different scales) using normalization methods such as Min-Max Normalization, Z-score Scaling or Unity Normalization prevents occurring of such a case [48]. To achieve a range of [0,1], Min-Max Normalization ($x_{new} = \frac{x-x_{min}}{x_{max}-x_{min}}$) or Unity Normalization ($x_{new} = \frac{x}{\sum x}$) where features are being re-scaled according to the percentage or the weight of each element while Z-score is based on mean value (μ) and standard deviation (σ) in $x_{new} = \frac{x-\mu}{\sigma}$ are proposed. According to [43] classification on EEG data using

Z-score gives the best results, hence this normalization technique was applied to our data. To keep the data balanced after normalising Equal100 data, the trials were randomly shuffled and 64 trials were extracted to be used further in the research. The risk of bias towards one case was minimised by balancing the data throughout all the subjects.

2.2 Feature Extraction

Before further analysis of the EEG data, feature extraction for finding meaningful information from the data is often applied. According to a systematic literature review of EEG classification from past 5 years found on Web of Science and PubMed databases [44], statistical measures of signal, power spectral density and wavelet decomposition are most common approaches for input formulation. One of the statistical measures used in this work was 1st order derivative, known as a gradient. It represents a rate of change of a function which often is presented as the slope of the function. It is also known as a measurement of the sensitivity towards the change of the function value. The generic formula known as Leibniz's notation of a derivative f' where a change of x is expressed as dx and derivative of y with respect to x is shown below:

$$f' = \frac{dy}{dx} \quad (3.1)$$

The derivative of x can be also represented as a limit function:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (3.2)$$

When we deal with real data and simple analytic forms of the derivatives dont exist, approximation of derivatives by finite differences is used. There are three main types: forward, backward and central differences defined respectively as:

$$\Delta_h[f](x) = f(x+h) - f(x). \quad (3.3)$$

$$\nabla_h[f](x) = f(x) - f(x-h). \quad (3.4)$$

$$\delta_h[f](x) = f\left(x + \frac{1}{2}h\right) - f\left(x - \frac{1}{2}h\right), \quad (3.5)$$

where h represents spacing within the data [49]. In our data, gradient is computed using central difference in the interior points and accurate one-sides (forward or backwards) differences at the boundaries.

The second feature extracted from the data was 2nd order derivative. It denotes the rate of change of the rate of change of a point x in the graph defined as well as *the rate of change of a*

3. Dataset

quantity is itself changing[¶]. It is simply a derivative of a derivative of a function f ($f'' = (f')'$). In Leibniz's notation it is represented as:

$$\frac{d^2y}{dx^2} = \frac{d}{dx} \left(\frac{dy}{dx} \right) \quad (3.6)$$

This again is applied using previously mentioned three different finite differences on our data. As the returned differences have the same shape as the input vector, the dimensionality of each observation changed from 32 channels to 96 features.

While these feature extraction techniques are used in a both approaches, features extracted from decomposed signal are presented below.

Feature	Formula	Description
Mean	$\mu = \frac{1}{N} \left(\sum_{i=1}^N x_i \right)$	Presents central value of a discrete set of numbers by summing all the values and dividing the number of values.
Median	$\text{Median} = \left(\frac{N+1}{2} \right)^{\text{th}} \text{ term, if } N = 2k+1: k \in \mathbb{Z}$ $\text{Median} = \frac{\left(\frac{N}{2} \right)^{\text{th}} \text{ term} + \left(\frac{N}{2} + 1 \right)^{\text{th}} \text{ term}}{2},$ $\text{if } N = 2k: k \in \mathbb{Z}$	Median represents a value in the middle of a discrete set, where the set has to be in order. Depending on the number of elements in the set (N) different formula will be used.
Variance	$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$	Variance measures how spread are elements in a set from their average value.
Standard Deviation	$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}}$	Standard deviation is a square root of variance. Unlike variance it is presented in the same units as the data.
Nth Percentile Value	$n = [P/100 * N]$	Nth percentile gives a value which represents the percentage of observations falling into this group.
Root Mean Square	$x_{rms} = \sqrt{\frac{1}{N} \left(\sum_{i=1}^N (x_i)^2 \right)}$	Root Mean Square is often used as a measure of the imperfection of estimator fitting to the data.
Zero Crossing Rate	$zcr = \frac{1}{T-1} \sum_{t=1}^{T-1} 1_{\mathbb{R}<0}(s_t s_{t-1}),$ <i>where s is a signal of length T and $1_{\mathbb{R}<0}$ is an indicator function</i>	Zero crossing rate is the rate at which signal changes its value from positive to negative and from negative to positive by crossing 0.
Energy	$D = \sqrt{\sum_{i=1}^N [d(i)]^2}$	In signal processing, energy is one of the main measures of signal data. It is calculated using Frobenius Norm.
Entropy	$S = - \sum_i P_i \log P_i$	Entropy calculate rate of producing information by a stochastic source of data. It can be treated as a measure of complexity of the data.

[¶]https://en.wikipedia.org/wiki/Second_derivative

3 Summary

In this chapter, the experiment settings and procedure were presented. Additionally, the initial findings from ccBrain Lab were used to formulate the hypothesis of this research. Finding reasoning behind specificity of such behaviour would help us understand the processes happening within our brains and its activity whenever we face a decision to be made.

Furthermore, preprocessing steps performed by CUBRIC neuroscientists were stated and followed by additional data formatting and normalisation. The last section presented various techniques used for feature extraction which are used in the following experiments.

Chapter 4

Frame-wise Approach

Due to the complexity of the signal data, binary classification is investigated firstly. As mentioned in a Chapter 3, the data is structured in a way where 32 channels correspond to features which later were extracted to 96 features for each frame in a sequence. Each case has 64 trials (for each subject) where a single trial consists of 350 timesteps.

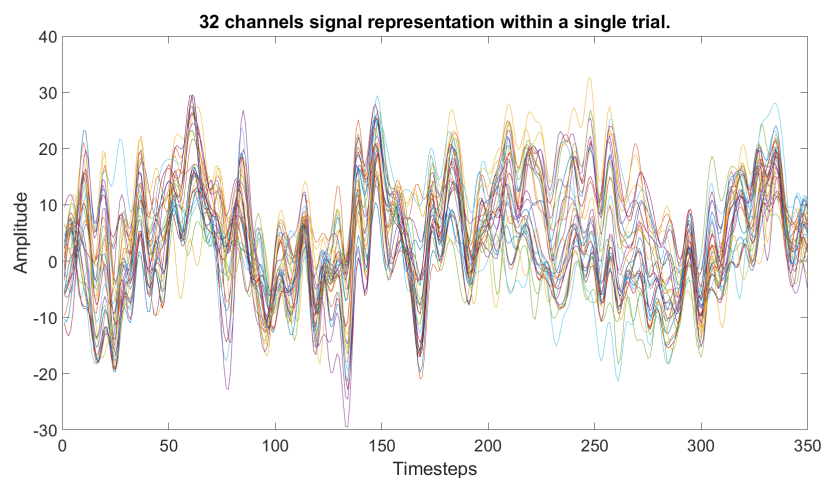


Figure 4.1: Signal presented in a single trial in Case1 (100vs100) from a single subject. 32 different channels are plotted on the graph along a time axis. The presented rectangle shows n-frame where the data is treated as a sentence of words (data from different channels in a particular time step).

The frame-wise approach considers every timestep (frame) as a single observation to be predicted. Each time frame in a sequence is a vector of 32 channels (Figure 4) and its extracted

features which belongs to the same class. For simplification, the following terminology will be used. Case1 will represent decision deadlock case between 100% and 100% probability of payoff, Case2 will stand for 20% versus 20% of payoff and finally, Case3 will represent 80% against 80% of probability of payoff. For each scenario (Case1 vs Case2, Case1 vs Case3 and Case2 vs Case3) classification is then performed.

1 Methodology

1.1 Subject Generic

For each experiment the data has been grouped where each subject G_i ($i \in \{1, 2, 3, \dots, 21\}$) represented a group. Each group contained 22400 samples with 96 dimensions (features) for each class (in binary case it gives 44800 data samples together). Using Leave One Group Out cross-validation approach, each classification was performed 21 times for a particular machine learning architecture where the model was trained on data from 20 subjects and tested on 1 unseen subjects data as shown in the Figure 4.3. Applying this validation technique enables an efficient way to utilize the data where each subject is used as a test dataset once.

The test dataset included as mentioned before 44800 samples with 350 frames. Each frame with its features is then predicted either as 0 or 1 to create 350 predictions. Using Majority Voting based on the dominance of predicted values, a trial was assigned a final prediction value (Figure 4.2). The accuracy of the prediction was then evaluated on 128 trials (two classes, each with 64 trials) based on the ground truth and predicted values (Equation 2.34).

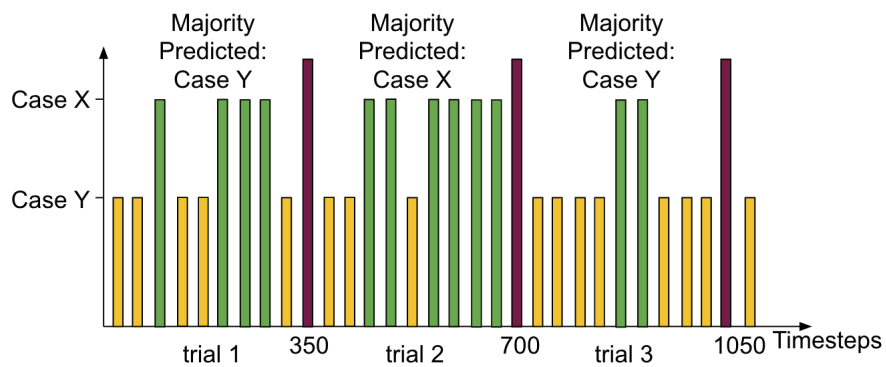


Figure 4.2: Figure captures majority voting on predicted frames within each trial (350 timesteps). Based on the majority of the predictions the trial is assigned the following prediction value.

4. Frame-wise Approach

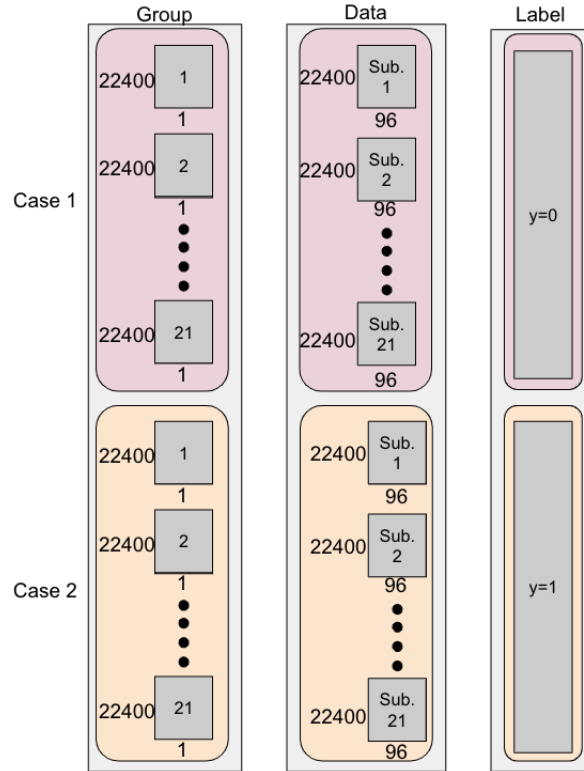


Figure 4.3: Figure presents Leave One Group Out cross-validation where each subject is assigned to a particular group (1,21) with 22400 observations (each having 96 features) from both cases to be fed to the estimator where the model will be tested on each group.

Using this methodology, classification was performed using Random Forest, k-Nearest Neighbor and Linear Discriminant Analysis. However, as the accuracy was limited and the dimensionality of the data is high, using dimensionality reducing technique to improve the performance was proposed. To ensure the information within the data was preserved Principal Component Analysis has been implemented. The modified data has been tested on previously mentioned classifiers for further comparison.

1.2 Subject Specific

Due to the complexity of the data and its subject specific nature, narrowing down the problem to only subject specific scenario was carried out. The experiment was performed on all 21 subjects with their 128 trials (form both cases). To better understand the data, get more metrics and solve an issue of a small dataset, k-fold cross-validation was performed. Once the observations have been grouped by sequences, the data has been shuffled and split into 4 folds. As the

number of folds had to be divisible by the number of trials and splitting data in half does not leave much data for training the model while 8 folds leave test dataset with only 16 sequences from both cases, 4-fold cross-validation was chosen. The test dataset followed the same idea of predicting values with majority voting. The result of all the folds from one subject was then averaged with additional standard deviation information. The classification was performed on the same models as in the subject generic scenario.

2 Results and Discussion

2.1 k-Nearest Neighbors

As k-NN is a non-parametric model, i.e. it makes no assumption about the data distribution hence no prior knowledge of the data is necessary and it has been applied to our data. k-NN relies on a feature space of the training data, thus it can be sensitive to the outliers. Although there is no training step this algorithm can be slow if brute force (compare one-by-one to each training instance) approach is chosen for finding k nearest neighbours. To avoid this issue, as mentioned in Section 1.2, space partitioning algorithm, Ball Tree has been applied. Since the feature space where data is projected needs to have some measure of the distance the euclidean metric system has been used. The most troublesome part of this model is choosing the value of k , which after a couple of iterations has been set to 5.

As presented in the table below (Figure 4.4) the model prediction is restricted. While comparing accuracy between subject generic (S.G.) and subject specific (S.S.) approach, on average subject generic case performs slightly better which can be caused by greater training dataset (enabling feature space to be more interpretable). Yet, the highest averaged accuracy using k-NN was achieved by Subject 3 (subject specific), with accuracy $\sim 67\%$. Despite the fact that *Case1vs2* in subject generic approach achieved overall better results than *Case1vs3* and *Case2vs3*, in subject specific methodology *Case1vs2* performs slightly worse and *Case1vs3* has higher accuracy. As the accuracy in subject specific scenario is averaged across the folds, standard deviation showed a divergence of the results of the folds where *Case2vs3* has the highest differences.

4. Frame-wise Approach

Subjects	k-NN								
	Subject Generic			Subject Specific					
	Case1vs2	Case1vs3	Case2vs3	Case1vs2		Case1vs3		Case2vs3	
	mean	mean	mean	mean	std.	mean	std.	mean	std.
1	0.484375	0.492188	0.5	0.515625	0.0716	0.4375	0.04941	0.382813	0.02591
2	0.539063	0.507813	0.46875	0.453125	0.09244	0.492188	0.07773	0.445313	0.05579
3	0.507813	0.523438	0.40625	0.671875	0.06442	0.578125	0.05182	0.460938	0.02591
4	0.515625	0.492188	0.523438	0.476563	0.02591	0.484375	0.10246	0.5	0.05413
5	0.554688	0.5	0.53125	0.46875	0.07967	0.476563	0.05579	0.554688	0.08942
6	0.5	0.453125	0.484375	0.390625	0.11158	0.539063	0.01353	0.523438	0.11771
7	0.554688	0.515625	0.492188	0.5	0.05413	0.507813	0.07773	0.460938	0.07453
8	0.46875	0.507813	0.570313	0.507813	0.06001	0.46875	0.04941	0.546875	0.10005
9	0.523438	0.5	0.5	0.5	0.05846	0.546875	0.10005	0.554688	0.07773
10	0.523438	0.507813	0.460938	0.507813	0.04622	0.53125	0.08839	0.5	0.09111
11	0.5	0.515625	0.453125	0.429688	0.02591	0.492188	0.04622	0.382813	0.06001
12	0.5	0.515625	0.484375	0.507813	0.08942	0.5	0.04941	0.492188	0.05579
13	0.492188	0.453125	0.460938	0.453125	0.08976	0.507813	0.05123	0.429688	0.03405
14	0.484375	0.46875	0.507813	0.382813	0.06766	0.515625	0.06442	0.4375	0.04941
15	0.453125	0.476563	0.476563	0.453125	0.15703	0.515625	0.01562	0.445313	0.09211
16	0.523438	0.476563	0.53125	0.460938	0.06001	0.414063	0.07118	0.429688	0.01353
17	0.476563	0.507813	0.492188	0.382813	0.07118	0.460938	0.07773	0.46875	0.07967
18	0.476563	0.46875	0.484375	0.453125	0.08414	0.578125	0.1661	0.421875	0.02706
19	0.539063	0.492188	0.492188	0.4375	0.09632	0.476563	0.04622	0.453125	0.03494
20	0.546875	0.460938	0.585938	0.5	0.05413	0.4375	0.05846	0.429688	0.04622
21	0.484375	0.484375	0.53125	0.570313	0.05579	0.46875	0.05413	0.453125	0.05634

Figure 4.4: Figure represents kNN for both, subject generic and subject specific case for all the subjects. In subject generic case, the following subject in the row was used as a test dataset during LOGO cross-validation.

Unfortunately, k-NN often suffers from the curse of dimensionality where although it works well for a small number of input variables, increased dimensionality of the data leads to a limited performance where k-NN struggles to predict the output of the unseen data point. Besides that, k-NN does not have the ability to prioritise attributes to decide what features are more valuable, each variable has the same *importance*, when predicting the class of the test data set.

2.2 Random Forest

Random Forest is an ensemble technique which has been implemented for a couple of reasons. It is more robust and efficient than most algorithms when dealing with large dimensional space or a great amount of training data. While a single decision tree tends to overfit the data, random forest combines the results of multiple decision trees to prevent it. As the model uses bagging (randomly sampling subsets with replacement) on both dataset and features of the trees in the forest, it enables the forest to reduce variance in comparison to regular decision trees (with high variance and low bias). Unlike k-NN, RF is capable of providing feature

4. Frame-wise Approach

importance measure which can indicate what variables are most significant among all.

Since the data is continuous, to define *the best split* Gini impurity has been applied together with CART algorithm (both are mentioned in Chapter 2, Section 1.1). As decision trees can be affected by outliers and noise in the data or simply overfit, a pre-pruning technique which specifies the minimum number of samples required to create a leaf node has been applied with a value of *min_samples_leaf* equal to 15. Also, the performance of the random forest is influenced by a number of estimators (trees) in the forest. The test was performed on 5, 10 and 15 estimators where although the difference in performance was minimal, 10 trees have been presented.

Subjects	Random Forest								
	Subject Generic			Subject Specific					
	Case1vs2	Case1vs3	Case2vs3	Case1vs2		Case1vs3		Case2vs3	
	mean	mean	mean	mean	std.	mean	std.	mean	std.
1	0.476563	0.476563	0.546875	0.4765625	0.080813	0.46875	0.054127	0.4375	0.054127
2	0.59375	0.507813	0.601563	0.5546875	0.034053	0.46875	0.049411	0.484375	0.064424
3	0.710938	0.609375	0.570313	0.8046875	0.13509	0.6171875	0.170269	0.40625	0.044194
4	0.554688	0.546875	0.492188	0.4296875	0.013531	0.484375	0.046875	0.421875	0.027063
5	0.601563	0.476563	0.5	0.6171875	0.143843	0.4453125	0.109096	0.6015625	0.080813
6	0.4375	0.546875	0.5	0.4296875	0.104524	0.3984375	0.060009	0.4921875	0.142136
7	0.46875	0.46875	0.523438	0.4296875	0.080813	0.4140625	0.034053	0.4453125	0.025911
8	0.492188	0.5625	0.507813	0.4453125	0.055792	0.4140625	0.025911	0.3828125	0.055792
9	0.46875	0.53125	0.492188	0.5078125	0.0340539	0.4375	0.096319	0.4296875	0.109096
10	0.5	0.539063	0.507813	0.40625	0.0441941	0.515625	0.089759	0.515625	0.071602
11	0.476563	0.515625	0.5	0.546875	0.068108	0.4296875	0.08378	0.4453125	0.025911
12	0.515625	0.507813	0.46875	0.4921875	0.1177071	0.4921875	0.16883	0.421875	0.071602
13	0.53125	0.523438	0.546875	0.4609375	0.025911	0.4140625	0.013532	0.4609375	0.05123
14	0.492188	0.476563	0.53125	0.40625	0.03125	0.5234375	0.060009	0.4296875	0.067658
15	0.5625	0.507813	0.515625	0.7265625	0.060009	0.421875	0.084143	0.4140625	0.046219
16	0.53125	0.492188	0.515625	0.453125	0.0349385	0.359375	0.034939	0.515625	0.051822
17	0.484375	0.507813	0.46875	0.421875	0.071602	0.4375	0.03125	0.4296875	0.040595
18	0.515625	0.554688	0.625	0.4921875	0.0462193	0.4296875	0.040595	0.46875	0.105974
19	0.484375	0.523438	0.453125	0.453125	0.0644235	0.4453125	0.040594	0.3984375	0.040595
20	0.53125	0.484375	0.484375	0.453125	0.015625	0.40625	0.022097	0.4609375	0.060009
21	0.523438	0.515625	0.5625	0.4921875	0.0259111	0.4375	0.038273	0.453125	0.046875

Figure 4.5: Figure represents RF for both, subject generic and subject specific case for all the subjects. In subject generic case, the following subject in the row was used as a test dataset during LOGO cross-validation.

Although, RF performs better than k-NN in subject generic scenario when it comes to subject specific, k-NN seems to interpret the data a little better than RF. Subject generic classification outperforms subject specific in all cases. In both S.G. and S.S. *Case1vs2* performs the best while *Case1vs3* is lower (yet the difference in overall performance in *Case1vs3* and *Case2vs3* is narrow). Again, Subject 3 achieves the best accuracy of $\sim 80\%$ (subject specific,

4. Frame-wise Approach

Case1vs2).

Feature importance as one of the strengths of RF has the potential to improve the performance of a model by reducing a number of features and preserving only the *important* ones. Features whose importance value is above a certain threshold are then used to transform the data and make the prediction. To find the most suitable threshold, feature importance for each variable was calculated. Based on the sample of these values with the index of the features in Figure 4.6 value 0.01 was chosen as the threshold. The number of features varied between 48 and 54. The experiment was performed on the subject specific scenario where the accuracy achieved by original data(with 96 dimensions) and modified data was compared for each subject in all cases.

Index	Feat. Importance	Index	Feat. Importance	Index	Feat. Importance	Index	Feat. Importance
1	0.009832912	25	0.010781358	49	0.010188368	73	0.009638605
2	0.010124693	26	0.010326619	50	0.009797038	74	0.010650419
3	0.010649185	27	0.011113144	51	0.011228225	75	0.010797136
4	0.011273169	28	0.011221928	52	0.010377281	76	0.010610304
5	0.010984972	29	0.010721182	53	0.009783267	77	0.01000131
6	0.011441831	30	0.011106282	54	0.009668446	78	0.009335636
7	0.010802143	31	0.011073185	55	0.01077811	79	0.011007929
8	0.01045691	32	0.011208717	56	0.010242421	80	0.010464232
9	0.010029401	33	0.009726331	57	0.009780975	81	0.010029689
10	0.011487619	34	0.009672194	58	0.00967221	82	0.009204029
11	0.011292942	35	0.009970752	59	0.010606819	83	0.010984466
12	0.010884635	36	0.010115503	60	0.009987035	84	0.010312483
13	0.01078322	37	0.010682821	61	0.009633407	85	0.009856391
14	0.009717154	38	0.010917102	62	0.011553987	86	0.009473838
15	0.011617582	39	0.010520216	63	0.010488489	87	0.010445912
16	0.011329239	40	0.010226414	64	0.010235294	88	0.010275079
17	0.010751683	41	0.009604222	65	0.009298715	89	0.009557632
18	0.010090953	42	0.010745104	66	0.009216658	90	0.009409122
19	0.011320848	43	0.010730136	67	0.009667175	91	0.010264364
20	0.011249434	44	0.010901977	68	0.010300639	92	0.010076403
21	0.010527922	45	0.010117182	69	0.010432116	93	0.009682303
22	0.010012039	46	0.009517185	70	0.010884197	94	0.01145246
23	0.011143019	47	0.011074115	71	0.010603378	95	0.010201105
24	0.010826281	48	0.010584808	72	0.010170917	96	0.010385717

Figure 4.6: Table represents an example of an index of each feature and its feature importance.

In spite of the conjectures, the performance of the model on the data with important features decreased by 1%. Based on it, further speculations of feature representation of the data can be developed.

4. Frame-wise Approach

Subjects	Case1vs2				Case1vs3				Case2vs3			
	Orig. Feat.		Feat. Import.		Orig. Feat.		Feat. Import.		Orig. Feat.		Feat. Import.	
	mean	std.	mean	std.	mean	std.	mean	std.	mean	std.	mean	std.
1	0.476563	0.06001	0.460938	0.02591	0.390625	0.08119	0.367188	0.06001	0.515625	0.14062	0.46875	0.07655
2	0.523438	0.08942	0.484375	0.05182	0.4375	0.04419	0.429688	0.04622	0.453125	0.06442	0.421875	0.03494
3	0.859375	0.09244	0.851563	0.06766	0.757813	0.07773	0.789063	0.05123	0.460938	0.05579	0.4375	0.04419
4	0.523438	0.08378	0.523438	0.07773	0.453125	0.05182	0.46875	0.04941	0.398438	0.02591	0.398438	0.01353
5	0.601563	0.02591	0.664063	0.04622	0.429688	0.04059	0.40625	0	0.6875	0.07655	0.75	0.05846
6	0.421875	0.03494	0.421875	0.01562	0.460938	0.06395	0.460938	0.04622	0.4375	0.0221	0.460938	0.02591
7	0.5	0.09632	0.40625	0.0221	0.460938	0.07773	0.46875	0.07655	0.460938	0.04622	0.429688	0.04622
8	0.40625	0.05846	0.398438	0.06766	0.507813	0.08664	0.515625	0.01562	0.445313	0.04622	0.453125	0.05182
9	0.453125	0.02706	0.421875	0.07812	0.53125	0.09632	0.539063	0.08378	0.382813	0.06395	0.40625	0.07655
10	0.382813	0.05579	0.40625	0.05846	0.515625	0.03494	0.570313	0.03405	0.414063	0.04059	0.429688	0.06395
11	0.515625	0.0716	0.484375	0.0716	0.429688	0.11561	0.421875	0.10005	0.484375	0.04688	0.445313	0.01353
12	0.539063	0.07453	0.484375	0.05634	0.476563	0.11131	0.484375	0.11375	0.5	0.07329	0.484375	0.03494
13	0.453125	0.06442	0.453125	0.03494	0.453125	0.04688	0.382813	0.04059	0.375	0.03125	0.382813	0.04059
14	0.375	0.04941	0.367188	0.06001	0.421875	0.01562	0.429688	0.02591	0.382813	0.08378	0.453125	0.12597
15	0.578125	0.10246	0.585938	0.11561	0.460938	0.06001	0.453125	0.08114	0.460938	0.10216	0.40625	0.09375
16	0.460938	0.03405	0.5	0.04941	0.484375	0.03494	0.390625	0.03494	0.515625	0.07812	0.492188	0.03405
17	0.460938	0.05579	0.4375	0.0221	0.492188	0.02591	0.539063	0.07453	0.429688	0.04059	0.375	0.08839
18	0.390625	0.06442	0.359375	0.08414	0.578125	0.05182	0.546875	0.02706	0.492188	0.09726	0.445313	0.02591
19	0.445313	0.06001	0.445313	0.04059	0.460938	0.07118	0.4375	0.04941	0.382813	0.01353	0.414063	0.06766
20	0.484375	0.04688	0.445313	0.06001	0.484375	0.06442	0.484375	0.12597	0.5625	0.07967	0.460938	0.06001
21	0.460938	0.06001	0.476563	0.06766	0.375	0.03827	0.40625	0.0221	0.398438	0.01353	0.367188	0.06001
Avg.	0.491072	0.101356	0.479911	0.107433	0.479167	0.076787	0.475819	0.089626	0.459078	0.071761	0.446801	0.075577

Figure 4.7: Figure represents subject specific prediction using Random Forest on orig. (original) data with 96 features and after thresholding feature importance to 0.01.

2.3 Linear Discriminant Analysis

While k-NN and RF are building models based on similarities in the data, Linear Discriminant Analysis (LDA) focuses on creating differences between the classes of the data. It attempts to explain the data based on linear combinations of the features within and between the classes. Implementing this algorithm was an attempt for finding a linear approach to explain and classify the dataset.

One of the advantages of this model is no hyperparameter tuning. However, the solver option can benefit the data where chosen for this experiment Singular Value Decomposition approach does not compute the covariance matrix, hence it is suitable for high dimensional data.

As presented in the table for both methodologies, subject generic and subject specific, the accuracy is restricted. The highest value was achieved by Subject 2 (~ 58%) in *Case1vs2* (S.G.). *Case1vs2* also performs the best in both subject generic and subject specific scenario. Standard deviation in S.S. indicates higher diversity of the results in *Case1vs3* and *Case2vs3*.

4. Frame-wise Approach

Subjects	Linear Discriminant Analysis								
	Subject Generic			Subject Specific					
	Case1vs2	Case1vs3	Case2vs3	Case1vs2		Case1vs3		Case2vs3	
	mean	mean	mean	mean	std.	mean	std.	mean	std.
1	0.53125	0.46875	0.539063	0.4140625	0.05579	0.3984375	0.04622	0.46875	0
2	0.578125	0.4375	0.546875	0.453125	0.01562	0.390625	0.05182	0.40625	0.05846
3	0.515625	0.507813	0.539063	0.421875	0.01562	0.453125	0.02706	0.421875	0.03494
4	0.515625	0.546875	0.515625	0.4296875	0.10452	0.4375	0.03827	0.40625	0.04941
5	0.5	0.476563	0.539063	0.453125	0.01562	0.4375	0.03125	0.4140625	0.05579
6	0.460938	0.507813	0.484375	0.4375	0.0221	0.4375	0.03827	0.40625	0.04941
7	0.539063	0.476563	0.515625	0.453125	0.01562	0.4609375	0.01353	0.46875	0.05846
8	0.476563	0.484375	0.421875	0.4296875	0.02591	0.4375	0.03125	0.453125	0.02706
9	0.554688	0.4375	0.53125	0.453125	0.01562	0.453125	0.01562	0.421875	0.03494
10	0.53125	0.492188	0.492188	0.4375	0.0221	0.4375	0.03125	0.421875	0.03494
11	0.46875	0.523438	0.554688	0.40625	0.05846	0.390625	0.01562	0.40625	0.04941
12	0.476563	0.421875	0.539063	0.4375	0.0221	0.453125	0.05182	0.359375	0.03494
13	0.398438	0.554688	0.515625	0.515625	0.13711	0.40625	0.04941	0.3984375	0.06766
14	0.570313	0.4375	0.4375	0.421875	0.03494	0.453125	0.01562	0.4296875	0.06395
15	0.554688	0.476563	0.4375	0.421875	0.03494	0.34375	0	0.375	0.05846
16	0.554688	0.523438	0.476563	0.421875	0.05182	0.421875	0.03494	0.421875	0.08119
17	0.539063	0.460938	0.5	0.4140625	0.02591	0.40625	0.05846	0.390625	0.06442
18	0.460938	0.546875	0.507813	0.4609375	0.02591	0.453125	0.01562	0.453125	0.10938
19	0.507813	0.4375	0.4375	0.46875	0.0221	0.4296875	0.02591	0.390625	0.05634
20	0.5	0.4375	0.4375	0.453125	0.01562	0.4375	0.04419	0.4375	0.03827
21	0.523438	0.476563	0.398438	0.421875	0.01562	0.4296875	0.02591	0.4296875	0.02591

Figure 4.8: Figure represents LDA for both, subject generic and subject specific case for all the subjects. In the subject generic case, the following subject in the row was used as a test dataset during LOGO cross-validation.

Based on the results of the introduced models across all the subjects, the general comparison of their performance can be created. Figure 4.9 presented below shows the average accuracy in both approaches across all the cases. While in subject generic case Random Forest achieves the highest results, in subject specific: k-NN performs overall slightly better than remaining architectures. Random Forest uses bootstrap aggregation which helps to reduce variance and it works well with big datasets hence it performed better than others. For subject specific scenario, k-NN that relies on the feature space was using the data from the same subject for training and testing procedure. Diverse brain activity and signal formulation seem to be subjective for each patient. This observation aligns with a neuroscientific notion of a brain fingerprint where each person has their own signature pattern [50]. According to the results, Linear Discriminant Analysis performs the worst which gives an assumption of data being too complex for a linear classifier such as LDA.

4. Frame-wise Approach

	Case	RF		k-NN		LDA	
		mean	std.	mean	std.	mean	std.
Subject Generic	1vs2	0.521577	0.058657	0.507068	0.028619	0.512277	0.042762
	1vs3	0.517857	0.03368	0.491443	0.021143	0.482515	0.039536
	2vs3	0.519717	0.042355	0.497024	0.039135	0.493676	0.046352
Subject Specific	1vs2	0.5	0.103305	0.477307	0.06333	0.43936	0.0241
	1vs3	0.450521	0.053496	0.496652	0.04221	0.42708	0.02805
	2vs3	0.453125	0.047978	0.465402	0.04897	0.41815	0.02766

Figure 4.9: Figure represents average accuracy and standard deviation of the performance across all the subjects in k-NN, RF and LDA for a subject generic and subject specific approach.

2.4 Dimensionality Reduction Model Enhancement

As the analysed data is high dimensional, it can be challenging for a model to explore and interpret it. An alternative way to deal with the dimensionality of the data is to use a dimensionality reduction technique such as PCA.

Described previously PCA, uses principle components to project the data onto itself while aiming to maximise variance across the data. In this case, 95% of *energy* was aimed to be maintained which means that the amount of variance to explain the data has to be greater than the specified percentage. While in subject generic methodology this concurred to reducing the number of features from 96 to 31, in subject specific scenario it varied across subjects and models. The following result from each machine learning technique (k-NN, RF and LDA) was presented in the tables below.

After applying PCA and classifying data using k-NN (Figure 4.10), the accuracy of the best performance (*Case1vs2*, S.S.) dropped to 60%. Although *Case1vs3* in S.S. gives respectively lower results, overall the model has slightly better data interpretation prediction in both, S.G. and S.S. scenarios across all the subjects. Standard deviation in subject specific scenario is less diverse across the subjects than in experiment without PCA, hence prediction of the folds varied less. This can be noticed especially by looking at *Case2vs3* in S.S.

Comparing the results in Random Forest (S.G.), *Case1vs2* and *Case1vs3* performs better for most of the subjects after dimensionality reduction (Figure 4.11). Previously mentioned accuracy of a Subject 3, S.S., ($\sim 80\%$) dropped down to $\sim 61\%$. For the majority of the subjects in S.S. the accuracy slightly improved, yet the average accuracy across the subjects shows a mild decrease. Also, standard deviation in subject specific scenario with PCA has lower diversity across the folds. Besides that, subject generic case still predicts better than

4. Frame-wise Approach

Subjects	k-NN + PCA								
	Subject Generic			Subject Specific					
	Case1vs2	Case1vs3	Case2vs3	Case1vs2		Case1vs3		Case2vs3	
	mean	mean	mean	mean	std.	mean	std.	mean	std.
1	0.484375	0.5	0.5390625	0.4375	0.07967	0.421875	0.10005	0.46875	0.0221
2	0.5	0.53125	0.453125	0.515625	0.07812	0.3828125	0.06766	0.4765625	0.06766
3	0.5	0.484375	0.4765625	0.6015625	0.09726	0.5859375	0.07118	0.46875	0.07655
4	0.53125	0.5390625	0.53125	0.4609375	0.07773	0.53125	0.10126	0.3984375	0.07773
5	0.546875	0.53125	0.5078125	0.5859375	0.05579	0.4609375	0.05123	0.578125	0.1279
6	0.4609375	0.484375	0.5234375	0.5	0.10126	0.46875	0.06629	0.4609375	0.06395
7	0.5390625	0.484375	0.53125	0.4140625	0.05123	0.4921875	0.10452	0.5234375	0.1091
8	0.5	0.4921875	0.53125	0.4609375	0.06766	0.3984375	0.04622	0.4921875	0.03405
9	0.5390625	0.5234375	0.5	0.515625	0.06442	0.484375	0.04688	0.484375	0.15068
10	0.515625	0.453125	0.4921875	0.421875	0.0716	0.4765625	0.06001	0.4375	0.06629
11	0.5234375	0.5	0.53125	0.5078125	0.08942	0.484375	0.04688	0.4765625	0.04622
12	0.515625	0.53125	0.4921875	0.5234375	0.06001	0.484375	0.09244	0.4609375	0.11131
13	0.5234375	0.484375	0.453125	0.4296875	0.09472	0.40625	0.07655	0.4296875	0.06001
14	0.46875	0.53125	0.4296875	0.4453125	0.05579	0.3671875	0.04622	0.53125	0.06629
15	0.5078125	0.5234375	0.4609375	0.5	0.05846	0.390625	0.0716	0.53125	0.03827
16	0.5234375	0.4921875	0.546875	0.421875	0.05182	0.390625	0.05634	0.4140625	0.06395
17	0.4921875	0.5234375	0.5	0.4296875	0.05579	0.3828125	0.06766	0.5078125	0.07118
18	0.5546875	0.4609375	0.53125	0.5390625	0.06766	0.5546875	0.10216	0.375	0.07967
19	0.546875	0.5078125	0.46875	0.46875	0.03125	0.53125	0.04419	0.421875	0.03494
20	0.4765625	0.5234375	0.5078125	0.53125	0.05846	0.4453125	0.02591	0.4921875	0.06766
21	0.484375	0.46875	0.5546875	0.4609375	0.03405	0.4921875	0.02591	0.46875	0.05846

Figure 4.10: Figure represents k-NN+PCA for both, subject generic and subject specific case for all the subjects. In subject generic case, the following subject in the row was used as a test dataset during LOGO cross validation.

subject specific after applying PCA.

Linear Discriminant Analysis with PCA performed on average a little better ($\sim 2\%$) in subject specific scenario (Figure 4.12). The prediction in subject generic case before and after using PCA is fairly comparable in where 11 out of 21 subjects performed better after using PCA in *Case1vs2* and *Case1vs3*, but only 8 subjects achieved higher score after dimensionality reduction. The average across all the subjects shows that 2 out of 3 cases (*Case1vs3* and *Case2vs3*) increased in their performance after introducing PCA to the model.

4. Frame-wise Approach

Subjects	Random Forest + PCA								
	Subject Generic			Subject Specific					
	Case1vs2	Case1vs3	Case2vs3	Case1vs2		Case1vs3		Case2vs3	
	mean	mean	mean	mean	std.	mean	std.	mean	std.
1	0.507813	0.476563	0.476563	0.570313	0.025911	0.539063	0.060009	0.53125	0.058463
2	0.53125	0.539063	0.539063	0.539063	0.071175	0.429688	0.046219	0.421875	0.071603
3	0.59375	0.5625	0.460938	0.609375	0.207877	0.554688	0.077733	0.476563	0.074527
4	0.53125	0.492188	0.5625	0.4375	0.110485	0.460938	0.025911	0.414063	0.040595
5	0.59375	0.492188	0.515625	0.421875	0.015625	0.507813	0.071175	0.476563	0.12573
6	0.5	0.476563	0.484375	0.429688	0.060009	0.484375	0.051822	0.4375	0.058463
7	0.507813	0.53125	0.484375	0.453125	0.046875	0.398438	0.025911	0.53125	0.038273
8	0.5	0.570313	0.515625	0.53125	0.058463	0.515625	0.046875	0.445313	0.05123
9	0.484375	0.546875	0.539063	0.507813	0.102162	0.539063	0.040595	0.46875	0.058463
10	0.476563	0.492188	0.460938	0.421875	0.015625	0.507813	0.092108	0.445313	0.117707
11	0.515625	0.5625	0.53125	0.484375	0.068108	0.460938	0.080813	0.398438	0.034054
12	0.546875	0.507813	0.40625	0.492188	0.135091	0.554688	0.086645	0.46875	0.044194
13	0.546875	0.507813	0.453125	0.460938	0.034054	0.4375	0.038273	0.476563	0.040595
14	0.53125	0.476563	0.5	0.46875	0.076547	0.4375	0.044194	0.390625	0.056337
15	0.554688	0.507813	0.546875	0.554688	0.067658	0.484375	0.051822	0.460938	0.040595
16	0.515625	0.46875	0.5	0.578125	0.056337	0.421875	0.015625	0.421875	0.057622
17	0.53125	0.546875	0.546875	0.570313	0.097265	0.445313	0.034054	0.421875	0.034939
18	0.453125	0.46875	0.5	0.414063	0.060009	0.539063	0.067658	0.429688	0.067658
19	0.484375	0.4375	0.515625	0.460938	0.046219	0.476563	0.055792	0.40625	0.022097
20	0.539063	0.546875	0.445313	0.421875	0.056337	0.445313	0.060009	0.484375	0.10246
21	0.5	0.484375	0.492188	0.46875	0.058463	0.429688	0.040595	0.414063	0.063948

Figure 4.11: Figure represents RF+PCA for both, subject generic and subject specific case for all the subjects. In the subject generic case, the following subject in the row was used as a test dataset during LOGO cross-validation.

Subjects	Linear Discriminant Analysis + PCA								
	Subject Generic			Subject Specific					
	Case1vs2	Case1vs3	Case2vs3	Case1vs2		Case1vs3		Case2vs3	
	mean	mean	mean	mean	std.	mean	std.	mean	std.
1	0.507813	0.507813	0.476563	0.4375	0.0221	0.484375	0.01562	0.4296875	0.03405
2	0.554688	0.492188	0.492188	0.4375	0.0221	0.453125	0.01562	0.40625	0.03125
3	0.578125	0.484375	0.484375	0.453125	0.02706	0.453125	0.06811	0.4453125	0.02591
4	0.5	0.46875	0.546875	0.453125	0.02706	0.4609375	0.01353	0.4609375	0.04059
5	0.445313	0.539063	0.445313	0.4375	0	0.4609375	0.04622	0.40625	0.04419
6	0.492188	0.484375	0.453125	0.453125	0.01562	0.421875	0.01562	0.390625	0.05634
7	0.5	0.523438	0.515625	0.453125	0.02706	0.453125	0.06811	0.453125	0.01562
8	0.554688	0.5	0.5	0.421875	0.02706	0.4375	0.03827	0.390625	0.01562
9	0.523438	0.515625	0.476563	0.390625	0.06442	0.421875	0.03494	0.453125	0.01562
10	0.476563	0.46875	0.539063	0.421875	0.03494	0.453125	0.01562	0.40625	0.0221
11	0.539063	0.5	0.507813	0.4140625	0.04059	0.4375	0	0.4453125	0.06001
12	0.460938	0.5	0.414063	0.46875	0.13441	0.4375	0.03827	0.421875	0.04688
13	0.578125	0.492188	0.484375	0.484375	0.12002	0.421875	0.03494	0.4375	0.03827
14	0.492188	0.585938	0.507813	0.4375	0.0221	0.4453125	0.05579	0.4453125	0.01353
15	0.460938	0.453125	0.507813	0.453125	0.03494	0.4375	0.03827	0.390625	0.01562
16	0.4375	0.5625	0.5	0.390625	0.05634	0.40625	0.05846	0.4140625	0.05123
17	0.5	0.429688	0.476563	0.40625	0.04419	0.46875	0.04419	0.453125	0.01562
18	0.492188	0.570313	0.484375	0.390625	0.08414	0.4921875	0.02591	0.4765625	0.07118
19	0.507813	0.539063	0.492188	0.390625	0.05182	0.4375	0.0221	0.421875	0.03494
20	0.539063	0.492188	0.523438	0.4375	0.0221	0.4375	0.03827	0.4765625	0.03405
21	0.53125	0.585938	0.484375	0.4375	0.03827	0.421875	0.04688	0.4375	0.0221

Figure 4.12: Figure represents LDA+PCA for both, subject generic and subject specific case for all the subjects. In the subject generic case, the following subject in the row was used as a test dataset during LOGO cross-validation.

4. Frame-wise Approach

	Case	RF+PCA		k-NN+PCA		LDA+PCA	
		mean	std.	mean	std.	mean	std.
Subject Generic	1vs2	0.521205	0.034363	0.51116	0.02636	0.50818	0.039135
	1vs3	0.509301	0.036614	0.50335	0.025121	0.5093	0.041101
	2vs3	0.498884	0.038522	0.50298	0.034222	0.49107	0.029557
Subject Specific	1vs2	0.490327	0.059199	0.484375	0.0526	0.431919	0.02632
	1vs3	0.479539	0.046966	0.458705	0.06081	0.44494	0.02081
	2vs3	0.448661	0.038392	0.471354	0.04739	0.431548	0.02618

Figure 4.13: Figure represents average accuracy and standard deviation of the performance across all the subjects in k-NN, RF and LDA for subject generic and subject specific approach after performing dimensionality reduction using PCA.

3 Result Overview

In all 6 models and 13 experiments presented above, various methods were used for analysing the data. Unfortunately, the results are restricted in all cases. The first approach of subject generic idea on k-NN, RF and LDA were used which resulted in RF achieving better scores as it deals best with high dimensional data and large datasets (Figure 4.16). Proposed importance feature did not influence models performance. Despite the expectations, subject specific approach didn't improve the overall accuracy. However, while subject specific models were trained on the subset of 33600 frames (96 trials), in the subject generic method the model used LOGO cross-validation where training dataset consisted of 20 subjects, i.e. 896000 observations (2560 trials).

As one of the characteristics of high dimensional data is its complexity, the dimensionality reduction was tested out. The overall results are shown in Figure 4.16. No significant improvement was experienced. Since k-NN is prone to curse of dimensionality it showed small improvement after applying PCA to the data. The lowest-performing model LDA, in both approaches, with and without PCA, gave limited prediction outcomes proving that this problem is not linearly separable. Interestingly, in majority of the experiments *Case1vs2* had the highest accuracy showing that Case 100% versus 100% against Case 20% versus 20% is the most distinguishable which aligns with our hypothesis, Chapter 3 Figure3.3.

4. Frame-wise Approach

3.1 Significant Cognition Period

As in both of our methodologies we use frame-wise prediction with majority voting on a trial (which includes 350 frames/observations) the value of each trial was assigned by the dominating output across a trial. However, by observing a number of frames being correctly predicted in each trial the theory of analysing data as a sequence with a prediction pattern was shaped. To comprehend and investigate this idea, the number of correct predictions per each frame was tracked across all the subjects in k-NN, S.S. scenario.

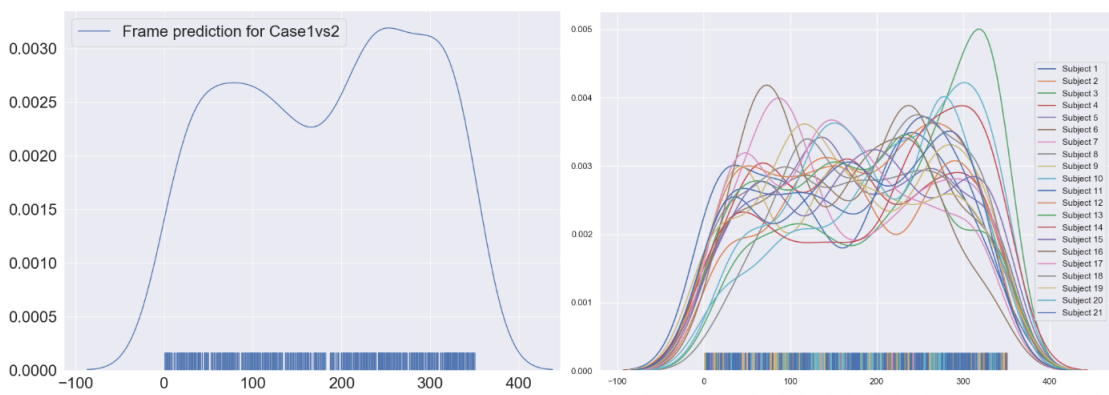
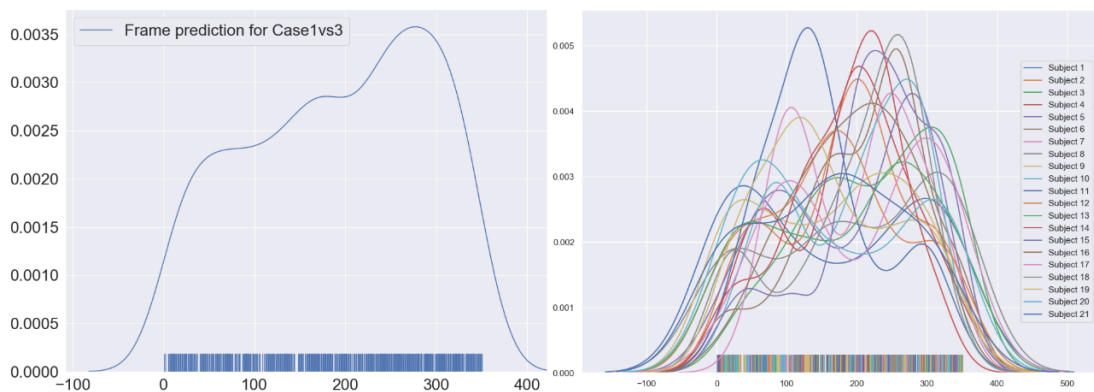


Figure 4.14: The figure presents Case1vs2 with two visualisations of the density of correctly predicted frames across the sequence. On the right side each subject has its distribution of the observations which is merged into one, across subject signal prediction pattern in the plot on the left.



4. Frame-wise Approach

Figure 4.15: The figure presents Case1vs3 with two visualisations of the density of correctly predicted frames across the sequence. On the right side each subject has its distribution of the observations which is merged into one, across subject signal prediction pattern in the plot on the left.

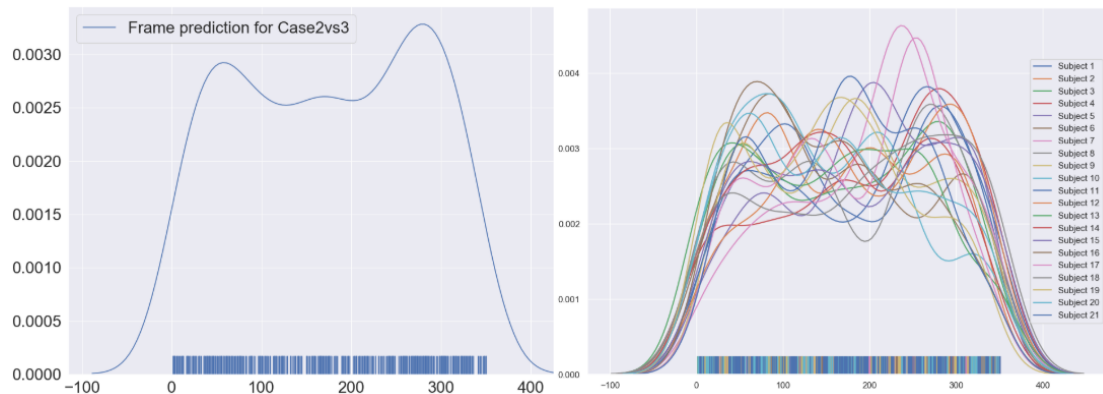


Figure 4.16: The figure presents Case2vs3 with two visualisations of the density of correctly predicted frames across the sequence. On the right side, each subject has its distribution of the observations which is merged into one, across subject signal prediction pattern in the plot on the left.

Based on the presented results, in all 3 scenarios (*Case1vs2*, *Case1vs3*, *Case2vs3*), there is a certain pattern across sequence. Every figure presents the density of correctly predicted frames within a sequence of 350 frames. For each frame number of corrected predictions was calculated and thresholded for enhanced visualisation. As test dataset consists of 128 trials in subject specific case, the threshold was set to half of this value (64 trials). Presented figures include two visualisations where on the right side univariate distribution was presented for each subject. Then the results across all the patients were merged to find cross subject sequence prediction pattern. As a result, all three scenarios present patterns of frames in a sequence that contribute to higher performance (sequence prediction). This essentially gives us theory on cognition pattern being time dependant where generic pattern across the subject is shared.

Besides that, among all scenarios, the end of a sequence contributes to the prediction the most. Yet, in *Case1vs2* and *Case2vs3* there is information carried out at the beginning of the sequence while in *Case1vs3* the information builds up across the sequence. If the data would be

treated as a sequence where information would be carried across the frames using a memory mechanism to add up for the final prediction, the data could be discriminated satisfactorily. This assumption leads us to the next step of the sequential approach.

4 Summary

This chapter presents the first approach for addressing the problem where each frame within the signal is predicted separately. A single observation is made of 32 channels values from a particular time frame t with additionally concatenated features (first and second-order derivative) extracted from it. Using majority voting of these predictions, the final outcome for every trial sequence was defined. Two methodologies: subject generic and subject specific are used in this approach. The subject generic methodology uses Leave One Group Out cross-validation to efficiently utilize the data and ensure each subjects data would be used as test data.

The second approach deals with a single subject's data to investigate subject specificity aspect of it. As each subject has 64 trials in a case, for binary classification the total amount of samples is equal to 44800 (128 trials). Using 4-fold cross-validation the model was able to get more insight into the data despite the small size of it.

Due to the complexity and high dimensionality of the feature space, dimensionality reduction technique: PCA was applied to both methodologies. Based on the results of these experiments, classical machine learning techniques turned out to be inefficient. While the subject generic approach scored slightly higher, we presume it is due to the difference of training samples (subject specific: 33600 training observations, subject generic: 896000 samples). The results of LDA model showed the complexity of the data and its inability to be linearly separated. As Random Forest deals the best with large dimensionality and size of the data it performed slightly better than others. PCA didn't provide significant improvement where in fact Random Forests score was slightly lower while k-NN due to the smaller feature space performed a little bit better. According to the observations, Case1vs2 is the most distinguishable which lines up with the observations made by ccBrain Lab.

Using feature importance ranking significant cognition period was observed. By tracking correctly predicted frames within the trials for each case (across all the subjects) the generic pattern for making a decision in a timeline was introduced. These conclusions give us the grounds to suspect time-dependencies in our data.

Chapter 5

Sequential Approach

In Chapter 4 frame-wise prediction ended up being a fiasco, yet after further analysis signal data presented a unique prediction pattern within a sequence. To perform a further investigation, the experiments for sequence classification were carried out in subject specific scenario with 4-fold cross-validation like in frame-wise approach. Two methodologies are introduced in this section where different signal representation is used in classification.

1 Methodology

1.1 Self Learnt Features with Neural Networks

While in frame-wise approach majority voting across the frames indicated the final prediction value of the sequence, no information between the frames was shared. Based on the findings the beginning and the end of the signal sequence contributes the most to its correct classification. Using memory mechanism where information from the previous frames $(n-1), (n-2), \dots, 1$ where $\{n = 1, 2, \dots, 350\}$, $n \in \mathbb{N}$ is passed to the following n^{th} frame could improve the prediction of the signal. Using time dependencies for sequential data classification is a common technique for preserving the memory context as mentioned in a Chapter 2 Section 2.1.

For each trial, a sequence of frames is input to the model to predict the value of the sequence using a many-to-one relationship. Previously introduced group of deep learning architectures which act as a chain of the same modules falls into the group of Recurrent Neural Networks where Vanilla Recurrent Neural Network, Long-Short Term Memory and Gated Recurrent Unit are implemented.

1.2 Hand-Crafted Features

Hitherto, data analysis was performed in the time domain where the signals from all the channels were presented in a time-amplitude manner. As mentioned previously in [44] another common approach for signal analysis uses frequency of the signal for an input formulation (Figure 2.18).

The Fourier Transform (FT) decomposes a signal into the frequencies it consists of yet it has zero resolution in the time domain. As we presume the data to have time dependencies, preserving time component is crucial. To achieve it discrete wavelet decomposition was applied. While various wavelets have been tested, Daubechies 2 (db2) is one of the wavelet commonly used for feature detection and it is effective with noisy data *.

To decompose signal the level of the decomposition has to be specified. Using Equation 5.1 the maximum level was computed using length of the input vector and filters length (db2) to result in max_level equal to 6. Signal from each channel was then decomposed into 6 different levels of approximate and detail coefficients which wavelet decomposition vector consists of (Figure 5.1).

$$max_level = \lfloor \log_2 \left(\frac{data_len}{filter_len - 1} \right) \rfloor \quad (5.1)$$

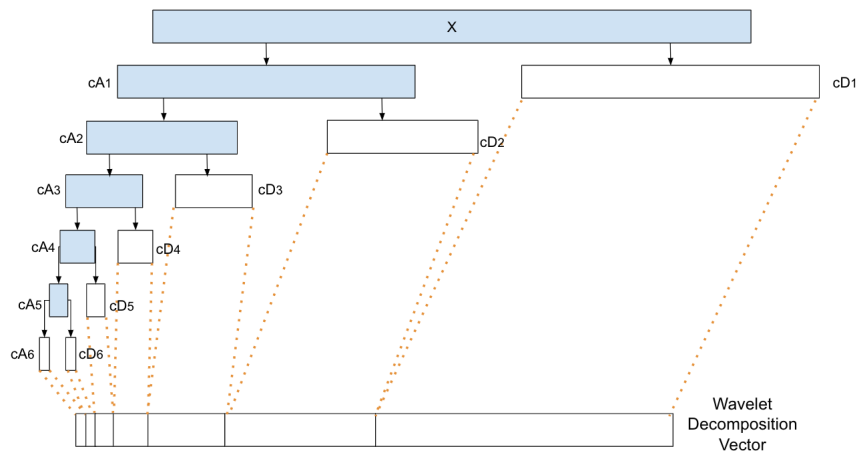


Figure 5.1: Representation of wavelet decomposition from data X into 6 different levels of wavelet coefficients with denoted as cD(1-6) and cA(1-6), where c - coefficient, D - Detail/ A - Approximation and the number represents level of decomposition.

*<https://www.mathworks.com/help/wavelet/gs/choose-a-wavelet.html>

5. Sequential Approach

As a result, from previously presented sample trial with 32 channels of the signal in Figure 4.1 data has been transformed to frequency-time domain and produced following data representation:

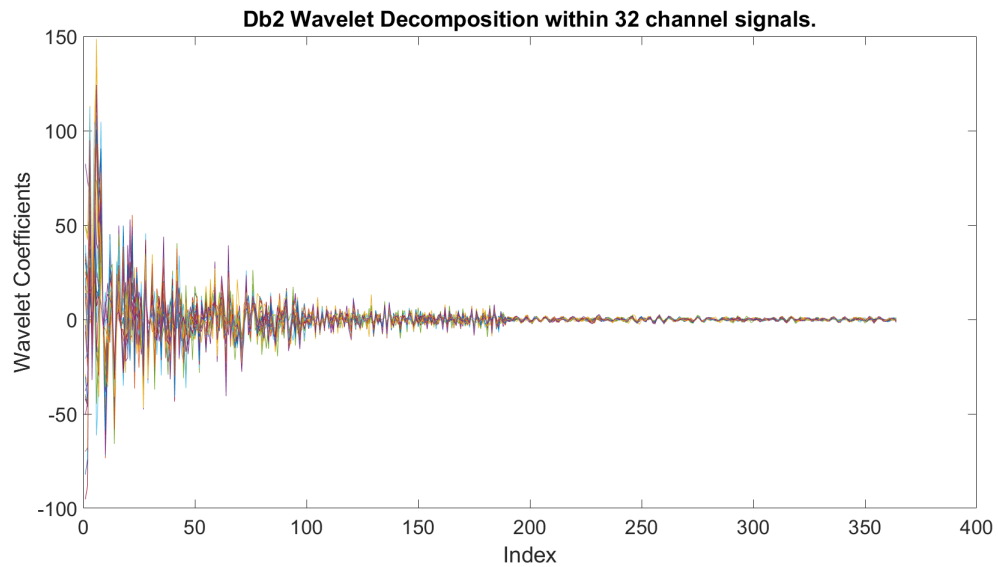


Figure 5.2: Transformation of sample signals from a single trial presented in Figure 4.1 to frequency-time domain.

Following the structure of the wavelet decomposition vector presented in Figure 5.1 deepest levels of decomposition with the highest frequency components are situated in a front part of the vector. For further analysis of the coefficients, different sub-bands of a signal were visualised. From previously presented decomposed signals, channel one was visualised below, Figure 5.3. While $cA6$, $cD6$, $cD5$, $cD4$, and $cD3$ show great variety in coefficients (for this specific signal decomposition (37,-20), (44,-67), (49,-27), (20,-36) and (14,-14) respectively), the remaining detail coefficients from level 1 and 2 are not that informative with margin (5,-5) and (1,-1).

To perform sequential classification, all channels within an observation have to be included. To avoid redundant features and curse of the dimensionality, coefficients with higher diversity of the frequency components from $cA6$, $cD6$, $cD5$, $cD4$, and $cD3$ are included further in experiments.

5. Sequential Approach

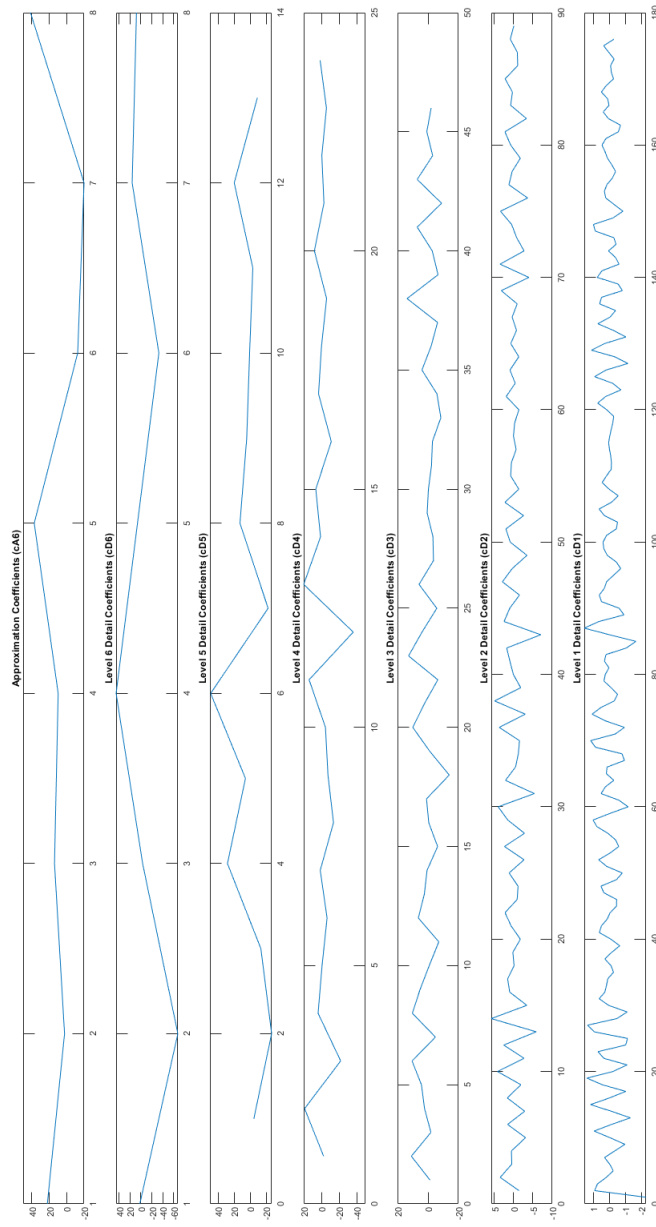


Figure 5.3: Figure presents an example of a signal decomposition with detail wavelet coefficients in 6 different levels. Each subplot represents different part of the wavelet decomposition vector, which are denoted by $cD(1-6)$ and $cA6$, where c - coefficient, D - Detail/ A - Approximation and the number represents level of decomposition. Sub-bands $cA6$, $cD6$, $cD5$, $cD4$, and $cD3$ present high frequency signal decomposition.

While changing signal representation to gather insightful information from the data, additional feature extraction is performed. In Chapter 3 second set of presented features including: frobenius norm [51], entropy, zero crossing, mean crossing, 5th percentile value, 25th percentile value, 75th percentile value, 95th percentile value, mean, median, standard deviation, variance and Root Mean Square value were calculated for each of the sub-bands resulting in 65 features for each channel signal. To achieve connectivity between the channels within the trial, features from 32 channels were flattened into a single observation to be then fed into a classifier. Due to the high dimensional space of the data, Random Forest and Fully Connect Neural Network were tested.

2 Results and Discussion

2.1 Self Learnt Features with Neural Networks

2.1.1 Recurrent Neural Network

As mentioned before in Chapter 2, Section 2.1 Vanilla Recurrent Neural Network is an architecture that re-uses cell (group of units) across the sequence. A number of units highly depends on the feature space, where various instances are examined. The optimal number of units with tanh activation in our case turned out to be 48. Moreover, to reduce the complexity of the model, regularization technique, dropout of 0.2 (where randomly chosen 20% of units within a layer are deactivated to prevent overfitting) was added to the architecture. The final, dense layer which outputs 1 value $\{0,1\}$ with sigmoid activation function was used.

During the learning process, the model compares the ground truth label with the predicted output which is done using loss function. In our case, where the output is binary $\{0,1\}$, binary cross-entropy (Equation 2.22) was implemented. To update the weights of the model an optimizer such as Adam [52] is used. The model can learn in a stochastic or batch-size way where the weights update is done after every single observation (stochastic) or a specified number of samples (a batch). The stochastic approach was chosen in this architecture. The optimal number of iterations through the entire dataset during the training process, known as well as a number of epochs, was set to 10 where the higher number of iterations caused even larger overfitting issue.

5. Sequential Approach

Subjects	Case1vs2				Case1vs3				Case2vs3			
	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc
1	0.88532	0.15844	0.61719	0.04622	0.99808	0.22101	0.52344	0.08378	1.10636	0.18916	0.53906	0.05579
2	0.89139	0.04902	0.53906	0.06395	0.94097	0.16188	0.53906	0.12377	1.12742	0.20987	0.46094	0.08081
3	1.09224	0.1267	0.54688	0.04688	1.30319	0.27296	0.45313	0.04688	1.12955	0.0573	0.44531	0.03405
4	1.19649	0.21118	0.53906	0.06001	1.13984	0.23361	0.53125	0.06988	1.05317	0.10406	0.51563	0.03494
5	0.86458	0.09832	0.5625	0.09632	1.16634	0.12134	0.45313	0.07813	1.23737	0.17735	0.39844	0.04622
6	1.25189	0.27288	0.41406	0.08942	1.33943	0.27089	0.48438	0.02706	1.39575	0.17069	0.41406	0.08081
7	1.17529	0.24885	0.52344	0.05579	1.00189	0.28699	0.59375	0.09632	1.09692	0.19469	0.53906	0.03405
8	0.81628	0.11095	0.57813	0.04688	0.85296	0.14174	0.58594	0.04622	1.00775	0.13275	0.51563	0.09504
9	1.02233	0.21463	0.53906	0.08942	1.103	0.09988	0.45313	0.05634	1.09973	0.27724	0.49219	0.09211
10	1.47318	0.41967	0.52344	0.08378	1.02889	0.07674	0.54688	0.05182	1.00936	0.08812	0.5	0.03125
11	0.95625	0.20439	0.5625	0.03827	0.97392	0.26364	0.57031	0.08081	0.99326	0.11196	0.46094	0.06001
12	1.05105	0.23725	0.47656	0.04622	1.01171	0.20456	0.52344	0.04622	1.10076	0.26852	0.53125	0.06988
13	1.16982	0.14343	0.5625	0.03125	1.11356	0.18709	0.51563	0.08414	1.10346	0.15368	0.57031	0.08081
14	1.18472	0.30087	0.50781	0.06001	1.09978	0.29095	0.46875	0.07329	1.02961	0.32193	0.55469	0.05123
15	0.87349	0.04376	0.5625	0.06629	0.98939	0.1377	0.53125	0.09632	1.08392	0.14721	0.52344	0.09472
16	1.16667	0.31758	0.51563	0.14741	1.03133	0.3156	0.59375	0.07967	0.97334	0.13977	0.55469	0.04622
17	0.87937	0.18674	0.60938	0.07813	1.20642	0.2909	0.48438	0.14741	1.07718	0.23175	0.53125	0.13441
18	1.03127	0.22365	0.54688	0.04688	1.21998	0.3304	0.50781	0.08942	1.20509	0.27201	0.51563	0.09244
19	1.05508	0.17879	0.52344	0.01353	1.04742	0.3964	0.60156	0.11348	0.96886	0.15793	0.57813	0.09244
20	1.03715	0.26392	0.55469	0.05579	1.01534	0.16276	0.57031	0.08378	0.92319	0.18483	0.57031	0.05123
21	1.48612	0.11672	0.4375	0.03827	1.18629	0.26505	0.53125	0.12885	1.29074	0.14035	0.4375	0.03827

Figure 5.4: Vanilla Recurrent Neural Network results in the subject specific scenario where for each case 21 subject recordings are presented. Each subject experiment was performed on 4-cross validation folds which were averaged at the end to produce average loss, average accuracy and standard deviation of both metrics.

Although the results are tight, *Case2vs3* remains less distinguishable than others and achieves lowest scores. Across all the cases, Subjects 5,6, and 21 struggled with predicting data. *Case1vs2* and *Case1vs3* perform similarly, yet on average *Case1vs2* scores higher which supports initial observations presented in Figure(3.3). The highest accuracy was achieved by Subject 1 in *Case1vs2*: 61.719%. Average loss value presents the average error between the predicted value and the actual label which model tried to minimise. Yet, the smallest value of the loss is 0.816 while in its peak it is 1.486. Although many hyperparameters were tweaked, the model struggled to learn effectively. As the classification of long sequences can be challenging for architectures such as RNN which often suffer from exploding/vanishing gradient, Long Short-Term Memory architecture was implemented.

2.1.2 Long-Short Term Memory

Long-Short Term Memory uses the cell unit to improve long term dependencies within the sequence. After multiple tests, the optimal architecture giving the highest results was finally established. The architecture had 3 stacked LSTM layers with 144, 96 and 48 units respectively.

5. Sequential Approach

Each of them was followed by a dropout layer of 50% with two dense layers at the end (10 and 1 output space values using sigmoid activation function).

Similarly to RNN architecture, the model was trained using binary cross-entropy and weights were adjusted thanks to Adam optimizer. While previously the model used a stochastic learning approach, LSTM performed slightly better using 32-batch size training method. The optimal level of epochs was set to 15.

Subjects	Case1vs2				Case1vs3				Case2vs3			
	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc
1	1.37566	0.22745	0.54688	0.03494	1.33663	0.21759	0.57813	0.04688	1.25826	0.28613	0.5	0.06629
2	1.2541	0.14798	0.44531	0.04622	1.23718	0.33389	0.53125	0.07655	1.34451	0.29376	0.44531	0.06395
3	1.04754	0.16439	0.65625	0.0625	1.18755	0.23115	0.61719	0.09727	1.31776	0.25699	0.5625	0.09375
4	1.36784	0.27799	0.57031	0.07118	1.29874	0.13555	0.52344	0.10216	1.50537	0.26886	0.4375	0.13441
5	1.09359	0.31124	0.64063	0.06811	1.48186	0.23995	0.5625	0.06629	1.32254	0.52321	0.61719	0.16883
6	1.82859	0.15153	0.46875	0.03827	1.44349	0.3408	0.5625	0.04941	2.03953	0.32544	0.44531	0.05123
7	1.43794	0.18743	0.55469	0.08942	1.57701	0.08808	0.52344	0.0406	1.72079	0.39994	0.47656	0.08942
8	1.18641	0.21247	0.55469	0.05123	1.28994	0.17944	0.50781	0.05579	1.28239	0.18399	0.50781	0.0406
9	1.20657	0.13597	0.59375	0.03827	1.44736	0.31547	0.55469	0.10452	1.43901	0.27263	0.5625	0.08558
10	1.23316	0.36211	0.60938	0.08414	0.9424	0.07848	0.67969	0.05123	1.21809	0.25115	0.60938	0.04688
11	1.29098	0.25422	0.57813	0.08414	1.29651	0.26976	0.52344	0.06766	1.65672	0.09444	0.42969	0.06001
12	1.17716	0.11093	0.53906	0.03405	1.33349	0.25836	0.48438	0.02706	1.35321	0.0807	0.49219	0.06766
13	1.44472	0.29515	0.53906	0.03405	1.68045	0.35912	0.45313	0.10005	1.38336	0.37667	0.54688	0.10482
14	1.5049	0.1904	0.55469	0.04622	1.59969	0.16612	0.49219	0.0406	1.6399	0.51611	0.53906	0.11771
15	1.14808	0.37395	0.61719	0.10216	1.13453	0.14231	0.61719	0.08081	1.26731	0.4492	0.55469	0.16883
16	1.31827	0.11275	0.53125	0.03827	1.61258	0.44348	0.42188	0.14406	1.53322	0.43721	0.46875	0.10126
17	1.68176	0.2085	0.46875	0.04941	1.8125	0.12247	0.4375	0.0221	1.63326	0.27928	0.49219	0.07453
18	1.60186	0.18774	0.46094	0.07453	1.61907	0.40105	0.50781	0.09727	1.6675	0.45396	0.52344	0.09727
19	1.58774	0.47319	0.5	0.12303	0.95876	0.05359	0.64844	0.05123	1.25826	0.36876	0.60156	0.11561
20	1.59692	0.22948	0.52344	0.06766	1.49231	0.24596	0.53125	0.05846	1.28519	0.091	0.58594	0.02591
21	1.28115	0.1998	0.5625	0.05846	1.50355	0.38568	0.51563	0.11159	1.57638	0.09638	0.48438	0.03494

Figure 5.5: Long-Short Term Memory results in subject specific scenario where for each case 21 subject recordings are presented. Each subject experiment was performed on 4-cross validation folds which were averaged at the end to produce average loss, average accuracy and standard deviation of both metrics.

This experiment resulted in performance improvement. In the majority of the subjects across all the cases, LSTM was able to predict slightly better than RNN. *Case1vs2* was the most distinguishable, yet the highest accuracy was achieved by Subject 10 in *Case1vs3*. *Case2vs3* improved more wherein LSTM 3 subjects scored above 60% with the average accuracy. The loss remained high, yet in many cases, there is a visible increase, such as Subject 6 in *Case2vs3*: 2.0395.

5. Sequential Approach

2.1.3 Gated Recurrent Unit

To test an alternative approach for dealing with long term dependencies, Gated Recurrent Unit was implemented. The optimal architecture across all the subjects was designed using 3 GRU layers with 144, 96 and 48 units respectively. Each GRU layer was followed by dropout of 50%. The last two were dense layers which had 10 and 1 neurons respectively, with sigmoid activation function.

Hyperparameters for learning process were chosen similarly to LSTM, where the model used binary cross-entropy loss function with Adam optimization technique which updated weights every 32 observations (32 batch training). The number of epochs was set to 20.

Subjects	Case1vs2				Case1vs3				Case2vs3			
	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc
1	0.995779	0.14931	0.578125	0.034939	1.293651	0.240475	0.515625	0.0625	1.318149	0.367074	0.492188	0.067658
2	0.933145	0.205437	0.507813	0.097265	1.097892	0.209526	0.53125	0.03125	1.085391	0.183189	0.570313	0.046219
3	1.369537	0.539891	0.601563	0.148848	1.422591	0.320097	0.578125	0.046875	1.573206	0.406411	0.507813	0.046219
4	1.315055	0.238645	0.609375	0.078125	1.404083	0.099493	0.484375	0.03125	1.516791	0.496455	0.523438	0.046219
5	1.24214	0.239051	0.617188	0.055792	1.758024	0.398011	0.640625	0.015625	1.74665	0.516013	0.53125	0.069877
6	2.186382	0.417697	0.453125	0.051822	1.783863	0.136244	0.46875	0.125	1.990886	0.892239	0.46875	0.088388
7	1.257389	0.223679	0.53125	0.038273	1.595416	0.44473	0.375	0.125	1.74012	0.303723	0.484375	0.046875
8	1.488464	0.192771	0.5	0.11482	1.408899	0.330711	0.578125	0.015625	1.193617	0.231103	0.445313	0.115614
9	1.471383	0.279859	0.492188	0.040595	1.848878	0.443362	0.515625	0	1.655101	0.35908	0.507813	0.060009
10	1.655445	0.399979	0.53125	0.079672	1.46995	0.46178	0.53125	0.046875	1.730167	0.474075	0.5	0.085582
11	1.311033	0.25865	0.585938	0.063948	1.157358	0.208048	0.390625	0.109375	1.185919	0.486236	0.429688	0.117707
12	1.32196	0.269511	0.578125	0.113752	1.12316	0.081502	0.5	0.03125	1.564093	0.087136	0.398438	0.013532
13	1.703845	0.335704	0.492188	0.067658	1.885546	0.550523	0.53125	0.03125	1.37518	0.168838	0.492188	0.060009
14	1.436361	0.411987	0.585938	0.025911	1.401898	0.386807	0.578125	0.109375	1.512571	0.33648	0.539063	0.046219
15	1.473665	0.363169	0.554688	0.040595	1.247172	0.307042	0.515625	0.015625	1.235502	0.278684	0.546875	0.027063
16	1.349258	0.462685	0.585938	0.060009	1.242963	0.105816	0.4375	0.0625	1.662532	0.126295	0.429688	0.034054
17	1.340486	0.143047	0.585938	0.025911	1.5105	0.211021	0.40625	0.015625	1.435274	0.110539	0.515625	0.068108
18	1.547942	0.354183	0.476563	0.040595	1.728835	0.344583	0.515625	0.015625	1.5571	0.389951	0.554688	0.060009
19	1.245215	0.240114	0.632813	0.025911	1.464257	0.23234	0.53125	0.046875	1.34151	0.246831	0.617188	0.060009
20	1.329804	0.268472	0.59375	0.038273	1.344786	0.296803	0.6875	0.046875	1.184307	0.203928	0.585938	0.060009
21	1.428484	0.124156	0.578125	0.056337	1.790361	0.368102	0.546875	0.015625	1.862559	0.267739	0.367188	0.067658

Figure 5.6: Gated Recurrent Unit results in the subject specific scenario where for each case 21 subject recordings are presented. Each subject experiment was performed on 4-cross validation folds which were averaged at the end to produce average loss, average accuracy and standard deviation of both metrics

The results of GRU subject wise showed an improvement in average accuracy across the subjects, yet in comparison to LSTM subject specific accuracy improved in the majority of subjects only for *Case1vs2*. The highest accuracy was achieved by Subject 20 in *Case1vs3* and the smallest average loss value belonged to Subject 2 in *Case1vs2*. In *Case2vs3* only one subject scored this time above 60%.

5. Sequential Approach

The average value across the subjects was calculated and presented below (Figure 5.7). In all cases, architectures were unstable where generalising the problem and extracting valuable information from the data remained challenging (even though regularisation techniques such as dropout were used). Models behave during training and prediction changed and varied from subject to subject and also within the subjects which indicate huge instability of the models where high loss gives an assumption that models are uncertain of the prediction. While there were a few subjects which performed significantly better than others, the overall results were limited. The complexity of the data and its subject specificity increases the complexity of experiments. The alternative approach for input formulation mentioned in [44] was performed using signal transformation.

	Case1vs2				Case1vs3				Case2vs3			
	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc
RNN	1.07428	0.18184	0.53534	0.04728	1.08427	0.11847	0.52679	0.04691	1.09583	0.11035	0.50707	0.05114
LSTM	1.365	0.20287	0.54836	0.05637	1.39455	0.22151	0.53683	0.06425	1.46203	0.2045	0.51823	0.05702
GRU	1.40013	0.24901	0.5558	0.05013	1.47524	0.2394	0.51711	0.07422	1.49841	0.24098	0.50037	0.06049

Figure 5.7: Figure presents averaged across subjects results from sequential approach using RNN, LST and GRU.

2.2 Hand-Crafted with Wavelets

2.2.1 Random Forest

Proposed input formulation is formed by decomposing signal using Discrete Wavelet Transform (DWT). As mentioned previously in this chapter in Section 1.2, the signal from each channel was decomposed into 6 sub-bands using Daubechies 2 wavelet resulting in 364 wavelet coefficients. For cA6, cD6, cD5, cD4, and cD3 sub-bands 13 features were extracted, resulting in 65-dimensional feature space of the signal. Concatenated and flattened representations of channels within a sequence (a single trial) were used as a single observation with 2080 features. From classical approaches, Random Forest which handles high-dimensional data was implemented using Gini Impurity together with CART algorithm. The architecture was based on 20 estimators. Additional pre-pruning to prevent outliers where minimum leaf sample of value 30 was applied.

5. Sequential Approach

Subjects	Random Forest						Random Forest Feature Importance					
	Case1vs2		Case1vs3		Case2vs3		Case1vs2		Case1vs3		Case2vs3	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
1	0.42969	0.04622	0.5	0.07967	0.46094	0.07118	0.46875	0.05846	0.36719	0.06001	0.4375	0.04941
2	0.55469	0.08378	0.48438	0.05634	0.49219	0.04622	0.46094	0.05579	0.53125	0.03827	0.50781	0.05123
3	0.60156	0.13689	0.65625	0.03827	0.47656	0.04059	0.69531	0.13509	0.66406	0.07773	0.45313	0.01563
4	0.4375	0.09111	0.39844	0.08081	0.57031	0.02591	0.42188	0.04688	0.48438	0.01563	0.46875	0.05846
5	0.58594	0.04622	0.4375	0.0221	0.51563	0.06442	0.57813	0.06442	0.53125	0.05846	0.57813	0.06442
6	0.53125	0.09632	0.38281	0.14384	0.45313	0.05634	0.52344	0.10452	0.44531	0.07773	0.52344	0.04059
7	0.40625	0.07329	0.40625	0.08558	0.36719	0.08081	0.4375	0	0.46094	0.10683	0.42188	0.03494
8	0.57813	0.12204	0.55469	0.03405	0.50781	0.04622	0.5	0.09632	0.53906	0.12956	0.55469	0.09726
9	0.5	0.07329	0.47656	0.09211	0.54688	0.08119	0.52344	0.04622	0.55469	0.01353	0.53906	0.09726
10	0.42969	0.08081	0.51563	0.11158	0.40625	0.06629	0.53125	0.09632	0.57031	0.07453	0.49219	0.08081
11	0.46094	0.04622	0.48438	0.02706	0.51563	0.10005	0.40625	0.08558	0.57031	0.10452	0.47656	0.08664
12	0.42969	0.1091	0.42188	0.04688	0.40625	0.09111	0.46094	0.06001	0.45313	0.02706	0.46094	0.07453
13	0.55469	0.04622	0.40625	0.13441	0.45313	0.05182	0.49219	0.07453	0.46094	0.1717	0.5	0.11049
14	0.53125	0.03827	0.46875	0.03827	0.50781	0.02591	0.51563	0.10005	0.48438	0.05634	0.45313	0.02706
15	0.53906	0.08942	0.50781	0.03405	0.47656	0.06001	0.625	0.1449	0.51563	0.08414	0.48438	0.0716
16	0.52344	0.07453	0.44531	0.05579	0.42188	0.06442	0.53906	0.08081	0.45313	0.04688	0.46875	0.15149
17	0.53125	0.09632	0.45313	0.04688	0.50781	0.08664	0.46094	0.02591	0.45313	0.02706	0.53125	0.138
18	0.42188	0.08976	0.50781	0.04059	0.42969	0.02591	0.42188	0.09504	0.5	0.07967	0.4375	0.03827
19	0.45313	0.09504	0.39063	0.04688	0.53125	0.04941	0.48438	0.14741	0.41406	0.05123	0.5	0.11482
20	0.53125	0.06629	0.39844	0.04622	0.49219	0.04622	0.48438	0.03494	0.50781	0.12377	0.47656	0.08942
21	0.49219	0.04059	0.49219	0.07118	0.45313	0.0716	0.45313	0.09504	0.59375	0.04941	0.47656	0.03405
Across sub	0.50112	0.05902	0.46615	0.06358	0.47582	0.04982	0.49926	0.06793	0.5026	0.0656	0.48772	0.03951

Figure 5.8: Figure presents results table of subject specific Random Forest and Random Forest with implemented Feature Importance across all the cases. The average accuracy and its standard deviation across the subject were calculated for all cases and presented at the bottom of the table.

As a result, comparing to the previous approach with frame-wise prediction in subject specific scenario, slight improvement has been achieved. Subject 3 scored the highest, achieving 65.625% of accuracy. In most of the cases, standard deviation of the accuracy across the folds remains low. To boost the performance of the model and avoid redundant information, feature importance has been applied with a threshold of 0.04. Although a slight increase is visible and the highest accuracy of 69.5% was achieved, the overall models' performance remained unsatisfactory.

After calculating feature importance across the sequence, the feature occurrence was recorded and separated using 32 bins where each bin represented a different channel in the brain. While the feature importance varies across the cases as shown in Figures 5.9, 5.10 and 5.11 some of the channels in their neighbourhood share similar number of activated features which could indicate some sort of connectivity between them. This idea will be explored in the further part of this document.

5. Sequential Approach

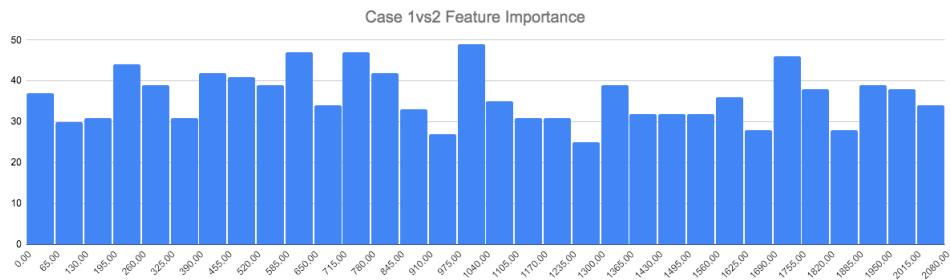


Figure 5.9: Figure presents feature importance within a sequence where each bin corresponds to features extracted from individual channels in Case 1vs2.

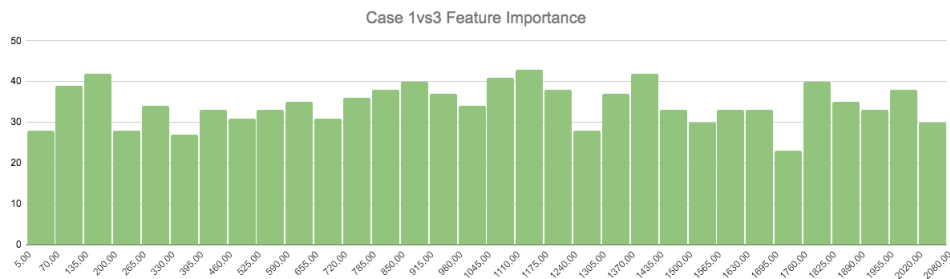


Figure 5.10: Figure presents feature importance within a sequence where each bin corresponds to features extracted from individual channels in Case 1vs3.

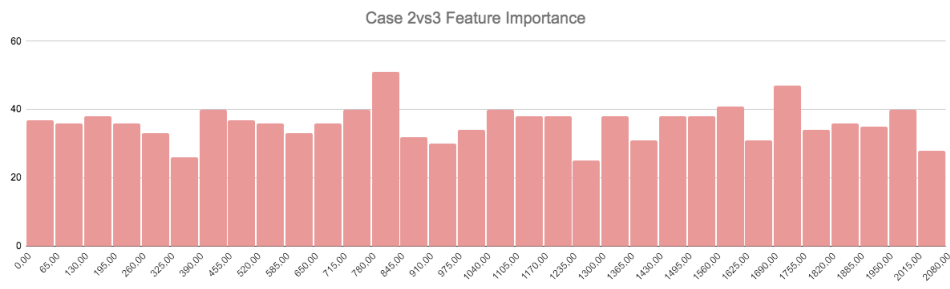


Figure 5.11: Figure presents feature importance within a sequence where each bin corresponds to features extracted from individual channels in Case 2vs3.

5. Sequential Approach

2.2.2 Fully Connected Neural Network

Signal data after transformation and additional modification create high-dimensional space which is then concatenated across all the channels within a trial. The dimensionality of that size is usually really challenging for classical models, hence fully connected neural network was implemented. As mentioned in the introduction to deep learning in Chapter 2, neural networks can be powerful techniques that are capable of learning from complex data. Variations of architectures where different activation functions (such as relu and tanh), number of hidden layers and units were examined.

The proposed architecture consists of 3 hidden layers with sigmoid activation function and 1000, 100 and 1 units respectively. To enable the model to generalise the problem and spread out weights without focusing on specific units dropout of 50% was applied after the first 2 layers. The model was trained throughout 10 epochs using binary cross-entropy loss function and Adam optimizer for weights update.

Subject	Case1vs3				Cas1vs2				Case2vs3			
	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc	Avg. Loss	Std. Loss	Avg. Acc	Std. Acc
1	0.96007	0.09123	0.41406	0.04622	0.8897	0.08974	0.46875	0.09375	0.94588	0.1977	0.5	0.06629
2	0.82938	0.10523	0.59375	0.09375	0.82169	0.10698	0.52344	0.09472	0.82157	0.10985	0.4375	0.05846
3	0.86923	0.07791	0.49219	0.04059	0.85064	0.121	0.54688	0.06442	0.99252	0.1078	0.47656	0.07118
4	1.00276	0.16749	0.5	0.09632	0.8127	0.11911	0.55469	0.03405	0.97424	0.07054	0.48438	0.05182
5	0.89036	0.0657	0.57031	0.08081	1.15775	0.3494	0.39063	0.08414	0.85078	0.11928	0.52344	0.11771
6	1.15352	0.23467	0.50781	0.09974	1.32281	0.30908	0.49219	0.08378	1.21418	0.24537	0.46875	0.05846
7	0.96501	0.13067	0.53906	0.05579	1.03944	0.18294	0.47656	0.07773	1.04751	0.11355	0.54688	0.05634
8	1.03118	0.19723	0.42969	0.04059	0.81711	0.03177	0.50781	0.06766	0.89921	0.13326	0.50781	0.06001
9	1.0751	0.13095	0.5	0.06629	1.01572	0.11177	0.44531	0.06001	1.03153	0.15624	0.51563	0.05182
10	1.0022	0.22199	0.57813	0.04688	0.90297	0.11324	0.59375	0.07967	1.08927	0.1161	0.50781	0.06766
11	0.95998	0.12416	0.49219	0.03405	0.89452	0.15728	0.51563	0.05182	1.01469	0.10126	0.40625	0.0221
12	0.69958	0.10404	0.63281	0.08942	0.84162	0.17599	0.46875	0.07329	0.79752	0.09896	0.50781	0.04622
13	0.87944	0.17719	0.53906	0.08942	1.10671	0.15853	0.46094	0.04622	1.05352	0.05631	0.39844	0.01353
14	0.97541	0.21861	0.51563	0.07813	0.90634	0.16495	0.57031	0.01353	0.83273	0.07706	0.55469	0.04622
15	0.8652	0.10688	0.45313	0.07813	0.82134	0.06745	0.46875	0.09632	0.87987	0.13344	0.46094	0.08942
16	0.87119	0.12983	0.53906	0.08081	1.00287	0.17901	0.48438	0.10482	0.77908	0.0516	0.50781	0.06766
17	0.89569	0.17874	0.5625	0.05846	1.23401	0.12448	0.51563	0.06442	1.14287	0.2936	0.48438	0.0716
18	1.05536	0.18434	0.49219	0.05579	1.02467	0.09265	0.42969	0.07453	0.88718	0.21309	0.48438	0.09244
19	0.88501	0.19903	0.48438	0.08414	0.81145	0.10578	0.53125	0.03827	0.93938	0.13557	0.49219	0.08081
20	1.02753	0.07271	0.46875	0.04941	1.04487	0.33367	0.5	0.06629	1.15063	0.27288	0.48438	0.06811
21	1.11599	0.15439	0.48438	0.08976	1.12258	0.29241	0.46094	0.10216	1.03549	0.20825	0.51563	0.08119
Across Sub.	0.95282	0.10411	0.51377	0.05276	0.9734	0.14744	0.49554	0.04746	0.97046	0.12007	0.48884	0.03839

Figure 5.12: Figure presents feature importance within a sequence where each bin corresponds to features extracted from individual channels in Case2vs3.

Although, neural networks usually perform well in high-dimensional space, in our case the model was trained only on 96 samples (subject specific) which makes generalisation of that problem and classification challenging. The architecture is unstable which was the case

even before applying dropout to the model. Standard deviation of accuracy shows fluctuations and average loss indicates how uncertain the model is about the predictions. Across all the subjects, *Case1vs3* achieved the smallest loss and highest accuracy. Subject 12 in *Case1vs3* scored the highest with accuracy of 63.281%.

Based on the findings, the overall sequential approach for classification of this EGG data performs slightly better. The first methodology is an extension of subject specific scenario in frame-wise approach where prediction is made on the whole sequence (many to one relationship) and the same input formulation. The second though presents different input formulation where signal decomposition using DWT gave another insight into the data. Using feature importance in the random forest the assumption of cross-channel connectivity can be made where finding correlations between the channels could form additional information crucial for the classification.

3 Summary

Sequential approach chapter presents experiments, were two methodologies of input formulation: self learnt and hand-crafted features were presented. To test time-dependencies within the data, models with memory mechanism such as Vanilla RNN, LSTM and GRU were implemented. The data for these experiments were featured in the same manner as in the frame-wise approach.

The second methodology uses alternative input formulation where signal decomposition using Discrete Wavelet Transform was applied. Out of a variety of different family wavelets, db2 has been chosen. The depth of the multilevel decomposition has been calculated based on the length of the data and the mother wavelet. Wavelet decomposition into approximation and detail coefficient result in wavelet decomposition vector of all the coefficients. To reduce the dimensionality of the signal 4 deepest levels (cA6, cD6, cD5, cD4, and cD3) were used for further implementation. Various techniques of feature extraction: frobenius norm; entropy; zero-crossing rate; mean crossing rate; 5th, 25th, 75th and 95th percentile value; mean; median; standard deviation; variance and Root Mean Square value were calculated for each of the sub-bands resulting in 65 features extracted for each channel signal. The data from 32 channels (within the same trial sequence) has been flattened and fed to Random Forest and Fully Connected Neural Network.

5. *Sequential Approach*

While the results of the models in the first methodology show a slight improvement over classical methods, LSTM and GRU are structured efficiently for long-term dependencies which enabled them to score higher than RNN. While GRU achieved the highest score 55.58% (*Case1vs2*), overall LSTM performed slightly better across all the cases. Nevertheless, the models' performance was in general poor and unstable where average loss values showed models uncertainty about the predictions.

An alternative input formulation using Random Forest scored slightly better than Random Forest in frame-wise approach (S.S.). Applied feature importance didn't show any significant improvement. Yet, by recording feature importance across the channels, the theory of cross-channel connectivity was formulated. Further experiments using Fully Connected Neural Network were an attempt for creating an architecture which would be able to better analyse and understand high dimensional feature space of each observation (2080 features). Unfortunately, the models' performance was poor and unstable.

Chapter 6

Conclusions

In Chapter 4 frame-wise approach for classification of the data has been introduced where classical machine learning techniques were examined. While results for subject generic scenario with LOGO cross validation turned out to be tight, subject specific scheme was implemented. Yet, due to the little amount of data, subject specific case performance was lower. The attempt to enhance models performance using dimensionality reduction didn't show significant improvement. Using LDA the data proved to be too complex to be linearly separable. Further data investigation provided interesting insight onto significant cognition period during the process of decision making.

In the following Chapter 5, sequential approach for data classification has been investigated. While deep learning techniques which use memory mechanism to carry out information across the sequence have shown to be slightly better than previous approach, the results remained narrow. Alternative input formulation was introduced by performing signal decomposition using Wavelets Transform. Four deepest levels of produced wavelet coefficients were explored and used for feature extraction. Although, the results were not what has been originally expected, feature importance using Gini Impurity in Random Forest gave interesting observation of possible relationship between the channels.

1 Contributions

Based on our data analysis and further investigation, the data has been proven to be complex and subjective which makes classification arduous. Common for medical domain, small number of data observations increases difficulty level of the problem where generalisation is the key. Yet, five main contributions which have a potential to influence further research have been documented.

- **Problem formulation.**

Extensive data analysis gave us an insight into the data and its complexity. We formulated hypothesis to investigate ccBrain Lab's findings as an attempt to comprehend and understand how humans face decision making deadlocks.

- **Benchmarking.**

The initial benchmark for the classification has been set. Furthermore, reasoning behind unsuccessful architectures and approaches was attempted to answer.

- **Linking results to the initial hypothesis.**

Based on initial findings from ccBrain Lab, the hypothesis of Case1, Case2 and Case3 being distinguishable were made. Although, the results were poor, binary classification of Case1 vs2 was the most discriminative proving these two cases to be more distinguishable than others (which was also observed by ccBrain Lab, Figure 3.3).

- **Significant cognition period.**

As EEG we have been provided with are time-series data, examining time dependencies was essential. For each case there was a certain prediction pattern across all the subjects which indicates singularity of a brain activity. This cognition pattern varied depending on the case showing specificity of time dependencies for each of them.

- **Cross-channel connectivity theory.**

By visualising signal sequences in time domain across the subjects in different channels (see Appendix), certain correlation among them can be assumed. Using presented EEG cap visualisation of channel mapping (Figure 3.4), signals from different channels can be compared and localised. Based on these visualisations we can see channels from frontal lobe (5, 10, 6, 7, 8, 9, 1, 14, 13, 12, 11) sharing similar signal pattern. Same applies

for channels in the middle section of the brain (15, 16, 17, 18, 2, 22, 21, 20, 19) where the pattern is distinguishable. While the signals from these two sections of the brain are somewhat similar, the lower part of parietal and occipital lobe share very contrasting pattern. This cross-channel connectivity can additionally be seen after applying feature importance to the model fed with features from decomposed signal data.

2 Future Work

Contributions we have made to this research can be linked where cross-channel connectivity and significant cognition period could be modeled together. Information transmission within the brain is possible thanks to neurotransmitters passed between synapses of neurons. If we would like to highlight every action potential, firing neurons would create so-called *net* of neurons connected via synapses. The appearance of the *net* would change over the time. By sampling individual *nets* at different time stamps we could capture relationship between the channels at the particular timestep and then throughout the whole sequence.

One of the crucial steps would be defining measurement methodology for calculating similarities between the channels. Cross-correlation is a technique which measures similarity as a function of displacement known as sliding dot product [53]. Another way to define connection between the channels could be by using graph neural networks. The representation of the relationship between the channels at time t could be then extended across the whole sequence creating time-series connectivity structure to perform video classification. Using some sort of channel connectivity measurement (for instance identity matrix or graph neural networks) to represent relationship between the channels a model could create a prediction pattern at time t . This could be then extended across the whole sequence creating time-series connectivity structure to perform video classification.

Bibliography

- [1] R. S. G. Britain, *Machine Learning: The Power and Promise of Computers that Learn by Example: an Introduction*. Royal Society, 2017.
- [2] R. Bruffaerts, “Machine learning in neurology: what neurologists can learn from machines and vice versa,” *Journal of Neurology*, vol. 265, no. 11, pp. 2745–2748, Nov 2018. [Online]. Available: <https://doi.org/10.1007/s00415-018-8990-9>
- [3] R. Hastie and R. M. Dawes, *Rational choice in an uncertain world: The psychology of judgment and decision making*. Sage, 2010.
- [4] W. Zajkowski, D. Krzemiński, J. Barone, L. Evans, and J. Zhang, “Reward certainty and preference bias selectively shape voluntary decisions,” *bioRxiv*, 2019. [Online]. Available: <https://www.biorxiv.org/content/early/2019/11/07/832311>
- [5] M. Yadava, P. Kumar, R. Saini, P. P. Roy, and D. Prosad Dogra, “Analysis of eeg signals and its application to neuromarketing,” *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 19 087–19 111, Sep 2017. [Online]. Available: <https://doi.org/10.1007/s11042-017-4580-6>
- [6] M. M. Moore, “Real-world applications for brain-computer interface technology,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 162–165, June 2003.
- [7] C. Mhl, B. Allison, A. Nijholt, and G. Chanel, “A survey of affective brain computer interfaces: principles, state-of-the-art, and challenges,” *Brain-Computer Interfaces*, vol. 1, no. 2, pp. 66–84, 2014. [Online]. Available: <https://doi.org/10.1080/2326263X.2014.912881>

- [8] A. Rustichini, “Chapter4 - neuroeconomics:: Formal models of decision making and cognitive neuroscience,” in *Neuroeconomics*, P. W. Glimcher, C. F. Camerer, E. Fehr, and R. A. Poldrack, Eds. London: Academic Press, 2009, pp. 33 – 46. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B978012374176900004X>
- [9] D. R. Wheeler, “Brand loyalties: qualitative, quantitative, or both?” *Journal of the Academy of Marketing Science*, vol. 2, no. 4, pp. 651–658, 1974. [Online]. Available: <https://doi.org/10.1007/BF02729459>
- [10] J. Zhang and J. B. Rowe, “The neural signature of information regularity in temporally extended event sequences,” *NeuroImage*, vol. 107, pp. 266 – 276, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1053811914010131>
- [11] C. Bishop, *Pattern recognition and machine learning (Information science and statistics)*. Springer, 2006.
- [12] J. Han, M. Kamber, and J. Pei. (2012) Data mining concepts and techniques, third edition. [Online]. Available: http://www.amazon.de/Data-Mining-Concepts-Techniques-Management/dp/0123814790/ref=tmm_hrd_title_0?ie=UTF8&qid=1366039033&sr=1-1
- [13] Z. R. Yang, *Machine learning approaches to bioinformatics*. World scientific, 2010, vol. 4.
- [14] L. Breiman, *Classification and Regression Trees*. New York: Routledge, 1984.
- [15] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip *et al.*, “Top 10 algorithms in data mining,” *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.
- [16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer New York Inc., 2001.
- [17] R. Weber, H.-J. Schek, and S. Blott, “A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces,” in *VLDB*, vol. 98, 1998, pp. 194–205.
- [18] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, 1987.

- [19] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, “Singular value decomposition and principal component analysis,” in *A practical approach to microarray data analysis*. Springer, 2003.
- [20] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, “Fisher discriminant analysis with kernels,” in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*. Ieee, 1999.
- [21] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [22] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity. 1943.” *Bulletin of mathematical biology*, vol. 52 1-2, pp. 99–115; discussion 73–97, 1990.
- [23] A. M. Turing, “Computers & thought,” E. A. Feigenbaum and J. Feldman, Eds. Cambridge, MA, USA: MIT Press, 1995, ch. Computing Machinery and Intelligence, pp. 11–35. [Online]. Available: <http://dl.acm.org/citation.cfm?id=216408.216410>
- [24] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. The MIT Press, 2012.
- [25] J. Patterson and A. Gibson, *Deep Learning: A Practitioner’s Approach*. Beijing: O’Reilly, 2017. [Online]. Available: <https://www.safaribooksonline.com/library/view/deep-learning/9781491924570/>
- [26] J. J. Hopfield, “Artificial neural networks,” *IEEE Circuits and Devices Magazine*, vol. 4, no. 5, pp. 3–10, Sept 1988.
- [27] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, 1958.
- [28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–, Oct. 1986. [Online]. Available: <http://dx.doi.org/10.1038/323533a0>
- [29] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 5 2015.

- [30] S. Sathasivam and W. A. T. W. Abdullah, “Logic learning in hopfield networks,” Tech. Rep. arXiv:0804.4075, Apr 2008, comments: To appear in Mod. Appl. Sci. [Online]. Available: <http://cds.cern.ch/record/1101626>
- [31] J. J. Hopfield, “Neurocomputing: Foundations of research,” J. A. Anderson and E. Rosenfeld, Eds. Cambridge, MA, USA: MIT Press, 1988, ch. Neural Networks and Physical Systems with Emergent Collective Computational Abilities, pp. 457–464. [Online]. Available: <http://dl.acm.org/citation.cfm?id=65669.104422>
- [32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: <http://dx.doi.org/10.1162/neco.1997.9.8.1735>
- [33] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: <http://arxiv.org/abs/1412.3555>
- [34] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of recurrent network architectures,” in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2342–2350.
- [35] P. Malmivuo, J. Malmivuo, and R. Plonsey, *Bioelectromagnetism: principles and applications of bioelectric and biomagnetic fields*. Oxford University Press, USA, 1995.
- [36] D. Millet, “The origins of eeg,” in *7th Annual Meeting of the International Society for the History of the Neurosciences (ISHN)*, 2002.
- [37] J. C. Henry, “Electroencephalography: Basic principles, clinical applications, and related fields, fifth edition,” *Neurology*, 2006. [Online]. Available: <https://n.neurology.org/content/67/11/2092.2>
- [38] . Niedermeyer, Ernst, . Lopes da Silva, F. H., and I. Ovid Technologies, *Electroencephalography: basic principles, clinical applications, and related fields*, 5th ed. Lippincott Williams & Wilkins, 2005.
- [39] M. Teplan *et al.*, “Fundamentals of eeg measurement,” *Measurement science review*, 2002.

- [40] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Snchez, “A survey on deep learning in medical image analysis,” 2017. [Online]. Available: <http://arxiv.org/abs/1702.05747>
- [41] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. C. Courville, Y. Bengio, C. Pal, P. Jodoin, and H. Larochelle, “Brain tumor segmentation with deep neural networks,” *CoRR*, vol. abs/1505.03540, 2015. [Online]. Available: <http://arxiv.org/abs/1505.03540>
- [42] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen, “Deep convolutional neural networks for multi-modality isointense infant brain image segmentation,” *NeuroImage*, vol. 108, pp. 214–224, 2015. [Online]. Available: <https://doi.org/10.1016/j.neuroimage.2014.12.061>
- [43] X. Zhang, L. Yao, D. Zhang, X. Wang, Q. Z. Sheng, and T. Gu, “Multi-person brain activity recognition via comprehensive EEG signal analysis,” *CoRR*, vol. abs/1709.09077, 2017. [Online]. Available: <http://arxiv.org/abs/1709.09077>
- [44] A. Craik, Y. He, and J. L. Contreras-Vidal, “Deep learning for electroencephalogram (eeg) classification tasks: a review,” *Journal of neural engineering*, vol. 16, no. 3, p. 031001, 2019.
- [45] R. N. Bracewell and R. N. Bracewell, *The Fourier transform and its applications*. McGraw-Hill New York, 1986, vol. 31999.
- [46] H. Taneja, *Advanced Engineering Mathematics*. IK International Pvt Ltd, 2008, vol. 2.
- [47] I. Daubechies, “The wavelet transform, time-frequency localization and signal analysis,” *IEEE transactions on information theory*, vol. 36, no. 5, pp. 961–1005, 1990.
- [48] S. Patro and K. K. Sahu, “Normalization: A preprocessing stage,” *arXiv preprint arXiv:1503.06462*, 2015.
- [49] I. R. Khan and R. Ohba, “New finite difference formulas for numerical differentiation,” *Journal of Computational and Applied Mathematics*, vol. 126, no. 1-2, pp. 269–276, 2000.
- [50] E. S. Finn, X. Shen, D. Scheinost, M. D. Rosenberg, J. Huang, M. M. Chun, X. Papademetris, and R. T. Constable, “Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity,” *Nature neuroscience*, 2015.

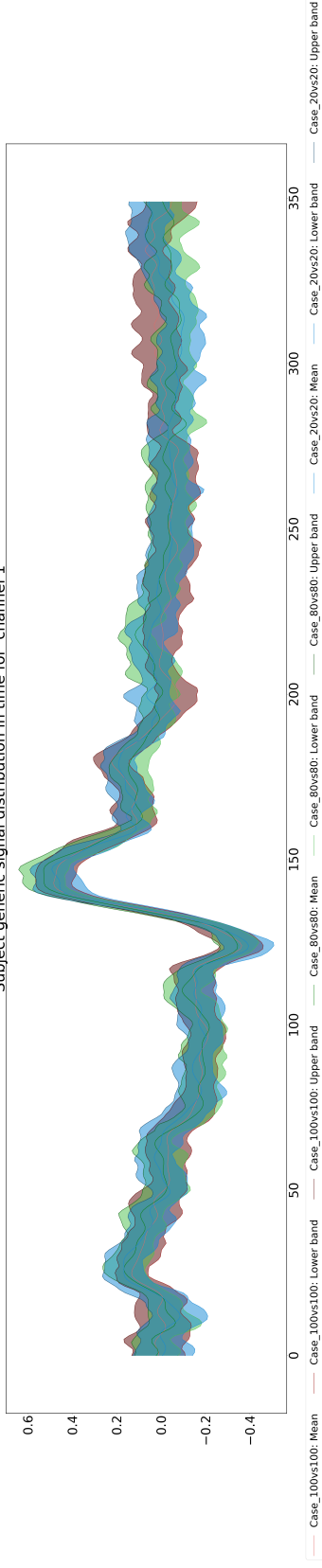
Bibliography

- [51] R. Dubey, S. Samantaray, A. Tripathy, B. C. Babu, and M. Ehtesham, “Wavelet based energy function for symmetrical fault detection during power swing,” in *2012 Students Conference on Engineering and Systems*. IEEE, 2012, pp. 1–6.
- [52] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [53] J. Y. Stein, “Function evaluation algorithms,” *Digital Signal Processing: A Computer Science Perspective*, pp. 605–618, 2000.

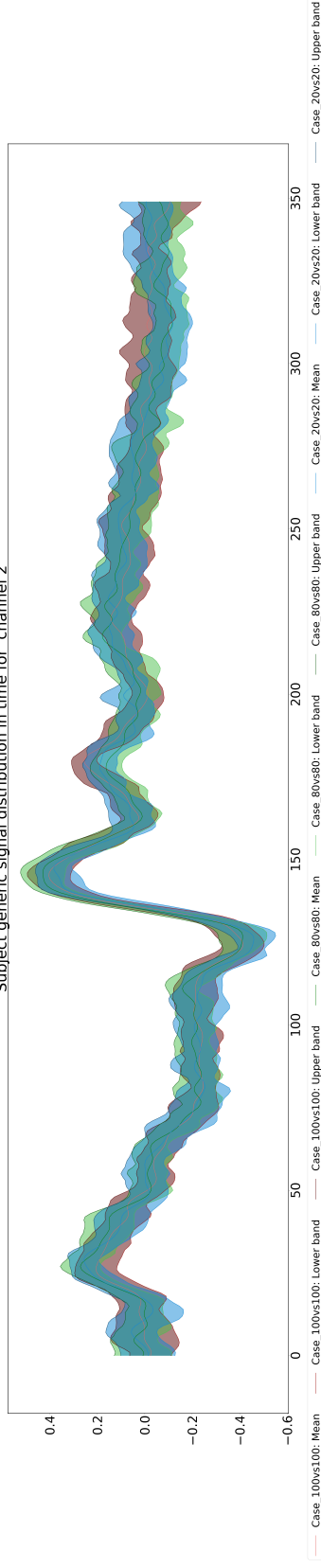
Appendix

Figures attached in Appendix present 32 channels where the average and standard deviation across all the subjects (and their trials) were calculated for every time frame. Each visualisation presents generic signal distribution across three different cases for a particular channel.

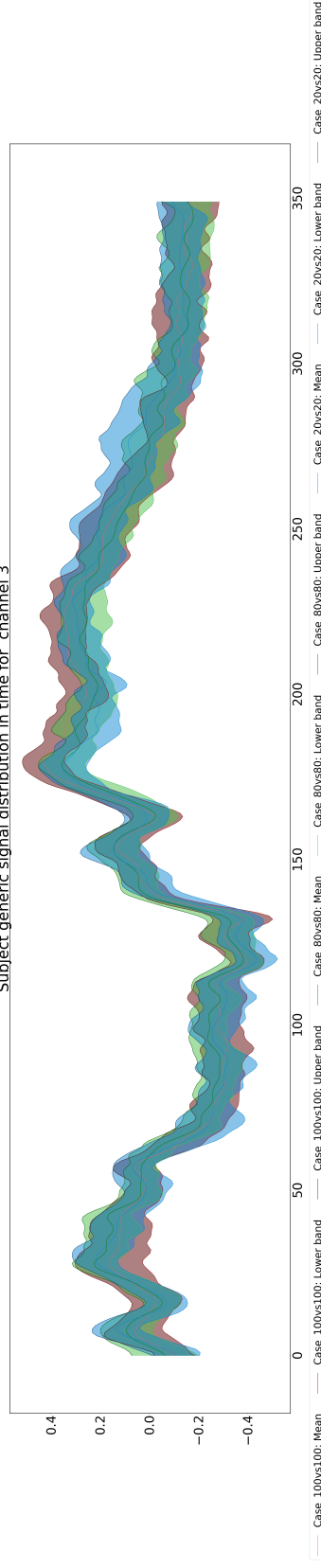
Subject generic signal distribution in time for channel 1



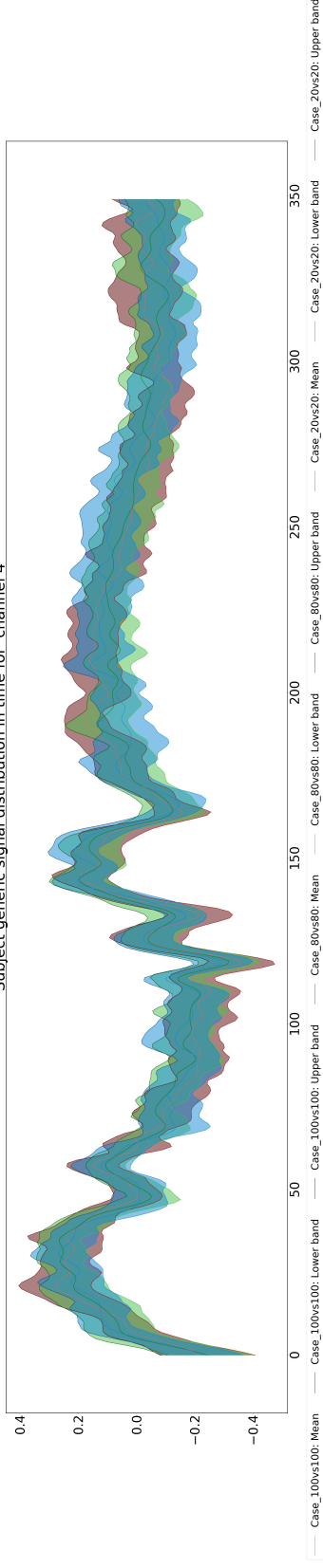
Subject generic signal distribution in time for channel 2



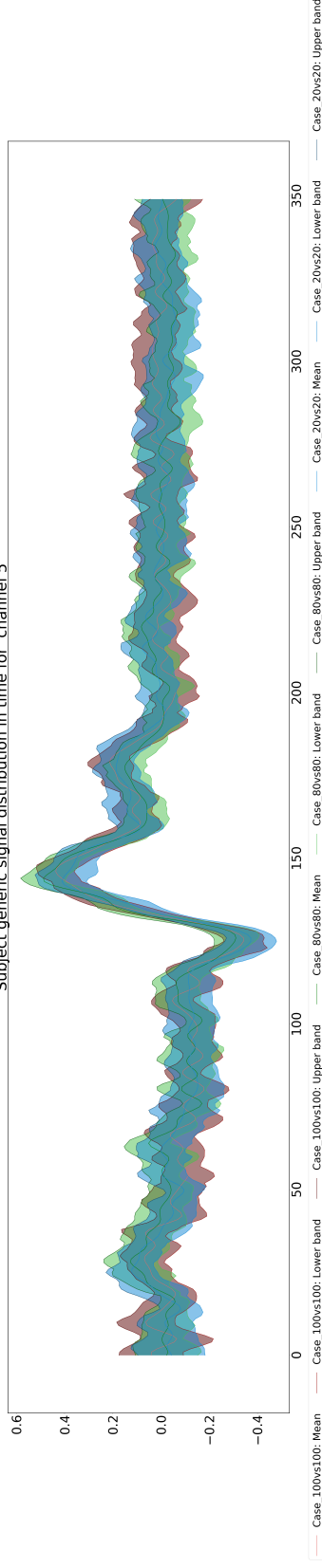
Subject generic signal distribution in time for channel 3



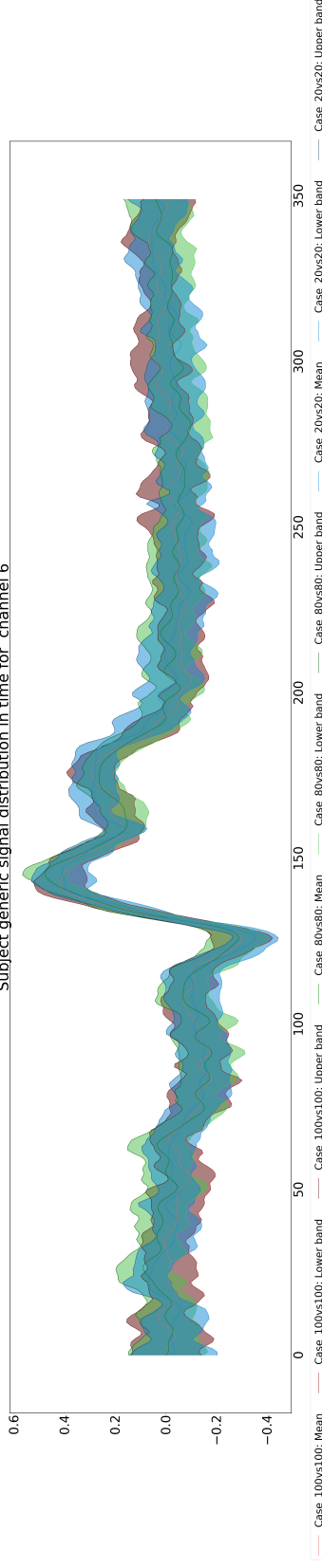
Subject generic signal distribution in time for channel 4



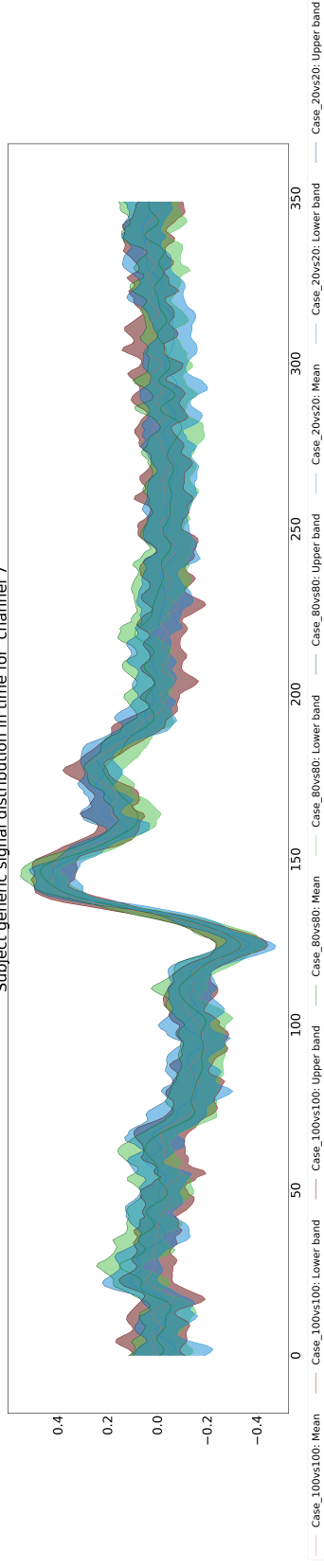
Subject generic signal distribution in time for channel 5



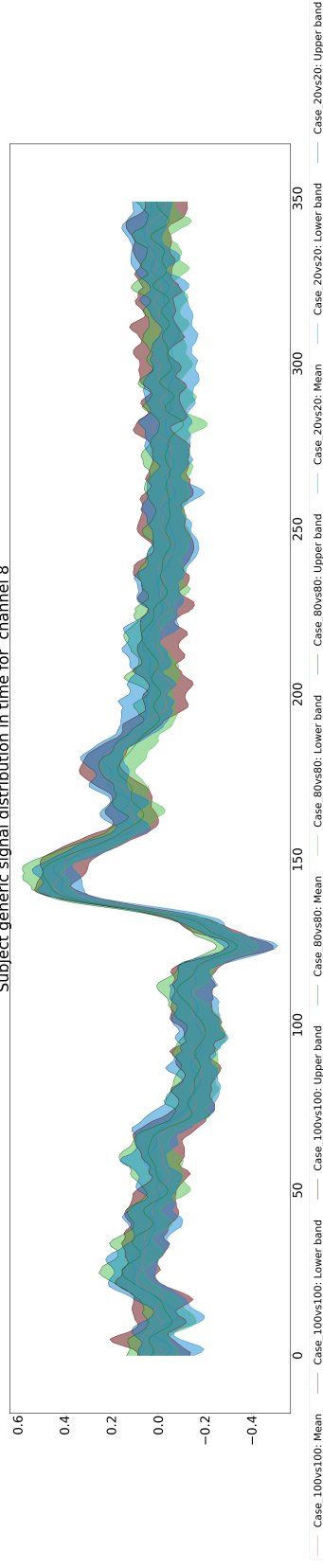
Subject generic signal distribution in time for channel 6



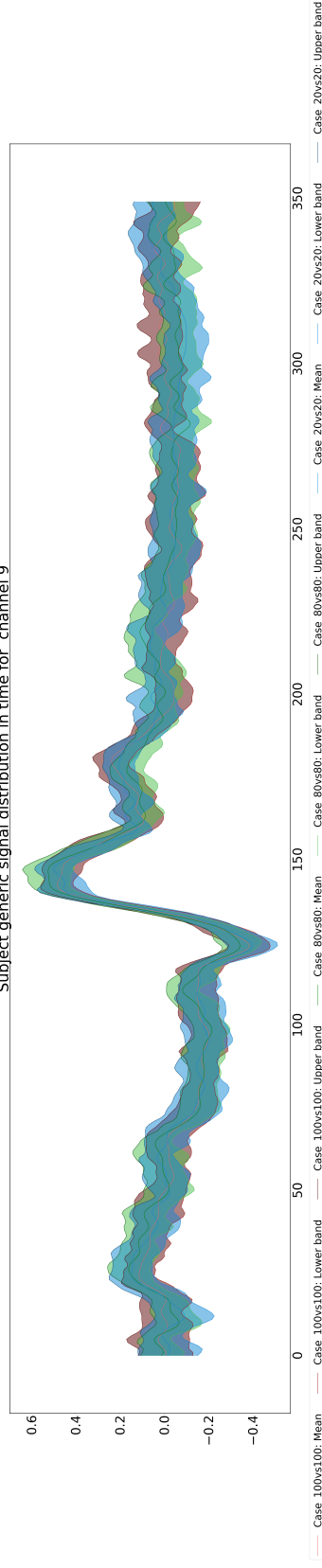
Subject generic signal distribution in time for channel 7



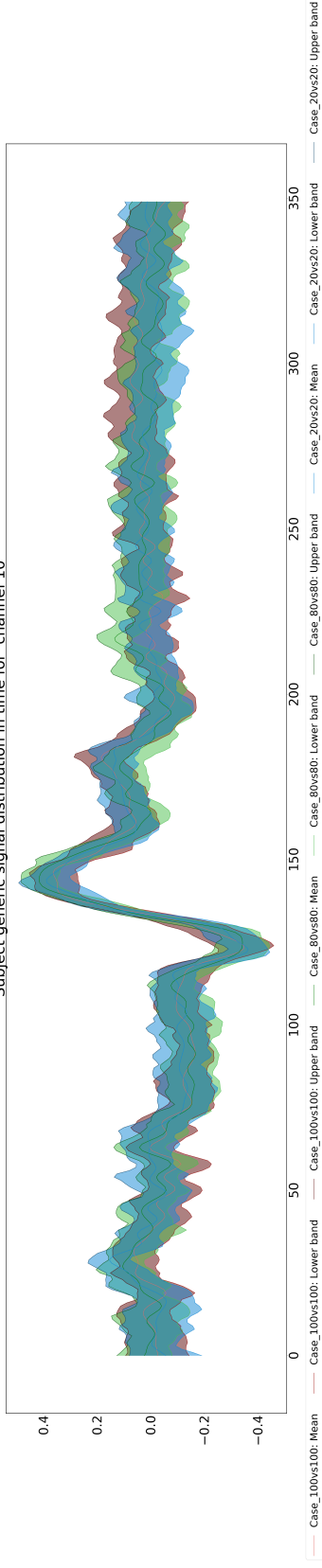
Subject generic signal distribution in time for channel 8



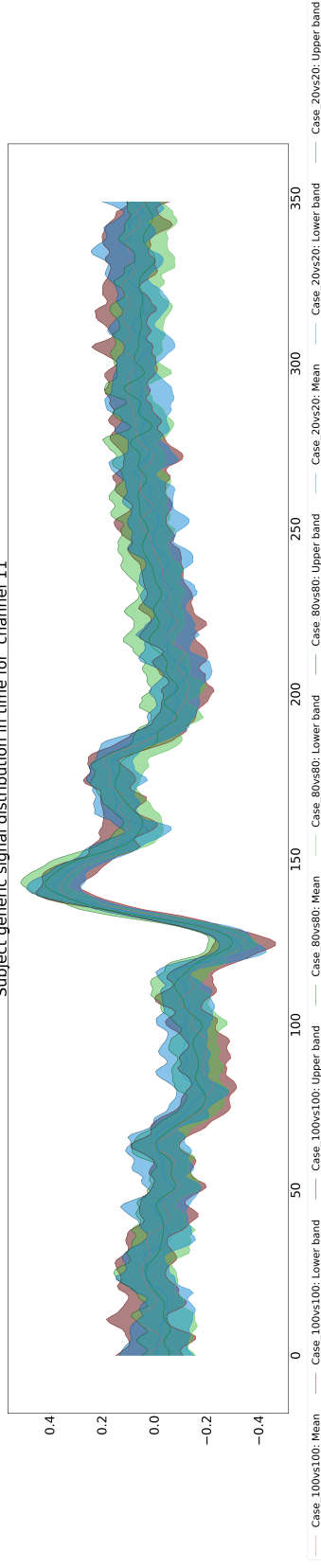
Subject generic signal distribution in time for channel 9



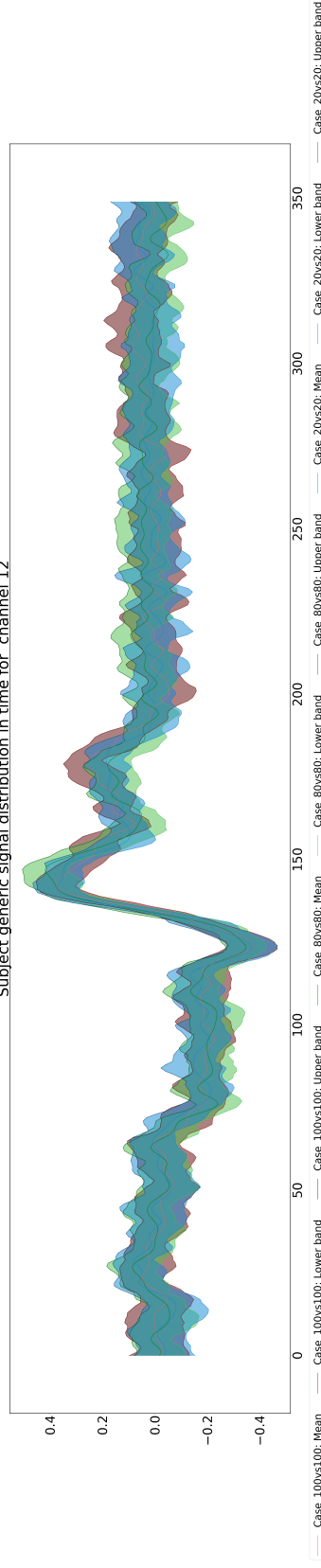
Subject generic signal distribution in time for channel 10



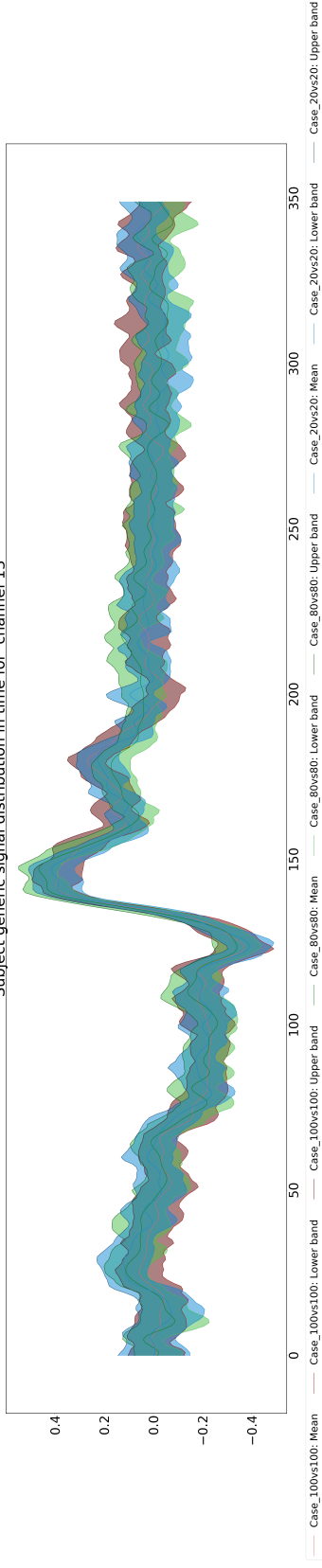
Subject generic signal distribution in time for channel 11



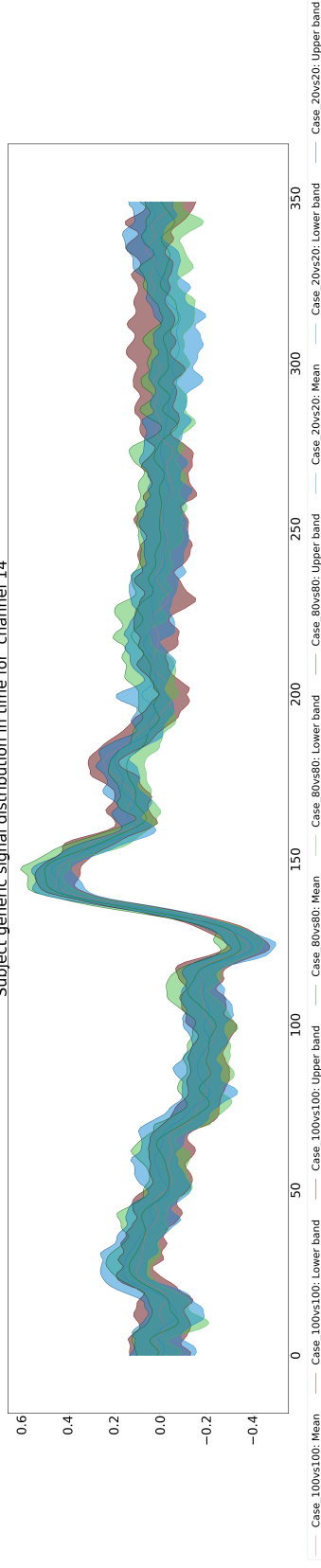
Subject generic signal distribution in time for channel 12



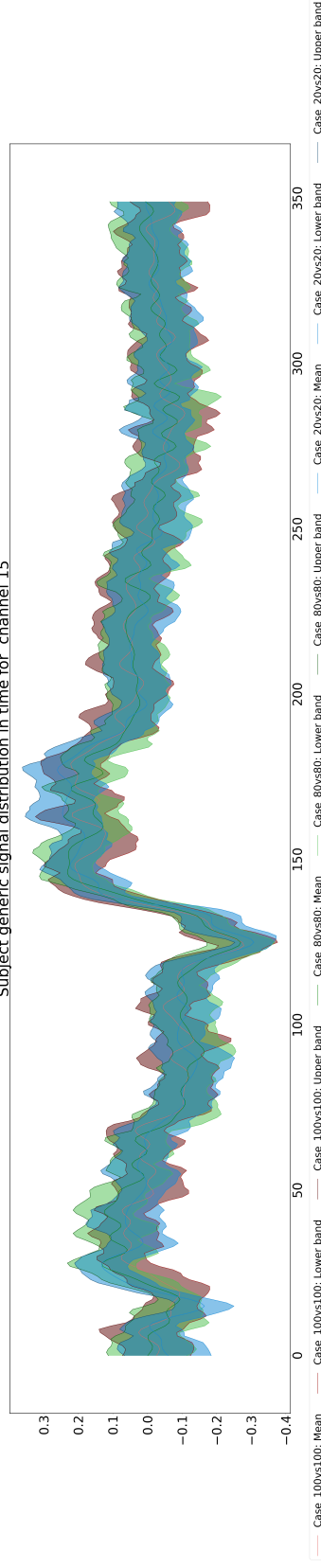
Subject generic signal distribution in time for channel 13



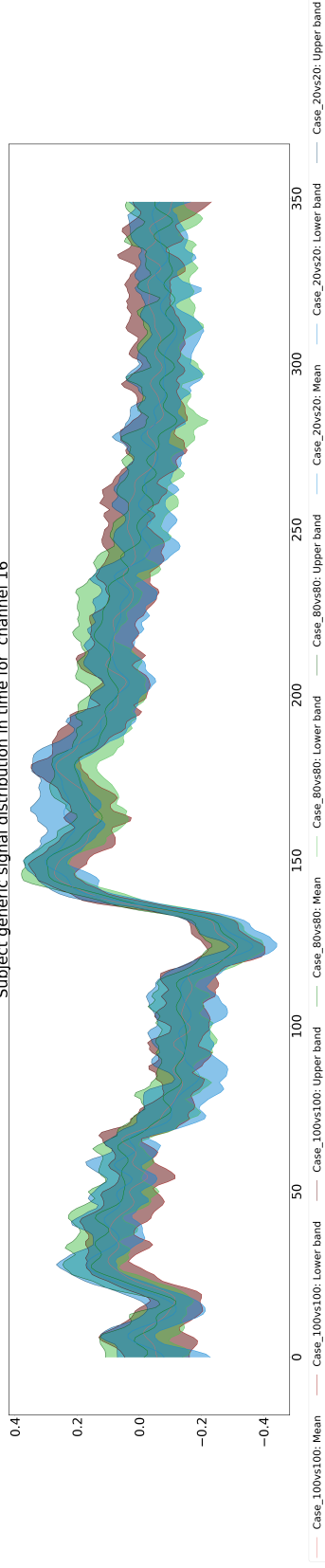
Subject generic signal distribution in time for channel 14



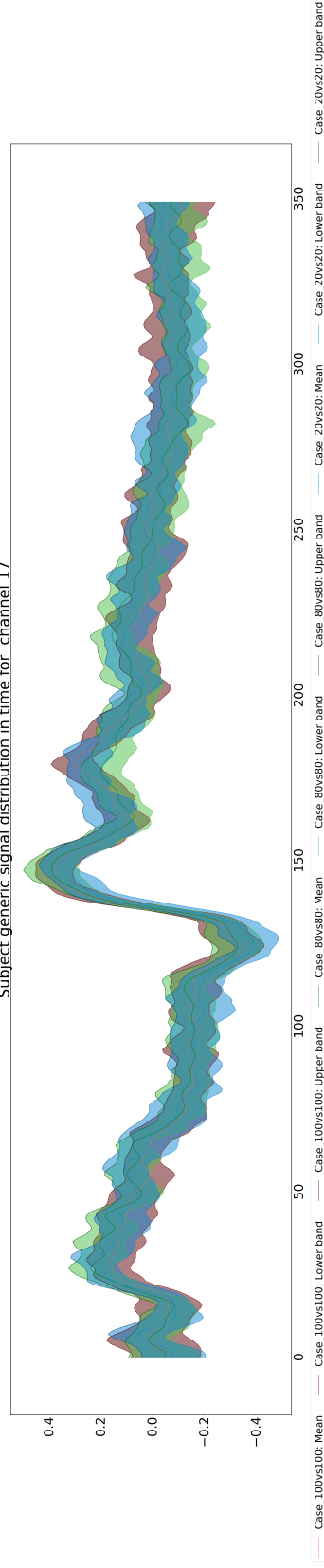
Subject generic signal distribution in time for channel 15



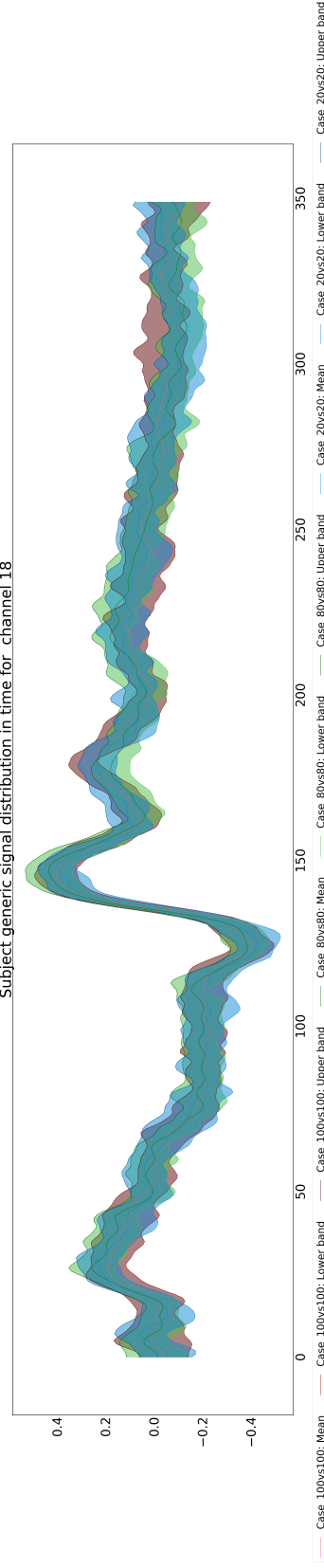
Subject generic signal distribution in time for channel 16



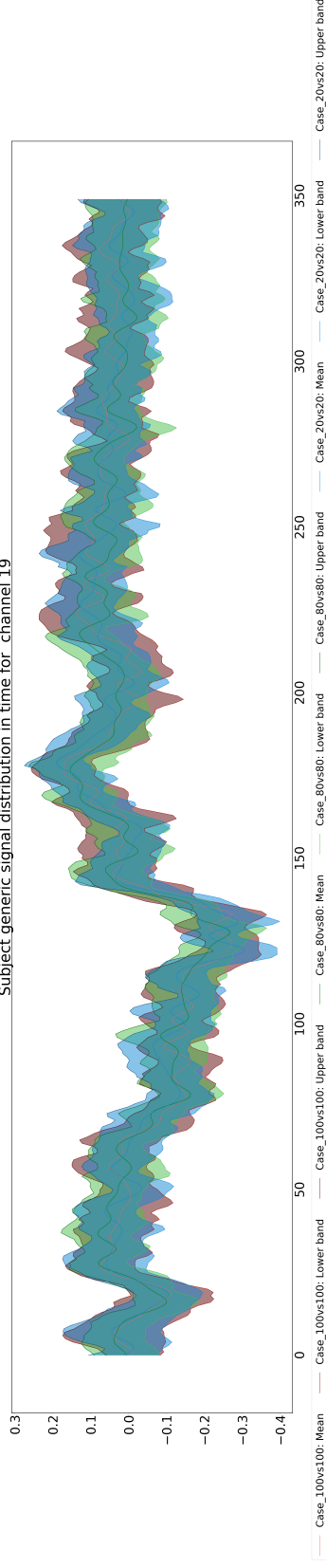
Subject generic signal distribution in time for channel 17



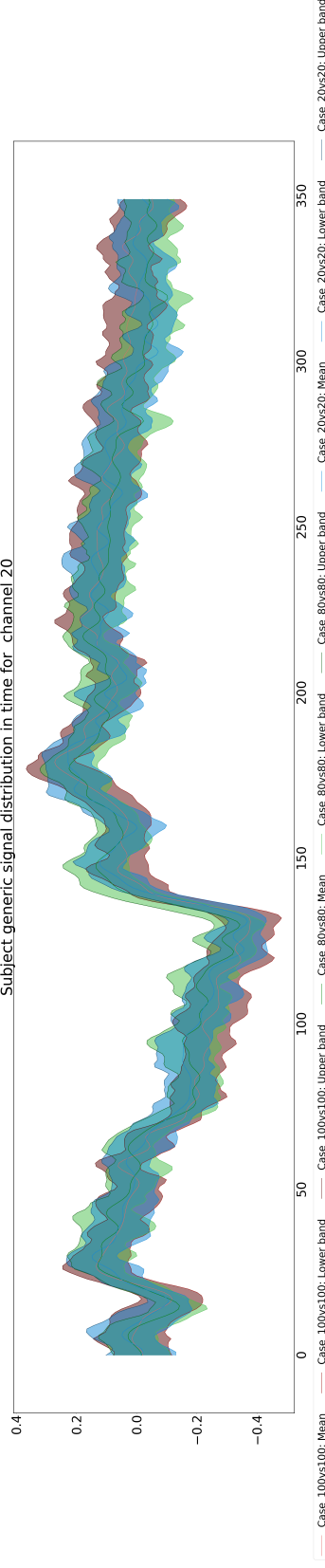
Subject generic signal distribution in time for channel 18



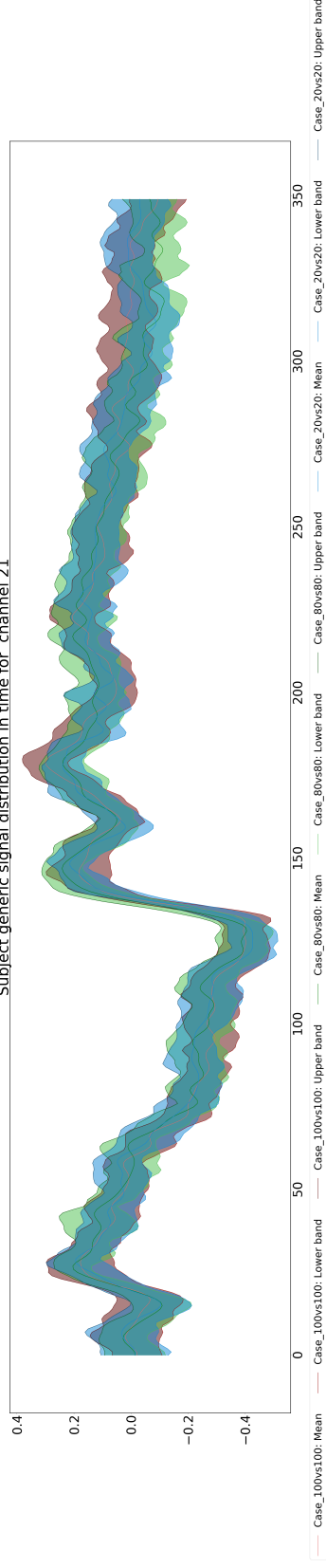
Subject generic signal distribution in time for channel 19



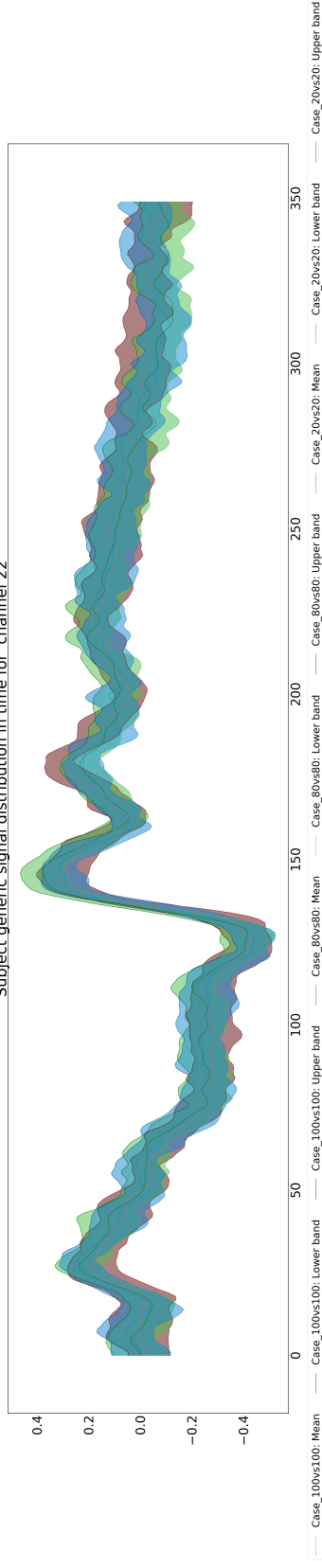
Subject generic signal distribution in time for channel 20



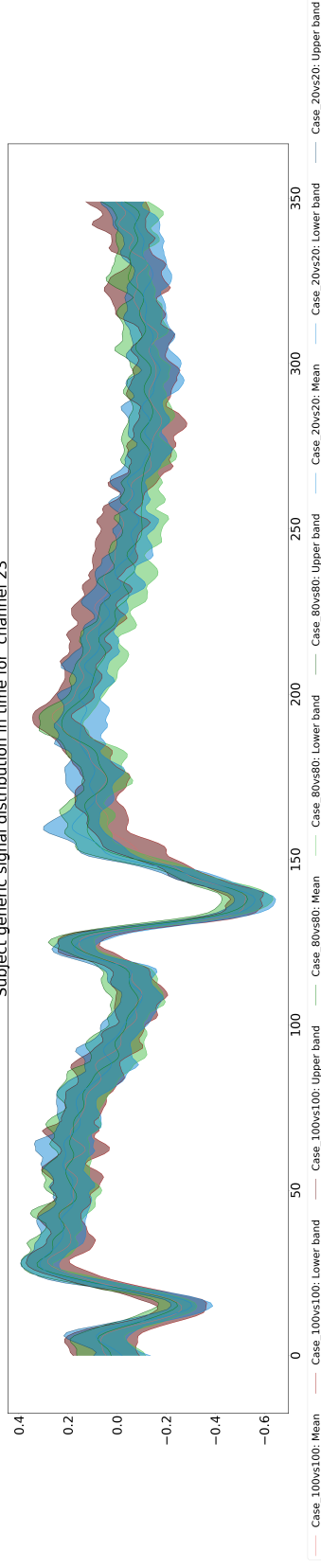
Subject generic signal distribution in time for channel 21



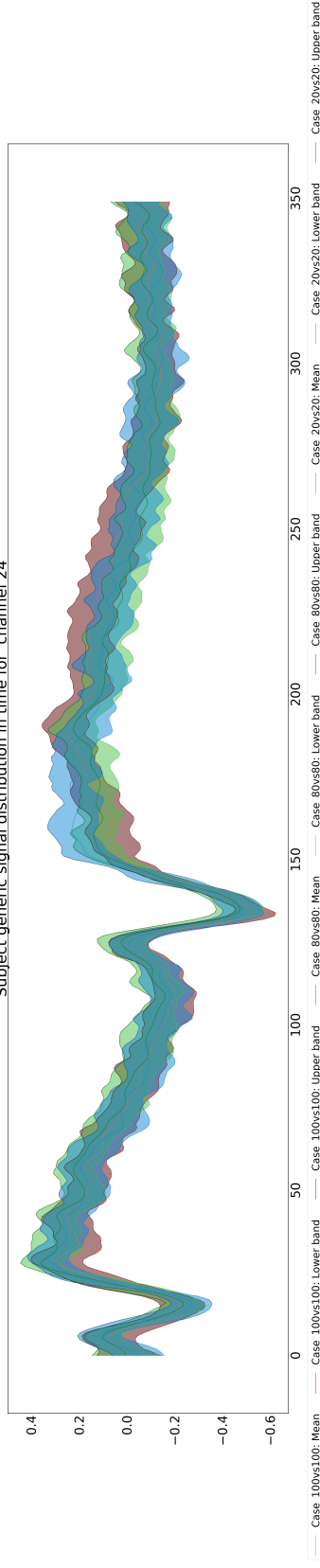
Subject generic signal distribution in time for channel 22



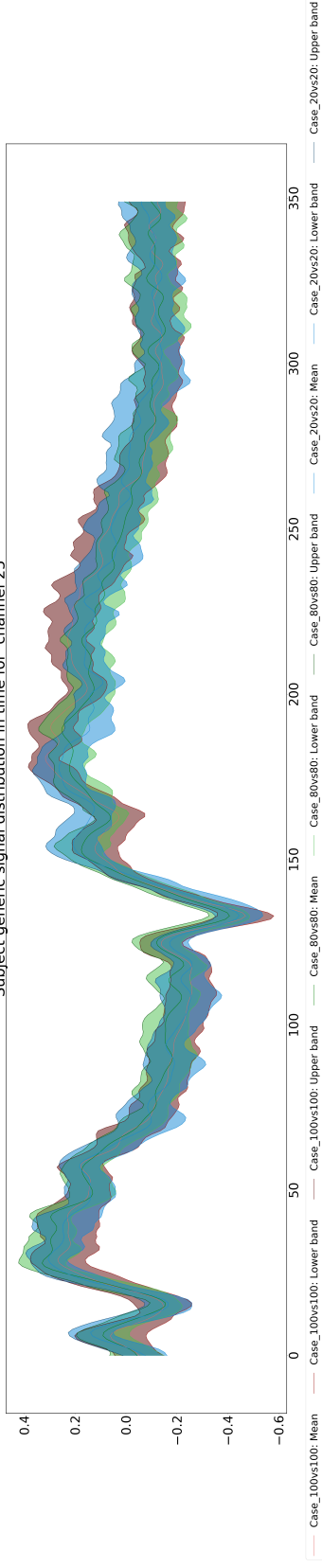
Subject generic signal distribution in time for channel 23



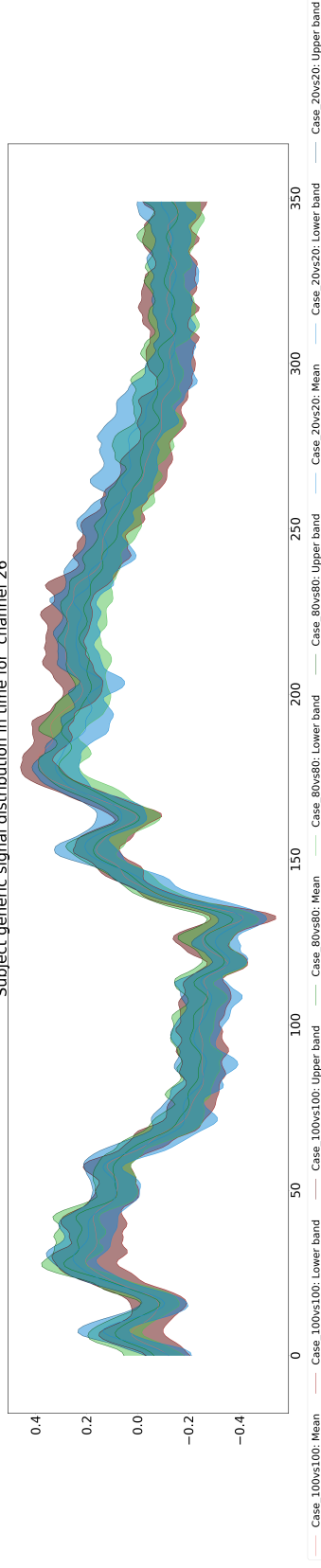
Subject generic signal distribution in time for channel 24



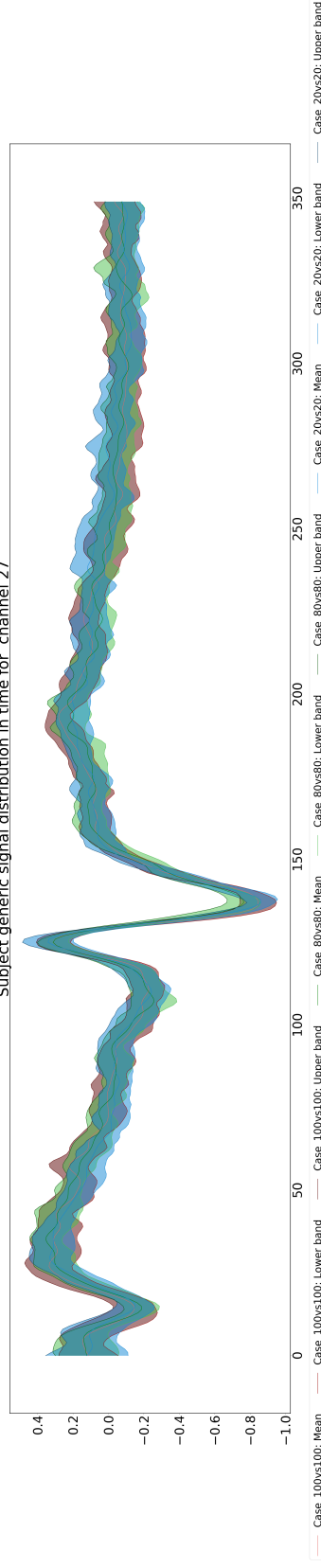
Subject generic signal distribution in time for channel 25



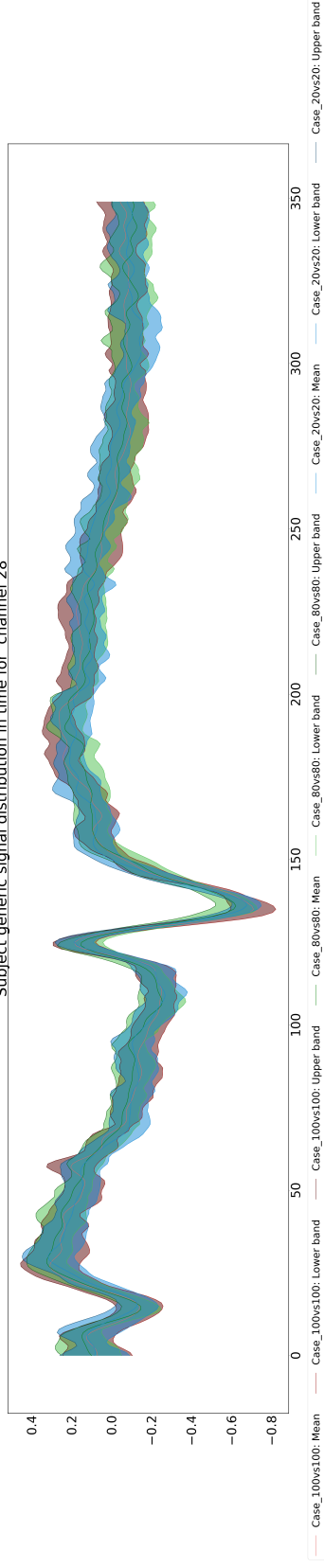
Subject generic signal distribution in time for channel 26



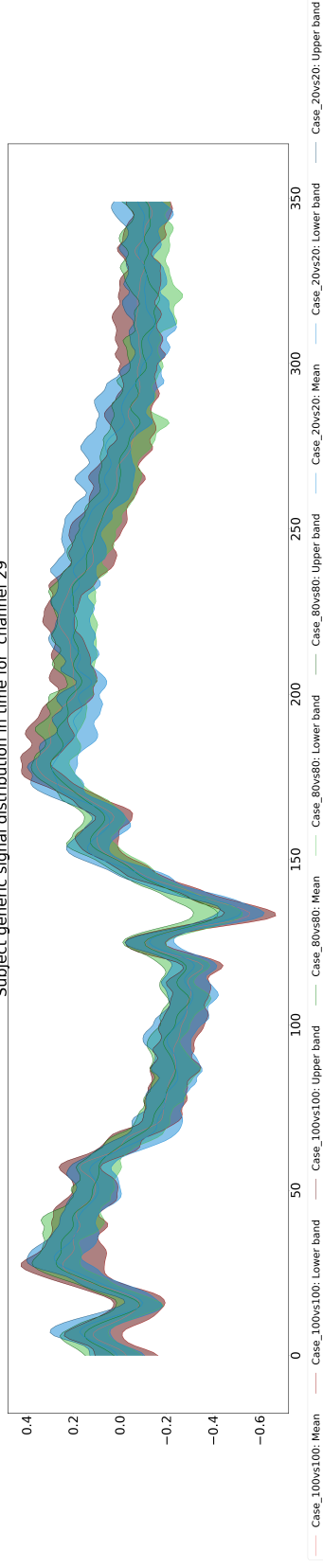
Subject generic signal distribution in time for channel 27



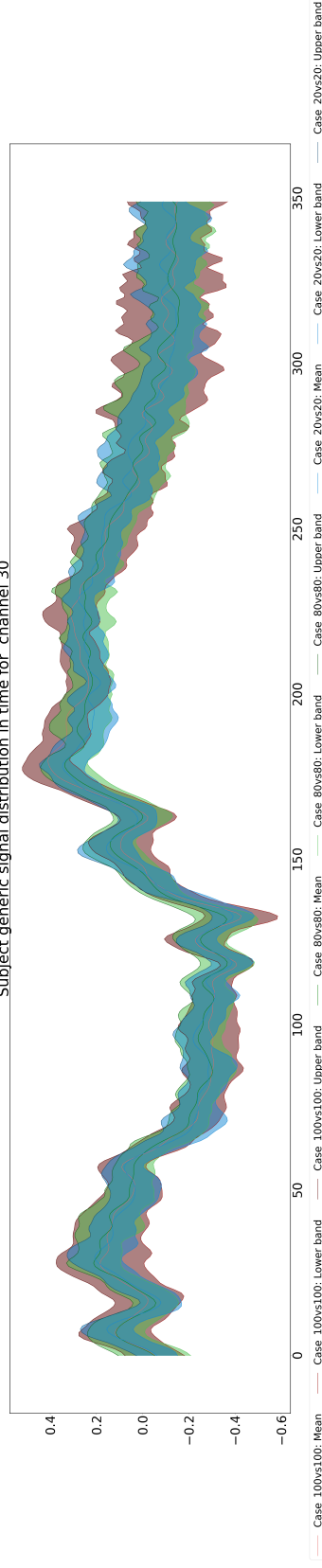
Subject generic signal distribution in time for channel 28



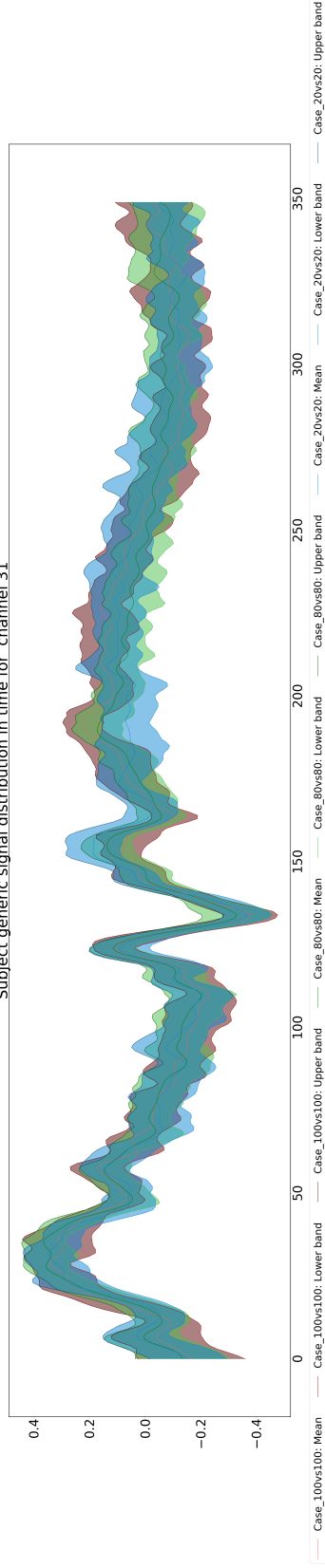
Subject generic signal distribution in time for channel 29



Subject generic signal distribution in time for channel 30



Subject generic signal distribution in time for channel 31



Subject generic signal distribution in time for channel 32

