Faculty of Technology, Design & Environment

Assessment cover

OXFORD BROOKES UNIVERSITY

Module No:	**COMP6013**	Module title:	**Bsc Computing Project**
Degree Programme :	**Bsc (Hons) Computer Science**		
Project title :	**Real-time preictal detection through the application of machine learning to Electroencephalogram signals.**		
Supervisor :	**Kashinath Basu**		
Due date and time:	**19 Apr 2024 - 13:00**		
Estimated total time to be spent on assignment:	**90 hours per student**		

Student No:	Student Name:
19066041	**William Riddell**

Statement of Compliance *(please tick to sign)*

I declare that the work submitted is my own and that the work I submit is fully in accordance with the University regulations regarding assessments *(www.brookes.ac.uk/uniregulations/current)*

Regulations governing the deposit and use of Oxford Brookes University Projects and Dissertations

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Library.

Statement of Permission *(please tick to sign)*

Thank you for agreeing to have your work (i.e. dissertation and associated deliverables) deposited in the Repository's Undergraduate eDissertation collection.

By depositing this work, you agree to the following Terms and Conditions:

You confirm that:

a. you are the copyright owner and/or have the right to grant us licence to hold your work in our institutional repository (currently RADAR).

We agree to:

b. add the work to the COMP6013 Collection so that it is available to Oxford Brookes University members for the lifetime of the institutional repository.
c. convert the work if necessary for long term preservation.

We reserve the right to remove third party copyright material that we may identify in the work.
We reserve the right to remove the work for any legal or administrative reason.

Use of AI Tools

You are required to use this form to declare which AI tools you have used and how you have used them. Please complete the form and attach it to your submission as an Appendix, if you have used such tools.

School of Engineering, Computing & Mathematics

LEARNING OUTCOMES

On successful completion of this module, students will be able to achieve the module following learning outcomes (LOs):	
1	Create, design, manage, plan, carry out, and evaluate a project involving the solution of a practical problem set in an appropriate social and economic context, taking into account other relevant factors such as risk
2	Apply practical and analytical skills acquired in the programme to the investigation of a substantial topic
3	Apply the scientific method and report findings using accepted formalisms
4	Identify and utilise trustworthy information sources, such as the ACM Digital Library to develop a coherent understanding of issues in the domain
5	Demonstrate the ability to carry out a substantial piece of work independently and critically evaluate the student’s achievements and their own personal development
6	Use appropriate technologies such as online libraries and databases to find, critically evaluate and utilise both non-specialist and technical information pertinent to the project
7	Demonstrate an awareness of and work in a manner guided by the legal, professional, ethical, security and social issues relevant to the IT and telecommunications industry

Engineering Council AHEP4 LOs assessed (from S1 2022-23):	
B3	Select and apply appropriate computational and analytical techniques to model broadly-defined problems, recognising the limitations of the techniques employed
B4	Select and evaluate technical literature and other sources of information to address broadly-defined problems
B5	Design solutions for broadly-defined problems that meet a combination of societal, user, business and customer needs as appropriate. This will involve consideration of applicable health & safety, diversity, inclusion, cultural, societal, environmental and commercial matters, codes of practice and industry standards
B6	Apply an integrated or systems approach to the solution of broadly-defined problems
B7	Evaluate the environmental and societal impact of solutions to broadly-defined problems
B8	Identify and analyse ethical concerns and make reasoned ethical choices informed by professional codes of conduct
B9	Use a risk management process to identify, evaluate and mitigate risks (the effects of uncertainty) associated with a particular project or activity
B10	Adopt a holistic and proportionate approach to the mitigation of security risks
B13	Select and apply appropriate materials, equipment, engineering technologies and processes
B15	Apply knowledge of engineering management principles, commercial context, project management and relevant legal matters
B17	Communicate effectively with technical and non-technical audiences

FORMATIVE FEEDBACK OPPORTUNITIES

Your supervisor will give you the following formative feedback:
• Weekly, during project supervision meetings • Written feedback on Proposal (See Appendix A) • Written feedback on Progress Report (See Appendix B) • Feedback on presentation draft

SUMMATIVE FEEDBACK DELIVERABLES

Deliverable description and instructions	**Weighting out of 100%**
Presentation (see Appendix C) comprising: a) presentation of software, with video URL b) project slides c) summary poster (i.e. the final project slide)	**10%**
Final Report (see Appendix D) comprising: a) written dissertation b) software artefact URL link to source code	**90%**

ASSIGNMENT IN DETAIL

See Handbook Appendices A – D for assignment details and marking grid.

Real-time preictal detection through the application of machine learning to Electroencephalogram signals.

William Riddell

April 17, 2024

Word Count: 10,000

Supervised by Kashinath Basu

Faculty of Technology, Design and Environment
School of Engineering, Computing and Mathematics

Contents

1 Abstract **10**

2 Acknowledgements **10**

3 Introduction **10**
3.1 Background 11
3.2 Aim and Objectives 11
3.3 Project Requirements 12

4 Background Review **12**
4.1 Datasets 12
4.2 Machine Learning (ML) Models 17

5 Project Methodology and Development **22**
5.1 Methodology 22
5.1.1 Approach 22
5.1.2 Development System 23
5.1.3 Version Control 23
5.1.4 Software and Libraries Used 24
5.1.5 Schedule 24
5.2 Development 25
5.2.1 Electroencephalogram (EEG) Data Extraction 25
5.2.2 Short-Time Fourier Transform (STFT) 27
5.2.3 Notch Filter 28
5.2.4 Synthesize Data 30
5.2.5 Convolutional Neural Network (CNN) and tuning methodology 32
5.2.6 Real Time Simulation 38
5.3 Problems Encountered 39

6 Review **41**
6.1 Objectives and Requirements 41
6.2 Results 42
6.3 Future Advancements 42

7 Professional Issues and Risks **42**
7.1 Professional Issues . . . 42
7.2 Risks . . . 44
7.2.1 Risk Matrix . . . 44
7.2.2 Risk Assessment . . . 46

8 Conclusion **47**

9 Final Thoughts **47**

10 Bibliography **48**

A Results **51**
A.1 Tables . . . 51
A.2 Plots . . . 60

B Misc. **60**

C Accessed Material **63**

D Presentation, Poster, and Ethics Form **63**

List of Figures

1 Third Order Butterworth bandpass filter . . . 18
2 Fourier transform equation . . . 18
3 Cross-Correlation . . . 19
4 Maximal Lyapunov (λ) exponent . . . 20
5 Phase-locking Synchrony (SPLV) extraction from the wavelet transformation . . . 20
6 Gantt Chart . . . 25
7 Extracted raw EEG data. (Science 2024) . . . 25
8 Diagram of the 10-20 International System. (*21 electrodes of International 10-20 system for EEG.svg* 2010) . . . 26
9 Head of extracted EEG data in CSV format. The shown data is approx 1/5 second and each column represents an single EEG node's data. . . . 27
10 Short-Time Fourier Transform (STFT) Window Extraction . . . 27

11 An STFT window with high frequency resolution. 29
12 Sliding Window technique for synthesizing data spectrogram images 32
13 Calculate the desired window size. "desired_size" is the length of the windows in seconds. "sampling_rate" is the sampling rate of the recorded data. 32
14 Calculate the "step" or "speed" of the sliding window. Can also be described as the offset between window starting points. "target" is the desired number of windows. 32
15 Produces an set of pairs, where an pair describes the start and ending index of an single window. The set length is the same as "target" 32
16 Categorical Cross-Entropy 33
17 Optimization function performance comparison. 34
18 Screenshot of running real-time simulation. 39
19 An STFT window with high time resolution. 63

List of Tables

1 The Datasets analysed 13
2 Channel Characteristics 14
3 Temporal properties 15
4 Subject properties 16
5 Suggested applications 17
6 Results from (Savadkoohi, Oladunni & Thompson 2020) showing results for the Time Domain, Frequency Domain, and Time-Frequency Domain 19
7 Cross-Correlation results. 20
8 Non-Linear interdependence results. 21
9 Dynamical Entrainment; difference of short-term Lyapunov exponents (DSTL) results. 21
10 Phase-locking Synchrony (SPLV) results. 21
11 Entropy of Phase Difference results. 21
12 Distribution of Wavelet Coherence results. 21
13 Cross-Correlation results. 22
14 List of Libraries used 24
15 List of Short-Time Fourier Transform (STFT) parameters . . 28

16 List of Short-Time Fourier Transform (STFT) parameters choices 28
17 Recording information for each subject. Valid represents if an subject has enough data, and is therefore suitable for training. 31
18 The possible values for each tuning variable. 36
19 Objectives taken from 3.2. 41
20 Requirements taken from 3.3. 41
21 Risk Matrix 44
22 Risk Assessment 46
23 Time duration metrics for each model in format %H:%M:%S.%f. 51
24 Accuracy and Loss metrics (average across epochs) during model training. 52
25 Accuracy metrics obtained during testing. 53
26 Generated Confusion Matrix during the model tuning process. 55
27 Generated Normalized Confusion Matrix during the model tuning process. 57
28 F1 metrics obtained during testing. 58
29 Precision metrics obtained during testing. 59
30 Recall metrics obtained during testing. 60

Acronyms

AI Artificial Intelligence. 10

ANN Artificial Neural Network. 40

CHB-MIT Children's Hospital Boston. 22, 24–26, 30, 39, 42, 60

CNN Convolutional Neural Network. 5, 10, 17, 19–22, 27, 32–35

CSV Comma-separated values. 6, 26, 27

DSTL Dynamical Entrainment; difference of short-term Lyapunov exponents. 7, 18, 20, 21

EDF European Data Format. 24, 25

EEG Electroencephalogram. 5, 6, 10–12, 22, 24–28, 38, 39, 41, 60

EEGs Electroencephalograms. 11, 12, 27

kNN K Nearest Neighbor. 17–19

LOOCV Leaving One Out Cross-Validation. 34

LR Logistical Regression. 17, 19–22

ML Machine Learning. 5, 10–12, 15, 17, 19, 22–24, 28, 30, 41, 43

OOM Out of Memory. 36, 40

OOP Object-oriented Programming. 26

SPLV Phase-locking Synchrony. 6, 7, 18, 20, 21

STFT Short-Time Fourier Transform. 5–8, 10, 18, 22–24, 27–29, 34, 39

SVM Support Vector Machine. 17–21

1 Abstract

"Epilepsy is one of the most common serious brain conditions, affecting over 70 million people worldwide." (Thijs, Surges, O'Brien & Sander 2019) Antiepileptic Drugs have been developed and are currently in widespread use, however "one third of people with epilepsy will not have adequate seizure control with the current medications [available Antiepileptic Drugs (AEDs)]. For these patients the situation has improved very little in the last few decades" (Galanopoulou, Buckmaster, Staley, Moshé, Perucca, Engel Jr, Löscher, Noebels, Pitkänen, Stables et al. 2012). Furthering this, 35% of epilepsy patients become resistant to AED medication (Moghim & Corne 2014). This paints a bleak situation for people suffering with epilepsy; "There is an urgent need for more effective and better-tolerated treatments" (Löscher & Schmidt 2011).

This project describes a non-intrusive solution which classifies the current state of an patient in real time. An preprocessing and classification pipeline has been developed which classifies Electroencephalogram (EEG) signals into one of three categories (interictal, preictal, or ictal) through the use of an Convolutional Neural Network (CNN) and Short-Time Fourier Transform (STFT). This project has also tuned an selection of models, with 15 different models boasting 100% accuracy against an validation dataset.

2 Acknowledgements

I would like to thank my project supervisor, Kashinath Basu for his support throughout the project. Our weekly meetings gave invaluable advice and direction which helped me stay on track and encouraged me to develop a project which I am incredibly proud of.

3 Introduction

Over the last 20 years, Artificial Intelligence (AI) has seen a large evolution through the use of Machine Learning (ML); the statistical analysis of data which leads to the unveiling of characteristics and connections. (Awad & Khanna 2015). There has been a large uptake of applying ML techniques to biomedical data, increasing the speed and accuracy of prediction, detection, diagnosis, and prognosis.

Electroencephalograms (EEGs) measure the electrical signals in the brain. EEGs have a great use in giving an insight into the inner workings of the brain, for example allowing us to pick up abnormalities preceding and during their occurrence. "A seizure is a burst of uncontrolled electrical activity between brain cells (also called neurons or nerve cells) that causes temporary abnormalities in muscle tone or movements (stiffness, twitching or limpness), behaviours, sensations or states of awareness." (Medicine n.d.) Due to this, monitoring the brain's electrical activity through the use of an EEG, and applying analysis through an ML model may allow us to detect the preictal period. "An automated accurate prediction of seizures will significantly improve the quality of life of patients and reduce the burden on caregivers" (Acharya, Hagiwara & Adeli 2018)

3.1 Background

"Because of their unpredictable nature, uncontrolled seizures represent a major personal handicap and source of worry for patients. In addition, persistent seizures constitute a considerable burden on healthcare resources." (Assi, Nguyen, Rihana & Sawan 2017) Due to this both medication and surgery are available to applicable patients, although with 30% patients being refractory to drug therapy, and an equally bleak surgery success rate; 75% in lesional cases, and 50% in nonlesional cases for temporal lobe cases along with 60% in lesional cases and merely 35% in nonlesional for frontal lobse cases (Assi et al. 2017), a large population of patients would therefore greatly benefit from a prediction system in their daily life.

3.2 Aim and Objectives

This project will aim to develop an consistent ML model trained to classify either preictal, interictal and ictal periods. The model will have to achieve a high degree of accuracy ($\geq 90\%$) when being applied to EEG data in real-time. Furthermore a real-time simulation will need to be developed along with an ML pipeline and model parameter tuning.

Objectives

1. Research and find an suitable dataset allowing for the classification of the interictal, preictal, and ictal periods.

2. Research and find an suitable ML model approach.

3. Create a data preperation and preprocessing pipeline which extracts the dataset into an ML format suitable for training and testing.

4. Undergo parameter tuning and model architecture tuning of the selected ML model approach.

5. Produce a simulation interface which streams EEG data in real time through the preprocessing pipeline and into the optimal model. The simulation should show the current classification to the user every second.

3.3 Project Requirements

The formal requirements of this project are stated in

1. Develop and preprocessing pipeline which accepts raw EEG data.

2. Tune an number of models to find the optimal variable configuration.

3. Ensure that the optimal model has an accuracy of $\geq 90\%$ against an validation dataset.

4. From EEG data extraction to prediction, the process must take at most 1 second.

5. Create an real-time simulation mimicking an real life application.

4 Background Review

4.1 Datasets

(Wong, Simmons, Rivera-Villicana, Barnett, Sivathamboo, Perucca, Ge, Kwan, Kuhlmann, Vasa et al. 2023) reviews 10 datasets available to download. It evaluates the way the EEGs were physically setup on the subject, the subjects themselves and the data's properties. Wong et al. also states their opinion on what tasks suit what dataset, with the main two tasks being either detection or prediction.

Dataset
University of Bonn
CHB-MIT Scalp EEG
Melbourne-NeuroVista seizure trial (Neurovista Ictal)
Kaggle UPenn and Mayo Clinic's Seizure Detection Challenge
Neurology and Sleep Centre Hauz Khas
Kaggle American Epilepsy Society Seizure Prediction Challenge
Kaggle Melbourne-University AES-MathWorks-NIH Seizure Prediction Challenge
TUH EEG Seizure Corpus (TUSZ)
Siena Scalp EEG
Helsinki University Hospital EEG

Table 1: The Datasets analysed

Within these datasets Wong et al. was also able to find the way the EEG nodes were positioned on the subject's cranium, along with whether the EEG nodes were either placed intracranial or extracranial. Wong et al. also the number of channels that are contained in the raw EEG data for each dataset.

Dataset	**Number of channels**	**Placement method**	**Type of signal**
University of Bonn	1	International 10–20 system, Intracranial	Scalp/Intracranial EEG
CHB-MIT Scalp EEG	18	International 10–20 system/Nomenclature	Scalp EEG
Melbourne-NeuroVista seizure trial (NeuroVista Ictal)	16	Intracranial	Intracranial EEG
Kaggle UPenn and Mayo Clinic's Seizure Detection Challenge	16–76	Intracranial	Intracranial EEG
Kaggle American Epilepsy Society Seizure Prediction Challenge	16	Intracranial	Intracranial EEG
Neurology and Sleep Centre Hauz Khas	1	International 10–20 System	Scalp EEG
Kaggle Melbourne-University AES-MathWorks-NIH Seizure Prediction Challenge Data	16	Intracranial	Intracranial EEG
TUH EEG Seizure Corpus (TUSZ)	23–31	International 10–20 system / Nomenclature	Scalp EEG
Helsinki University Hospital EEG	19	International 10–20 system	Scalp EEG
Siena Scalp EEG	20/29	International 10–20 system/Nomenclature	Scalp EEG

Table 2: Channel Characteristics

Wong et al. also noted along with this data that the "University of Bonn dataset contains a mixture of both scalp and intracranial EEG data where scalp EEG from healthy subjects was taken, while intracranial EEG was taken from subjects with a history of seizures." (Wong et al. 2023). This may present a skew on the ML model during training.

Dataset	**Noncontinuous data**	**Short-term continuous data**	**Continuous data**
University of Bonn	Yes	No	No
CHB-MIT Scalp EEG	No	Yes	Yes
Melbourne-NeuroVista seizure trial (Neurovista Ictal)	N/A	N/A	N/A
Kaggle UPenn and Mayo Clinic's Seizure Detection Challenge	Yes	No	No
Kaggle American Epilepsy Society Seizure Prediction Challenge	Yes	No	No
Neurology and Sleep Centre Hauz Khas	Yes	No	No
Kaggle Melbourne-University AES-MathWorks-NIH Seizure Prediction Challenge Data	Yes	No	No
TUH EEG Seizure Corpus (TUSZ)	No	Yes	No
Helsinki University Hospital EEG	No	Yes	No
Siena Scalp EEG	No	Yes	No

Table 3: Temporal properties

Wong et al. ordered the datasets into groups, either continuous or non continuous data. For the continuous data they separated out datasets which record for less that 24 hours in a single go, these were classified as "Short-term continuous" data.

Dataset	Number of subjects	Subject type
University of Bonn	10	Human
CHB-MIT Scalp EEG	23	Human
Melbourne-NeuroVista seizure trial (NeuroVista Ictal)	12	Human
Kaggle UPenn and Mayo Clinic's Seizure Detection Challenge	12	Human & Canine
Kaggle American Epilepsy Society Seizure Prediction Challenge	7	Human & Canine
Neurology and Sleep Centre Hauz Khas	10	Human
Kaggle Melbourne-University AES-MathWorks-NIH Seizure Prediction Challenge Data	3	Human
TUH EEG Seizure Corpus (TUSZ)	642	Human
Helsinki University Hospital EEG	79	Human
Siena Scalp EEG	14	Human

Table 4: Subject properties

Wong et al. also was able to identify the number of subjects within each dataset. Within the two "Kaggle" datasets there are Canine subjects, making them unsuitable for this project.

Within the review, they also produced tables displaying the segment information for each dataset, breaking down the recording length and frequency, along with the number of events and segments. This information should not weight into which dataset suits the idea of preictal prediction so shall be left out in this background review. Wong et al. also discussed the idea of the class imbalance problem, where the number and length of each ictal period is unbalanced. Two datasets, "University of Bonn" and the "Neurology and Sleep Centre Hauz Khas" have addressed this issue and have balanced their data between ictal, preictal, interictal and nonictal periods.

Taking the research into account Wong et al. suggested which dataset suits either prediction or detection. "Since the aim of seizure prediction is to forecast impending seizures, EEG recordings that include preictal and interictal data should be used for the study, while the aim of seizure detection is to detect ongoing seizure events, hence, EEG recordings that contain ictal and interictal data should be used." (Wong et al. 2023).

Dataset	**Application**
University of Bonn	Seizure detection
CHB-MIT Scalp EEG	Seizure detection/Prediction
Melbourne-NeuroVista seizure trial (NeuroVista Ictal)	Seizure detection/Prediction
Kaggle UPenn and Mayo Clinic's Seizure Detection Challenge	Seizure detection
Kaggle American Epilepsy Society Seizure Prediction Challenge	Seizure prediction
Neurology and Sleep Centre Hauz Khas	Seizure detection/Prediction
Kaggle Melbourne-University AES-MathWorks-NIH Seizure Prediction Challenge Data	Seizure prediction
TUH EEG Seizure Corpus (TUSZ)	Seizure detection/Prediction
Helsinki University Hospital EEG	Seizure detection/Prediction
Siena Scalp EEG	Seizure detection/Predictio

Table 5: Suggested applications

4.2 Machine Learning (ML) Models

A series of papers have been reviewed with the following ML model types and feature extraction processes being used:

- Machine Learning (ML) models
  - K Nearest Neighbor (kNN).
  - Support Vector Machine (SVM).
  - Logistical Regression (LR).
  - Convolutional Neural Network (CNN).

- Preprocessing Pipelines
  - Time domain, using the third order Butterworth bandpass filter.
  - Frequency domain using the Fourier transform.
  - Time-Frequency domain using Wavelet Decomposition.
  - Cross-correlation.
  - Non-linear Interdependence.
  - Dynamical Entrainment; difference of short-term Lyapunov exponents (DSTL).
  - Phase-locking Synchrony (SPLV).
  - Entropy of the phase difference.
  - Wavelet Coherence.
  - Short-Time Fourier Transform (STFT)

K Nearest Neighbor (kNN) and Support Vector Machine (SVM)

(Savadkoohi et al. 2020) built a feature extraction process that extracted the "time domain" using the Butterworth filter (1-70 Hz) 1, the "frequency domain" using a Fourier transform 2, and the "time-frequency domain" using Wavelet decomposition for the entire dataset. The resulting data is then split into its 5 brain wave bands; Delta, Theta, Alpha, Beta, and Gamma. From these 4 variables were extracted, mean, variance, skewness, and kurtosis, leading to 60 total extracted features.

$$y(n) = \sum_{i=0}^{N} a_i \cdot x(n-i) + \sum_{j=1}^{N} b_j \cdot y(n-j)$$

Figure 1: Third Order Butterworth bandpass filter

$$\widehat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\ e^{-i2\pi\xi x}\,dx.$$

Figure 2: Fourier transform equation

A prediction was calculated from an kNN and SVM model for each separate domain. The results are shown in 6 showing the potential of each model.

Model	Accuracy %		
	TD	FD	T-FD
Support Vector Machine (SVM)	99.5	100	100
K Nearest Neighbor (kNN)	99.5	99	99.5

Table 6: Results from (Savadkoohi et al. 2020) showing results for the Time Domain, Frequency Domain, and Time-Frequency Domain

The classification times however were not included in the report. Extraction in real-time may not be applicable for an approach so extensive in its feature extraction.

Logistical Regression (LR), Support Vector Machine (SVM), and Convolutional Neural Network (CNN)

Mirowski et al. used the (Freiburg 2024) dataset containing intracranial recordings and measured the performance of an LR, SVM, and CNN ML models. They compiled 6 different preprocessing pipelines and tested the selected models against each. The preprocessing pipelines are as follows:

- The Cross-correlation between pairs of EEG channels were calculated with delays ranging from -0.5 to 0.5. The delays allowed for the propagation and processing time of brainwaves. Cross-correlation describes the amount f has to be shifted along the x axis to equal g3. Only the maximum value of the cross-correlation values were retained for training.

$$(f \star g)(\tau) \triangleq \int_{-\infty}^{\infty} \overline{f(t)} g(t+\tau)\, dt$$

Figure 3: Cross-Correlation

- Non-linear interdependence "which measures the distance, in state-space, between time-delay embedded trajectories of two EEG channels" was also extracted. This is an bivariate feature which “measures the Euclidian distance, in reconstructed state-space, between trajectories described by two EEG channels”.

- The third method was Dynamical Entrainment; difference of short-term Lyapunov exponents (DSTL), where an Lyapunov exponent 4 describes the rate of separation for two trajectories, which in this report was based off “a common measure of the chaotic nature of a signal” (Mirowski, Madhavan, LeCun & Kuzniecky 2009)

$$\lambda = \lim_{t\to\infty} \lim_{|\delta \mathbf{Z}_0|\to 0} \frac{1}{t} \ln \frac{|\delta \mathbf{Z}(t)|}{|\delta \mathbf{Z}_0|}$$

Figure 4: Maximal Lyapunov (λ) exponent

- The last 3 features were Phase-locking Synchrony (SPLV), Entropy, and Coherence of the phase difference. These were extracted from the the wavelet transformation The wavelet transformation extracts the frequency-specific and time-dependent phases.

$$SPLV_{a,b}(f) = \frac{1}{N} \sum_{l=1}^{N} e^{i\left(\phi_{a,f}(l) - \phi_{b,f}(l)\right)}$$

Figure 5: Phase-locking Synchrony (SPLV) extraction from the wavelet transformation

The following results show the number of patients with perfect seizure prediction results “(no false positives, all seizures predicted)” from (Mirowski et al. 2009) for each model, for each preprocessing pipeline.

Cross-Correlation		
LR	CNN	SVM
4	9	4
19%	43%	19%

Table 7: Cross-Correlation results.

Non-Linear interdependence		
LR	CNN	SVM
3	10	5
19%	48%	24%

Table 8: Non-Linear interdependence results.

DSTL
SVM
1
5%

Table 9: Dynamical Entrainment; difference of short-term Lyapunov exponents (DSTL) results.

SPLV		
LR	CNN	SVM
10	13	7
48%	62%	33%

Table 10: Phase-locking Synchrony (SPLV) results.

Entropy of Phase Difference		
LR	CNN	SVM
9	11	7
43%	52%	33%

Table 11: Entropy of Phase Difference results.

Distribution of Wavelet Coherence		
LR	CNN	SVM
11	15	8
52%	71%	38%

Table 12: Distribution of Wavelet Coherence results.

These results show the strength of an CNN when compared to other ML models as it's obtained the highest results across all tables. LR however can also achieve strong results when paired with the correct preprocessing pipeline. Both methods would be applicable for this project assuming that the time from recording to final prediction is less that an second.

Convolutional Neural Network (CNN)

Another CNN approach was implemented by Truong et al. (Truong, Nguyen, Kuhlmann, Bonyadi, Yang, Ippolito & Kavehei 2018) who applied an Short-Time Fourier Transform (STFT) transform 10 on 30 seconds of EEG data. Interference was then removed from the produced matrices by applying an notch filter. The final result was then fed into an CNN producing these results:

No. of seizures	Sensitivity (%)	False Positive Rate (/h)
59	81.4 ± 0.0	0.06 ± 0.00

Table 13: Cross-Correlation results.

5 Project Methodology and Development

5.1 Methodology

5.1.1 Approach

From the datasets discussed in 4.1 the Children's Hospital Boston (CHB-MIT) EEG dataset has been chosen due to the large amount of continuous data, suitable for extracting the preictal period 5. The dataset also has EEG nodes positioned on the subject's scalps, fitting with the use case of this project. Furthermore, the CHB-MIT EEG dataset has produced summary plain-text files for each of the subject's recordings, stating when every seizure began and ended allowing for an straightforward methodology when configuring the EEG tuning and training environment.

Through the analysis of various ML models an CNN will be used for this project. Both (Truong et al. 2018) and (Mirowski et al. 2009) shows the potential CNN have when being used in preictal prediction. The approach described in (Truong et al. 2018) was able to achieve similar or better results in comparison with other preprocessing pipelines and / or ML model configurations while using an less computationally intense preprocessing method. This method was based around an STFT extraction and is described in (Truong et al. 2018). This may allow for the classification to be done in real time, meeting the project's objectives 3.2. To ensure this requirement is met an real-time simulation will be developed, showing the raw EEG data and the final prediction every second.

5.1.2 Development System

Due to the linear nature of this project an Waterfall methodology will be used. Waterfall ensures that previous stages of the software are complete before moving on, this will keep the project on track and moving in an forward direction, crucial for the time-limited aspect of this project. Furthermore, due to the analysis undergone through literature reviews the direction and therefore the goals of the project are clearly defined; this further supports the choice of an Waterfall approach.

5.1.3 Version Control

This project will be under version control through the use of Git, along with GitHub as an remote repository. Both the dataset and the trained models will not be under version control. The use of an version control system and remote repository ensures that if anything goes awry an backup of all commits are stored remotely, securing any progress made in the project. Git also gives the benefit of many features; notably branches and stashing, allowing for developing on ideas, features, and bugs without affecting the "main" code, and merging allows for the combination of said branches.

5.1.4 Software and Libraries Used

Name:	Use case:
Google Cloud	Used to download the CHB-MIT EEG dataset
Keras (TensorFlow)	Used to build, train and test the ML models.
Matplotlib	Used in the real-time simulation to plot the Spectrogram and CHB-MIT EEG data.
MNE	Provides classes for loading EEG data from both CSV and EDF files. Also used for running an Short-Time Fourier Transform (STFT) extraction.
NumPy	Utilized functions and classes to store and manipulate loaded data.
Pandas	Utilized functions and classes to store and manipulate loaded data.
scikit-learn	Provides functions used to produce performance metrics for trained ML models.
nicegui.py	An framework which allows for web-application development.

Table 14: List of Libraries used

5.1.5 Schedule

The schedule describes the way the Waterfall methodology has affected the project; it shows how it's been broken down into smaller, distinct stages with progress being blocked until the previous stage has been completed. The overlap in March and April is when the models are being tuned, therefore the only active development is on the real-time simulation.

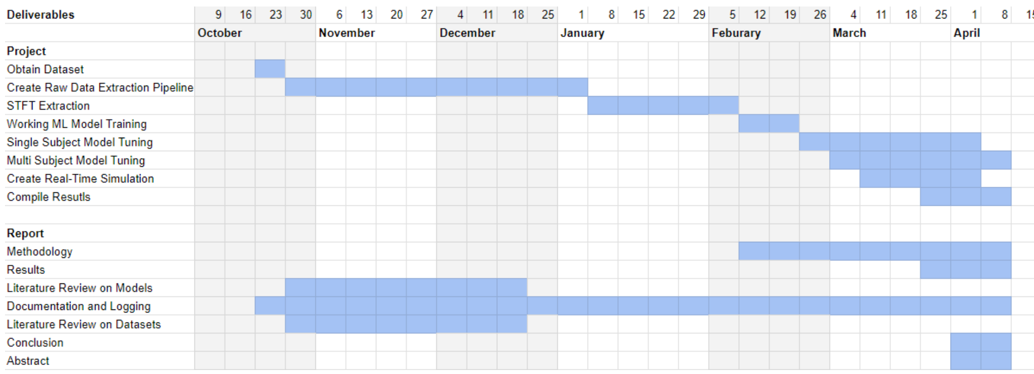


Figure 6: Gantt Chart

5.2 Development

5.2.1 Electroencephalogram (EEG) Data Extraction

The European Data Format (EDF) (Kemp, Värri, Rosa, Nielsen & Gade 1992) is a file format which was designed for the archival and exchange of EEG recordings (Kemp & Roessen 2013) and was the format of the CHB-MIT EEG dataset. EDF file's contain raw EEG signals plotted against an time axis and contain metadata such as the name and recording frequency for each node. 7 shows an plot of raw EEG data.

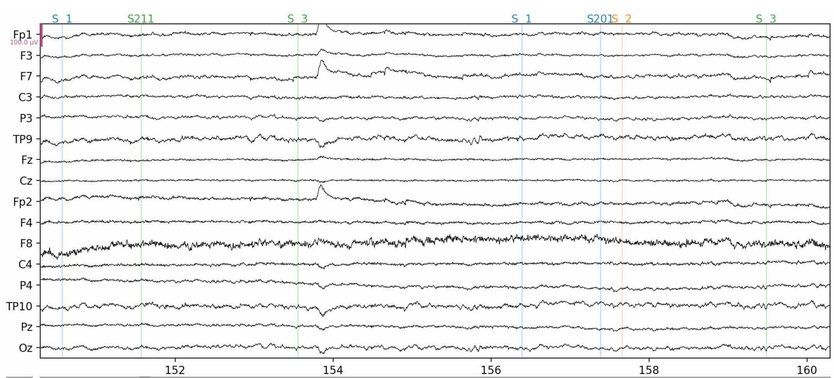


Figure 7: Extracted raw EEG data. (Science 2024)

Each node's name describes it's position within the internationally accepted 10-20 system shown in 8. The 10-20 system was popularized due to it's ability to cover all brain regions in an method proportional to the skull size and shape. This ensures that the inter-electrode spacing is equal, allowing for consistent measurements to be taken regardless of subject. (Morley, Hill & Kaditis 2016)

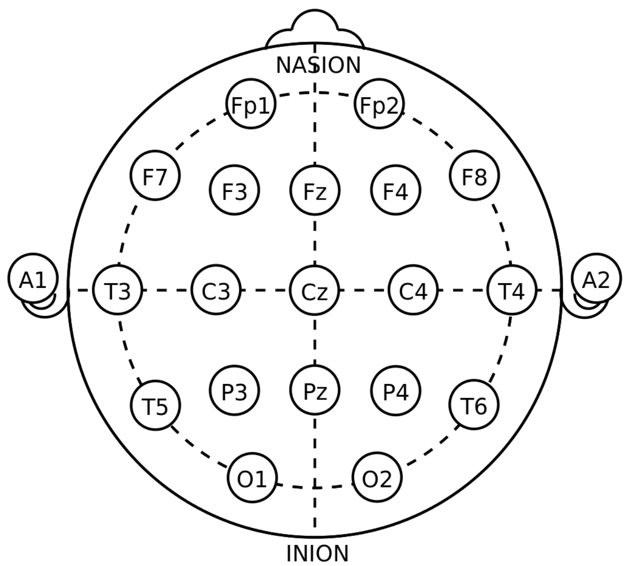


Figure 8: Diagram of the 10-20 International System. (*21 electrodes of International 10-20 system for EEG.svg* 2010)

The CHB-MIT dataset also contained summary files (see 3). From these summary files an Object-oriented Programming (OOP) representation of the dataset was created. This OOP representation was generated for every recording and shows which subject it belonged too, the name of each EEG node within the recording, also representing each node's position in the 10-20 system, the recording frequency of the EEG, and any seizure's start and end times contained within that recording. This OOP representation allowed for the labelling of the ictal, preictal and interictal periods.

For this project the EEG signals were extracted and stored in an CSV file. This allowed for an more flexible configuration of tuning scenarios such as EEG node combinations or subject specific tuning, further discussed in **??**. Within the CHB-MIT EEG dataset each EEG configuration was not equal as some recordings contained nodes not present in other recordings. Due to this the common channels were extracted to allow for an fair comparison of the results. There were 17 channels with the names: FP1-F7, F7-T7,T7-P7, P7-O1, FP1-F3, F3-C3, C3-P3, P3-O1, FP2-F4, F4-C4, C4-P4, P4-O2, FP2-F8, F8-T8, P8-O2, FZ-CZ, CZ-PZ. An example of the extracted data can be seen at 9. For each recording 3 CSV files were created, "/ictal/master.csv", "/preictal/master.csv" and "/interictal/master.csv". If a period appeared more than once in an single recording it was concatenated to it's corresponding "master.csv" file.

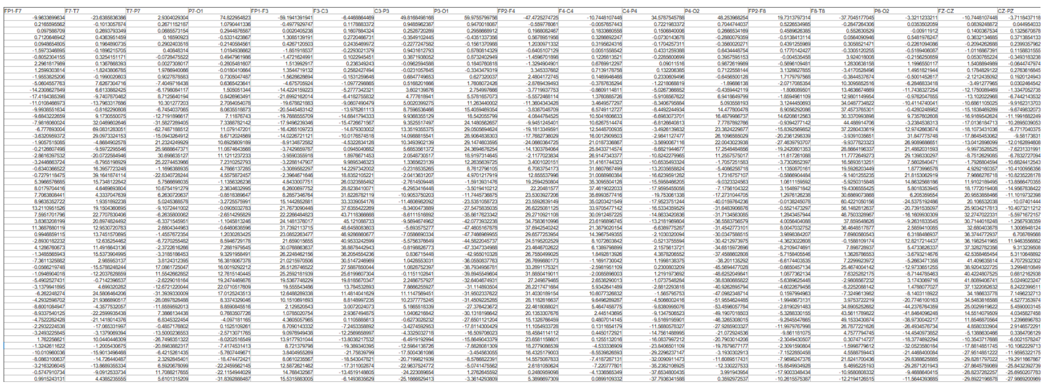


Figure 9: Head of extracted EEG data in CSV format. The shown data is approx 1/5 second and each column represents an single EEG node's data.

5.2.2 Short-Time Fourier Transform (STFT)

STFT extraction is applied against STFT is applied to the data collected from the EEGs to produce an spectrogram of length 30 seconds. These images contains only a single class and were the input into the CNN. STFT can be expressed mathematically;

$$\mathbf{STFT}\{x(t)\}(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-i\omega t}\,dt$$

Figure 10: Short-Time Fourier Transform (STFT) Window Extraction

Where $w(\tau)$ is an window function and $x(t)$ is the input signal from the EEGs.

This can also be explained as taking an Fourier transform of the EEG signals after an window function has been applied, and then sliding an window across the result. The sliding window transforms the one-dimensional output from the Fourier transform into two-dimensional data allowing for visual analysis.

There are various parameters for an STFT transformation:

Window Function	Used to isolate signal currently undergoing analysis. Optimal functions have low to no artefacts left in the signal and creates no discontinuities at section boundaries.
Window Size	Changes the size of the window function. This affects the resolution of both time and frequency, leading to the uncertainty principal; either variable will be in high resolution. See 11 and 19
Time Step (step size or hop size)	This is the distance between windows. Influences window over or underlap, as well as directly affecting computational load.

Table 15: List of Short-Time Fourier Transform (STFT) parameters

For this project these variables were selected:

Window Function	Sine Window: $w[n] = \sin\left(\frac{\pi n}{N}\right) = \cos\left(\frac{\pi n}{N} - \frac{\pi}{2}\right), \quad 0 \leq n \leq N$.
Window Size	7680. The length of the input data.
Time Step (step size or hop size)	3840. This leads to an window overlap.

Table 16: List of Short-Time Fourier Transform (STFT) parameters choices

See 19 for an Spectrogram produced with these parameters.

5.2.3 Notch Filter

"Power line interference may severely corrupt neural recordings at 50/60 Hz and harmonic frequencies. The interference is usually non-stationary and can vary in frequency, amplitude and phase." (Keshtkaran & Yang 2014) This poses a large issue when training an ML model against EEG recordings as the interference may affect the model's ability to pick up on characteristics, or may mislead the model.

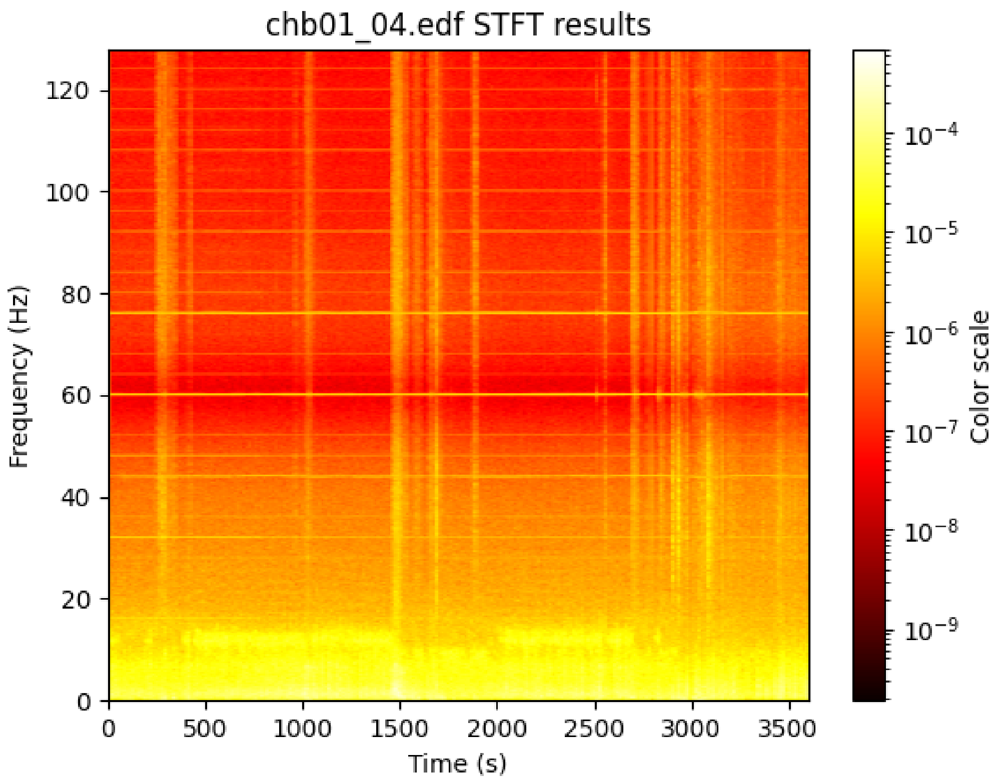


Figure 11: An STFT window with high frequency resolution.

11 clearly shows power line interference with an horizontal bands around 60 Hz. (Keshtkaran & Yang 2014) discusses the different methods for removing power line interference, although due to the objectives stated in 3.2, an compute-intensive method needed to be avoided to allow the project to meet the real-time requirement. Through analysis done by Keshtkaran et al. an notch filter is fairly computationally light, although it does however have drawbacks such as not entirely removing the interference (Keshtkaran & Yang 2014). An notch filter was therefore applied to the raw data during the application of the STFT function.

5.2.4 Synthesize Data

Within the CHB-MIT dataset there was an class imbalance with the interictal period dwarfing all other classes for every subject. An visualization of the data can be see bellow in 17 which was generated from the extracted data. All subjects have very little data for the preictal and ictal classes, with most of them having under 30 seconds of recorded data. If an model was trained off synthesized data for these subjects, regardless of synthesization method, the results would be greatly misleading and the models would not work in an real-life application. Therefore it was decided that only subjects "chb06" and "chb12" will be utilized for the training and tuning of ML models within this project.

	Recordings			Line Count			Length (Seconds)		
	Ictal	Pre	Inter	Ictal	Pre	Inter	Ictal	Pre	Inter
chb01	7	7	42	456	9334	37352484	1.78	36.46	145908.14
chb02	3	3	36	15	3118	32489237	0.059	12.18	126911.08
chb03	7	7	38	416	8201	35004081	1.625	32.04	136734.69
chb04	3	3	42	2385	15164	143801293	9.32	59.23	561723.8
chb05	5	5	39	568	7344	35927152	2.22	28.69	140340.44
chb06	7	7	18	17803	41291	61439324	69.54	161.3	239997.36
chb07	3	3	19	331	21128	61769049	1.29	82.53	241285.35
chb08	5	5	20	929	11739	18420150	3.63	45.86	71953.71
chb09	3	3	19	6648	19716	62519544	25.19	77.02	244216.97
chb10	7	7	25	461	26621	46068086	1.8	103.99	179953.46
chb11	3	3	35	812	3682	32052414	3.17	14.38	125204.74
chb12	10	10	21	8058	10643	18532198	31.48	41.57	72391.40
chb13	8	8	33	4435	8988	30391012	17.32	35.11	118714.90
chb14	7	7	26	1608	12433	23940969	6.28	48.57	93519.41
chb15	14	14	40	7160	12350	36843574	27.97	48.24	143920.21
chb16	6	6	19	4381	5832	17495378	17.11	22.78	68341.3
chb17	3	3	21	299	7678	1934639	1.17	29.99	75572.03
chb18	6	6	36	329	9142	32822357	1.29	35.71	128212.33
chb19	3	3	30	242	5657	27569463	0.95	22.1	107693.21
chb20	6	6	29	2569	6147	25421628	10.04	24.01	99303.23
chb21	4	4	33	207	7451	30240352	0.81	29.11	118126.34
chb22	3	3	31	210	7004	28557334	0.82	27.36	111552.09
chb23	3	3	9	12020	6111	24455749	46.95	23.87	95530.27

Table 17: Recording information for each subject. Valid represents if an subject has enough data, and is therefore suitable for training.

An sliding window method was used to synthesize the data for subject “chb06” and “chb12”. The sliding window method moves an window of 30s across the data at speed x. Variable speed allows for the generation of an specific number of windows, see 12 for an diagram of this process. 13, 14, and 15 were taken from the project’s implementation and describe the algorithm used to synthesize the data.

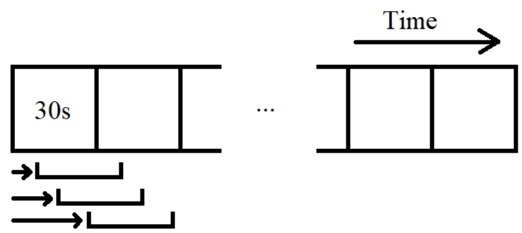


Figure 12: Sliding Window technique for synthesizing data spectrogram images

$$\text{window\_size} = \text{desired\_size} \times \text{sampling\_rate} = 30 \times 256$$

Figure 13: Calculate the desired window size. "desired_size" is the length of the windows in seconds. "sampling_rate" is the sampling rate of the recorded data.

$$\text{step} = \frac{\text{total\_rows} - \text{window\_size}}{\text{target} - 1}$$

Figure 14: Calculate the "step" or "speed" of the sliding window. Can also be described as the offset between window starting points. "target" is the desired number of windows.

$$\text{windows} = \{(\lfloor i \times \text{step} \rfloor, \lfloor i \times \text{step} + \text{window\_size} \rfloor) \mid i \in \{0, 1, 2, ..., \text{target} - 1\}\}$$

Figure 15: Produces an set of pairs, where an pair describes the start and ending index of an single window. The set length is the same as "target"

5.2.5 Convolutional Neural Network (CNN) and tuning methodology

Overview

Due to the lack of suitable subjects discussed in 5.2.4 this project focused on tuning an CNN and it's architecture. The number of convolutional blocks and dense layers, as well as the dense layer's complexity were setup as the model's architecture tuning variables, this is further disused in 5.2.5 along with the other variables used for tuning. The loss function and optimizer however were not changed between model iterations and were set to Categorical Cross-Entropy and the Adam Optimizer respectively. Categorical Cross Entropy 16 is a common loss function for multi class classification applications such as this project, therefore all models within this project utilized the same Categorical Cross Entropy function allowing for the comparison of the loss metrics.

$$H(y, \hat{y}) = -\sum_{i} y_i \log(\hat{y}_i)$$

Figure 16: Categorical Cross-Entropy

The Adam optimizer combines two different versions of stochastic gradient descent; Adaptive Gradient Algorithm and Root Mean Square Propagation. Both of these optimizer functions handle the learning rate for each node within an neural network allowing for more efficient training. The two functions manage the learning rates differently with Adaptive Gradient Algorithm being more applicable for natural language and computer vision problems, Root Mean Square Propagation however suites online learning and non-stationary problems (Brownlee 2021). Adam combines these two approaches, realizing the benefit of each of them and configures the learning rates accordingly. The performance of Adam when compared to other optimizer functions can be seen in 17 (Brownlee 2021).

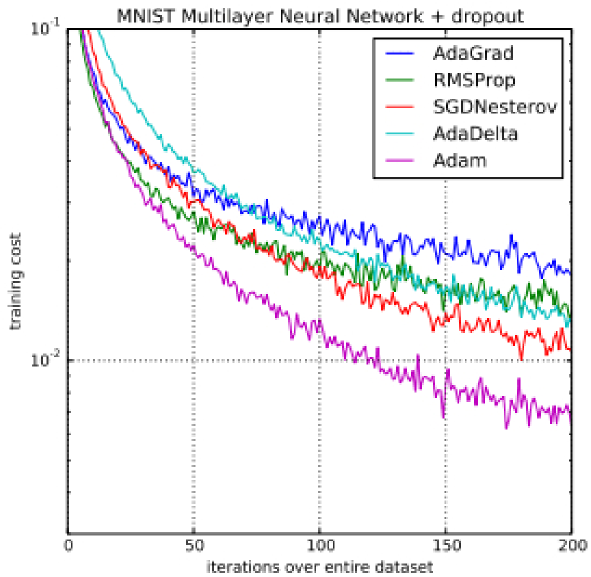


Figure 17: Optimization function performance comparison.

With only a single machine and the time available specified in **??**, applying Leaving One Out Cross-Validation (LOOCV) for all architecture and variable configurations would of been impossible. Therefore an testing training split ratio needed to be decided on. The time costs of running testing, especially on complex CNN models were high, consequently an large split for testing would be impractical as it would further drive up the training testing times. Due to the nature of the way the data was synthesized as well, an smaller testing split would be representative of all the available data for each class, this therefore lead the project to use an 80% training and 20% testing split.

Architecture

The CNN was dynamically built in accordance to the current variable configuration. By default an single 2 dimensional convolutional block was included with an input shape of $(17 \times 3841 \times 2)$ such that the model would accept an 30 second STFT window. This convolutional block had 32 filters with an kernel size of $(3 \times 3 \times 3)$. The dynamic convolutional blocks were then built, increasing the number of filters following the formula $32 \times (2^i)$ where i

is the convolutional block index. Each convolutional block is followed by an batch normalization, and after all convolutional blocks an 2 dimensional Max Pooling layer with an pool size of (2×2) is applied, along with an Flatten layer preparing the data to be passed to the dense layers.

After the convolutional blocks and related layers have been added the dense layers are built. Again, an single dense layer is added by default which has an output shape of $(1, 3)$, preceding this dense layer is the dynamically generated dense layers. These layers have an set output shape specified in the architecture tuning variables.

Listing 1: Function which dynamically builds an CNN model from the specified parameters. Logging has been removed.

```python
    def compile_model(self, num_conv_layers=4, num_dense_layers=4, dense_layer_size=64):
        self.num_conv_layers = num_conv_layers
        self.num_dense_layers = num_dense_layers
        self.dense_layer_size = dense_layer_size

        if num_dense_layers < 1 or num_conv_layers < 0:
            raise ValueError

        model = models.Sequential()

        # add convolutional layers
        model.add(layers.Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(17, 3841, 2)))
        for i in range(1, num_conv_layers):
            model.add(layers.Conv2D(32 * (2**i), kernel_size=(3, 3), activation='relu'))
            model.add(layers.BatchNormalization())

        model.add(layers.MaxPooling2D(pool_size=(2, 2)))
        model.add(layers.Flatten())

        # add dense layers
        for i in range(num_dense_layers-1):
            model.add(layers.Dense(dense_layer_size, activation='relu'))

```

```
        model.add(layers.Dense(3, activation='softmax'))
        model.compile(loss='categorical_crossentropy',
           optimizer='adam', metrics=['accuracy'])
```

Tuning Variables 18 shows the different variables and their possible values used for model tuning. An model was trained for possible configurations of these variables.

Model Parameters			
Num. Conv. Layers	2	3	
Num. Dense Layers	4	8	12
Dense Layer Size	64	128	256
Architecture Parameters			
Epochs	1	8	32
Batch Size	16	32	48

Table 18: The possible values for each tuning variable.

The number of convolutional layers had to be limited; as the filter size increased exponentially with each layer an Out of Memory (OOM) error would occour when the filter size was too large. Therefore only 2 and 3 convolutional layers were tested. The number of epochs were also limited due to training time. With this configuration of tuning variables, 162 models were required to be trained.

Metrics For each model an plain text file was created during it's tuning process. This plain text file recorded the start and end times of each process, including its model compiling, model training and model testing. The file also contained metrics such as the training and testing accuracy and loss, the final confusion matrix and the normalized confusion matrix. An classification report was also generated which calculated the precision, recall, F1-score and support for each class. The classification report also calculated the accuracy for the model. An example of this file can be seen in 2. The results are available along with an discussion in 6.2.

Listing 2: Logging file programmatically generated during the tuning process. Training accuracy and loss lines have been trimmed for readability.

```
------------------------------------------------
```

```
                    current configuration

                conv layers count   :   2
                    dense layer count   :   4
                        dense layer size    :   128

                            epochs  :   32
                                batch_size  :   32



dataset generation start : 2024-04-06 21:00:09.473945
one hot matrix:
['interictal', 'preictal', 'ictal']
{'interictal': 0, 'preictal': 1, 'ictal': 2}
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
target subjects : ['chb06']
dataset generation end : 2024-04-06 21:00:09.666774
compiling model start : 2024-04-06 21:00:09.669026
compiling model end : 2024-04-06 21:00:09.847257
training start : 2024-04-06 21:00:09.848724
training end : 2024-04-07 01:59:22.883511
training accuracy: [0.974943995475769,
   0.997187077999115, 0.9978642463684082,
   0.9973433613777161, 1.0...
training loss: [1.4391484260559082,
   0.01381457969546318, 0.019184233620762825,
   0.043409526348114014...
saving model start : 2024-04-07 01:59:22.893082
saving model end : 2024-04-07 01:59:34.687428
cmatrix prediction start : 2024-04-07 01:59:34.691563
cmatrix prediction end : 2024-04-07 02:02:52.020467
Confusion Matrix:
[[1626    0    0]
 [   0 1543    0]
 [   0    0 1631]]
Normalized Confusion Matrix:
[[1. 0. 0.]
```

```
 [0. 1. 0.]
 [0. 0. 1.]]
Classification Report:
              precision    recall  f1-score   support

  interictal       1.00      1.00      1.00      1626
    preictal       1.00      1.00      1.00      1543
       ictal       1.00      1.00      1.00      1631

    accuracy                           1.00      4800
   macro avg       1.00      1.00      1.00      4800
weighted avg       1.00      1.00      1.00      4800
```

5.2.6 Real Time Simulation

An real-time simulation was built to show the potential of the preprocessing and prediction pipeline, proving it's ability to output an prediction from raw EEG data each second. The real-time solution was built with "NiceGui.py" and the solution to setup model tuning was adapted to work in this new environment, implementing each step as defined above. The NiceGui solution has an hard-coded path to an saved keras model. This model is loaded before the simulation begins and is used to run the predictions. 18 contains an image of the real-time simulation.

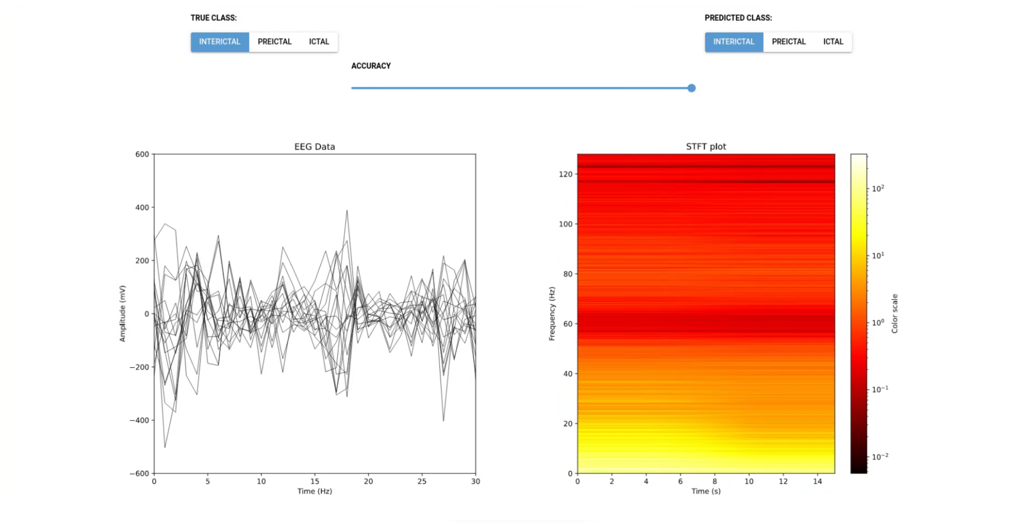


Figure 18: Screenshot of running real-time simulation.

Within the simulation you can toggle which class the EEG data is being pulled from, this then replaces the last 30 seconds of streamed data with data extracted from the newly selected class. This in turn refreshed the two plots on the webpage, with the left plot representing the signals from each EEG node, and the right plot being the produced STFT window used for prediction. The predicted class then appears in the top right, and the accuracy bar, representing the accuracy as a percentage is updated.

5.3 Problems Encountered

There have been many issues faced throughout the duration of this project. From the very start managing bodies of datasets would not respond, this impacted my decision making when selecting one of the datasets discussed in 4.1. Furthermore, once the CHB-MIT dataset was selected there were discrepancies within the summary files and the extracted data with channel names not matching. The CHB-MIT dataset also did not record the EEG values in the same manner for all subjects, this lead to some data points from EEG nodes to be discarded as those specific EEG channels were not used for all subjects. On average this meant over 25% of recorded data had to be removed. Of what was recorded, some subjects had very little data for certain classes with an overabundance of recorded interictal data. Having such little data for certain classes decreases the confidence of the synthesized data, as even though enough data points are available, the synthesized data may not be representative of an real-life situation.

This project's approach changed substantially across its development. Originally development attempted to extract numeric and statistical features for each recorded point, more akin to traditional ANN methodologies, however this approach was abandoned as it would not of been able to meet the real-time objective of this project.

Tuning the models raised more problems than anything else in this project, with OOM issues being an common problem occurring throughout the model tuning process and the development of the real-time simulation. All code affected with OOM problems had to be carefully re-written to remove any memory leaks as well as implementing generator functions which for example would open the dataset paths used for training only when required. Some OOM errors were not avoidable however. Certain model architecture configurations would lead to memory errors with convolutional block complexity growing exponentially only an maximum of 3 convolutional blocks were able to be used. Dense layer count and complexity on top also added to the problem.

There were also many failed attempts at tuning the models. An reoccurring error was missing data points or even classes when tuning, this lead to many trained models being invalid as they would only predict 2, or even 1 class due to it's training data. Even when tuning was seemingly working this issue would appear with no exceptions being raised or problems being apparent. Over 250 trained models have been discarded due to this problem alone, this has drastically reduced the results avaliable for discussion.

6 Review

6.1 Objectives and Requirements

Index	Objective	Achieved
1	Research and find an suitable dataset allowing for the classification of the interictal, preictal, and ictal periods.	True
2	Research and find an suitable ML model approach.	True
3	Create a data preperation and preprocessing pipeline which extracts the dataset into an ML format suitable for training and testing.	True
4	Undergo parameter tuning and model architecture tuning of the selected ML model approach.	True
5	Produce a simulation interface which streams EEG data in real time through the preprocessing pipeline and into the optimal model. The simulation should show the current classification to the user every second.	True

Table 19: Objectives taken from 3.2.

Index	Requirements	Achieved
1	Develop and preprocessing pipeline which accepts raw EEG data.	True
2	Tune an number of models to find the optimal variable configuration.	True
3	Ensure that the optimal model has an accuracy of $\geq 90\%$ against an validation dataset.	True
4	From EEG data extraction to prediction, the process must take at most 1 second.	True
5	Create an real-time simulation mimicking an real life application.	True

Table 20: Requirements taken from 3.3.

The above tables correspond to the items in 3.3 and 3.2. Upon review, even with the tuning issues discussed in 7.1, this project was able to achieve all of it's objectives and requirements.

6.2 Results

6.3 Future Advancements

The risks proposed are for a final seizure detection product which is available to consumers and healthcare services. The current scope of the project does not contain the risks stated in the following sections and therefore the development goals of the project have not been changed to negate any of these risks.

7 Professional Issues and Risks

7.1 Professional Issues

1. Legal Issues:

    - If the project was to be implemented into an seizure prediction system then GDPR legislation will have to be upheld; if the model is utilizing online learning then the data being stored will not be anonymous, meaning the data will have to be deleted when required and avaliable to send to the user on their request. If the data however was truly anonymous, akin to the CHB-MIT dataset used in this project, then the data does not fall under GDPR. Another legal issue could arise if an seizure was not predicted correctly leading to an user suing the company producing the prediction system. This could be solved through an agreement which the end user has to accept, stating the producers of the product, including the developers are not liable.

2. Social Issues:

    - People from all walks of life have epilepsy, and due to financial costs some may not be able to afford the prediction system. This will be an issue that may not be solvable by the producers of the software, and instead may need to rely on funding such as an governmental body, healthcare service or charity. The producers can make the software open source allowing for users to fit it onto their own hardware, although that will take time, money, and knowledge for the individual to setup an equivalent system.

3. Ethical Issues:

    - The final product should attempt to achieve the same accuracy regardless of age or gender, although due to the differences in the way each persons brain functions and it's characteristics, this may be a difficult task to achieve. In an ideal world either an generic model would exist, or models would be trained off datasets specific to age or gender allowing for common characteristics between individuals in those groups to be picked up on, however there is an significant lack of data currently available make this unfeasible.

4. Environmental Issues:

    - EEG nodes are comparatively inexpensive to their possible benefits, due to the low price of EEG nodes the creation of an prediction device won't have a large material cost. The ML model tuning process however will have a large energy cost. This will need to be taken into account when setting up an system that tunes models at an industry scale. Steps will have to be taken to minimise the energy consumption of the system.

5. Intellectual Property Issues:

    - There will be an issue when deciding who owns the recorded data. If the company who produces the final product owns it then they can utilize the large amount of data to further refine and train more advanced models, allowing them to tackle other issues such as the ones discussed in "Social Issues", although this should be a choice for the end user. Another issue may be patenting the product and which components should or can be patented or copyrighted.

6. Accessibility Issues:

    - As discussed epileptic patients come from all walks of life which means the final device needs to have an accessible interface, allowing everyone to have a clear indication of the state of the device, including the preictal period alert or even if the device is on. Different people will need different alert methods, and the final device should be extensible, allowing the final patient to fit it to their disability. Some variants of the device should have any combination of light, sound and vibration alerts, as well as notifications to any of their other devices. This will allow anyone, regardless of disability to be alerted if an preictal period has been detected.

7.2 Risks

7.2.1 Risk Matrix

The numbers in each cell of the risk matrix corresponds to the items in 7.1.

Probability	Harm Severity			
	Minor	Marginal	Critical	Catastrophic
Certain				
Likely				
Possible				
Unlikely			4, 6	3, 5, 7
Rare				1, 2

Table 21: Risk Matrix

7.2.2 Risk Assessment

No.	Risk	Impact	Mitigation Strategy
1	Personal data could be leaked through an attack	Fail to adhere to GDPR. Criminal Offence	Implement encryption for data as well as increasing security measures
2	Personal data is leaked by unauthorized employee	Fail to adhere to GDPR. Criminal Offence	Tighten access controls. Educate employees about password and general security
3	Personal data is leaked by an authorized employee	Fail to adhere to GDPR. Criminal Offence	Employ education techniques stating the importance of adhering to GDPR as well as minimizing access to sensitive data.
4	False Negatives for the individual	End users will be unprepared for their seizures and may be caught in an unfavourable situation depending on reliance on the final product.	Implement cross validation techniques. Request all missed seizures to be logged or automatically detected for further training and inspection.
5	False Positives for the individual	A patient may experience un-needed stress or stop important activities due to false positives. Could have measurable knock on effects for an individual.	False positives should be recorded which will allow for further development of the model.
6	False Negatives in a healthcare environment	Staff may not be prepared for an seizure, increasing reaction times	See “False Negatives for the individual”
7	False Positives in a healthcare environment	Staff may waste time preparing for a seizure which never occours, where their help may be needed elsewhere	See “False Positives for the individual”

Table 22: Risk Assessment

8 Conclusion

asdf

9 Final Thoughts

10 Bibliography

References

21 electrodes of International 10-20 system for EEG.svg (2010). Accessed: 2024-04-10.
URL: *https://commons.wikimedia.org/wiki/File:21_electrodes_of_International_10-20_system_for_EEG.svg*

Acharya, U. R., Hagiwara, Y. & Adeli, H. (2018), 'Automated seizure prediction', *Epilepsy & Behavior* **88**, 251–261.

Assi, E. B., Nguyen, D. K., Rihana, S. & Sawan, M. (2017), 'Towards accurate prediction of epileptic seizures: A review', *Biomedical Signal Processing and Control* **34**, 144–157.

Awad, M. & Khanna, R. (2015), *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, Apress.
URL: *https://books.google.co.uk/books?id=qPGQnAEACAAJ*

Brownlee, J. (2021), 'Gentle introduction to the adam optimization algorithm for deep learning', `^1^`. Accessed: 2024-04-14.

Freiburg, S. P. P. (2024), 'Eeg database', `https://epilepsy.uni-freiburg.de/freiburg-seizure-prediction-project/eeg-database`. Accessed: 2024-04-07.

Galanopoulou, A. S., Buckmaster, P. S., Staley, K. J., Moshé, S. L., Perucca, E., Engel Jr, J., Löscher, W., Noebels, J. L., Pitkänen, A., Stables, J. et al. (2012), 'Identification of new epilepsy treatments: issues in preclinical methodology', *Epilepsia* **53**(3), 571–582.

Kemp, B. & Roessen, M. (2013), 'European data format now supports video', *Sleep* **36**(7), 1111–1111.

Kemp, B., Värri, A., Rosa, A. C., Nielsen, K. D. & Gade, J. (1992), 'A simple format for exchange of digitized polygraphic recordings', *Electroencephalography and clinical neurophysiology* **82**(5), 391–393.

Keshtkaran, M. R. & Yang, Z. (2014), 'A fast, robust algorithm for power line interference cancellation in neural recording', *Journal of neural engineering* **11**(2), 026017.

Löscher, W. & Schmidt, D. (2011), 'Modern antiepileptic drug development has failed to deliver: ways out of the current dilemma', *Epilepsia* **52**(4), 657–678.

Medicine, J. H. (n.d.), 'Types of seizures, what is a seizure?'.
URL: *https://www.hopkinsmedicine.org/health/conditions-and-diseases/epilepsy/types-of-seizures*

Mirowski, P., Madhavan, D., LeCun, Y. & Kuzniecky, R. (2009), 'Classification of patterns of eeg synchronization for seizure prediction', *Clinical neurophysiology* **120**(11), 1927–1940.

Moghim, N. & Corne, D. W. (2014), 'Predicting epileptic seizures in advance', *PloS one* **9**(6), e99334.

Morley, A., Hill, L. & Kaditis, A. (2016), '10-20 system eeg placement', *European Respiratory Society, European Respiratory Society* .

Savadkoohi, M., Oladunni, T. & Thompson, L. (2020), 'A machine learning approach to epileptic seizure prediction using electroencephalogram (eeg) signal', *Biocybernetics and Biomedical Engineering* **40**(3), 1328–1341.

Science, N. D. (2024), 'What is eeg? - neural data science in python'. Accessed: 2024-04-10.
URL: *https://neuraldatascience.io/7-eeg/about_eeg.html*

Thijs, R. D., Surges, R., O'Brien, T. J. & Sander, J. W. (2019), 'Epilepsy in adults', *The lancet* **393**(10172), 689–701.

Truong, N. D., Nguyen, A. D., Kuhlmann, L., Bonyadi, M. R., Yang, J., Ippolito, S. & Kavehei, O. (2018), 'Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram', *Neural Networks* **105**, 104–111.

Wong, S., Simmons, A., Rivera-Villicana, J., Barnett, S., Sivathamboo, S., Perucca, P., Ge, Z., Kwan, P., Kuhlmann, L., Vasa, R. et al. (2023), 'Eeg datasets for seizure detection and prediction—a review', *Epilepsia Open* .

A Results

Model key: [convolution block count].[dense layer count].[dense layer size]/[batch size].[epochs]

A.1 Tables

Model	Compiling	Saving	Testing
2.4.128/16.1	0:00:00.166984	0:00:13.026372	0:02:29.318160
2.4.128/16.32	0:00:00.180040	0:00:11.992944	0:02:36.181157
2.4.128/16.8	0:00:00.183991	0:00:12.106965	0:02:28.278624
2.4.128/32.1	0:00:00.177680	0:00:11.880910	0:02:58.134548
2.4.128/32.32	0:00:00.178231	0:00:11.794346	0:03:17.328904
2.4.128/32.8	0:00:00.166262	0:00:12.443413	0:02:50.145996
2.4.128/48.1	0:00:00.169285	0:00:14.551886	0:02:57.700320
2.4.128/48.8	0:00:00.182113	0:00:11.717746	0:02:31.724462
2.4.64/16.1	0:00:00.186607	0:00:05.879324	0:02:20.833553
2.4.64/16.32	0:00:00.167169	0:00:05.977051	0:02:40.019918
2.4.64/16.8	0:00:00.166427	0:00:06.424271	0:02:55.648672
2.4.64/32.1	0:00:00.162784	0:00:05.895562	0:02:30.918573
2.4.64/32.32	0:00:00.173991	0:00:05.902261	0:03:07.540450
2.4.64/32.8	0:00:00.164715	0:00:05.895772	0:02:32.670039
2.4.64/48.1	0:00:00.161886	0:00:05.935214	0:02:59.123882
2.4.64/48.32	0:00:00.168396	0:00:05.906412	0:02:40.036783
2.4.64/48.8	0:00:00.160545	0:00:05.846560	0:02:46.982105

Table 23: Time duration metrics for each model in format %H:%M:%S.%f.

Model	Training Accuracy	Training Loss
2.4.128/16.1	0.9832265377044678	0.8224268555641174
2.4.128/16.32	0.9993602503091097	0.02814496521023102
2.4.128/16.8	0.9976428672671318	0.11281578045975693
2.4.128/32.1	0.975464940071106	1.4378010034561157
2.4.128/32.32	0.9989679381251335	0.04745509265126133
2.4.128/32.8	0.9962754622101784	0.18717686411091886
2.4.128/48.1	0.9669219255447388	2.318654775619507
2.4.128/48.8	0.9958652406930923	0.2898520281258925
2.4.64/16.1	0.9799447655677795	0.6853911876678467
2.4.64/16.32	0.9992593247443438	0.024666143968849415
2.4.64/16.8	0.9969331100583076	0.08956567873635081
2.4.64/32.1	0.9619211554527283	1.725386619567871
2.4.64/32.32	0.9987660832703114	0.05414362490514908
2.4.64/32.8	0.9953247904777527	0.21542102738881397
2.4.64/48.1	0.9655154347419739	2.379462718963623
2.4.64/48.32	0.9989125896245241	0.07438817310655671
2.4.64/48.8	0.994374118745327	0.31704871635884047

Table 24: Accuracy and Loss metrics (average across epochs) during model training.

Model	Testing Accuracy
2.4.128/16.1	0.99
2.4.128/16.32	1.00
2.4.128/16.8	1.00
2.4.128/32.1	1.00
2.4.128/32.32	1.00
2.4.128/32.8	1.00
2.4.128/48.1	1.00
2.4.128/48.8	1.00
2.4.64/16.1	1.00
2.4.64/16.32	1.00
2.4.64/16.8	1.00
2.4.64/32.1	1.00
2.4.64/32.32	1.00
2.4.64/32.8	1.00
2.4.64/48.1	1.00
2.4.64/48.32	1.00
2.4.64/48.8	1.00

Table 25: Accuracy metrics obtained during testing.

Model	Confusion Matrix		
	1593	33	0
2.4.128/16.1	0	1543	0
	0	0	1631
	1626	0	0
2.4.128/16.32	0	1543	0
	0	0	1631
	1625	1	0
2.4.128/16.8	0	1543	0
	0	0	1631
	1626	0	0
2.4.128/32.1	0	1543	0
	0	0	1631
	1626	0	0
2.4.128/32.32	0	1543	0
	0	0	1631
	1626	0	0
2.4.128/32.8	0	1543	0
	0	0	1631
	1626	0	0
2.4.128/48.1	0	1543	0
	0	0	1631
	1626	0	0
2.4.128/48.8	0	1543	0
	0	0	1631
	1626	0	0
2.4.64/16.1	0	1543	0
	0	0	1631
	1626	0	0
2.4.64/16.32	0	1543	0
	0	0	1631
	1626	0	0
2.4.64/16.8	0	1543	0
	0	0	1631
	1626	0	0
2.4.64/32.1	5	1538	0
	0	0	1631

	1626	0	0
2.4.64/32.32	0	1543	0
	0	0	1631
	1626	0	0
2.4.64/32.8	0	1543	0
	0	0	1631
	1626	0	0
2.4.64/48.1	0	1543	0
	0	0	1631
	1626	0	0
2.4.64/48.32	0	1543	0
	0	0	1631
	1626	0	0
2.4.64/48.8	0	1543	0
	0	0	1631

Table 26: Generated Confusion Matrix during the model tuning process.

Model	Normalized Confusion Matrix		
	1.0	0.02093909	0.0
2.4.128/16.1	0.0	0.97906091	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.128/16.32	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.000647668394	0.0
2.4.128/16.8	0.0	0.999352332	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.128/32.1	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.128/32.32	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.128/32.8	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.128/48.1	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.128/48.8	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.64/16.1	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.64/16.32	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.64/16.8	0.0	1.0	0.0
	0.0	0.0	1.0
	0.9969344	0.0	0.0
2.4.64/32.1	0.0030656	1.0	0.0
	0.0	0.0	1.0

	1.0	0.0	0.0
2.4.64/32.32	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.64/32.8	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.64/48.1	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.64/48.32	0.0	1.0	0.0
	0.0	0.0	1.0
	1.0	0.0	0.0
2.4.64/48.8	0.0	1.0	0.0
	0.0	0.0	1.0

Table 27: Generated Normalized Confusion Matrix during the model tuning process.

Model	Interictal	Preictal	Ictal	Macro Average	Weighted Average
2.4.128/16.1	0.99	0.99	1.00	0.99	0.99
2.4.128/16.32	1.00	1.00	1.00	1.00	1.00
2.4.128/16.8	1.00	1.00	1.00	1.00	1.00
2.4.128/32.1	1.00	1.00	1.00	1.00	1.00
2.4.128/32.32	1.00	1.00	1.00	1.00	1.00
2.4.128/32.8	1.00	1.00	1.00	1.00	1.00
2.4.128/48.1	1.00	1.00	1.00	1.00	1.00
2.4.128/48.8	1.00	1.00	1.00	1.00	1.00
2.4.64/16.1	1.00	1.00	1.00	1.00	1.00
2.4.64/16.32	1.00	1.00	1.00	1.00	1.00
2.4.64/16.8	1.00	1.00	1.00	1.00	1.00
2.4.64/32.1	1.00	1.00	1.00	1.00	1.00
2.4.64/32.32	1.00	1.00	1.00	1.00	1.00
2.4.64/32.8	1.00	1.00	1.00	1.00	1.00
2.4.64/48.1	1.00	1.00	1.00	1.00	1.00
2.4.64/48.32	1.00	1.00	1.00	1.00	1.00
2.4.64/48.8	1.00	1.00	1.00	1.00	1.00

Table 28: F1 metrics obtained during testing.

Model	Interictal	Preictal	Ictal	Macro Average	Weighted Average
2.4.128/16.1	1.00	0.98	1.00	0.99	0.99
2.4.128/16.32	1.00	1.00	1.00	1.00	1.00
2.4.128/16.8	1.00	1.00	1.00	1.00	1.00
2.4.128/32.1	1.00	1.00	1.00	1.00	1.00
2.4.128/32.32	1.00	1.00	1.00	1.00	1.00
2.4.128/32.8	1.00	1.00	1.00	1.00	1.00
2.4.128/48.1	1.00	1.00	1.00	1.00	1.00
2.4.128/48.8	1.00	1.00	1.00	1.00	1.00
2.4.64/16.1	1.00	1.00	1.00	1.00	1.00
2.4.64/16.32	1.00	1.00	1.00	1.00	1.00
2.4.64/16.8	1.00	1.00	1.00	1.00	1.00
2.4.64/32.1	1.00	1.00	1.00	1.00	1.00
2.4.64/32.32	1.00	1.00	1.00	1.00	1.00
2.4.64/32.8	1.00	1.00	1.00	1.00	1.00
2.4.64/48.1	1.00	1.00	1.00	1.00	1.00
2.4.64/48.32	1.00	1.00	1.00	1.00	1.00
2.4.64/48.8	1.00	1.00	1.00	1.00	1.00

Table 29: Precision metrics obtained during testing.

Model	Interictal	Preictal	Ictal	Macro Average	Weighted Average
2.4.128/16.1	0.98	1.00	1.00	0.99	0.99
2.4.128/16.32	1.00	1.00	1.00	1.00	1.00
2.4.128/16.8	1.00	1.00	1.00	1.00	1.00
2.4.128/32.1	1.00	1.00	1.00	1.00	1.00
2.4.128/32.32	1.00	1.00	1.00	1.00	1.00
2.4.128/32.8	1.00	1.00	1.00	1.00	1.00
2.4.128/48.1	1.00	1.00	1.00	1.00	1.00
2.4.128/48.8	1.00	1.00	1.00	1.00	1.00
2.4.64/16.1	1.00	1.00	1.00	1.00	1.00
2.4.64/16.32	1.00	1.00	1.00	1.00	1.00
2.4.64/16.8	1.00	1.00	1.00	1.00	1.00
2.4.64/32.1	1.00	1.00	1.00	1.00	1.00
2.4.64/32.32	1.00	1.00	1.00	1.00	1.00
2.4.64/32.8	1.00	1.00	1.00	1.00	1.00
2.4.64/48.1	1.00	1.00	1.00	1.00	1.00
2.4.64/48.32	1.00	1.00	1.00	1.00	1.00
2.4.64/48.8	1.00	1.00	1.00	1.00	1.00

Table 30: Recall metrics obtained during testing.

A.2 Plots

B Misc.

Listing 3: Example Summary file from the CHB-MIT EEG dataset.

```
File Name: chb06_03.edf
File Start Time: 03:09:42
File End Time: 7:09:42
Number of Seizures in File: 0

File Name: chb06_04.edf
File Start Time: 07:09:51
File End Time: 10:50:52
Number of Seizures in File: 2
Seizure 1 Start Time: 327 seconds
Seizure 1 End Time: 347 seconds
Seizure 2 Start Time: 6211 seconds
```

```
Seizure 2 End Time: 6231 seconds

File Name: chb06_05.edf
File Start Time: 10:51:20
File End Time: 14:51:20
Number of Seizures in File: 0

File Name: chb06_06.edf
File Start Time: 14:51:23
File End Time: 18:51:23
Number of Seizures in File: 0

File Name: chb06_07.edf
File Start Time: 18:51:31
File End Time: 22:51:31
Number of Seizures in File: 0

File Name: chb06_08.edf
File Start Time: 22:51:39
File End Time: 26:51:39
Number of Seizures in File: 0

File Name: chb06_09.edf
File Start Time: 02:51:47
File End Time: 6:51:47
Number of Seizures in File: 1
Seizure 1 Start Time: 12500 seconds
Seizure 1 End Time: 12516 seconds

File Name: chb06_10.edf
File Start Time: 06:51:54
File End Time: 10:51:54
Number of Seizures in File: 1
Seizure 1 Start Time: 10833 seconds
Seizure 1 End Time: 10845 seconds

File Name: chb06_12.edf
File Start Time: 14:52:10
File End Time: 18:52:10
Number of Seizures in File: 0
```

```
File Name: chb06_13.edf
File Start Time: 18:52:20
File End Time: 22:52:20
Number of Seizures in File: 1
Seizure 1 Start Time: 506 seconds
Seizure 1 End Time: 519 seconds

File Name: chb06_14.edf
File Start Time: 22:52:35
File End Time: 26:52:35
Number of Seizures in File: 0

File Name: chb06_15.edf
File Start Time: 02:52:43
File End Time: 6:52:43
Number of Seizures in File: 0

File Name: chb06_16.edf
File Start Time: 06:52:51
File End Time: 7:43:21
Number of Seizures in File: 0

File Name: chb06_17.edf
File Start Time: 07:45:51
File End Time: 11:45:51
Number of Seizures in File: 0

File Name: chb06_18.edf
File Start Time: 11:45:55
File End Time: 13:58:03
Number of Seizures in File: 1
Seizure 1 Start Time: 7799 seconds
Seizure 1 End Time: 7811 seconds

File Name: chb06_24.edf
File Start Time: 08:23:24
File End Time: 12:23:24
Number of Seizures in File: 1
Seizure 1 Start Time: 9387 seconds
```

```
Seizure 1 End Time: 9403 seconds
```

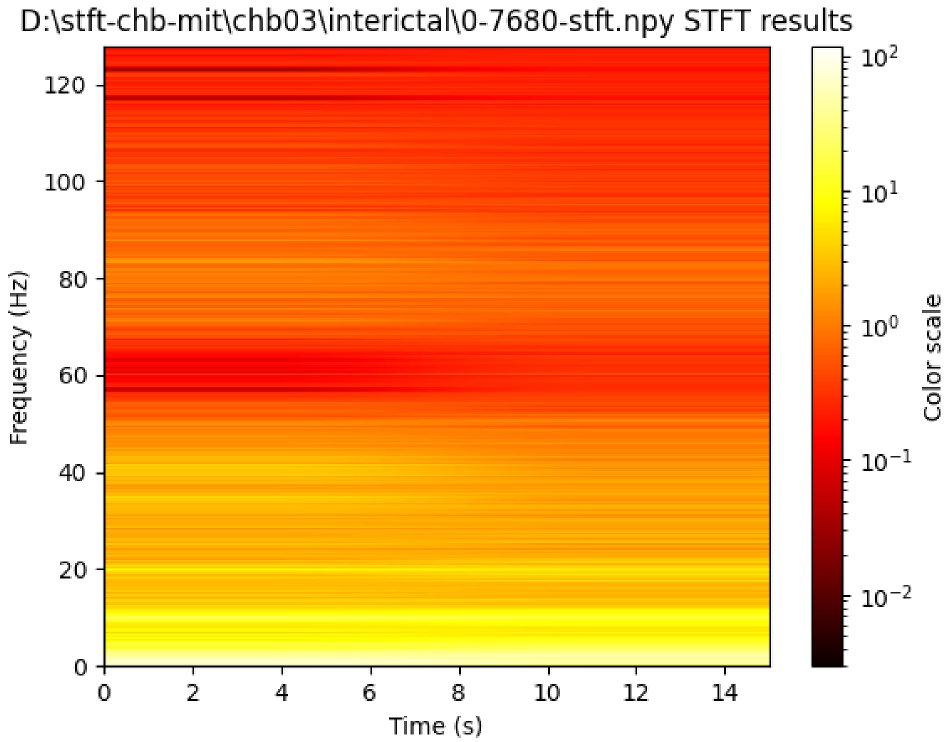


Figure 19: An STFT window with high time resolution.

C Accessed Material

D Presentation, Poster, and Ethics Form

OXFORD BROOKES UNIVERSITY

Real-time preictal detection through the application of machine learning to Electroencephalogram signals.

Final Major Project

William Riddell - 19066041

Contents

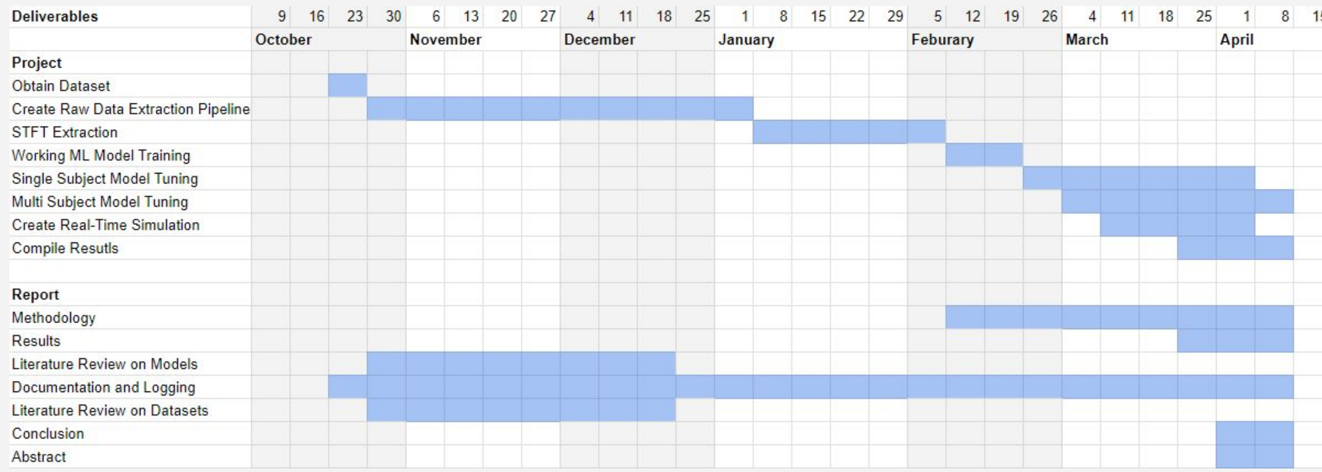
- Epilepsy
- Order of work
- Solution
  - Dataset
  - Preprocessing
  - Convolutional Neural Network
  - Post Processing
  - System Results
- Conclusion

Epilepsy

- “Epilepsy is one of the most common serious brain conditions, affecting over 70 million people worldwide.” - Thijs, R.D., Surges, R., O'Brien, T.J. and Sander, J.W., 2019. Epilepsy in adults. The Lancet, 393(10172), pp.689-701.
- “One third of people with epilepsy will not have adequate seizure control with the current medications [available Antiepileptic Drugs (AEDs)]. For these patients the situation has improved very little in the last few decades” - Galanopoulou, A.S., Buckmaster, P.S., Staley, K.J., Moshé, S.L., Perucca, E., Engel Jr, J., Löscher, W., Noebels, J.L., Pitkänen, A., Stables, J. and White, H.S., 2012. Identification of new epilepsy treatments: issues in preclinical methodology. Epilepsia, 53(3), pp.571-582.
- “Of those treated with AEDs, 35% become resistant to medication” - Moghim, N. and Corne, D.W., 2014. Predicting epileptic seizures in advance. PloS one, 9(6), p.e99334.
- “There is an urgent need for more effective and better-tolerated treatments to control drug-resistant seizures, as well as for innovative therapies to prevent, stop, or reverse the development of epilepsy and epilepsy-related comorbidities” - Löscher, W. and Schmidt, D., 2011. Modern antiepileptic drug development has failed to deliver: ways out of the current dilemma. Epilepsia, 52(4), pp.657-678.

Order of Work

- Emphasis on literature review findings instead of personal experiments.
- Increased the time allocated for hyper-parameter tuning due to the training times for a single model.
- Added time to allow for further developments, such as subject agnostic model training or node configuration tuning.

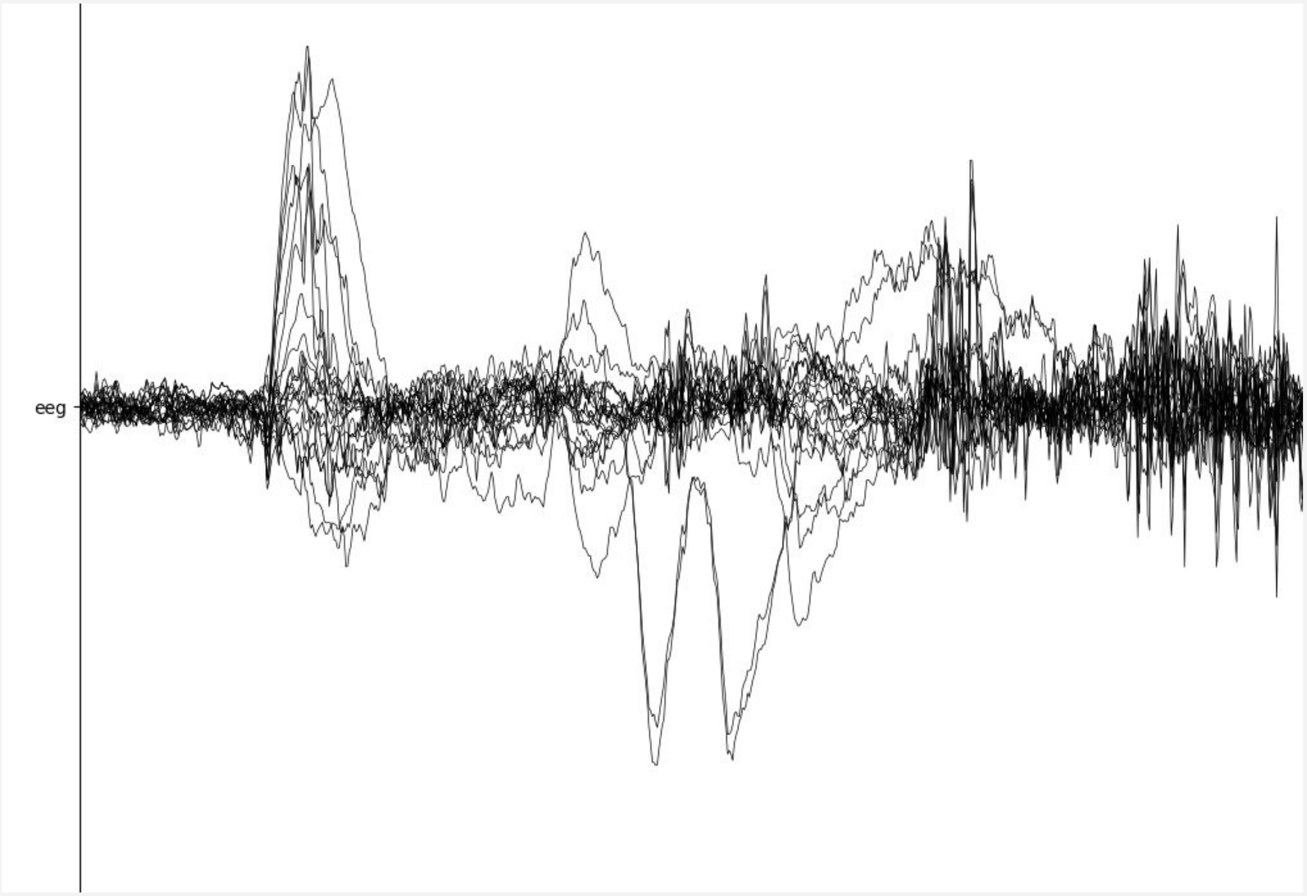


The Solution

CHB-MIT EEG

- 23 unique subjects.
- 10-20 International System for Scalp EEG.
- 17 common channels between all subjects.
- Continuous recordings with labeled seizure times.
- Open source; hosted through Google Cloud.

Boston Children's Hospital
Until every child is well

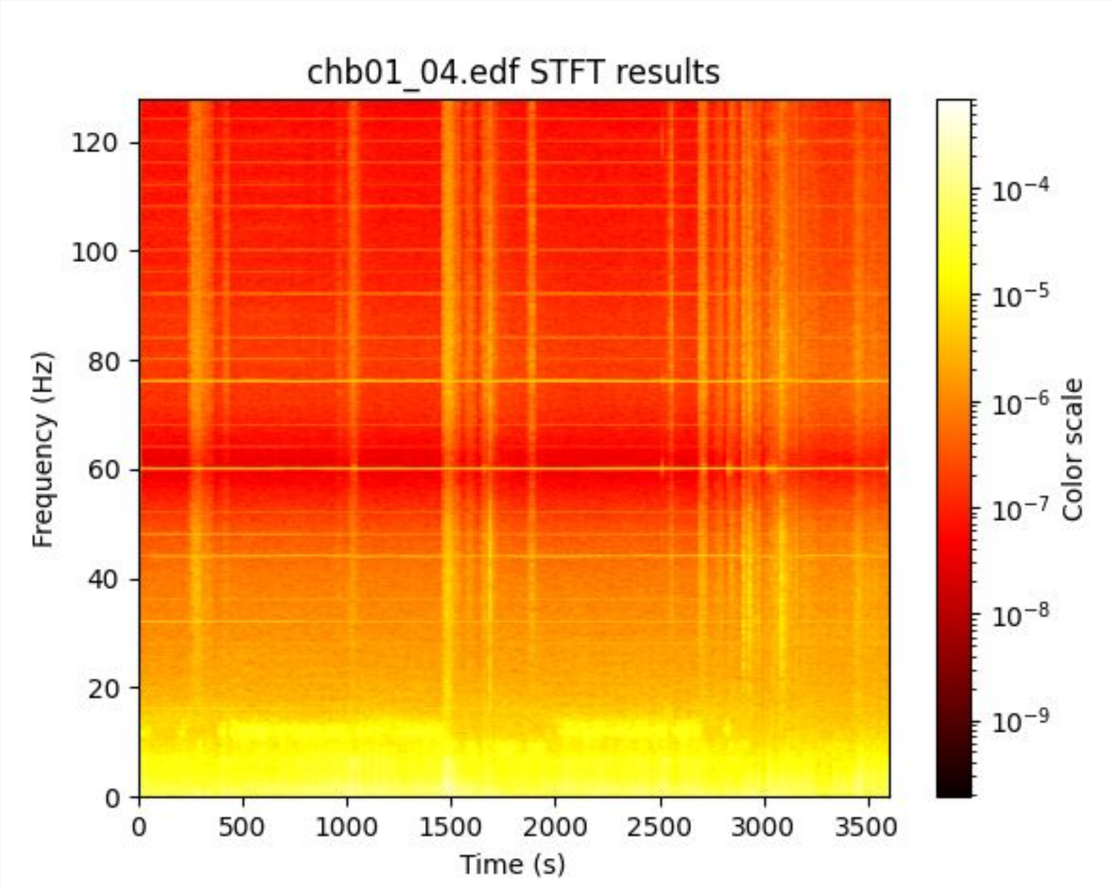


A Butterfly representation of subject 1's EEG recording.
https://github.com/N3utra1/COMP6013_Dissertation.git

Preprocessing

- Short-time Fourier transform.
- Band Pass Filter.
- Class balancing.
- Approach influenced by "Convolutional neural networks for seizure prediction and scalp electroencephalogram".

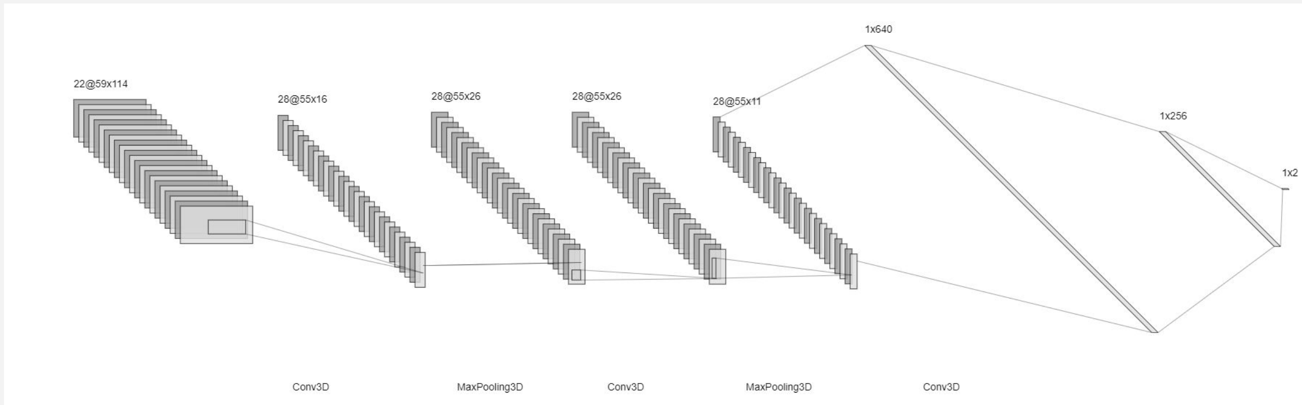
Truong, N.D., Nguyen, A.D., Kuhlmann, L., Bonyadi, M.R., Yang, J., Ippolito, S. and Kavehei, O., 2018. Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram. Neural Networks, 105, pp.104-111.



Heat map of a Short-time Fourier transform result. Seizure between 1467-1494 seconds..
https://github.com/N3utra1/COMP6013_Dissertation.git

The Convolution Neural Network

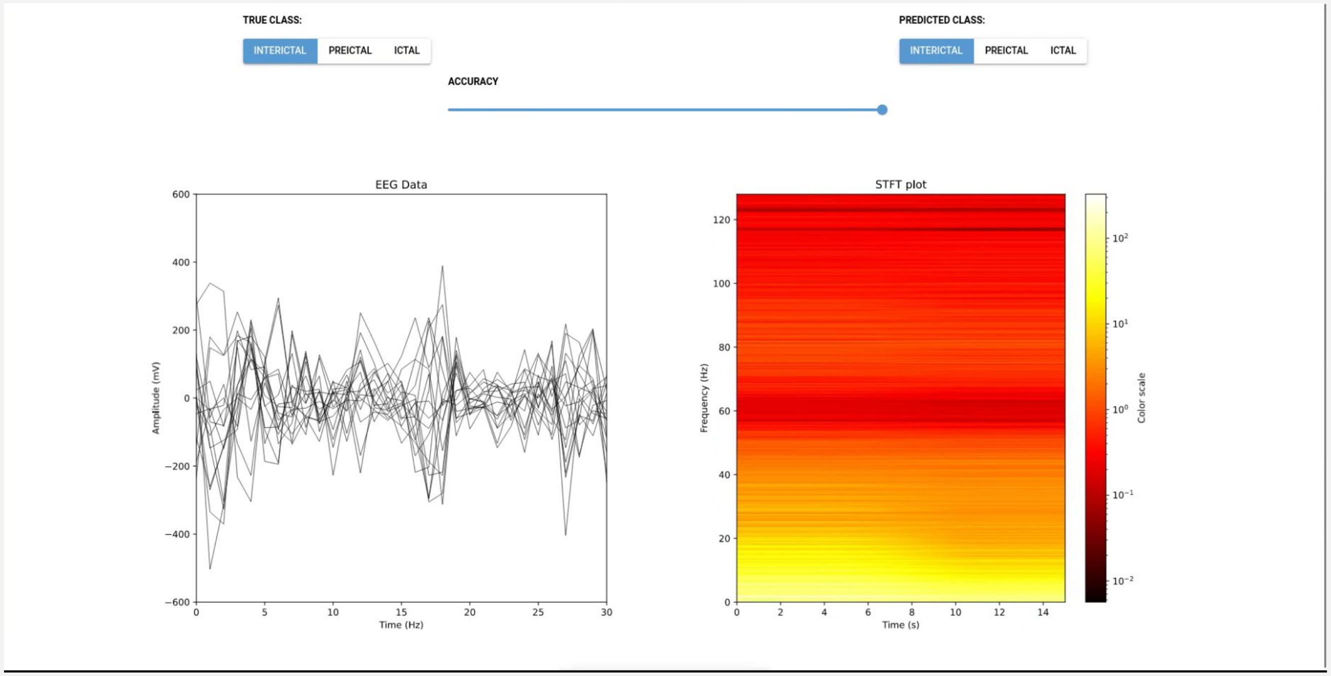
- The CNN is created using Python 3 along with the TensorFlow and Keras.
- The solution is an image recognition approach to the problem.
- CNN model architecture has been tuned, along with the hyper-parameters.



Representation of an model architecture. Created using https://alexlenail.me/NN-SVG/LeNet.html

Real Time Simulation

- Shows the preprocessing pipeline's ability to run in real-time.
- User selects an class in the top left and the model shows it's prediction in the top right.
- The plot on the left shows the last 30 seconds of raw EEG data. 256 new data points for each EEG node arrive each second.
- The right plot is the STFT spectrogram image being inputted into the ML model.



Model Results

- Models trained against a single subject's data, allowing the CNN to pick up on their characteristics.
- The models were tested against 20% of the dataset (unseen).
- 14 models were perfect models with 100% accuracy against the test dataset.
- This shows the potential this approach has for real-time preictal prediction.

Model	Training Accuracy	Training Loss
2.4.128/16.1	0.9832265377044678	0.8224268555641174
2.4.128/16.32	0.9993602503091097	0.02814496521023102
2.4.128/16.8	0.9976428672671318	0.11281578045975693
2.4.128/32.1	0.975464940071106	1.4378010034561157
2.4.128/32.32	0.9989679381251335	0.04745509265126133
2.4.128/32.8	0.9962754622101784	0.18717686411091886
2.4.128/48.1	0.9669219255447388	2.318654775619507
2.4.128/48.8	0.9958652406930923	0.2898520281258925
2.4.64/16.1	0.9799447655677795	0.6853911876678467
2.4.64/16.32	0.9992593247443438	0.024666143968849415
2.4.64/16.8	0.9969331100583076	0.08956567873635081
2.4.64/32.1	0.9619211554527283	1.725386619567871
2.4.64/32.32	0.9987660832703114	0.05414362490514908
2.4.64/32.8	0.9953247904777527	0.21542102738881397
2.4.64/48.1	0.9655154347419739	2.379462718963623
2.4.64/48.32	0.9989125896245241	0.07438817310655671
2.4.64/48.8	0.994374118745327	0.31704871635884047

Table 24: Accuracy and Loss metrics (average across epochs) during model training.

Results table taken from report.

Model key: [convolution block count].[dense layer count].[dense layer size]/[batch size].[epochs]

Wrapping up

Future Advancements

- Tuning STFT parameters.
- Further tuning of model architecture.
- Testing different EEG node configurations.
- “From a large number of recent studies, it is clear that BCI is an up-and-coming research area.” - Zabcikova, M., Koudelkova, Z., Jasek, R. and Lorenzo Navarro, J.J., 2022. Recent advances and current trends in brain-computer interface research and their applications. International Journal of Developmental Neuroscience, 82(2), pp.107-123.

Real-time preictal detection through the application of machine learning to Electroencephalogram signals.

William Riddell Final year project.
Supervisor: Kashinath Basu

Introduction

“A seizure is a burst of uncontrolled electrical activity… that causes temporary abnormalities in muscle tone or movements, behaviors, sensations or states of awareness.” (Medicine n.d.). “Persistent seizures constitute a considerable burden on healthcare resources." (Assi, Nguyen, Rihana & Sawan 2017). Therefore “An automated accurate prediction of seizures will significantly improve the quality of life of patients and reduce the burden on caregivers” (Acharya, Hagiwara & Adeli 2018). As uncontrolled electrical activity causes seizures, a method has been devised that applies image recognition through the use of machine learning to images generated from EEG data. This method can be run in real-time, outputting a prediction each second, with the ability to predict seizures up to 20 minutes before onset.

Literature

10 datasets, 6 models, and 10 preprocessing pipelines were evaluated. Results varied with a preprocessing pipeline using “Distribution of Wavelet Coherence” along with a CNN (Mirowski et al. 2009) obtaining some of the highest results.

Another approach was using a pre-processing pipeline with Short Time Fourier Transforms (Truong, Nguyen, Kuhlmann, Bonyadi, Yang, Ippolito & Kavehei 2018). This achieved respectable results of 81.4% sensitivity and 0.06 false positives/hour.

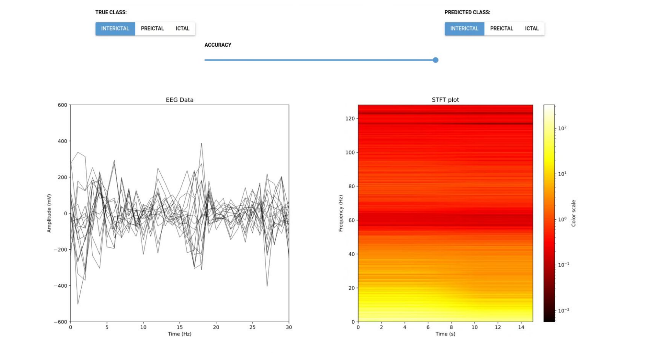
Ethical Issues

As this project is only training ML models ethical issues are very small, however, if this is embedded in a seizure prediction system the false positive rate will have to be very low; if the system alerts the user and is incorrect, it will cause a large amount of unneeded stress for the end user as they prepare for a seizure which has been incorrectly predicted.

Methodology

This project used a Short-Time Fourier-Transform which creates an image from the electrical signals in the brain, recorded with an EEG. Interference is then removed by using a notch filter. For training the machine learning models, data was synthesized by using a sliding window method to balance the dataset, however, for prediction, the images can be directly fed into a model which classifies the results.

The project was programmed in Python 3 and used TensorFlow and MNE as the main libraries.



Real Time Simulation. EEG data left, Spectrogram (STFT) image right. Model’s predicted class top right.

Results

This project has trained over 15 models which obtained 100% accuracy during testing. These models are trained against an individual, allowing the neural network to pick up on their specific characteristics. Any of these models can then be used in conjunction with the real-time simulation which has also been developed.
The real-time simulation displays the model's ability to classify the current state of the subject each second, alerting the subject if a seizure is imminent. This shows the potential this project has to greatly alleviate stress for people suffering from seizures.

Conclusions

This project was able to achieve its objectives, developing a model with an accuracy >90%, along with a preprocessing pipeline that can process the raw data in under a second. The research and development done allows for many future advancements;

- Tuning STFT window parameters directly changes the properties of the model’s input images.
- Implementing online learning, such that the model is trained against a subject at all times, further reduces false positives and inaccurate predictions.
- EEG Node configurations should also be tested, possibly showing that the same results can be obtained by using fewer EEG nodes.

With hindsight and more time, other preprocessing pipelines and ML model types would have been integrated into the solution. As the code is open source, structuring the code such that anyone could integrate the future advancements discussed above would allow for many different variable configurations to be tested against many different model types and preprocessing approaches.

References

https://drive.google.com/file/d/1VGSPdRVlAAZp02pls3qw4FANiYk2peBG/view?usp=sharing

https://github.com/N3utra1/COMP6013_Dissertation

Faculty of Technology, Design and Environment - Ethics Review Form E1

- This form should be completed jointly by the **Supervisor and Student** who is undertaking a research/major project which involves human participants.
- It is the **Supervisor** who is responsible for exercising appropriate professional judgement in this review.
- Before completing this form, please refer to the University **Code of Practice for the Ethical Standards for Research involving Human Participants**, available at http://www.brookes.ac.uk/Research/Research-ethics/ and to any guidelines provided by relevant academic or professional associations.
- Note that the ethics review process needs to fully completed and signed **before fieldwork commences**.

(i) **Project Title:**

Real-time preictal detection through the application of machine learning to Electroencephalogram signals.

(ii) **Name of Supervisor and School in which located:**

Kashinath Basu - Faculty of Technology, Design and Environment School of Engineering, Computing and Mathematics

(iii) **Name of Student and Student Number:**

William Riddell - 19066041

(iv) **Brief description of project outlining where human participants will be involved (30-50 words):**

No Human participants will be involved in the process of the project. An pre-recorded dataset will be downloaded containing EEG data against unnamed subjects. An AI will be trained off extracted features from this dataset.

		Yes	No
1.	Does the study involve participants who are unable to give informed consent (e.g. children, people with learning disabilities)?		X
2.	If the study will involve participants who are unable to give informed consent (e.g. children under the age of 18, people with learning disabilities), will you be unable to obtain permission from their parents or guardians (as appropriate)?		X
3.	Will the study require the cooperation of a gatekeeper for initial access to groups or individuals to be recruited (e.g. students, members of a self-help group, employees of a company)?		X
4.	Are there any problems with the participants' right to remain anonymous, or to have the information they give not identifiable as theirs?		x

5.	Will it be necessary for the participants to take part in the study without their knowledge/consent at the time? (e.g. covert observation of people in non-public places?)		x
6.	Will the study involve discussion of or responses to questions the participants might find sensitive? (e.g. own traumatic experiences)		x
7.	Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants?		x
8.	Will blood or tissue samples be obtained from participants?		x
9.	Is pain or more than mild discomfort likely to result from the study?		x
10.	Could the study induce psychological stress or anxiety?		x
11.	Will the study involve prolonged or repetitive testing of participants?		x
12.	Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?		x
13.	Will deception of participants be necessary during the study?		x
14.	Will the study involve NHS patients, staff, carers or premises?		x

Signed:	K. Basu	**Supervisor**
Signed:	William Riddell	**Student**
Date:	17/04/24	

What to do now:

1. If you have answered **'no'** to all the above questions:

   (a) The student must **submit** the completed and fully signed E1 form to their **Dissertation Module Leader via Moodle.**
   (b) The student must keep a copy of the E1 form which must be bound into their dissertation as an appendix.
   (c) The supervisor must keep a copy of the E1 form as they are responsible for monitoring compliance during the fieldwork.

1. If you have answered **'yes'** to **any** of the above questions:

(a) The supervisor and student must complete the TDE E2 form available at http://www.brookes.ac.uk/Research/Research-ethics/Ethics-review-forms/
(b) Note that the information in the E2 must be in **sufficient detail** for the ethical implications to be clearly identified.
(c) The signed E2 and signed E1 Form must be emailed to Bridget Durning (bdurning@brookes.ac.uk) who is the Faculty Research Ethics Officer (FREO) for review. Please allow **at least two weeks** for this review process.
(d) If/when approved the FREO will issue an E3 Ethics Approval Notice.
(e) The student must send the E1, E2 and E3 Notice **to the Dissertation Module Leader**.
(f) The student must also keep copies which must be bound into their dissertation as an appendix.
(g) The supervisor must keep a copy of documentation to monitor compliance during field work.

1. If you answered ‘yes’ to any of questions 1-13 and ‘yes’ to question 14, an application must be submitted to the appropriate NHS research ethics committee. This is an onerous and time consuming process so the supervisor should liaise early with the FREO if the student is considering this.