

TDK-dolgozat

Gercuj Bence



Pázmány Péter Catholic University
Faculty of Information Technology and Bionics



Hungarian Research Network
Institute for Computer Science and Control

TDK REPORT

Optimal feature and channel selection of motor imagery-based EEG data in brain-computer interface applications

Bence Gercuj

Molecular Bionics Engineering BSc, admission 2021/2022/01

2024. 11. 13.

Supervisors:

Csaba Márton Köllöd, PhD, research fellow; Anh Tuan Hoang, research associate;
Dr. István Ulbert, institute director; Dr. Zsolt János Viharos, senior research fellow

Abstract

Electroencephalography (EEG) is a widely used, inexpensive and non-invasive technique for measuring and recording electrical brain activity, particularly in brain-computer interfaces (BCI). This method enables BCI users with certain motor impairments to operate and control particular devices and software that assist in recovery, rehabilitation, or in many other activities. However, low signal-to-noise ratio, the need for task- and user-specific EEG layouts, and several other notable issues all present challenges when the recorded EEG data is involved in machine learning-based classification tasks. Methods in feature selection and feature extraction attempt to mitigate these issues by reducing data dimensionality. Feature extraction methods perform extraction and transformation of data so that further insight could be gained. These methods include but are not limited to statistical metrics like mean and variance, band power, Hjorth parameters, and so forth. Based on their origin, features can belong to one out of several domains. These are time domain, frequency domain, time-frequency domain, or other spatial extensions. On the other hand, feature selection is applied post-extraction to further refine the data and enhance classification accuracy. Feature selection techniques include filter methods like Minimum Redundancy Maximum Relevance, wrapper methods that incorporate classifiers, embedded methods like Lasso, and hybrid methods. The aim of this research is to provide an optimal channel and feature selection solution for motor imagery-based brain-computer interface applications. This report outlines the basic concepts of electroencephalography, feature extraction and feature engineering. Furthermore, it presents the findings on optimal feature and channel selection for binary EEG classification using convolutional neural networks obtained by the combined use of the Bionic Applications and Adaptive, Hybrid Feature Selection frameworks. Additionally, a number of features are evaluated and the potential of channel-specific feature sets is investigated. Results show that a task-specific subset of EEG channels can be selected and used while improving or preserving classification accuracy, despite heavily reducing dimensionality. Additionally, the efficiency of several feature extraction techniques are established based on their

contribution to improving classification accuracy.

Contents

Front page	i
Abstract	iii
1 Introduction	1
2 Literature review	4
2.1 Feature engineering	4
2.1.1 Feature selection methods in detail	6
2.2 Feature extraction in EEG	13
2.3 Feature and channel selection in EEG	15
3 Proposed solution	17
3.1 Solution framework	17
3.1.1 Bionic Applications	17
3.1.2 Adaptive, Hybrid Feature Selection	18
3.1.3 MNE-Python	21
3.1.4 MNE-Features	21
3.2 Methodology	22
3.2.1 Recorded data	22
3.2.2 Data preprocessing, feature extraction	22
3.2.3 Feature selection	24
3.2.4 Analysis of intermediate results	25
3.2.5 Evaluation of channel selection	26
3.2.6 Evaluation of feature selection	27
4 Results	29
4.1 Channel Selection	29
4.1.1 Pilot B	29
4.1.2 Pilot C	32

4.2	Feature selection	35
4.2.1	Pilot B	35
4.2.2	Pilot C	38
5	Conclusion and future research	42
6	Appendix	44
	Literature	75

Chapter 1

Introduction

Our brains are estimated to have up to 10^{10} neurons, each connected to approximately 10^4 other neurons. [1] Through complex biological mechanisms, these neurons exhibit electrical activity. This electrical activity can be presented as a dipole model for each individual neuron, as seen in Figure 1.1. The current flowing through the dipole produces an electrical and a magnetic field. When neural activity occurs locally en masse, the summated electrical and magnetic fields can be measured by specialized hardware. The non-invasive measurement of magnetic activity from the scalp is done via magnetoencephalography (MEG). Depending on the proximity to the neurons, and by extension the degree of invasiveness, recording the electric activity is done via either electroencephalography (EEG) or electrocorticography (ECoG). ECoG measurement is conducted from the surface of the brain, greatly increasing spatial resolution and signal-to-noise ratio, albeit requiring an invasive operation for electrode placement. EEG, on the other hand, requires no surgical procedure, sacrificing data quality in the process.

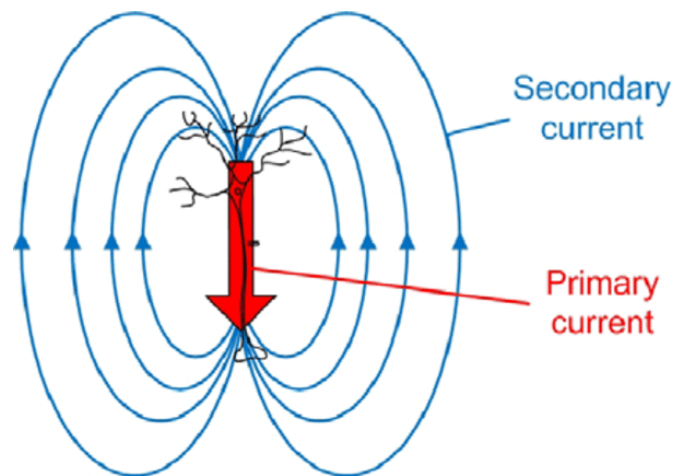


Figure 1.1: Current dipole model of a neuron by Becker [2]

Since its inception in the early 20th century, electroencephalography has been used for measuring and recording brain activity from the scalp. It sees major use in clinical applications to this day, such as sleep observation, epilepsy monitoring, intensive care, and so forth. While other methods, such as MEG, provide increased spatial resolution compared to EEG, its low cost and relatively simple deployment makes it a widely used tool in several other fields of application as well. One of these fields is brain-computer interfaces (BCI), particularly motor imagery-based BCI designed to aid in the rehabilitation of or improving quality of life for patients suffering from certain motor disabilities. Using a combination of EEG and machine learning, BCIs enable individuals to control external devices, like intelligent prosthetics, that further enhance rehabilitation [3], [4].

However, commonly used EEG layouts can produce high volumes of data that contain irrelevant or redundant segments regarding the task at hand. Additionally, the growth of data dimensions results in either greater data size and complexity or increased sparsity. This is usually referred to as the curse of dimensionality, a concept introduced by mathematician Richard Bellman [5]. In machine learning, all of these factors reduce classifier performance by decreasing model accuracy and increasing training time.

Several dimensionality reduction techniques exist that attempt to address the issues mentioned above. In machine learning, these can be categorized into two groups: feature extraction and feature selection. Feature extraction is a group of methods that transforms the original data, creating a new set of features in the process. Feature selection methods, in contrast, do not transform the input data. Instead, the original set of features is filtered, preserving relevant features and removing unnecessary ones [6].

The solution presented in this report is the combined use of the Bionic Applications (BA) BCI framework [7]–[9] and the Adaptive, Hybrid Feature Selection algorithm (AHFS) [10] to establish an optimal set of channels and features for certain imaginary mental tasks including motor imagery (MI). Data is obtained by live recording of two individuals living with tetraplegia. This database is then processed offline. The BA framework enables the complete pipeline processing of EEG data, including Fully Automated Statistical Thresholding for EEG artifact Rejection (FASTER) [11] and feature extraction. Following these steps, the AHFS algorithm performs feature selection on the generated data. The results are processed, then a set of channels are evaluated using the EEGNet Fusion classifier [12] included in the BA framework. The established sets of channels were able to preserve accuracy at 69% in one case and increase from 58.7% to 60.7% in the other, while significantly lowering data dimensionality by eliminating

86% of channels. Additionally, a subset of features are assessed using a shallow neural network. In the first case, a minor increase from 58.6% to 58.8% accuracy compared to the complete dataset was achieved, however with a 91% feature count reduction. In the second case, the accuracy increase was much notable, from 60.1% to 76.7% with an 85% column-wise dimensionality reduction.

In summary, the abundant amounts of data in EEG applications increase the need for dimensionality and complexity reduction. The aim of this research is to achieve this task by using feature extraction and selection. By removing irrelevant and redundant measurements, the speed and accuracy by which machine learning-based brain-machine interfaces can be trained and used increase. This results in enhanced quality of life for patients suffering from certain motor dysfunctions.

The report aims to provide the necessary details for understanding the results. In Chapter 2, the relevant literature on feature engineering, generally and specifically for EEG, along with state-of-the-art solutions is shown. In Chapter 3, the proposed solution and methodology is presented. In Chapter 4, the results obtained are examined and compared with findings in literature. Finally, in Chapter 5, the conclusion of the report is provided.

Chapter 2

Literature review

This chapter provides a review of the literature relevant to this topic. In Section 1, feature extraction and selection as a concept is presented with existing applications and solutions. In Section 2, feature extraction and specific features used in EEG is shown. Lastly, in Section 3, feature and channel selection in the context of EEG is discussed.

2.1. Feature engineering

Feature engineering is the process of preparing and transforming data in machine learning applications so that classification algorithms can work with and produce improved metrics, such as decreased loss or increased accuracy. While the terms of feature selection, feature extraction and feature engineering define different methods, feature selection and feature extraction can be recognized as processes within feature engineering, thus these are often mixed and used interchangeably [13]. In a more concise manner, these two methods are considered dimensionality reduction techniques.

Feature selection algorithms preserve the original dataset by producing a subset of features that enhances classifier performance. These methods can be categorized as filter, wrapper, embedded, and hybrid methods. Filter methods are mainly statistical functions that perform selection prior to classification, like chi-square. Wrapper methods, on the other hand, include a learning algorithm in the selection process, thereby taking a more active role in the classification process. Such algorithm can be the Sequential Forward Selection that begins with an empty subset and gradually increases its size. Embedded methods work by integrating feature selection within the classifier itself. An example is a linear classifier with Lasso regularization [14]. Finally, hybrid methods combine different methods, such as combining filter and wrapper methods.

Feature extraction algorithms, in contrast with feature selection methods, transform the input data to generate new features in such a manner that provides new insights for the classifier to exploit and thus improve [15]. While not all extraction techniques reduce the dimensionality directly, their combined use with feature selection these two methods serve as powerful tools.

In 2022, Pudjihartono et al. [16] reviewed the use of feature selection in disease risk prediction as a tool for removing redundant and negligible-impact single nucleotide polymorphisms. Amongst the more advanced filter methods, the Mutual Information Feature Selection [17], Minimum Redundancy Maximum Relevance [18], Conditional Mutual Information Maximization [19], Fast Correlation Based Filter [20] were noted as algorithms that take into account the redundancy between selected and candidate features. Exhaustive techniques included BOOST [21], FastEpistasis [22] and TEAM [23] that focused on making pairwise comparisons, albeit without taking redundancy into account. Relief-based methods such as Relief [24], Relief-F [25], TURF [26], SURF [27], SURF* [28], MultiSURF [29] and MultiSURF* [30] were highlighted as valuable algorithms that score features based on the ability of a feature to make a difference between similar samples that are of different class. The authors note for filter methods that while the independence from the learning algorithm means a more generalized approach, resulting in less overfitting, the potential to improve accuracy by fine-tuning to the specific classifier is lost.

For wrapper methods, an exhaustive search strategy would prove to be too expensive in terms of computational cost and time. Given this aspect, an emphasis is placed on implementing heuristic search strategies that find an optimal subset of features within the realm of computational feasibility. These algorithms can be, but are not limited to, randomized [31], sequential [32], genetic [33], ant colony [34], and so forth.

On the other hand, embedded methods are techniques that implement regularization, such as decision tree-based methods and algorithms, are considered to be in the field of embedded methods. Examples for the former include decision trees, random forest, gradient boosting and Lasso or elastic net-regularized linear classifiers like Support Vector Machine and logistic regression for the latter. In general, regularization works by reducing unnecessary feature coefficients, thereby reducing their impact in classification.

According to the no-free-lunch theorem, there is no universally optimal feature selection technique. Instead, hybrid methods should be considered that offer to mitigate the disadvantage of each method with the advantage of the other. For instance, combining

filter and wrapper methods would enable the use of a wrapper method on a large dataset by applying a filter method beforehand, thereby reducing its size to make per-iteration classification computationally viable.

In the field of stock market forecasting with machine learning, Htun et al. [35] surveyed the use of feature selection and feature extraction in 2023. From the field of filter methods, the authors highlighted correlation- and distance-based measures such as Pearson Correlation Coefficient, Spearman Rank Correlation, Linear Correlation, Rank Correlation, and the Euclidean and Manhattan measures. Additionally, the Relief family of algorithms was highlighted. For wrapper methods, Recursive Feature Elimination was noted. Finally, amongst the embedded methods mentioned were Random Forest, Support Vector Machine and Logistic Regression. Outside of the three main field of methods, information theory-based algorithms were discussed. These were the Forward Selection Minimal Redundancy Maximal Relevance [36] and Conditional Mutual Information Maximization [37] solutions. Concerning feature extraction, the authors highlighted the usage of Principal Component Analysis [38] and Autoencoders [39] alongside Linear Discriminant Analysis [40].

2.1.1 Feature selection methods in detail

Forward Feature Selection Algorithm

Amiri et al. [41] showcased the Forward Feature Selection Algorithm, which is based on mutual information. This method is initialized with an empty subset of selected feature indices, the set of all feature indices and the calculated vector of feature-target mutual information. The algorithm starts by selecting the feature with the highest mutual information with the target, or class, vector. Then, the mutual information value is nullified and the feature is added to the subset of selected features. The process repeats until the desired size of the selected features-subset is reached.

0. Initialization: feature index set F , feature-target mutual information vector M , empty selected features-subset S , desired number of selected features k
1. Select feature $f \in F$ such that $f \equiv \text{argmax}(M)$
2. Let subset $S = S + \{f\}$, value $M[f] = NULL$
3. If $\text{size}(S) \geq k$ then continue, else repeat step 1
4. Return: S

Mutual Information-based Feature Selection

Battiti [42] investigated the use of mutual information for feature selection in a supervised neural network classification application. The author argued that contemporarily available methods, such as Principal Component Analysis, do not stay invariant during a possible transformation of the variables. Additionally, linear dependence-based measures, such as correlation, cannot take into account arbitrary relationship between variables. As a response to these raised issues, the author highlighted mutual information and argued that these problems do not apply on it.

The algorithm itself works by initializing an empty subset of the selected feature indices, the set of feature indices, a penalty modifier for redundancy, the calculated vector of feature-target mutual information, and the calculated upper triangle matrix of feature-feature mutual information. The process begins by selecting the feature with the highest feature-target mutual information value. Then, all remaining candidates are scored by subtracting the sum of their mutual information with all selected features weighted by the penalty term from their mutual information with the target variable. The feature with the highest score is selected, and the process repeats until the previously defined number of selected features is not met.

0. Initialization: feature index set F , feature-target mutual information vector MT , feature-feature mutual information matrix MF , empty selected features-subset S , desired number of selected features k , redundancy penalty modifier β
1. Select feature $f \in F$ such that $f \equiv \text{argmax}(MT)$
2. Let subset $S = S + \{f\}$
3. For each candidate feature $f_{c_i} \in F$ and $f_{c_i} \notin S$ calculate and assign redundancy subtracted from value $MT[f_{c_i}]$
4. Select candidate feature f_{c_i} such that $f_{c_i} \equiv \text{argmax}(f_c)$
5. Let subset $S = S + \{f_{c_i}\}$
6. If $\text{size}(S) < k$ then continue, else repeat step 3
7. Return: S

Modified Mutual Information-based Feature Selection

Amiri et al. [41] argued that the feature selection solution presented by Battiti does not take into account the number of selected features, thus greatly diminishing the value of the candidate feature-target mutual information in higher dimensions. As a solution, the authors modified the scoring step by taking the mean of the candidate feature's mutual information with all selected features instead of its sum. Otherwise, the algorithm remained the same.

0. Initialization: feature index set F , feature-target mutual information vector MT , feature-feature mutual information matrix MF , empty selected features-subset S , desired number of selected features k , redundancy penalty modifier β
1. Select feature $f \in F$ such that $f \equiv \text{argmax}(MT)$
2. Let subset $S = S + \{f\}$
3. For each candidate feature $f_{c_i} \in F$ and $f_{c_i} \notin S$ calculate and assign redundancy value subtracted from value $MT[f_{c_i}]$
4. Select candidate feature f_{c_i} such that $f_{c_i} \equiv \text{argmax}(f_c)$
5. Let subset $S = S + \{f_{c_i}\}$
6. If $\text{size}(S) < k$ then continue, else repeat step 3
7. Return: S

Linear Correlation-based Feature Selection

Amiri et al. [41] proposed the replacement of the mutual information measure in the Modified Mutual Information-based Feature Selection algorithm with linear correlation. While no change in the algorithm occurs, and the potential inferiority of correlation as a measure as noted by Battiti is a disadvantage, the authors argued that correlation has a lower computational complexity than mutual information.

0. Initialization: feature index set F , feature-target correlation vector CT , feature-feature correlation matrix CF , empty selected features-subset S , desired number of selected features k , redundancy penalty modifier β
1. Select feature $f \in F$ such that $f \equiv \text{argmax}(CT)$

2. Let subset $S = S + \{f\}$
3. For each candidate feature $f_{c_i} \in F$ and $f_{c_i} \notin S$ calculate and assign redundancy value subtracted from value $CT[f_{c_i}]$
4. Select candidate feature f_{c_i} such that $f_{c_i} \equiv \text{argmax}(f_c)$
5. Let subset $S = S + \{f_{c_i}\}$
6. If $\text{size}(S) < k$ then continue, else repeat step 3
7. Return: S

Fast Correlation Based Filter

Yu et al. [20] proposed a fast correlation-based filter method that aims to address a number of issues with other contemporary feature selection methods, comparing it with Relief and CFS, a subset search algorithm. The authors argued that while Relief is a scalable algorithm with linear time complexity, it does not address the problems of redundancy between selected features. On the other hand, CFS was found to be having too high degrees of computational complexity, albeit effective with handling feature redundancy.

The authors noted that for an underlying measure, symmetry as a mathematical property is desired so that redundancy between features can be quantified, alongside normalization. In this respect, symmetrical uncertainty as an information theory-based measure was selected.

The algorithm is initialized with the set of feature indices, an empty subset of selected feature indices, a filtering threshold parameter, the calculated vector of feature-target symmetrical uncertainty, and the calculated upper triangle matrix of feature-feature symmetrical uncertainty. The process begins by filtering out all features with feature-target symmetrical uncertainty below the threshold, so that only the high enough symmetrical uncertainty values remain. Next, the features are ordered in a decreasing manner based on their feature-target symmetrical uncertainty values. The feature with the highest of such value is defined to be the predominant one. The next feature in the list is examined by comparing the two feature's feature-feature symmetrical uncertainty, and the candidate's feature-target symmetrical uncertainty. If the former is higher than the latter, indicating high redundancy, then the candidate feature is removed and the next in line is examined. Once the list ends, the role of predominance is given to the next feature in the list. Features that were previously predominant are not examined anymore. This process repeats until all available features were given the predominant role.

0. Initialization: feature index set F , feature-target symmetrical uncertainty vector SUT , feature-feature symmetrical uncertainty matrix SUF , empty selected features-subset S , filtering threshold parameter δ
1. Let subset $F = F - \{argwhere(SUT \leq \delta)\}$ and $S = S - \{argwhere(SUT \leq \delta)\}$
2. Order subset F and SUT by SUT descending
3. Let predominant feature $f_d \equiv F[0]$
4. For each candidate feature $f_i \in F$ where $index(f_i) > index(f_d)$ let subset $F = F - \{argwhere(SUT[f_i] < SUF[f_d, f_i])\}$
5. If feature $F[index(f_d)+1]$ is not null then let predominant feature $f_d \equiv F[index(f_d)+1]$ and repeat step 4, else continue
6. Let subset $S = F$
7. Return: S

Fast Correlation Based Filter / K

Senliol et al. [43] proposed a modified version of the Fast Correlation Based Filter method. The primary issue that the authors raise with the Fast Correlation Based Filter algorithm is that it may remove too many features given highly correlated inputs. This is due to the primacy of the predominant feature and the lack of a halting mechanism for selecting a specifically desired number of features. Senliol et al. introduced temporary predominance that only enables the predominant feature to remove one feature and then make it transfer its predominance. Additionally, feature elimination occurs backwards in the sorted feature-target symmetrical uncertainty vector, thereby first removing the features with low class correlation first. The authors claim that their method outperformed the original Fast Correlation Based Filter algorithm and had comparable performance to the Minimum Redundancy Maximum Relevance feature selection technique.

0. Initialization: feature index set F , feature-target symmetrical uncertainty vector SUT , feature-feature symmetrical uncertainty matrix SUF , empty selected features-subset S , desired number of selected features k , filtering threshold parameter δ
1. Let subset $F = F - \{argwhere(SUT \leq \delta)\}$ and $S = S - \{argwhere(SUT \leq \delta)\}$

2. Order subset F and SUT by SUT descending
3. Let predominant feature $f_d \equiv F[0]$
4. For each candidate feature $f_i \in F$ where $index(f_i) > index(f_d)$ let removed feature $f_r = argwhere(SUT[f_i] < SUT[f_d, f_i])[-1]$
5. Let subset $F = F - \{f_r\}$
6. If feature $F[index(f_d) + 1]$ is not null and $size(S) < k$ then let predominant feature $f_d \equiv F[index(f_d) + 1]$ and repeat step 4, else continue
7. Let subset $S = F$
8. Return: S

Fast Correlation Based Filter in Pieces

Egea et al. [44] suggested another modification to the Fast Correlation Based Filter, and by extension to the K variation. Recognizing both of their merits and shortcomings, the Fast Correlation Based Filter in Pieces attempts to lower the computational complexity of the K variant while improving the accuracy of the original method. It aims to do as such by dividing the feature space into pre-defined pieces after the features were sorted in a descending order according to their feature-target symmetrical uncertainty values. By recognizing the neighboring features' highly similar symmetrical uncertainty values, these features are grouped together and their redundancy to each other are measured. This is in contrary to the other two implementations where one feature's redundancy is quantified with every other feature. The outcomes presented show that the parameter handling the number of pieces plays a great role in the outcome, with lower piece counts resulting in faster albeit less accurate solutions and higher piece counts in the opposite. Thus, the challenge lies in finding the optimal parameter for every dataset.

0. Initialization: feature index set F , feature-target symmetrical uncertainty vector SUT , empty vicinity symmetrical uncertainty vector SUV , empty feature score vector FS , empty selected features-subset S , number of pieces p , desired number of selected features k , filtering threshold parameter δ
1. Let subset $F = F - \{argwhere(SUT \leq \delta)\}$ and $S = S - \{argwhere(SUT \leq \delta)\}$
2. Order subset F and SUT by SUT descending

3. For each candidate feature $f_i \in F$ let feature vicinity $f_v \equiv \text{transpose}(\text{reshape}(f_i, \text{shape} = \text{pieces} \times \frac{\text{samples}}{\text{pieces}}))$
4. Let $SU_{f_v} \equiv SU(f_v)$
5. Let vector $SUV = SUV + \{SU_{f_v}\}$
6. Order vector SUV by descending
7. Let feature score vector $size(FS) \equiv size(F)$
8. For index i in $size(FS)$, let $fs_i = \text{argwhere}(i \equiv F) + \text{argwhere}(i \equiv SUV)$
9. Order vector FS descending and let subset $S \equiv FS[:k]$
10. Return: S

Minimum Redundancy Maximum Relevance

Ding et al. [18] introduced the information theory-based Minimum Redundancy Maximum Relevance feature selection algorithm, originally for dimensionality reduction in microarray gene expression data. The authors recognized the deficiencies of then contemporary filter methods regarding handling redundancy and proposed a mutual information-based technique. In this spirit, they defined minimum redundancy and maximum relevance as important measures in this algorithm. Minimum redundancy refers to the concept of minimizing the sum of all mutual information values between a candidate feature and the already selected features divided by square of the selected feature subset size. In parallel, maximum relevance refers to maximizing the feature-target mutual information. For the final scoring of candidate features, either minimum redundancy can be subtracted from maximum relevance, or maximum relevance divided by minimum redundancy. While in theory any correlation measure could be applied, the authors chose mutual information so that the algorithm is able to quantify covering the same amount of information with a smaller subset of selected features in contrast with using the complete dataset.

The method begins with the set of feature indices, the empty subset of selected feature indices, and the calculated vector for feature-target mutual information and the upper triangle matrix of feature-feature mutual information. The process starts by selecting the feature with the highest feature-target mutual information as the first feature. Then, all candidate features are evaluated by calculating their maximum relevance and minimum redundancy in the manner described previously. Next, the score is established by either

method and the candidate feature with the highest score is added to the subset of selected features.

0. Initialization: feature index set F , feature-target mutual information vector MT , feature-feature mutual information matrix MF , empty selected features-subset S , desired number of selected features k
1. Select feature $f \in F$ such that $f \equiv \text{argmax}(MT)$
2. Let subset $S = S + \{f\}$
3. For each $f_{c_i} \in F$ and $f_{c_i} \notin S$ calculate and assign minimum redundancy subtracted from maximum relevance
4. Select candidate feature f_{c_i} such that $f_{c_i} \equiv \text{argmax}(f_c)$
5. Let subset $S = S + \{f_{c_i}\}$
6. If $\text{size}(S) \geq k$ then continue, else repeat step 3
7. Return: S

2.2. Feature extraction in EEG

Feature extraction has been present in the study of EEG for a long time. In 1977, for instance, Bodenstein et al. proposed an adaptive segmentation method that extracted the power spectrum and temporal information [45]. Since then, advances in all fields involved in EEG led to more advanced methods.

Rashid et al. [46] reviewed the state of EEG-based BCI systems and their research in 2020. They noted the issue of identifying the most suitable feature extraction technique for all EEG paradigms, like motor imagery, and steady-state evoked potential. The authors also reviewed noteworthy feature extraction methods. In the time domain, Autoregressive Modeling, Higuchi Fractal Dimension, Log Energy Entropy and Principal Component Analysis were mentioned. Frequency domain features included Fast-Fourier Transform and Welch's method based Power Spectral Density and Band Power. Noting the absence of temporal characteristics in spectral features and spectral information in time domain features, the authors assumed that time-frequency domain features can mitigate the lack of information found in the individual domains. Short-Time Fourier Transform, Continuous- and Discrete Wavelet Transform were of important time-frequency

domain features. Additionally, Common Spatial Pattern was found to be a powerful and widespread feature extraction method.

Wang et al. [47] investigated feature extraction techniques in EEG-based emotion classification applications in 2021. The most common features were found to be mostly frequency or time-frequency domain in origin, such as Short-Time Fourier Transform, Power Spectral Density, Wavelet Transform, Differential Entropy, Sample Entropy, Wavelet Entropy, Empirical Mode Decomposition, and basic statistical features. In the time domain, Hjorth parameters and Fractal Dimension were among the most common. Additionally, in the frequency and time-frequency domains, Higher-Order Spectrum, Higher-Order Crossover, Wavelet Packet Transform and Hilbert-Huang Transform were noted.

A. Singh et al. [48] introduced the spatio-temporal feature domain when reviewing motor imagery-based BCIs in 2021 with the sample covariance matrices extraction method, utilizing both spatial and spectral information. The authors also argued that with the arrival of deep neural networks, like EEGNet, the need for manual feature extraction had mostly subsided.

Mohdiwale et al. [49] found that the beta and gamma EEG frequency sub-bands are of importance for motor imagery applications in 2021. They also emphasized the importance of both basic and higher-order statistic measures.

Lastly, in 2023 A. K. Singh et al. [50] reviewed trends in EEG feature extraction methods and largely re-iterated over the findings mentioned previously. They grouped the extraction techniques into two groups; one-dimensional features and multi-dimensional features. In the one-dimensional time frequency domain Autoregressive Modeling, Higuchi Fractal Dimension, basic statistical features such as mean and variance, Detrended Fluctuation Analysis were mentioned. In the spectral domain, Fourier Transform, Power Spectral Density, Band Power and Hilbert-Huang Transform were highlighted. The final one-dimensional category was the decomposition domain. This domain saw the inclusion of Wavelet Transform and Empirical Mode Decomposition. Finally, in the multi-dimensional category, the Short-Time Fourier Transform in the time-frequency domain and the Common Spatial Pattern with its variations in the spatial domain were noted. The authors also expressed the advantages and disadvantages of each investigated method. The Higuchi Fractal Dimension was found to be an efficient method for non-linear, non-stationary data extraction, albeit lacking optimized accuracy. Basic statistical features, like mean and variance, were easy to implement and inexpensive to compute, however they might not be suitable for physiological signals. Power Spectral Density pro-

vided versatile features for increased accuracy with little computational expense, though it had difficulties with non-stationary signals. Lastly, Band Power is able to provide data for high accuracy classification, albeit it is computationally extensive.

2.3. Feature and channel selection in EEG

In a previously mentioned article by A. Singh et al. [48], the authors defined the main objective of channel selection as finding the optimal channel set for a certain BCI task. This set then provides an improved classification accuracy and reduced computation time. As the task of feature selection is similarly finding an optimal subset of the dataset, the authors argue that channel selection methods are derived from feature selection techniques. They highlight several algorithms used in feature selection of EEG datasets, such as Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis, Kernel Principal Component Analysis, Multilayer Autoencoders, and manifold learning-based methods.

Mohdiwale et al. [49] used a stochastic method called Harmony Search-based Feature Selection algorithm for motor imagery-based EEG signal classification. The authors used the BNCI Horizon 2020 dataset.

Guttmann-Flury et al. [51] established a common Core Channel Selection set for motor imagery, motor execution, steady state visual evoked potential and P300 paradigms. The results were obtained by reviewing several studies and applying Cohen’s d statistical measure.

Kabir et al. [52] decomposed motor imagery-based signals into four signal bands, on which the Common Spatial Pattern extraction method was applied. Then the generated features were put through selection via Correlation-based Feature Selection, Minimum Redundancy Maximum Relevance, and multi-Subspace Randomization Collaboration-based Feature Selection. The latter method provided the best classification accuracy for benchmark datasets. The authors used the BCI Competition III dataset.

Sun et al. [53] noted that EEG data contains complex channel correlations instead of simple proximities, thus more advanced selection criteria is necessary based on correlation and task characteristics amongst other parameters. The authors also noted the use of Principal Component Analysis, Independent Component Analysis and its combination with Wavelet Transform, and Autoencoders as dimensionality reduction techniques. When reviewing literature, Sun et al. highlighted the use of the Bispectrum and Euclidean Distance-based Channel Selection algorithm that significantly increased accuracy

in 18 out of 35 subjects. Additionally, the authors noted a voting-based channel selection algorithm that also significantly improved classification accuracy and reduced computational complexity.

In their recent article, Kabir et al. [54] applied the Relief-F feature selection algorithm following Common Spatial Pattern extraction on specific frequency subbands. The authors once again utilized the BCI Competition III dataset.

Chapter 3

Proposed solution

The purpose of this chapter is to present the proposed methodology and solution in detail. In Section 1, the Bionic Applications framework, the Adaptive, Hybrid Feature Selection algorithm, and the related tools are presented. In Section 2, a thorough examination of the methodology is shown.

3.1. Solution framework

3.1.1 Bionic Applications

The Bionic Applications framework [7]–[9] is an electroencephalographic and electromyographic signal processing and classification toolbox primarily designed for brain-computer interface application in the Cybathlon ETH Zürich competition [55], [56]. The Python-based solution contains database handling, feature extraction methods, artifact filtering, and various learning algorithms. These tools are all integrated in such a manner that enables the use of pipelining, effectively requiring no user input after setting up initial parameters.

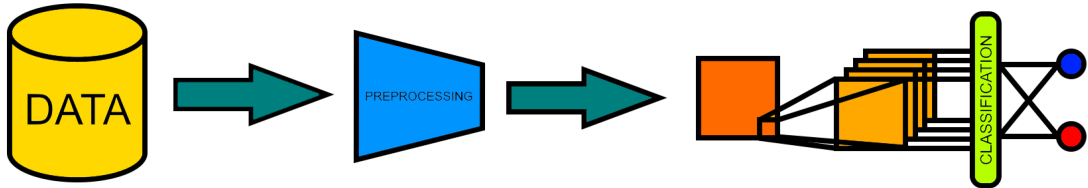


Figure 3.1: An example pipeline in the Bionic Applications framework.

3.1.2 Adaptive, Hybrid Feature Selection

Proposed by Viharos et al. [10], the Adaptive, Hybrid Feature Selection algorithm is a hybrid technique that encompasses filter and wrapper methods in order to produce an optimal subset of features. This enables the mitigation of both domain’s disadvantages, that is the reduction of input data dimensionality for the wrapper method and the fine tuning of the selected feature set to a specific classifier. The algorithm was implemented in the MATLAB environment containing the feature selection measures included in the original article, namely the Forward Feature Selection Algorithm, Modified Mutual Information-based Feature Selection, Linear Correlation-based Feature Selection, Fast Correlation Based Filter with its K variant, Minimum Redundancy Maximum Relevance and its alternative method, Joint Mutual Information Maximization with its normalized version, and an Euclidean distance-based feature selection solution. The wrapper method is based on Sequential Feature Selection with a shallow artificial neural network using Levenberg-Marquardt optimization [57]. The implementation itself is modular, allowing an indefinite number of filter measures can be included up to the point of computational feasibility. Similarly, the learning algorithm part of the wrapper method can be changed.

The algorithm is initialized with the input dataset and the desired number of features to be selected. In a parallelized approach, all filter methods process the dataset and individually produce one candidate for evaluation. After the filter methods are concluded, the candidate features are evaluated one by one combined with the already selected features. The feature that produced the lowest classification error is added to the subset of selected features. If the subset size matches the previously determined feature count, then the algorithm terminates, otherwise the process is repeated. A general overview is given in Figure 3.2.

Aside from its feature selection profile, the Adaptive Hybrid Feature Selection method provides several other insights. The inclusion of several filter algorithms provides a direct comparison between them. This enables the measurement of each method’s goodness in a sequential setting. Additionally, the sequential search strategy in the wrapper part provides a metric of feature importance, which would otherwise not be possible or difficult to establish with a stochastic search strategy or subgroup selection method.

Algorithm improvement, optimization

The Adaptive, Hybrid Feature Selection algorithm originally was implemented in the MATLAB environment. While a decent software package for high dimensionality nu-

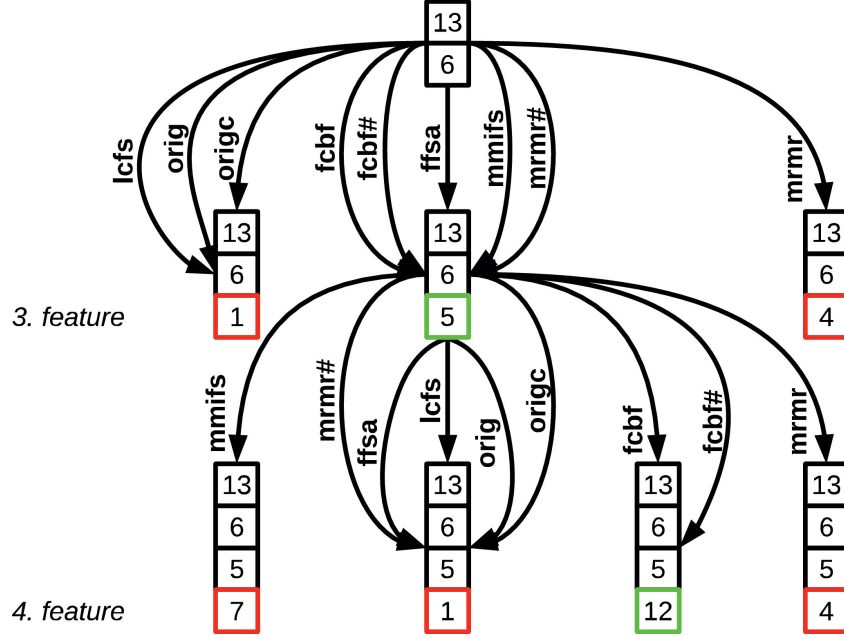


Figure 3.2: A general overview of one Adaptive, Hybrid Feature Selection iteration. Figure by Viharos et al. [10]

merical operations, several drawbacks exist. Amongst others, these are its licensing requirement and resulting low accessibility for researchers, lack of features such as GPU acceleration of certain optimizers, and relatively low popularity in the field of machine learning. To combat the issues raised, a decision was made to transfer the algorithm into the Python environment. This transfer is nearing its completion at the time of writing.

In its current version, both in the Python and MATLAB implementations, the most computationally intensive section of the solution is the Levenberg-Marquardt optimization-based shallow neural network. The goal of this iterative optimization method is to find the local minimum of a multivariate function, expressed as the sum of squares of non-linear functions [57]. To decrease computational complexity, the Hessian matrix is approximated by the Jacobian matrix. In the context of the learning algorithm, the size of this matrix depends on the input data dimensionality, with higher input size resulting in bigger matrices.

During algorithm usage it was established that running the software on CPU resulted in noticeable execution time increase when performing feature selection on high dimensional data. While initial iterations handle a low number of features, the bigger the dataset, the sooner a major slowdown occurs. Given that GPU-s are more suitable for large matrix calculations, we decided to implement a GPU version of this classifier. This was done using the PyTorch module [58]. The comparison of CPU and GPU calculation

time as a function of the Jacobi matrix size is presented in Figure 3.3. The matrix size is determined by the number of samples and free parameters in the neural network:

$$Jacobi\ size = s_{in} \cdot n_{layer_1} + \left(\sum_{i=2}^{s_{layers}-1} n_{layer_i} \cdot n_{layer_{i+1}} \right) + b, \quad (3.1)$$

where s_{in} is the input size, n_{layer_i} is the number of neurons in the i th layer, s_{layers} is the number of layers including the input and output layers, and b is the bias of the neural network.

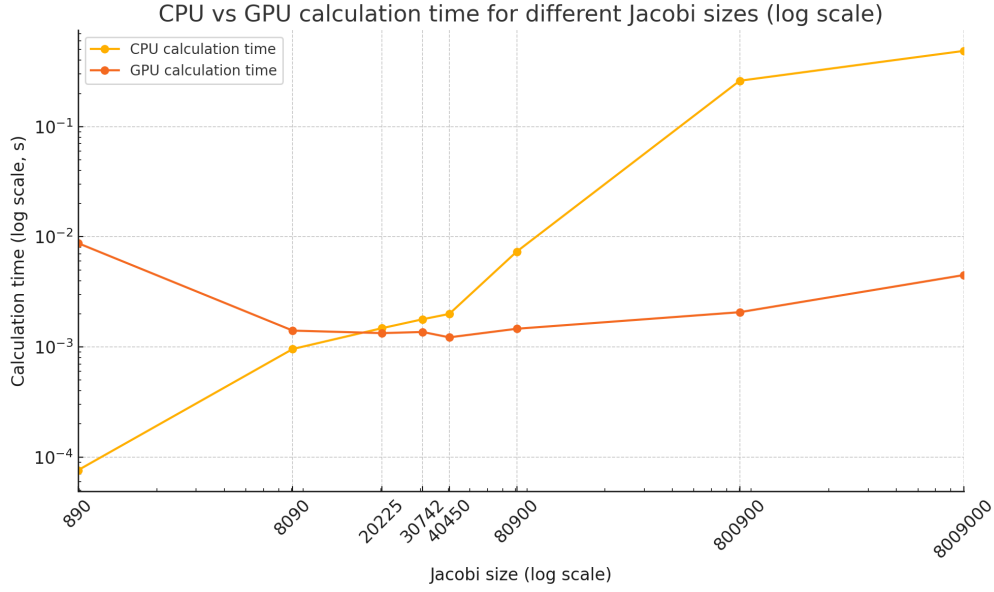


Figure 3.3: Calculation time of the Jacobi matrix with different sizes in logarithmic scale.

With these results, a threshold can be determined below which the CPU version of the classifier is used, and above which the GPU version is applied. This threshold lies between the 10^3 and 10^4 magnitudes. To showcase the superiority of the GPU implementation with large datasets, a comparison is made with the MATLAB implementation, which ran parallelized on CPU, on the MIT-BIH dataset [59]. The results are shown below in Figure 3.4.

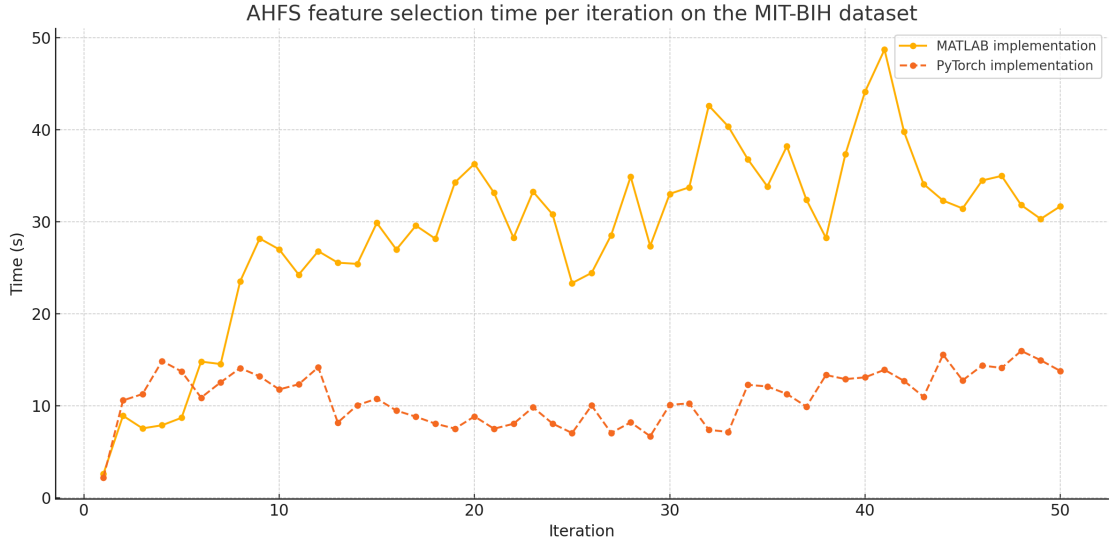


Figure 3.4: Comparison of AHFS implementations. PyTorch denotes the Python environment using the PyTorch module.

3.1.3 MNE-Python

MNE-Python [60] is a magnetoencephalographic and electroencephalographic data analyzing toolbox developed in the Python environment. It enables the creation of signal processing pipelines built on unified data formats, file handling, preprocessing, statistical evaluation, visualization, and more. It is well integrated in the Python ecosystem, itself relying on notable modules such as NumPy [61] and SciPy [62] amongst others. For our solution, raw data handling and file I/O operations were of most concern.

3.1.4 MNE-Features

MNE-Features [63] is a feature extraction package specifically designed for magnetoencephalographic and electroencephalographic signal. As its name suggests, the module is developed in such a way that it seamlessly handles raw data in MNE-Python formats. A similarly well integrated tool in the Python developer environment, it follows the standard scikit-learn [64] format for implementing class interfaces. For our purposes, the majority of univariate feature extraction methods contained in the module were applied.

3.2. Methodology

3.2.1 Recorded data

Data was obtained by measuring and recording signals from two middle aged men, referred to as pilot B and pilot C, suffering from tetraplegia performing distinct tasks related to the motor imagery paradigm. The imaginary tasks were as follows:

1. left hand movement
2. right hand movement
3. both legs movement
4. saying words starting with a given character
5. iterative subtraction
6. singing a song
7. spatial navigation in a well known place
8. 3D rotation of an object
9. resting period

Signals were recorded with an actiCHamp amplifier on the actiCAP headset manufactured by Brain Products GmbH [65]. The measurements produced electroencephalographic data with 63 channels.

Each recording, defined as experiment, produced several sessions. Every session included one out of the nine imagery tasks. The experiments are grouped together into a database, each experiment serving as a different subject.

3.2.2 Data preprocessing, feature extraction

The raw Functional Image Files (FIF) [66] were collected and separated based on which pilot provided the data. After this, the FIF measurements were converted into uniform-length units of data called epochs. In each raw recording, the epochs started from 500 milliseconds past the visual task assignment and ended at 5 seconds past the visual task assignment. In more elementary terms, the neuron activity that occurred between 500 milliseconds and 5 seconds past the flash on the pilot’s monitor that provided

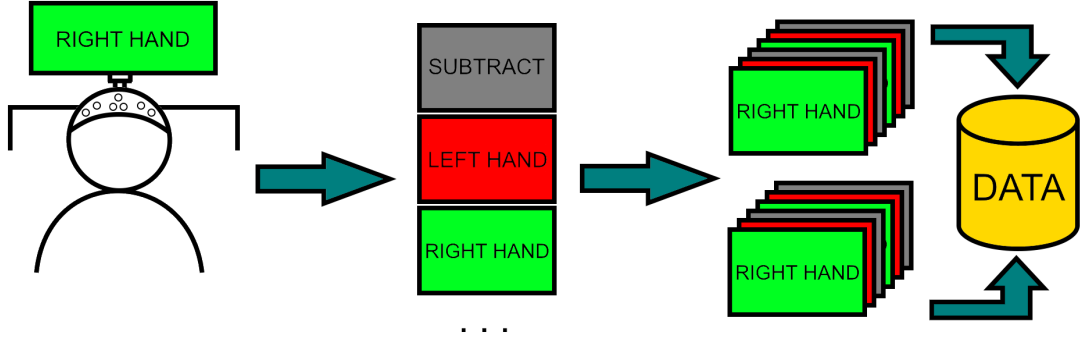


Figure 3.5: Process of EEG data recording.

the instructions were retained. The epochs that contained the desired tasks were kept and concatenated into a single object. For pilot B, tasks two (right hand movement) and five (iterative subtraction) were selected, while pilot C performed tasks three (both legs movement) and five (iterative subtraction). These two specific pair of classes were selected based on prior experience. In both cases, this meant producing a dataset for binary classification.

Following epoch extraction, Fully Automated Statistical Thresholding for EEG artifact Rejection (FASTER) [11] was performed. FASTER is a non-supervised preprocessing tool that performs artifact rejection and outlier removal with a high sensitivity and specificity. The data was windowed with 500 ms-sized windows and 100 ms shifts.

Past the preprocessing phase, MNE-Features applied feature extraction with a number of time- and frequency domain features. These features were:

- | | |
|---------------------------|------------------------------------|
| 1. Mean | 10. Approximate Entropy [68] |
| 2. Variance | 11. Sample Entropy [68] |
| 3. Standard Deviation | 12. Decorrelation Time [69] |
| 4. Peak-to-Peak Amplitude | 13. Hjorth Mobility [70] |
| 5. Skewness | 14. Hjorth Complexity [70] |
| 6. Kurtosis | 15. Higuchi Fractal Dimension [71] |
| 7. Root-Mean Squared | 16. Katz Fractal Dimension [71] |
| 8. Quantile | 17. Zero Crossings |
| 9. Hurst Exponent [67] | 18. Line Length [72] |

19. SVD Entropy [73]
20. SVD Fisher Information [73]
21. Band Energy [74]
22. Spectral Edge Frequency [75]
23. Spectral Slope [76]
24. Spectral Entropy [77]
25. Spectral Hjorth Mobility [69]
26. Spectral Hjorth Complexity [69]
27. Power Spectrum [69]

With feature extraction accomplished in a parallel execution, the new transformed data is combined with each column labelled according to its channel origin and transformation method. Then the results are converted into a Comma-Separated Value spreadsheet for feature selection.

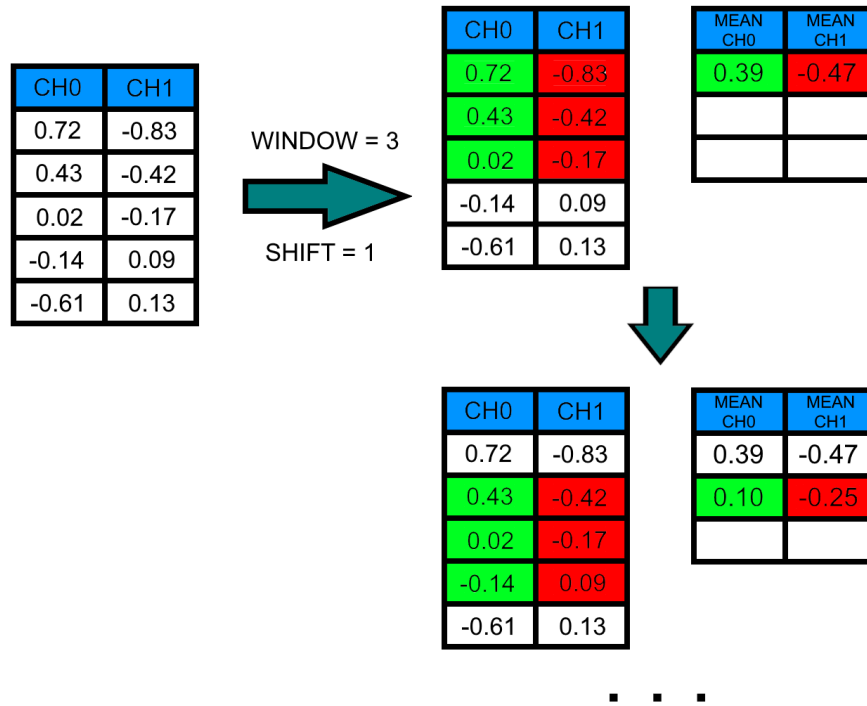


Figure 3.6: Example of windowed feature extraction.

3.2.3 Feature selection

The feature extracted dataset was given as input to the Adaptive, Hybrid Feature Selection algorithm's MATLAB implementation. Initially, the desired number of features to be selected was set at the full dataset size. Based on runtime loss statistics, this number was lowered in later runs. As the filter method measures require discrete input, the dataset, excluding the target variables, was discretized into 1200 equal-sized bins.

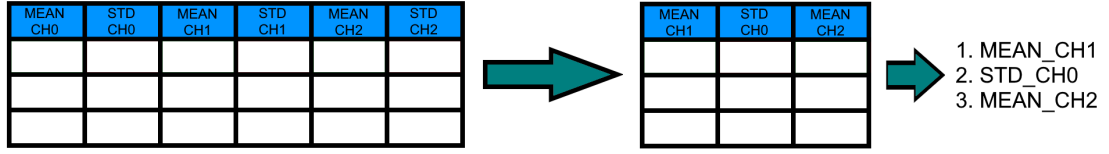


Figure 3.7: Feature selection process.

When the selection process concluded, the subset of the selected features was adjusted to match the number of iterations where the loss was at the minimum, as seen in Figure 3.8. This ensured that features that caused the error to increase were excluded from the final analysis.

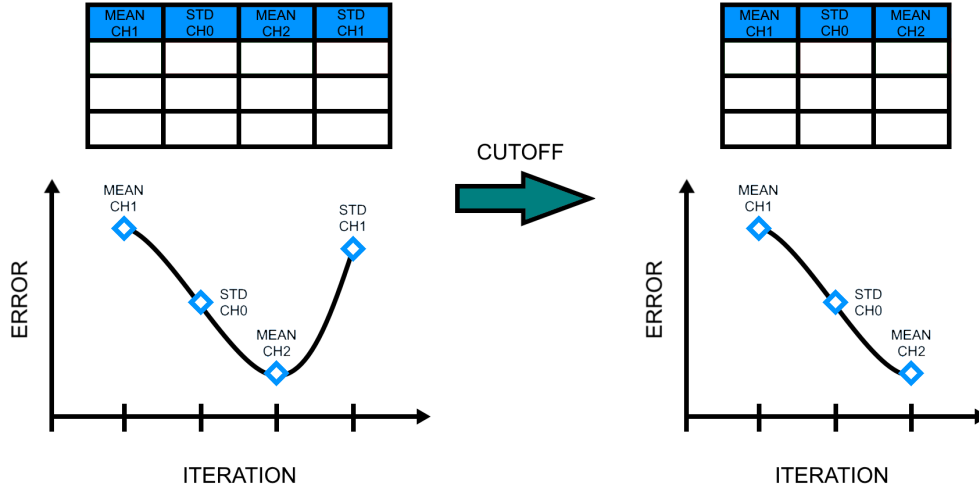


Figure 3.8: Final subset selection during feature selection.

3.2.4 Analysis of intermediate results

Given the subset of ordered, selected features, the value of each channel and feature type could be quantified. Recall that the Sequential Feature Selection wrapper method measured the importance of every feature by how early it was selected - the earlier, the more significant. In this spirit, the actual amount a feature contributed to channel and feature type importance is weighted as the following:

$$score = \frac{rank}{subset\ size} \quad (3.2)$$

where *score* is the measure of importance as a real number between 0 and 1, *rank* is the position of the feature occupied in the ordered subset reversed, and *subset size* is the size of the selected features subset.

In this manner, the score of importance for every channel and feature type is the sum of all feature scores where the feature belongs to the specific channel and feature type, as seen in Figure 3.9.

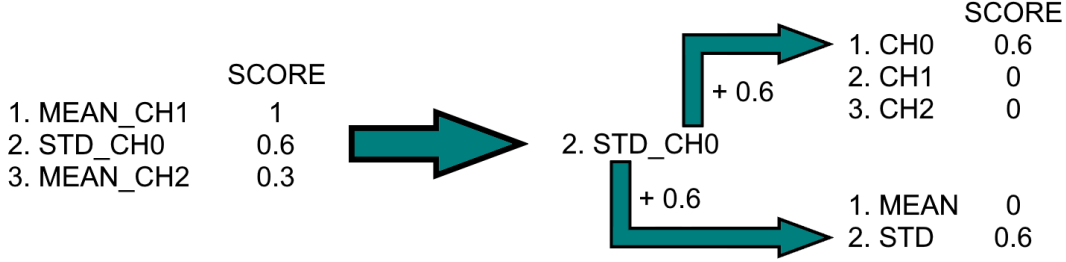


Figure 3.9: Scoring process.

Formally, the channel or feature type score is:

$$S_i = \sum_{f \in F_i}^{n_{F_i}} f \quad (3.3)$$

where S_i is the score of the i th channel or feature type, n_{F_i} is the number of features in the F_i feature group, $f \in F_i$ is the feature score for a feature in the F_i feature group, and F_i is the group of features originating from the i th channel or having the i th feature type.

Once the scores are calculated, the channels are set in a descending order accordingly and later evaluated.

3.2.5 Evaluation of channel selection

The Bionic Applications framework, with a pipeline containing a deep neural network, was used for channel subset evaluation. The learning algorithm in question is EEGNet Fusion [12], a convolutional neural network with primary use in cross-subject motor imagery signal classification. The classifier parameters were 500 epochs, 20 epoch patience for early stopping, 20% validation split, and 0.001 learning rate with 5 fold cross-validation. Additionally, early stopping is suspended for the first 40 epochs. This is referred to as the give up-period [78]. The TensorFlow-based [79] model was compiled with the Adam optimizer [80] given Sparse Categorical Cross-Entropy loss and accuracy as an additional metric. The model was created by transfer learning with the initial training done on all

subjects at once. After this, the model weights were frozen, then fine-tuned on each fold. The tuned version's mean per class accuracy was used as a goodness measure.

First, the full raw dataset with 63 channels was evaluated with the parameters defined above. Then, the scored and ordered channels were evaluated in successively expanding groups of three, that is the first subset contains the first three channels, the second the first six, the third the first nine, and so forth. This method is illustrated in Figure 3.10.

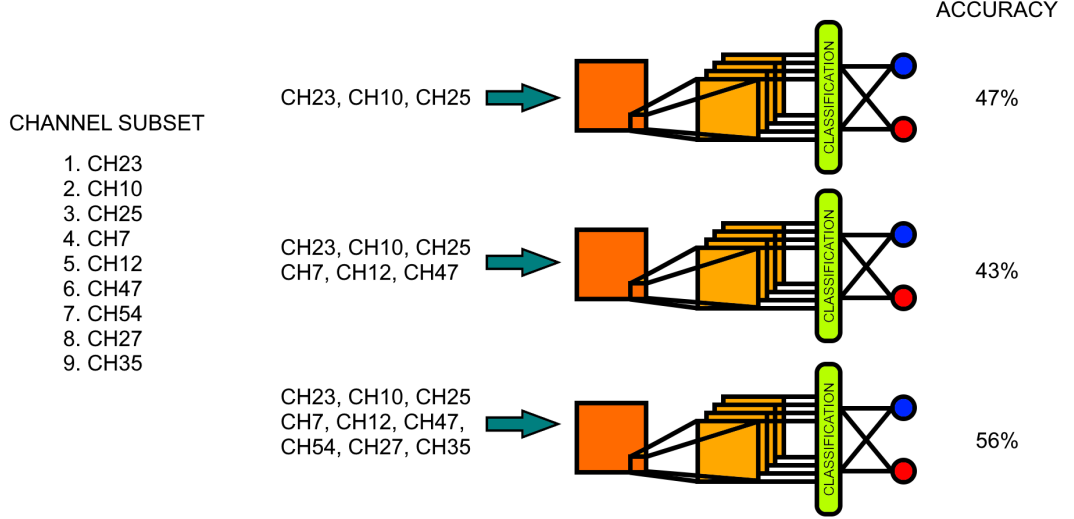


Figure 3.10: Channel evaluation process.

Following the neural network evaluation, the accuracies were compared and statistically evaluated.

3.2.6 Evaluation of feature selection

While wrapper feature selection methods are able to produce superior feature subsets compared to filter methods, their generalization capabilities suffer due to fine-tuning their results to a specific learning algorithm. Consequently, the need to evaluate the results of the Adaptive, Hybrid Feature Selection algorithm by a separate, independent classifier arises.

For this purpose, a custom built shallow neural network using the TensorFlow Python package was utilized. The network consisted of 2 hidden layers with 8 and 4 neurons, resulting in the architecture of $input\ size \times 8 \times 4 \times 2$. A 0-mean, 1-standard deviation normalization was performed on the input. The activation function for the hidden neurons was Leaky ReLU [81], and for the output neurons it was the softmax function. The model

optimizer was Adam with Categorical Cross-Entropy loss and accuracy as an additional metric. The network was fitted on 5 folds in 500 epochs with a 20 epoch early stopping. Each fold had an 80% train and 20% test split.

The goodness of the feature selection solution was established by comparing the mean accuracy of the five folds with the complete feature extracted dataset's five fold mean accuracy. As a result of the low sample size, no statistical evaluation was made. In addition, all channels were mapped a set of features that were obtained in the feature selection process.

Chapter 4

Results

This chapter presents the results obtained by the implementation of the channel and feature selection solution described in this report. In Section 1, the results of channel selection are showcased and compared with findings in literature. In Section 2, the results of feature selection are introduced and contrasted with literature where applicable.

4.1. Channel Selection

4.1.1 Pilot B

For pilot B, the data contained 26 subjects with 1155 epochs after preprocessing. The feature extracted dataset was of size 47355×2395 . Feature selection produced 215 features with the Scaled Root Mean Squared Error (SRMSE) plotted in Figure 4.1. The formula for SRMSE is:

$$\sqrt{\frac{\sum_{i=1}^n (y_{pred_i} - y_{true_i})^2}{\frac{0.8^2}{2}}}, \quad (4.1)$$

where n is the length of the target vector, y_{pred_i} is the i th predicted target value, and y_{true_i} is the i th actual target value. The scaling occurs because the input data is scaled between 0.1 and 0.9 before neural network training.

The selected features provided the channel scores shown in the bar chart Figure 4.2 and listed in Table 6.1, Appendix. In short, the first 12 channels were selected for evaluation in successively increasing groups of three. These channels were, in descending score order, F7, C4, AF7, F8, C3, FT8, P7, CP3, CP5, FT9, FC6, and CP1.

The classification by the deep neural network was then performed, with each channel set providing the accuracies presented in Figure 4.3 and Table 4.1.

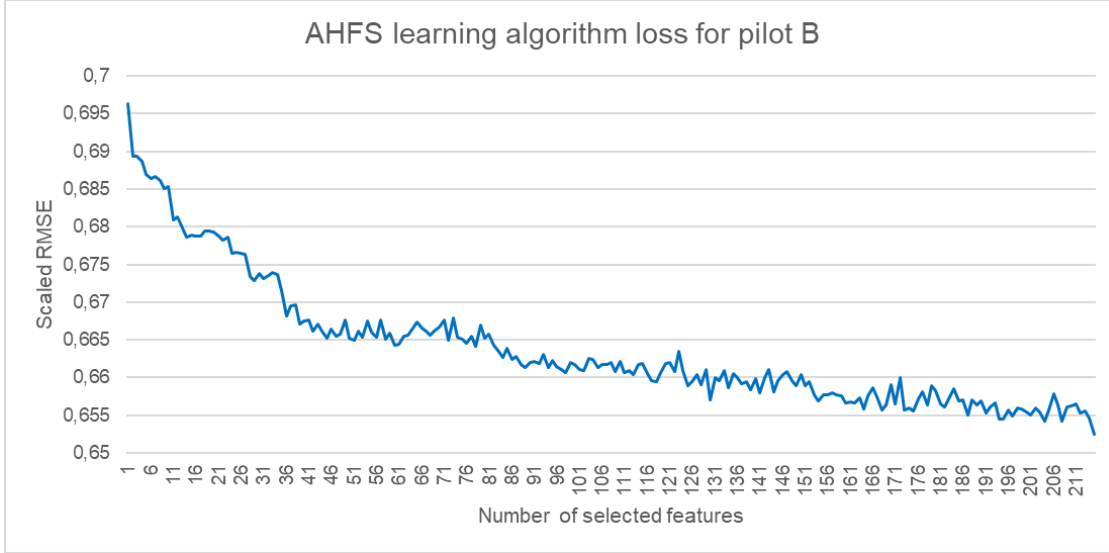


Figure 4.1: Pilot B AHFS loss per number of selected features.

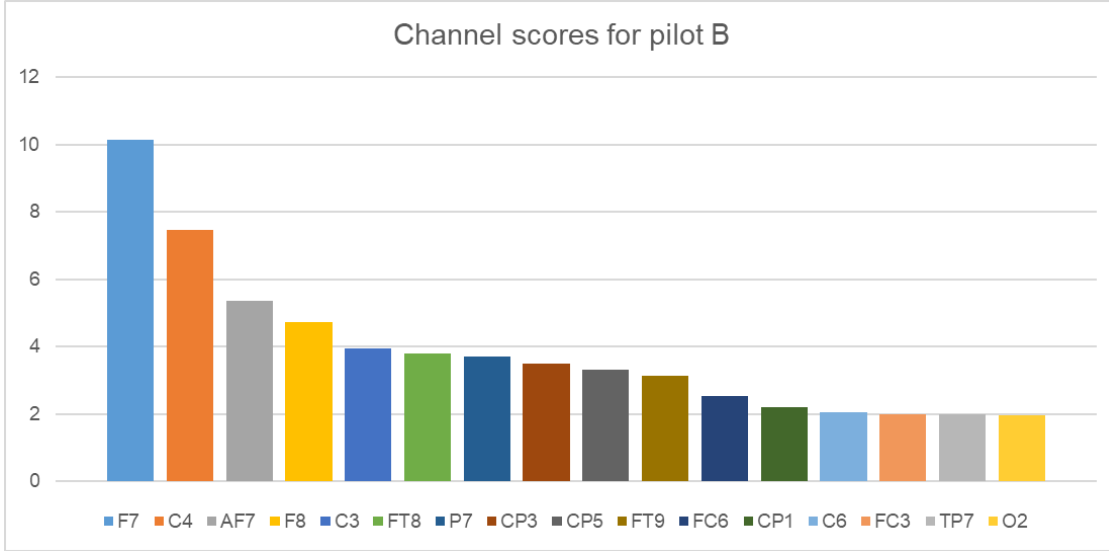


Figure 4.2: Pilot B channel scores.

As observable from the results, the mean of all accuracies in the 9 channel set outperformed all other groups, including the full 63 channel set. To verify this, two statistical tests were performed using the Pingouin Python module [82]. First, the normality of both series of accuracies were validated using the Shapiro-Wilk test [83] with a significance level of $\alpha = 0.05$. While the null hypothesis was accepted for the full channel set ($p > 0.6$), it was rejected for the 9 channel set ($p < 0.002$). For this reason, the Mann-Whitney U non-parametric test [84] was chosen with the null hypothesis defined as both distributions being equal, and the alternative hypothesis established as the 9 channel set having a stochastically greater underlying distribution than the 63 channel set. The

3 channels	F7, C4, AF7
6 channels	F7, C4, AF7, F8, C3, FT8
9 channels	F7, C4, AF7, F8, C3, FT8, P7, CP3, CP5
12 channels	F7, C4, AF7, F8, C3, FT8, P7, CP3, CP5, FT9, FC6, CP1

	63 channels	3 channels	6 channels	9 channels	12 channels
Accuracy	58,7%	55,9%	59,1%	60,7%	60,4%
Standard Deviation	3,6%	3,4%	4,0%	4,0%	3,4%

Table 4.1: Channel set components and classifier performance for each set, pilot B.

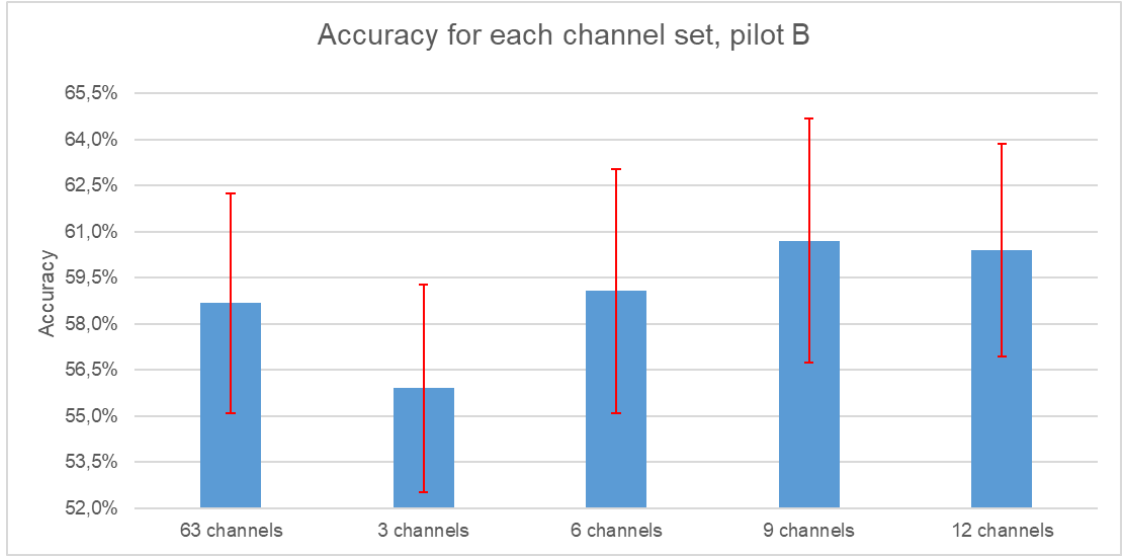


Figure 4.3: Final accuracy for each channel set for pilot B. The red intervals mark the standard deviation for each group.

null hypothesis was rejected with $p < 0.003$, and the alternative hypothesis was accepted. Therefore, it can be stated that channel selection significantly increased classification accuracy from 58.7% to 60.7%.

Mapping the channels on a visualized EEG channel layout, the heatmap in Figure 4.4 is produced. Compared with the Core Channel Selection made by Guttman-Flury et al. [51], channels F7, C4, F8, C3, P7, and CP3 overlapped with the core motor imagery paradigm set, channels FT8 and CP5 fell outside the authors' set and channel AF7 was not mentioned. However, it is important to note that the tasks performed in this case were not exclusively motor imagery in nature.

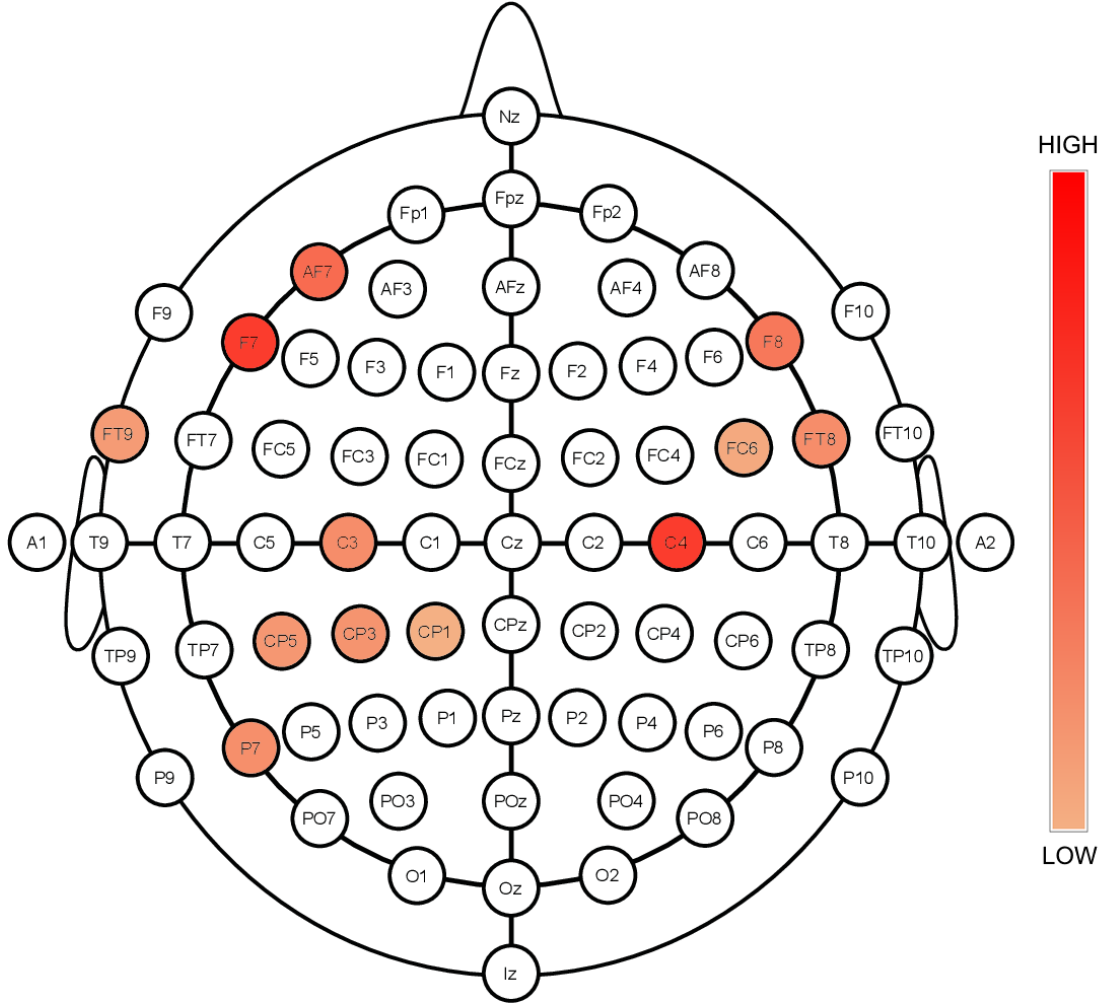


Figure 4.4: Layout of selected EEG channels with color scale based on score, pilot B. Original layout scheme by Hvaring et al. [85].

4.1.2 Pilot C

Concerning pilot C, the dataset consisted of 29 subjects with 1384 epochs in total post processing. After feature extraction, the data dimensions were 56744×2395 . The Adaptive, Hybrid Feature Selection algorithm concluded that 350 features were optimal. The error curve is plotted in Figure 4.5.

The channel scores obtained from the selected features-subset are presented in Figure 4.6 and listed in Table 6.2, Appendix. In a similar fashion to pilot B's case, the 12 highest scoring channels were selected for evaluation in the previously mentioned increasing group size manner. In summary, the channels in ranked order were Cz, C1, C3, CP5, T8, C5, TP7, Pz, FC6, CPz, CP1, and C6.

Following score determination, the sets were evaluated in the Bionic Applications framework using the EEGNet Fusion convolutional neural network. The results are pre-

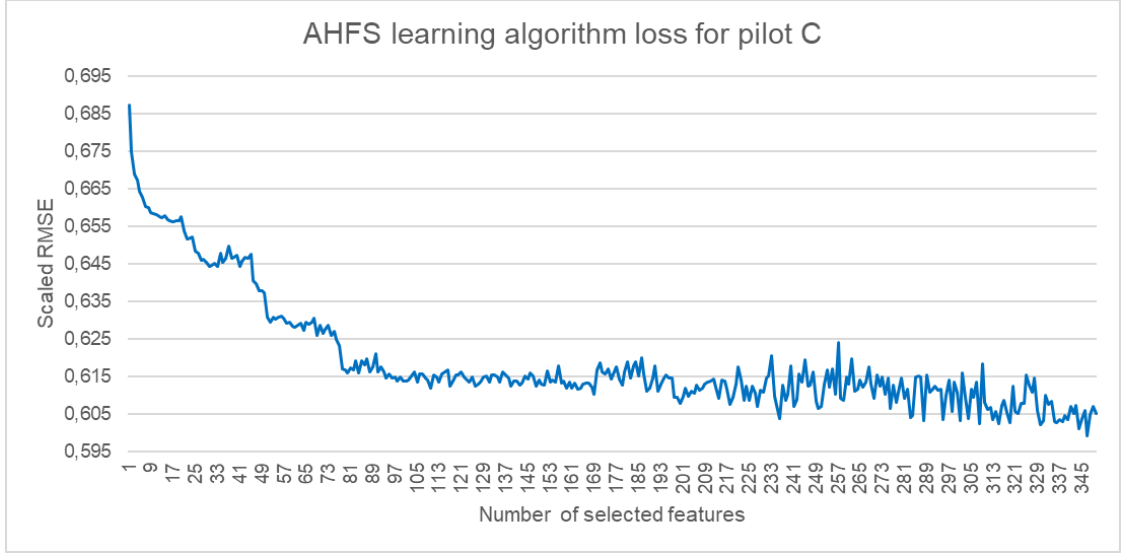


Figure 4.5: Pilot C AHFS loss per number of selected features.

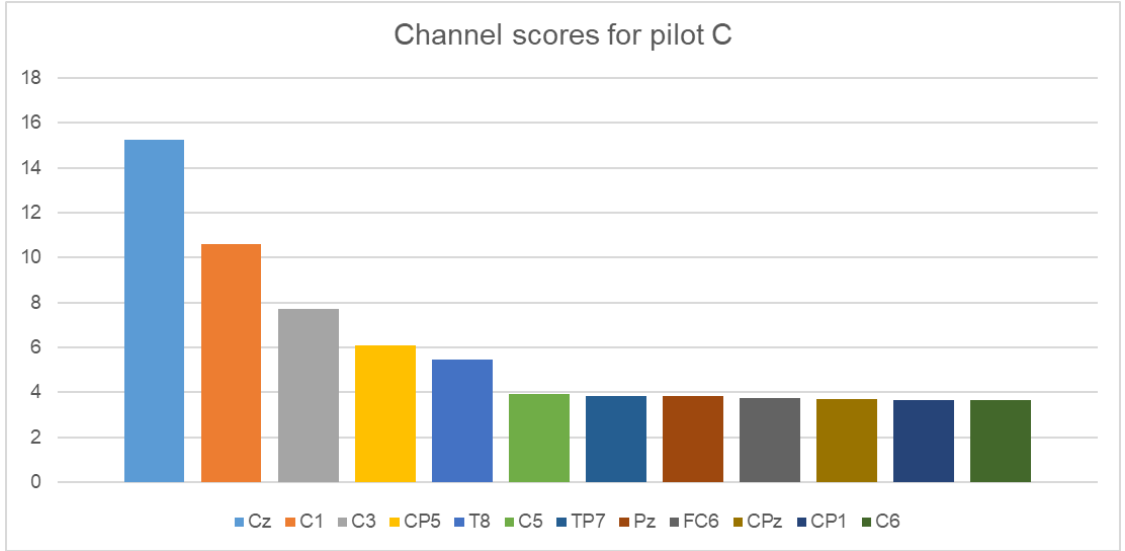


Figure 4.6: Pilot C channel scores.

sented in Figure 4.5.

The results show that channel selection offers a minimally increased, albeit de facto identical, accuracy compared to the 63 channel set version. For verification purposes, statistical tests were performed. The Shapiro-Wilk test ruled that both the full 63 channel accuracies and selective 9 channel accuracies were normally distributed with respective p-values of $p > 0.11$ and $p > 0.56$. Accordingly, the independent t-test [86] was performed with a confidence level of 95%. The null hypothesis was defined as both accuracies being equal to each other, while the alternative hypothesis stated that the full 63 channel set had a lower accuracy in comparison to the selected channel-subset. With a p-value

3 channels	Cz, C1, C3
6 channels	Cz, C1, C3, CP5, T8, C5
9 channels	Cz, C1, C3, CP5, T8, C5, TP7, Pz, FC6
12 channels	Cz, C1, C3, CP5, T8, C5, TP7, Pz, FC6, CPz, CP1, C6

	63 channels	3 channels	6 channels	9 channels	12 channels
Accuracy	69,6%	63,5%	68,2%	69,9%	68,1%
Standard Deviation	4,1%	4,9%	4,2%	4,0%	4,1%

Table 4.2: Channel set components and classifier performance for each set, pilot C.

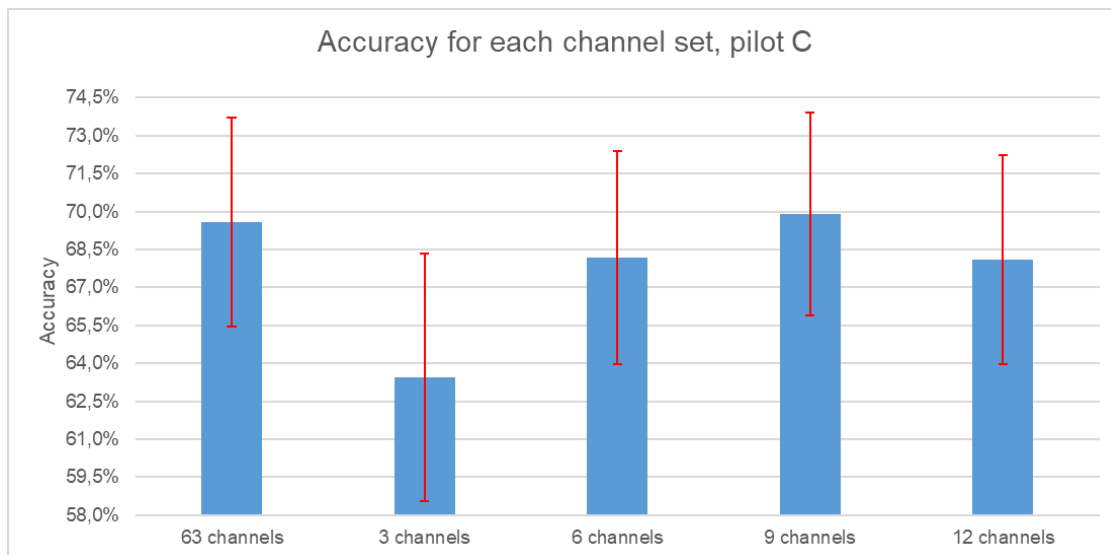


Figure 4.7: Final accuracy for each channel set for pilot C. The red intervals mark the standard deviation for each group.

of $p > 0.41$, the null hypothesis was accepted, confirming that the accuracy difference between the full and selected channel sets were insignificant. However, the elimination of 54 channels significantly reduced computational complexity at no cost to accuracy.

The optimal channels are shown in the EEG layout in Figure 4.8. Compared to the Core Channel Selection, channels C1, C3, and C5 match with the findings of Guttman-Flury et al. [51], channels Cz, CP5, T8, Pz, and FC6 do not, and channel TP7 was not available for comparison.

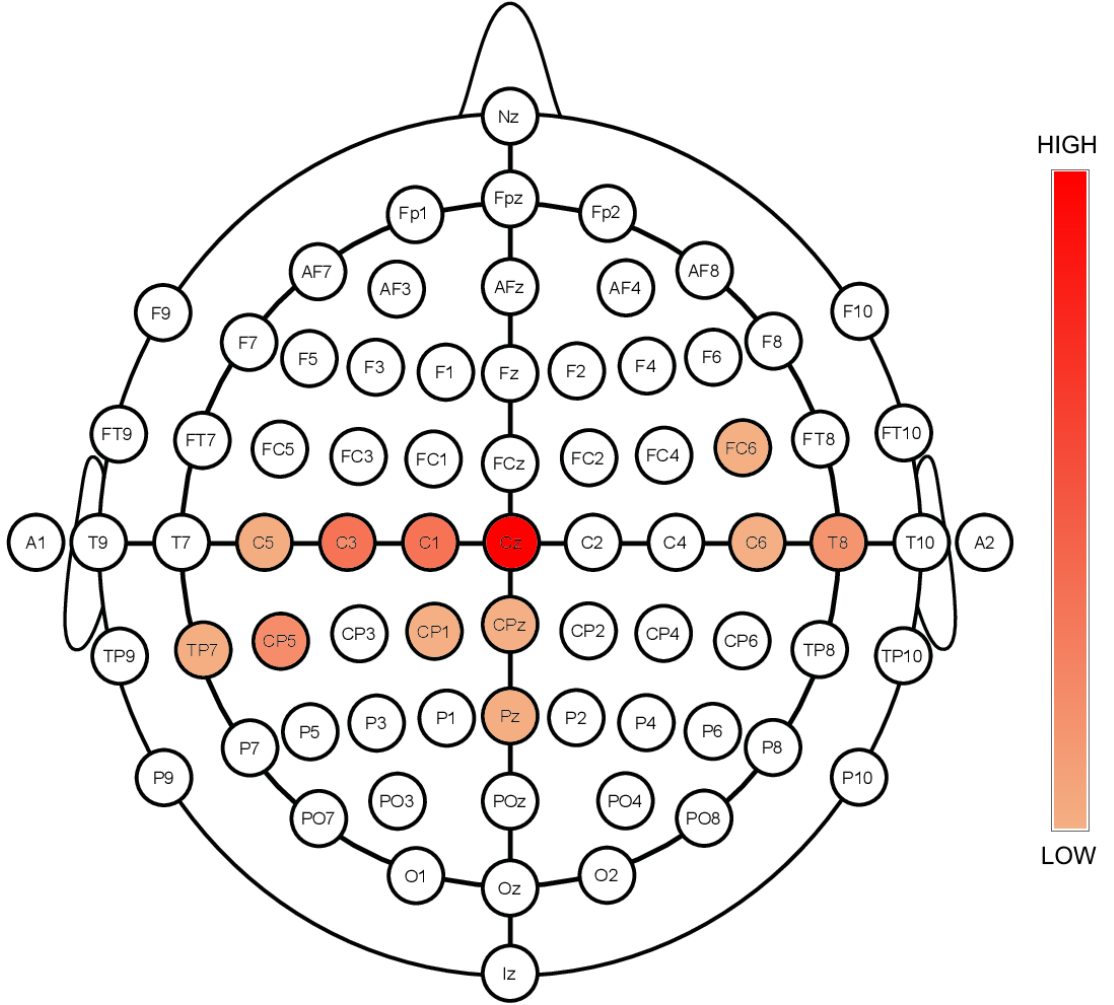


Figure 4.8: Layout of selected EEG channels with color scale based on score, pilot C. Original layout scheme by Hvaring et al. [85].

4.2. Feature selection

4.2.1 Pilot B

Similarly to channel selection, the number of selected features were chosen based on which produced the lowest learning algorithm error, which is identical to the results evaluated by channel selection. The feature extracted dataset of size 47355×2395 was compared with the post-selection dataset of size 47355×215 using the custom built shallow neural network.

The results are showcased in Figure 4.9 and detailed in Table 4.3.

The accuracy comparison shows that no significant increase in accuracy was obtained by performing feature selection. However, data complexity significantly decreased at no cost to accuracy.

	Complete dataset	215 features
Mean accuracy	58.60%	58.84%
Standard Deviation	0.65%	1.87%

Table 4.3: Classification accuracy for each feature set, pilot B.

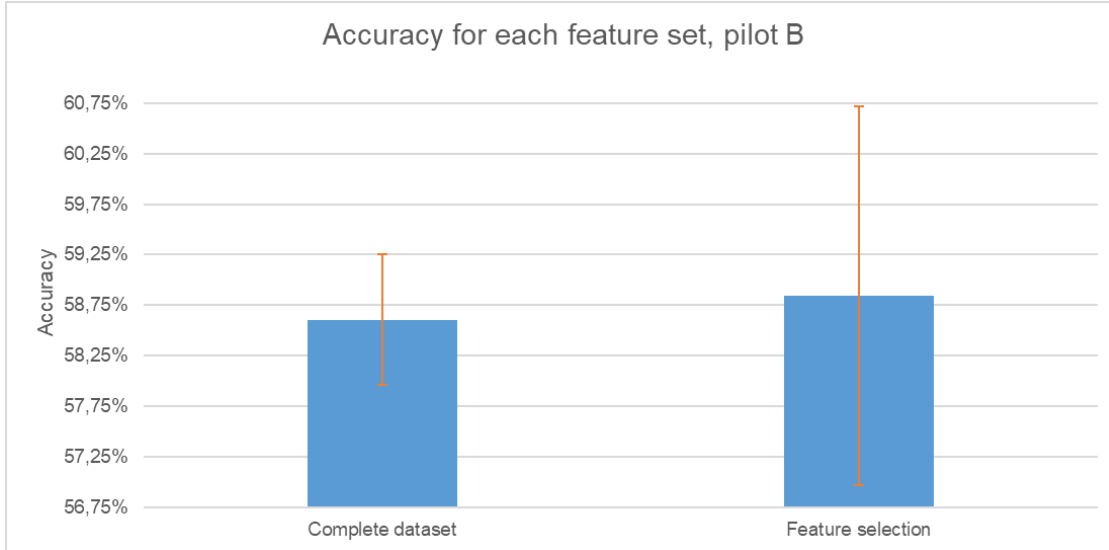


Figure 4.9: Final accuracy for each feature set for pilot B. The orange intervals mark the standard deviation for each group.

In a similar fashion to channel selection, each feature was scored in a weighted manner according to its rank. In Figure 4.10, the most prevalent features are shown. In Table 6.3, Appendix, all feature types are listed with their respective score.

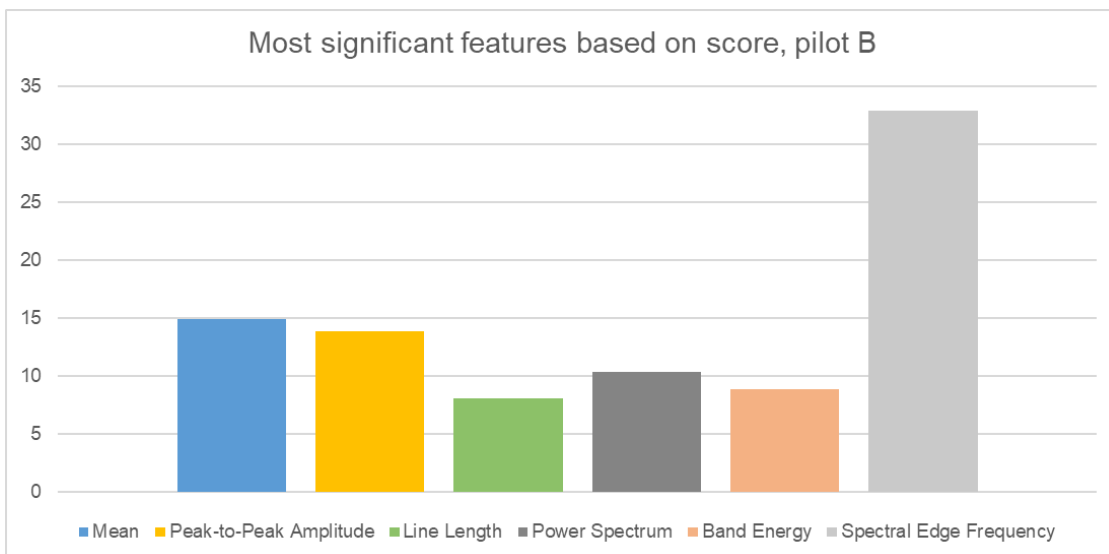


Figure 4.10: Significant features for pilot B.

As the results show, the most dominant feature was the Spectral Edge Frequency, followed by Mean, Peak-to-Peak Amplitude, Power Spectrum, Band Energy and Line Length. Half of the significant features were in the frequency domain, while the other half of features were in the time domain. Comparing the results with the findings in literature, an overlap is found with Rashid et al. [46] with the Welch’s method-based Power Spectrum and Band Power. While the Higuchi Fractal Dimension is noted, its relevance in the current evaluation was limited. Wang et al. [47] noted Sample Entropy, Hjorth Parameters, and Fractal Dimension, however their importance was low in this setting. On the other hand, Power Spectrum Density and basic statistical features were of higher relevance. The findings of A. K. Singh et al. [50] presented the Higuchi Fractal Dimension, basic statistical features, Power Spectral Density and Band Power. Aside from the Higuchi Fractal Dimension, the other noted features were of increased importance. Additionally, Mean as a basic statistical feature proved to be an appropriate extraction method, in contrast with the authors’ suggestion that statistical features may not be suitable for physiological signals.

In case of the Band Energy and Power Spectrum features, the importance of the delta (0.5 - 4 Hz), theta (4 - 8 Hz), alpha (8 - 13 Hz), beta (13 - 30 Hz), and gamma (30 - 100 Hz) bands were examined. Taking into account their combined occurrence, the weighted score of each band is shown in Figure 4.11 and presented in Table 4.4.

Band	Score
Delta	0
Theta	3.66
Alpha	7.64
Beta	4.11
Gamma	3.86

Table 4.4: Frequency band weighted scores, pilot B.

Based on the results, the delta band was of no significance whatsoever. The most relevant band was the alpha frequency interval, followed by beta, gamma and theta. The last three were roughly uniform in terms of score. Examining the literature, Mohdiwale et al. [49] highlighted the importance of the beta and gamma frequency bands for motor imagery-based EEG applications. In contrast, the current results show the relevance of the alpha band, with the beta and gamma bands playing a relatively less important role. However, similarly to the channel selection case, it is important to note that the tasks

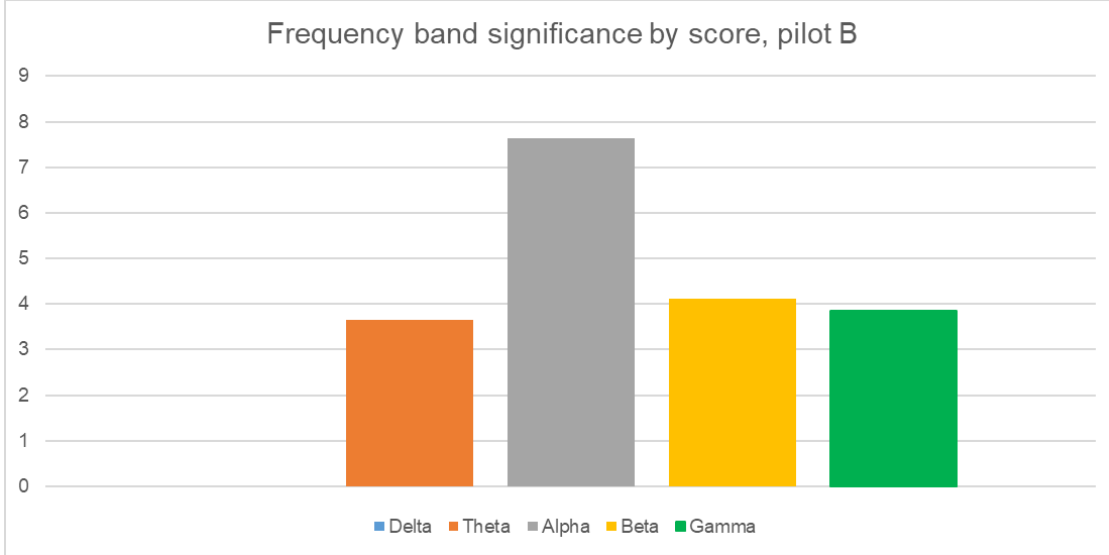


Figure 4.11: Frequency band significance based on weighted score for pilot B.

were not exclusively motor imagery in origin.

The subsequent use of feature extraction and selection also presented insights into what type of features suit a specific channel best. These results are showcased in Table 6.5. In summary, the feature types associated with each channel were, in almost all cases, the same across each channel.

4.2.2 Pilot C

As previously noted, the post-extraction dataset for pilot C had a dimensionality of 56744×2395 . Following feature selection, the size shrunk to 56744×350 . This version of the data was then evaluated using the TensorFlow-based custom shallow neural network.

The results of the evaluation are shown in Figure 4.12 and listed in detail in Table 4.5. The presented results show that the subset of the complete dataset performed significantly better in terms of classification accuracy.

	Complete dataset	350 features
Mean accuracy	60.05%	76.65%
Standard Deviation	0.77%	0.38%

Table 4.5: Classification accuracy for each feature set, pilot C.

The weighted scoring of all features is established in Table 6.4, Appendix. The most significant features are showcased in Figure 4.13.

Similarly to pilot B, the most relevant feature was Spectral Edge Frequency. Half of

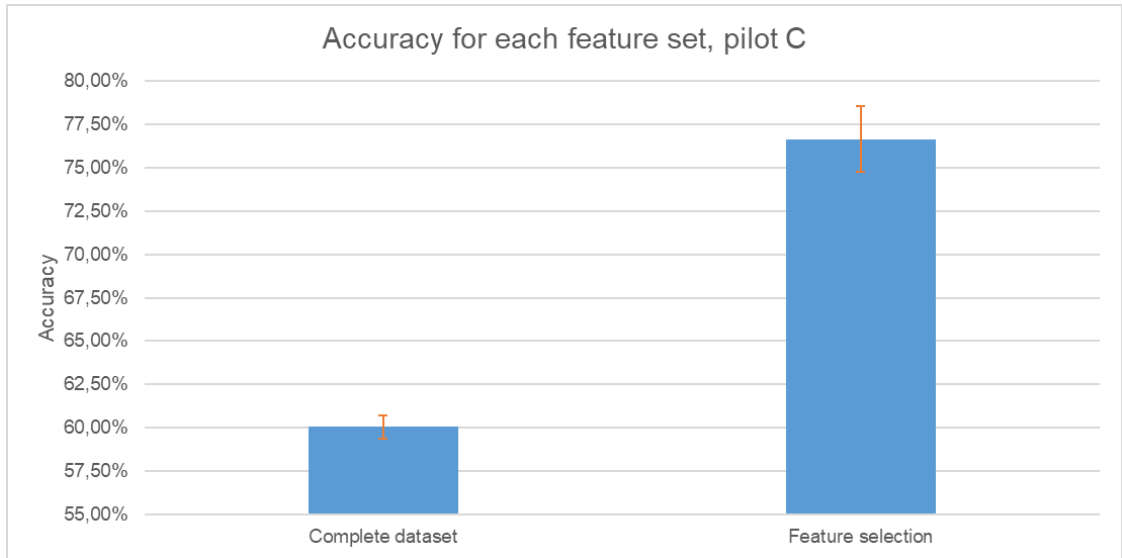


Figure 4.12: Final accuracy for each feature set for pilot C. The orange intervals mark the standard deviation for each group.

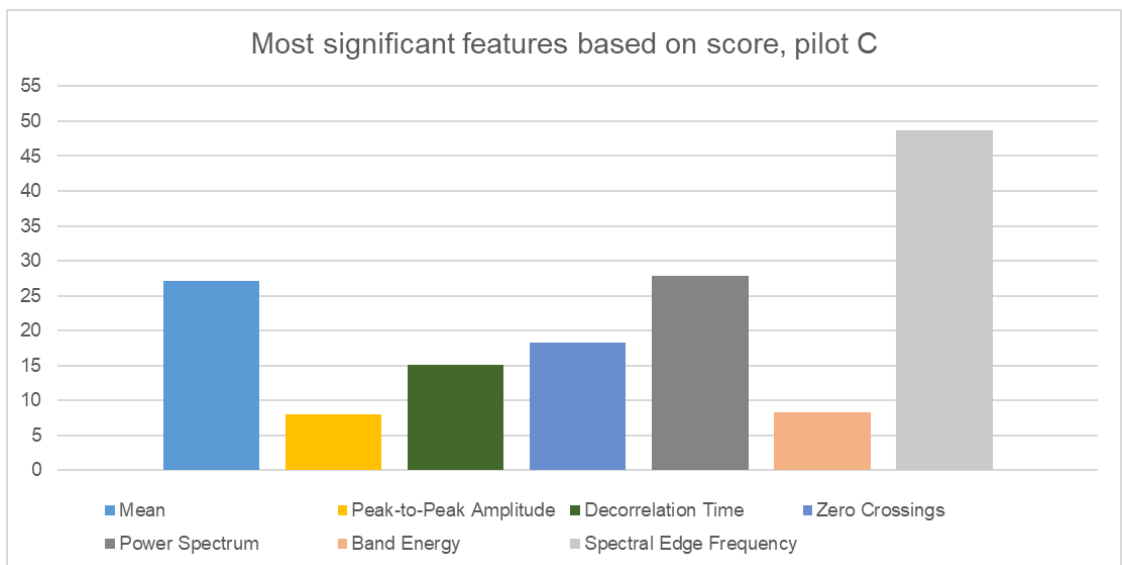


Figure 4.13: Significant features for pilot C.

that score were Power Spectrum and Mean, followed by Zero Crossings, Decorrelation Time, Peak-to-Peak Amplitude and Band Energy with increasingly lower scores. Unlike the case of pilot B, however, the most significant features produced by pilot C were chiefly in the time domain, with the exception of the Power Spectrum, Band Energy and Spectral Edge Frequency features. Overlapping with the findings, Rashid et al. [46] highlighted the Power Spectral Density and Band Power as similarly significant features. Wang et al. [47] noted basic statistical features and Power Spectral Density as features in common with the present results. Finally, A. K. Singh et al. [50] reiterated that basic statistical

features, Power Spectral Density, and Band Power are relevant. However, in neither case were Peak-to-Peak Amplitude, Decorrelation Time, Zero Crossings, or Spectral Edge Frequency mentioned as significant features. Additionally, the case for basic statistical functions serving as suitable features for physiological data was reinforced with these results as well.

The delta, theta, alpha, beta, and gamma frequency bands were scored alongside the features. The results are shown in Figure 4.14 and detailed in Table 4.6.

Band	Score
Delta	0
Theta	0
Alpha	29.04
Beta	4.67
Gamma	2.47

Table 4.6: Frequency band weighted scores, pilot C.

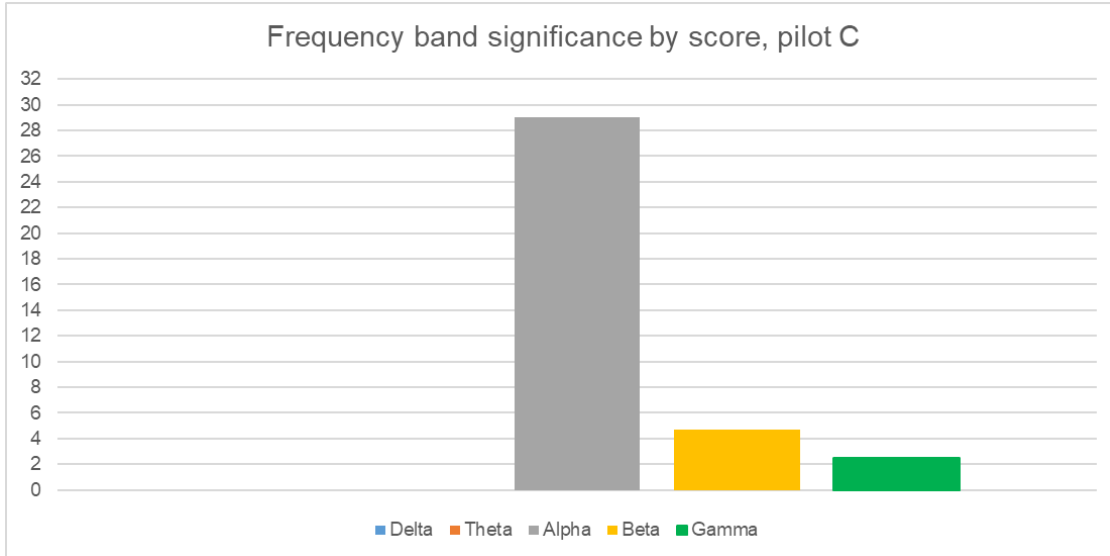


Figure 4.14: Frequency band significance based on weighted score for pilot C.

As the results show, the contrast in significance with the alpha band is remarkably more pronounced. Furthermore, neither the delta band, nor the theta band participated in enhancing classification performance at all. The role of beta and gamma bands were minor. In a similar fashion to pilot B's case, the findings of Mohdiwale et al. [49] that the beta and gamma bands play a significant role are somewhat challenged.

Feature sets were established for all channels, listed in Table 6.6, Appendix. Similarly

to pilot B, the majority of features were not unique to one specific channel but rather overlapping with others.

Chapter 5

Conclusion and future research

Electroencephalography (EEG) was briefly compared with other neural activity measurement techniques such as electrocorticography and magnetoencephalography. The issue of high data dimensionality in EEG recordings was recognized and the task of reducing it via feature extraction and selection laid out.

Feature extraction and selection were presented as feature engineering solutions. Feature selection methods were grouped into filter, wrapper, embedded and hybrid methods. A number of feature selection algorithms were showcased in detail. Feature extraction in EEG was further investigated, classifying features into time, frequency, time-frequency, spatial, spatio-temporal, and decomposition domains. The advantages and disadvantages of a number of methods were compared. Feature and channel selection in EEG were looked into.

The solution framework was laid out and its components presented in detail, including the Bionic Applications framework, the Adaptive, Hybrid Feature Selection algorithm, and so forth. The methodology was showcased from data origins through preprocessing and feature extraction to feature selection and evaluation.

The aim of this research was to provide an optimal channel and feature selection solution for motor imagery-based brain-computer interface applications. This report outlined the basic concepts of electroencephalography, feature extraction and feature engineering. The relevant literature in these topics were explored. The detailed methodology was presented and the obtained results evaluated and compared with literature.

While several authors argued that deep neural networks eliminate the need for feature selection, the results discussed in this report clearly show otherwise. Additionally, many algorithms are evaluated on the same popular datasets, whereas our databases are continuously expanded and updated.

For both pilots, a personalized set of channels were defined as the optimum that reduces data dimensionality and preserves or improves classification accuracy. In detail, accuracy was maintained at 69% for pilot C and increased from 58.7% to 60.7% for pilot B with a channel set size reduction of 86% in both cases. Two subsets of extracted features were also established that enabled optimal classification by a shallow neural network. The results were an accuracy increase from 60.1% to 76.7% with an 85% feature count reduction for pilot C and another albeit minor accuracy improvement from 58.6% to 58.8% with a 91% feature set size decrease. Additionally, each channel was assigned a set of features, opening the possibility of fine-tuning feature extraction to individual channels. To the best of our knowledge, no such undertaking is present in the relevant literature.

A detailed methodology and solution was established, enabling the development of integrated software solutions in the future. Further research should include a search strategy for establishing the number of channels necessary for an optimal subset based on the weighted scoring. For future brain-computer interface applications, the possibility of using dedicated hardware for online classification on feature extracted data may be considered.

Acknowledgement

This research has been supported by the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory and by the TKP2021-NKTA-01 NRDIO grant on "Research on cooperative production and logistics systems to support a competitive and sustainable economy".

On behalf of Project "Comprehensive testing of machine learning algorithms" we thank for the use of HUN-REN Cloud (<https://science-cloud.hu/>) that significantly helped us achieve the results presented here.

Chapter 6

Appendix

This chapter contains additional data that is necessary to showcase but otherwise not vital to present in place. The tables begin in the next page.

Table 6.5: Feature sets mapped to a specific channel, pilot B

Fp1	Peak-to-Peak Amplitude
	Line Length
	Band Energy, alpha band
	Band Energy, theta band
	Band Energy, beta band
	Standard Deviation
	Variance
	Spectral Edge Frequency
	Spectral Slope
	Mean
	Root-Mean Squared
	Band Energy, gamma band
	Spectral Hjorth Mobility
	Spectral Hjorth Complexity
Fz	Power Spectrum, alpha band
	Spectral Edge Frequency
	Higuchi Fractal Dimension
	Mean
	Decorrelation Time
	Spectral Edge Frequency
	Mean
	Zero Crossings
	Power Spectrum, alpha band
	Spectral Edge Frequency
FT9	Spectral Edge Frequency
	Band Energy, beta band
	Line Length
	Power Spectrum, beta band
	Mean
	Zero Crossings
	Peak-to-Peak Amplitude

	Variance
	Standard Deviation
	Spectral Slope
	Spectral Edge Frequency
	Band Energy, theta band
	Band Energy, alpha band
	Power Spectrum, theta band
	Root-Mean Squared
FC1	Power Spectrum, alpha band
	Higuchi Fractal Dimension
	Spectral Edge Frequency
	Decorrelation Time
C3	Peak-to-Peak Amplitude
	Line Length
	Variance
	Standard Deviation
	Band Energy, beta band
	Band Energy, alpha band
	Spectral Hjorth Mobility
	Spectral Edge Frequency
T7	Power Spectrum, alpha band
	Spectral Edge Frequency
	Higuchi Fractal Dimension
	Mean
	Power Spectrum, gamma band
	Hjorth Mobility
	Peak-to-Peak Amplitude
	Line Length
	Power Spectrum, gamma band
	Hjorth Mobility
	Spectral Edge Frequency
	Higuchi Fractal Dimension
	Sample Entropy
	Spectral Hjorth Mobility

	Katz Fractal Dimension
	Zero Crossings
	Peak-to-Peak Amplitude
	Spectral Slope
	Band Energy, gamma band
	Spectral Hjorth Complexity
	Band Energy, beta band
	Mean
	Spectral Slope
CP5	Peak-to-Peak Amplitude
	Mean
	Spectral Edge Frequency
	Variance
	Standard Deviation
	Line Length
CP1	Spectral Edge Frequency
	Mean
	Zero Crossings
Pz	Line Length
	Peak-to-Peak Amplitude
	Band Energy, beta band
	Band Energy, gamma band
	Spectral Hjorth Mobility
	Spectral Edge Frequency
P3	Peak-to-Peak Amplitude
	Spectral Edge Frequency
P7	Spectral Edge Frequency
	Mean
	Zero Crossings
O1	Power Spectrum, theta band
	Spectral Edge Frequency
	Peak-to-Peak Amplitude
	Zero Crossings
Oz	Spectral Edge Frequency

	Mean
	Zero Crossings
O2	Power Spectrum, theta band
	Spectral Edge Frequency
	Peak-to-Peak Amplitude
	Mean
	Zero Crossings
P4	Spectral Edge Frequency
	Zero Crossings
P8	Spectral Edge Frequency
TP10	Peak-to-Peak Amplitude
	Mean
	Spectral Edge Frequency
CP6	Line Length
	Mean
	Hjorth Mobility
	Spectral Edge Frequency
	Sample Entropy
	Power Spectrum, gamma band
	Band Energy, gamma band
	Spectral Slope
CP2	Spectral Edge Frequency
	Mean
	Line Length
Cz	Peak-to-Peak Amplitude
	Spectral Edge Frequency
C4	Peak-to-Peak Amplitude
	Spectral Slope
	Spectral Edge Frequency
T8	Peak-to-Peak Amplitude
	Spectral Edge Frequency
	Power Spectrum, alpha band
	Higuchi Fractal Dimension
	Line Length

	Band Energy, gamma band
FT10	Mean
	Spectral Edge Frequency
FC6	Spectral Edge Frequency
FC2	Spectral Edge Frequency
	Mean
	Zero Crossings
F4	Mean
	Spectral Edge Frequency
F8	Peak-to-Peak Amplitude
	Line Length
	Band Energy, beta band
	Band Energy, gamma band
	Spectral Edge Frequency
	Spectral Hjorth Mobility
Fp2	Mean
	Peak-to-Peak Amplitude
	Spectral Edge Frequency
AF7	Power Spectrum, alpha band
	Spectral Edge Frequency
AF3	Peak-to-Peak Amplitude
	Spectral Edge Frequency
AFz	Spectral Edge Frequency
	Mean
F1	Spectral Edge Frequency
F5	Power Spectrum, alpha band
	Peak-to-Peak Amplitude
	Spectral Edge Frequency
	Mean
FT7	Spectral Edge Frequency
FC3	Peak-to-Peak Amplitude
	Power Spectrum, gamma band
	Spectral Edge Frequency
	Line Length

	Band Energy, gamma band
C1	Line Length
	Higuchi Fractal Dimension
	Spectral Edge Frequency
	Peak-to-Peak Amplitude
	Power Spectrum, gamma band
C5	Mean
	Spectral Edge Frequency
TP7	Spectral Edge Frequency
CP3	Spectral Edge Frequency
	Mean
P1	Spectral Edge Frequency
	Zero Crossings
P5	Spectral Edge Frequency
	Zero Crossings
PO7	Mean
	Spectral Edge Frequency
	Peak-to-Peak Amplitude
PO3	Mean
	Spectral Edge Frequency
PO4	Mean
	Spectral Edge Frequency
PO8	Spectral Edge Frequency
P6	Spectral Edge Frequency
P2	Spectral Edge Frequency
	Mean
	Power Spectrum, theta band
CPz	Spectral Edge Frequency
	Mean
CP4	Spectral Edge Frequency
	Mean
TP8	Spectral Edge Frequency
	Mean
C6	Spectral Edge Frequency

	Mean
C2	Spectral Edge Frequency
FC4	Spectral Edge Frequency
FT8	Spectral Edge Frequency
	Mean
F6	Spectral Edge Frequency
AF8	Peak-to-Peak Amplitude
	Spectral Edge Frequency
AF4	Spectral Edge Frequency
	Peak-to-Peak Amplitude
F2	Spectral Edge Frequency
FCz	Mean
	Spectral Edge Frequency
	Peak-to-Peak Amplitude

Table 6.6: Feature sets mapped to a specific channel, pilot C

Fp1	SVD Entropy
	SVD Fisher Information
	Zero Crossings
	Spectral Edge Frequency
	Band Energy, gamma band
	Band Energy, beta band
	Hjorth Complexity
	Sample Entropy
	Approximate Entropy
	Mean
	Hjorth Mobility
	Root-Mean Squared
	Decorrelation Time
	Power Spectrum, beta band
	Katz Fractal Dimension
	Power Spectrum, alpha band

Fz	Power Spectrum, alpha band
	Spectral Edge Frequency
	Mean
	Band Energy, alpha band
	Decorrelation Time
	Zero Crossings
	Peak-to-Peak Amplitude
	Power Spectrum, beta band
	Spectral Entropy
F3	Power Spectrum, alpha band
	Spectral Edge Frequency
	Band Energy, alpha band
	Mean
	Decorrelation Time
	Peak-to-Peak Amplitude
	Spectral Entropy
	Hjorth Mobility
	Power Spectrum, beta band
	Zero Crossings
	Higuchi Fractal Dimension
F7	Power Spectrum, alpha band
	Spectral Edge Frequency
	Mean
	Decorrelation Time
	Hjorth Mobility
	Zero Crossings
	Band Energy, beta band
	Power Spectrum, beta band
	Peak-to-Peak Amplitude
	Hjorth Complexity
	Spectral Slope
	Power Spectrum, gamma band
	Katz Fractal Dimension
	Spectral Slope

	SVD Entropy
	Sample Entropy
FT9	Spectral Edge Frequency
	Mean
	Power Spectrum, alpha band
	Zero Crossings
	Decorrelation Time
	Higuchi Fractal Dimension
FC5	Spectral Edge Frequency
	Zero Crossings
	Mean
FC1	Power Spectrum, alpha band
	Spectral Edge Frequency
	Decorrelation Time
	Band Energy, alpha band
	Peak-to-Peak Amplitude
	Mean
	Zero Crossings
C3	Power Spectrum, alpha band
	Spectral Edge Frequency
	Decorrelation Time
	Peak-to-Peak Amplitude
	Mean
	Band Energy, alpha band
	Zero Crossings
T7	Spectral Edge Frequency
	Zero Crossings
	Mean
	Power Spectrum, alpha band
	Decorrelation Time
TP9	Power Spectrum, alpha band
	Spectral Edge Frequency
	Band Energy, alpha band
	Decorrelation Time

	Zero Crossings
	Mean
CP5	Spectral Edge Frequency
	Band Energy, beta band
	SVD Entropy
	Zero Crossings
	Band Energy, gamma band
	Power Spectrum, alpha band
	SVD Fisher Information
	Decorrelation Time
	Power Spectrum, beta band
	Hjorth Mobility
	Hjorth Complexity
	Line Length
	Katz Fractal Dimension
	Spectral Slope
	Mean
	Approximate Entropy
CP1	Mean
	Spectral Edge Frequency
	Zero Crossings
	Decorrelation Time
Pz	Mean
	Spectral Edge Frequency
	Power Spectrum, alpha band
	Zero Crossings
	Skewness
	Decorrelation Time
P3	Power Spectrum, alpha band
	Spectral Edge Frequency
	Decorrelation Time
	Band Energy, alpha band
	Peak-to-Peak Amplitude
	Peak-to-Peak Amplitude

	Power Spectrum, alpha band
	Spectral Edge Frequency
	Decorrelation Time
	Zero Crossings
	Band Energy, alpha band
	Mean
O1	Mean
	Spectral Edge Frequency
	Decorrelation Time
	Zero Crossings
Oz	Mean
	Peak-to-Peak Amplitude
	Spectral Edge Frequency
	Root-Mean Squared
	Band Energy, gamma band
	Zero Crossings
	Decorrelation Time
O2	Spectral Edge Frequency
	Zero Crossings
	Power Spectrum, alpha band
	Skewness
	Mean
	Higuchi Fractal Dimension
	Decorrelation Time
P4	Spectral Edge Frequency
	Power Spectrum, alpha band
	Zero Crossings
	Decorrelation Time
	Mean
P8	Mean
	Spectral Edge Frequency
	Power Spectrum, alpha band
	Zero Crossings
	SVD Entropy

	Decorrelation Time
	Higuchi Fractal Dimension
	Skewness
TP10	Spectral Edge Frequency
	Power Spectrum, alpha band
	Zero Crossings
	Decorrelation Time
	Mean
CP6	Mean
	Power Spectrum, alpha band
	Spectral Edge Frequency
	Decorrelation Time
	Zero Crossings
	Peak-to-Peak Amplitude
	Higuchi Fractal Dimension
CP2	Spectral Edge Frequency
	Zero Crossings
	Mean
	Peak-to-Peak Amplitude
Cz	Spectral Edge Frequency
	SVD Entropy
	Zero Crossings
	Power Spectrum, alpha band
	Higuchi Fractal Dimension
	Decorrelation Time
	Root-Mean Squared
	SVD Fisher Information
C4	Spectral Edge Frequency
	Peak-to-Peak Amplitude
	Mean
	Zero Crossings
	Decorrelation Time
T8	Spectral Edge Frequency
	Zero Crossings

T8

	Decorrelation Time
	Band Energy, gamma band
	Mean
FT10	Spectral Edge Frequency
	Mean
	Zero Crossings
FC6	Spectral Edge Frequency
	Mean
	Zero Crossings
FC2	Spectral Edge Frequency
	Mean
	Zero Crossings
	Peak-to-Peak Amplitude
	Decorrelation Time
	Power Spectrum, alpha band
F4	Spectral Edge Frequency
	Mean
	Zero Crossings
	Peak-to-Peak Amplitude
F8	Power Spectrum, alpha band
	Spectral Edge Frequency
	Zero Crossings
	Decorrelation Time
	Mean
Fp2	Mean
	Peak-to-Peak Amplitude
	Spectral Edge Frequency
	Zero Crossings
AF7	Mean
	Power Spectrum, alpha band
	Spectral Edge Frequency
	Decorrelation Time
	Zero Crossings
AF3	Spectral Edge Frequency

	Mean
	Zero Crossings
AFz	Power Spectrum, alpha band
	Spectral Edge Frequency
	Decorrelation Time
	Skewness
F1	Spectral Edge Frequency
	Mean
	Zero Crossings
F5	Spectral Edge Frequency
	Power Spectrum, alpha band
	Zero Crossings
	Mean
	Decorrelation Time
	Peak-to-Peak Amplitude
FT7	Mean
	Spectral Edge Frequency
	Skewness
	Zero Crossings
	Decorrelation Time
FC3	Spectral Edge Frequency
	Power Spectrum, alpha band
	Zero Crossings
	Decorrelation Time
C1	Spectral Edge Frequency
	Power Spectrum, alpha band
	Mean
	Peak-to-Peak Amplitude
	Decorrelation Time
	Zero Crossings
C5	Mean
	Power Spectrum, alpha band
	Spectral Edge Frequency
	Decorrelation Time

	Zero Crossings
TP7	Spectral Edge Frequency
	Mean
	Skewness
	Zero Crossings
CP3	Power Spectrum, alpha band
	Spectral Edge Frequency
	Mean
	Peak-to-Peak Amplitude
	Decorrelation Time
P1	Spectral Edge Frequency
	Mean
	Zero Crossings
	Peak-to-Peak Amplitude
P5	Spectral Edge Frequency
	Mean
	Power Spectrum, alpha band
	Peak-to-Peak Amplitude
	Zero Crossings
	Decorrelation Time
PO7	Power Spectrum, alpha band
	Spectral Edge Frequency
	Decorrelation Time
	Higuchi Fractal Dimension
	Zero Crossings
	Mean
PO3	Spectral Edge Frequency
	Zero Crossings
	Mean
PO4	Spectral Edge Frequency
	Mean
	Zero Crossings
PO8	Spectral Edge Frequency
	Mean

	Zero Crossings
P6	Spectral Edge Frequency
	Zero Crossings
	Mean
	Decorrelation Time
P2	Spectral Edge Frequency
	Skewness
	Mean
	Zero Crossings
CPz	Spectral Edge Frequency
	Peak-to-Peak Amplitude
	Zero Crossings
	Decorrelation Time
CP4	Spectral Edge Frequency
	Power Spectrum, alpha band
	Zero Crossings
	Decorrelation Time
TP8	Spectral Edge Frequency
	Mean
	Decorrelation Time
	Zero Crossings
C6	Spectral Edge Frequency
	Mean
	Decorrelation Time
	Power Spectrum, alpha band
C2	Spectral Edge Frequency
	Mean
	Power Spectrum, alpha band
FC4	Spectral Edge Frequency
	Decorrelation Time
	Peak-to-Peak Amplitude
	Mean
	Zero Crossings
	Spectral Edge Frequency

FT8

	Power Spectrum, alpha band
	Zero Crossings
F6	Spectral Edge Frequency
	Zero Crossings
	SVD Entropy
	SVD Fisher Information
	Power Spectrum, alpha band
	Decorrelation Time
AF8	Spectral Edge Frequency
	Power Spectrum, alpha band
	Mean
AF4	Spectral Edge Frequency
	Mean
F2	Spectral Edge Frequency
	Power Spectrum, alpha band
	Decorrelation Time
FCz	Mean
	Spectral Edge Frequency
	Power Spectrum, alpha band
	Decorrelation Time

Channel	Score
F7	10.14
C4	7.46
AF7	5.35
F8	4.73
C3	3.95
FT8	3.78
P7	3.69
CP3	3.47
CP5	3.32
FT9	3.14
FC6	2.53
CP1	2.19
C6	2.05
FC3	2.00
TP7	1.98
O2	1.97
P5	1.91
Pz	1.85
FT7	1.84
FC4	1.77
FC5	1.74
PO4	1.64
P3	1.63
T8	1.55
F3	1.47
TP8	1.45
P4	1.40
Fz	1.34
P1	1.24
Oz	1.16
CP6	1.14
Cz	1.14

Channel	Score
FC1	1.13
Fp1	1.11
T7	1.10
AF8	1.05
F5	1.03
TP10	0.96
TP9	0.96
O1	0.93
C1	0.93
F6	0.87
FCz	0.85
P8	0.85
C2	0.80
CP2	0.76
F1	0.70
PO3	0.68
CPz	0.65
CP4	0.65
P2	0.62
P6	0.61
Fp2	0.54
FC2	0.53
FT10	0.53
F2	0.52
F4	0.48
AF3	0.44
PO8	0.41
AFz	0.39
PO7	0.35
C5	0.33
AF4	0.25

Table 6.1: Channel scores for pilot B.

Channel	Score
Cz	15.24
C1	10.60
C3	7.70
CP5	6.08
T8	5.44
C5	3.93
TP7	3.84
Pz	3.83
FC6	3.74
CPz	3.69
CP1	3.65
C6	3.65
Fz	3.15
P1	3.13
FC1	3.11
F7	3.07
FC5	2.99
CP3	2.93
T7	2.88
C2	2.86
P5	2.81
CP6	2.73
FC3	2.72
F8	2.68
P7	2.59
FT9	2.59
FCz	2.55
TP9	2.50
F3	2.32
P2	2.29
CP2	2.29
P3	2.21

Channel	Score
AF7	2.19
Fp1	2.14
F1	2.12
F6	2.08
FT8	2.08
TP10	2.03
O1	1.97
P8	1.96
O2	1.95
Oz	1.92
PO8	1.79
TP8	1.77
FT10	1.76
AFz	1.74
F4	1.66
P4	1.63
PO4	1.63
PO3	1.61
FC2	1.61
C4	1.49
FC4	1.40
AF3	1.34
Fp2	1.33
F2	1.29
FT7	1.22
PO7	1.22
F5	1.17
AF8	1.05
P6	1.00
AF4	0.91
CP4	0.64

Table 6.2: Channel scores for pilot C.

Feature	Score
Spectral Edge Frequency	32.88
Mean	14.93
Peak-to-Peak Amplitude	13.84
Power Spectrum	10.38
Band Energy	8.88
Line Length	8.07
Higuchi FD	3.40
Variance	2.82
STD	2.77
Spectral Slope	2.76
Hjorth Mobility	1.73
Spectral Hjorth M.	1.51
Zero Crossings	1.34
Sample Entropy	1.03
RMS	0.74
Katz FD	0.51
Spectral Hjorth C.	0.30
Decorrelation Time	0.10
Skewness	0
Kurtosis	0
Quantile	0
Hurst Exponent	0
Approx. Entropy	0
Hjorth Complexity	0
SVD Entropy	0
SVD Fisher Info.	0
Spectral Entropy	0

Table 6.3: Feature scores for pilot B.

Feature	Score
Spectral Edge Frequency	48.76
Power Spectrum	27.89
Mean	27.09
Zero Crossings	18.25
Decorrelation Time	15.17
Band Energy	8.28
Peak-to-Peak Amplitude	8.02
SVD Entropy	3.79
SVD Fisher Info.	2.36
Hjorth Mobility	2.32
Skewness	2.11
Hjorth Complexity	1.79
Spectral Slope	1.30
Katz FD	1.30
Higuchi FD	1.29
RMS	1.24
Approx. Entropy	1.15
Sample Entropy	1.13
Spectral Entropy	0.94
Line Length	0.93
Quantile	0.37
Spectral Hjorth M.	0.02
Variance	0
STD	0
Kurtosis	0
Hurst Exponent	0
Spectral Hjorth C.	0

Table 6.4: Feature scores for pilot C.

Bibliography

- [1] G. R. Müller-Putz, “Electroencephalography,” *Handbook of clinical neurology*, vol. 168, pp. 249–262, 2020.
- [2] H. Becker, “Denoising, separation and localization of EEG sources in the context of epilepsy,” Theses, Université Nice Sophia Antipolis, Oct. 2014. [Online]. Available: <https://hal.science/tel-01077875>.
- [3] I. Iturrate, R. Chavarriaga, and J. del R. Millán, “Chapter 23 - general principles of machine learning for brain-computer interfacing,” in *Brain-Computer Interfaces*, ser. Handbook of Clinical Neurology, N. F. Ramsey and J. del R. Millán, Eds., vol. 168, Elsevier, 2020, pp. 311–328. DOI: <https://doi.org/10.1016/B978-0-444-63934-9.00023-8>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780444639349000238>.
- [4] L. Zhan Hong, A. Zourmand, J. Victor Patricks, and G. T. Thing, “Eeg-based brain wave controlled intelligent prosthetic arm,” in *2020 IEEE 8th Conference on Systems, Process and Control (ICSPC)*, 2020, pp. 52–57. DOI: [10.1109/ICSPC50992.2020.9305784](https://doi.org/10.1109/ICSPC50992.2020.9305784).
- [5] R. Bellman and R. Kalaba, “On adaptive control processes,” *IRE Transactions on Automatic Control*, vol. 4, no. 2, pp. 1–9, 1959.
- [6] J. Li, K. Cheng, S. Wang, *et al.*, “Feature selection: A data perspective,” *ACM Comput. Surv.*, vol. 50, no. 6, Dec. 2017, ISSN: 0360-0300. DOI: [10.1145/3136625](https://doi.org/10.1145/3136625). [Online]. Available: <https://doi.org/10.1145/3136625>.
- [7] C. Köllöd, A. Adolf, G. Márton, M. Wahdow, W. Fadel, and I. Ulbert, “Closed loop BCI system for Cybathlon 2020,” *Brain-Computer Interfaces*, vol. 10, no. 2-4, pp. 114–128, 2023, ISSN: 2326-263X. DOI: [10.1080/2326263X.2023.2254463](https://doi.org/10.1080/2326263X.2023.2254463).
- [8] C. M. Köllöd, A. Adolf, K. Iván, G. Márton, and I. Ulbert, “Deep comparisons of neural networks from the EEGNet family,” *Electronics*, vol. 12, no. 12, p. 2743, Jan. 2023, ISSN: 2079-9292. DOI: [10.3390/electronics12122743](https://doi.org/10.3390/electronics12122743).

- [9] A. Gramfort, M. Luessi, E. Larson, *et al.*, “MEG and EEG data analysis with MNE-Python,” *Frontiers in Neuroscience*, vol. 7, no. 267, pp. 1–13, 2013. DOI: 10.3389/fnins.2013.00267.
- [10] Z. J. Viharos, K. B. Kis, Á. Fodor, and M. I. Büki, “Adaptive, hybrid feature selection (ahfs),” *Pattern Recognition*, vol. 116, p. 107932, 2021, ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2021.107932>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320321001199>.
- [11] H. Nolan, R. Whelan, and R. Reilly, “Faster: Fully automated statistical thresholding for eeg artifact rejection,” *Journal of Neuroscience Methods*, vol. 192, no. 1, pp. 152–162, 2010, ISSN: 0165-0270. DOI: <https://doi.org/10.1016/j.jneumeth.2010.07.015>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165027010003894>.
- [12] K. Roots, Y. Muhammad, and N. Muhammad, “Fusion convolutional neural network for cross-subject eeg motor imagery classification,” *Computers*, vol. 9, no. 3, 2020, ISSN: 2073-431X. DOI: 10.3390/computers9030072. [Online]. Available: <https://www.mdpi.com/2073-431X/9/3/72>.
- [13] T. Verdonck, B. Baesens, M. Óskarsdóttir, and S. vanden Broucke, “Special issue on feature engineering editorial,” *Machine Learning*, vol. 113, no. 7, pp. 3917–3928, Jul. 2024, ISSN: 1573-0565. DOI: 10.1007/s10994-021-06042-2. [Online]. Available: <https://doi.org/10.1007/s10994-021-06042-2>.
- [14] N. Rao, R. Nowak, C. Cox, and T. Rogers, “Classification with the sparse group lasso,” *IEEE Transactions on Signal Processing*, vol. 64, no. 2, pp. 448–463, 2016. DOI: 10.1109/TSP.2015.2488586.
- [15] R. R. Zebari, A. M. Abdulazeez, Q. D. Zeebaree, D. A. Zebari, and J. N. Saeed, “A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction,” *Journal of Applied Science and Technology Trends*, vol. 1, no. 1, pp. 56–70, May 2020. DOI: 10.38094/jastt1224. [Online]. Available: <https://www.jastt.org/index.php/jasttpath/article/view/24>.
- [16] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O’Sullivan, “A review of feature selection methods for machine learning-based disease risk prediction,” *Frontiers in Bioinformatics*, vol. 2, 2022, ISSN: 2673-7647. DOI: 10.3389/fbinf.2022.927312. [Online]. Available: <https://www.frontiersin.org>.

org/journals/bioinformatics/articles/10.3389/fbinf.2022.927312.

- [17] R. Battiti, “Using mutual information for selecting features in supervised neural net learning,” *IEEE Transactions on neural networks*, vol. 5, no. 4, pp. 537–550, 1994.
- [18] C. Ding and H. Peng, “Minimum redundancy feature selection from microarray gene expression data,” in *Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003*, 2003, pp. 523–528. DOI: 10.1109/CSB.2003.1227396.
- [19] R. Schlittgen, “A weighted least-squares approach to clusterwise regression,” *AStA Advances in Statistical Analysis*, vol. 95, no. 2, pp. 205–217, 2011.
- [20] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 856–863.
- [21] X. Wan, C. Yang, Q. Yang, *et al.*, “Boost: A fast approach to detecting gene-gene interactions in genome-wide case-control studies,” *The American Journal of Human Genetics*, vol. 87, no. 3, pp. 325–340, 2010.
- [22] T. Schüpbach, I. Xenarios, S. Bergmann, and K. Kapur, “Fastepistasis: A high performance computing solution for quantitative trait epistasis,” *Bioinformatics*, vol. 26, no. 11, pp. 1468–1469, 2010.
- [23] X. Zhang, S. Huang, F. Zou, and W. Wang, “Team: Efficient two-locus epistasis tests in human genome-wide association study,” *Bioinformatics*, vol. 26, no. 12, pp. i217–i227, 2010.
- [24] K. Kira and L. A. Rendell, “The feature selection problem: Traditional methods and a new algorithm,” in *Proceedings of the tenth national conference on Artificial intelligence*, 1992, pp. 129–134.
- [25] I. Kononenko, “Estimating attributes: Analysis and extensions of relief,” in *European conference on machine learning*, Springer, 1994, pp. 171–182.
- [26] J. H. Moore and B. C. White, “Tuning relief for genome-wide genetic analysis,” in *European conference on evolutionary computation, machine learning and data mining in bioinformatics*, Springer, 2007, pp. 166–175.

- [27] C. S. Greene, N. M. Penrod, J. Kiralis, and J. H. Moore, "Spatially uniform relief (surf) for computationally-efficient filtering of gene-gene interactions," *BioData mining*, vol. 2, pp. 1–9, 2009.
- [28] C. S. Greene, D. S. Himmelstein, J. Kiralis, and J. H. Moore, "The informative extremes: Using both nearest and farthest individuals can improve relief algorithms in the domain of human genetics," in *European conference on evolutionary computation, machine learning and data mining in bioinformatics*, Springer, 2010, pp. 182–193.
- [29] R. J. Urbanowicz, R. S. Olson, P. Schmitt, M. Meeker, and J. H. Moore, "Benchmarking relief-based feature selection methods for bioinformatics data mining," *Journal of biomedical informatics*, vol. 85, pp. 168–188, 2018.
- [30] D. Granizo-Mackenzie and J. H. Moore, "Multiple threshold spatially uniform relief for the genetic analysis of complex human diseases," in *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics: 11th European Conference, EvoBIO 2013, Vienna, Austria, April 3-5, 2013. Proceedings 11*, Springer, 2013, pp. 1–10.
- [31] Y. Mao and Y. Yang, "A wrapper feature subset selection method based on randomized search and multilayer structure," *BioMed research international*, vol. 2019, no. 1, p. 9864213, 2019.
- [32] M. Xiong, X. Fang, and J. Zhao, "Biomarker identification by feature wrappers," *Genome research*, vol. 11, no. 11, pp. 1878–1887, 2001.
- [33] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 2, pp. 44–49, 1998.
- [34] R. Forsati, A. Moayedikia, R. Jensen, M. Shamsfard, and M. R. Meybodi, "Enriched ant colony optimization and its application in feature selection," *Neurocomputing*, vol. 142, pp. 354–371, 2014.
- [35] H. H. Htun, M. Biehl, and N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction," *Financial Innovation*, vol. 9, no. 1, p. 26, Jan. 2023, ISSN: 2199-4730. DOI: 10.1186/s40854-022-00441-7. [Online]. Available: <https://doi.org/10.1186/s40854-022-00441-7>.
- [36] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transac-*

- tions on pattern analysis and machine intelligence, vol. 27, no. 8, pp. 1226–1238, 2005.
- [37] X. V. Nguyen, J. Chan, S. Romano, and J. Bailey, “Effective global approaches for mutual information based feature selection,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’14, New York, New York, USA: Association for Computing Machinery, 2014, pp. 512–521, ISBN: 9781450329569. DOI: [10.1145/2623330.2623611](https://doi.org/10.1145/2623330.2623611). [Online]. Available: <https://doi.org/10.1145/2623330.2623611>.
 - [38] I. T. Jolliffe, *Principal component analysis for special types of data*. Springer, 2002.
 - [39] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.
 - [40] G. J. McLachlan, *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons, 2005.
 - [41] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, “Mutual information-based feature selection for intrusion detection systems,” *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1184–1199, 2011, Advanced Topics in Cloud Computing, ISSN: 1084-8045. DOI: <https://doi.org/10.1016/j.jnca.2011.01.002>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1084804511000038>.
 - [42] R. Battiti, “Using mutual information for selecting features in supervised neural net learning,” *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994. DOI: [10.1109/72.298224](https://doi.org/10.1109/72.298224).
 - [43] B. Senliol, G. Gulgezen, L. Yu, and Z. Cataltepe, “Fast correlation based filter (fcbf) with a different search strategy,” in *2008 23rd International Symposium on Computer and Information Sciences*, 2008, pp. 1–4. DOI: [10.1109/ISCIS.2008.4717949](https://doi.org/10.1109/ISCIS.2008.4717949).
 - [44] S. Egea, A. Rego Mañez, B. Carro, A. Sánchez-Esguevillas, and J. Lloret, “Intelligent iot traffic classification using novel search strategy for fast-based-correlation feature selection in industrial environments,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1616–1624, 2018. DOI: [10.1109/JIOT.2017.2787959](https://doi.org/10.1109/JIOT.2017.2787959).
 - [45] G. Bodenstein and H. Praetorius, “Feature extraction from the electroencephalogram by adaptive segmentation,” *Proceedings of the IEEE*, vol. 65, no. 5, pp. 642–652, 1977. DOI: [10.1109/PROC.1977.10543](https://doi.org/10.1109/PROC.1977.10543).

- [46] M. Rashid, N. Sulaiman, A. P. P. Abdul Majeed, *et al.*, “Current status, challenges, and possible solutions of eeg-based brain-computer interface: A comprehensive review,” *Frontiers in Neurorobotics*, vol. 14, 2020, ISSN: 1662-5218. DOI: 10.3389/fnbot.2020.00025. [Online]. Available: <https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2020.00025>.
- [47] J. Wang and M. Wang, “Review of the emotional feature extraction and classification using eeg signals,” *Cognitive Robotics*, vol. 1, pp. 29–40, 2021, ISSN: 2667-2413. DOI: <https://doi.org/10.1016/j.cogr.2021.04.001>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2667241321000033>.
- [48] A. Singh, A. A. Hussain, S. Lal, and H. W. Guesgen, “A comprehensive review on critical issues and possible solutions of motor imagery based electroencephalography brain-computer interface,” *Sensors*, vol. 21, no. 6, 2021, ISSN: 1424-8220. DOI: 10.3390/s21062173. [Online]. Available: <https://www.mdpi.com/1424-8220/21/6/2173>.
- [49] S. Mohdiwale, M. Sahu, G. R. Sinha, and V. Bhateja, “Statistical wavelets with harmony search- based optimal feature selection of eeg signals for motor imagery classification,” *IEEE Sensors Journal*, vol. 21, no. 13, pp. 14 263–14 271, 2021. DOI: 10.1109/JSEN.2020.3026172.
- [50] A. K. Singh and S. Krishnan, “Trends in eeg signal feature extraction applications,” *Frontiers in Artificial Intelligence*, vol. 5, 2023, ISSN: 2624-8212. DOI: 10.3389/frai.2022.1072801. [Online]. Available: <https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2022.1072801>.
- [51] E. Guttmann-Flury, X. Sheng, and X. Zhu, “Channel selection from source localization: A review of four eeg-based brain–computer interfaces paradigms,” *Behavior Research Methods*, vol. 55, no. 4, pp. 1980–2003, Jun. 2023, ISSN: 1554-3528. DOI: 10.3758/s13428-022-01897-2. [Online]. Available: <https://doi.org/10.3758/s13428-022-01897-2>.
- [52] M. H. Kabir, S. Mahmood, A. Al Shiam, A. S. Musa Miah, J. Shin, and M. K. I. Molla, “Investigating feature selection techniques to enhance the performance of eeg-based motor imagery tasks classification,” *Mathematics*, vol. 11, no. 8, 2023,

- ISSN: 2227-7390. DOI: 10.3390/math11081921. [Online]. Available: <https://www.mdpi.com/2227-7390/11/8/1921>.
- [53] C. Sun and C. Mou, "Survey on the research direction of eeg-based signal processing," *Frontiers in Neuroscience*, vol. 17, 2023, ISSN: 1662-453X. DOI: 10.3389/fnins.2023.1203059. [Online]. Available: <https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2023.1203059>.
 - [54] M. H. Kabir, N. I. Akhtar, N. Tasnim, *et al.*, "Exploring feature selection and classification techniques to improve the performance of an electroencephalography-based motor imagery brain-computer interface system," *Sensors*, vol. 24, no. 15, 2024, ISSN: 1424-8220. DOI: 10.3390/s24154989. [Online]. Available: <https://www.mdpi.com/1424-8220/24/15/4989>.
 - [55] R. Riener and L. J. Seward, "Cybathlon 2016," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 2792–2794. DOI: 10.1109/SMC.2014.6974351.
 - [56] *This is CYBATHLON — cybathlon.ethz.ch*, <https://cybathlon.ethz.ch/en/cybathlon>, [Accessed 12-11-2024].
 - [57] M. Lourakis and A. Argyros, "Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment?" In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, 2005, 1526–1531 Vol. 2. DOI: 10.1109/ICCV.2005.128.
 - [58] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *CoRR*, vol. abs/1912.01703, 2019. arXiv: 1912.01703. [Online]. Available: <http://arxiv.org/abs/1912.01703>.
 - [59] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001. DOI: 10.1109/51.932724.
 - [60] A. Gramfort, M. Luessi, E. Larson, *et al.*, "Meg and eeg data analysis with mne-python," *Frontiers in Neuroscience*, vol. 7, 2013, ISSN: 1662-453X. DOI: 10.3389/fnins.2013.00267. [Online]. Available: <https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2013.00267>.
 - [61] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/

- s41586-020-2649-2. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>.
- [62] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
 - [63] J.-B. Schiratti, J.-E. Le Douget, M. Le Van Quyen, S. Essid, and A. Gramfort, “An ensemble learning approach to detect epileptic seizures from long intracranial eeg recordings,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 856–860.
 - [64] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
 - [65] *ActiCHamp series / Brain Products GmbH*, <https://www.brainproducts.com/solutions/actichamp/>, [Accessed 11-11-2024].
 - [66] “Functional image file format description,” Elekta Neuroscience, Tech. Rep., 2011.
 - [67] K. Devarajan, B. Sanjeevirayar, V. Balasampath, J. E., and J. K., “Eeg-based epilepsy detection and prediction,” *International Journal of Engineering and Technology*, vol. 6, pp. 212–216, Jan. 2014. DOI: [10.7763/IJET.2014.V6.698](https://doi.org/10.7763/IJET.2014.V6.698).
 - [68] J. S. Richman and J. R. Moorman, “Physiological time-series analysis using approximate entropy and sample entropy,” *AJP Heart and Circulatory Physiology*, vol. 278, no. 6, H2039–H2049, Jun. 2000. DOI: [10.1152/ajpheart.2000.278.6.h2039](https://doi.org/10.1152/ajpheart.2000.278.6.h2039). [Online]. Available: <https://doi.org/10.1152/ajpheart.2000.278.6.h2039>.
 - [69] C. Teixeira, B. Direito, H. Feldwisch-Drentrup, *et al.*, “Epilab: A software package for studies on the prediction of epileptic seizures,” *Journal of Neuroscience Methods*, vol. 200, no. 2, pp. 257–271, 2011, ISSN: 0165-0270. DOI: <https://doi.org/10.1016/j.jneumeth.2011.07.002>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165027011003888>.
 - [70] N. Päävinen, S. Lammi, A. Pitkänen, J. Nissinen, M. Penttonen, and T. Grönfors, “Epileptic seizure detection: A nonlinear viewpoint,” *Computer Methods and Programs in Biomedicine*, vol. 79, no. 2, pp. 151–159, 2005, ISSN: 0169-2607. DOI: <https://doi.org/10.1016/j.cmpb.2005.04.006>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169260705001057>.

- [71] R. Esteller, G. Vachtsevanos, J. Echauz, and B. Litt, "A comparison of waveform fractal dimension algorithms," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 2, pp. 177–183, 2001. DOI: 10.1109/81.904882.
- [72] R. Esteller, J. Echauz, T. Tchong, B. Litt, and B. Pless, "Line length: An efficient feature for seizure onset detection," in *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2, 2001, 1707–1710 vol.2. DOI: 10.1109/IEMBS.2001.1020545.
- [73] S. J. Roberts, W. Penny, and I. Rezek, "Temporal and spatial complexity measures for electroencephalogram based brain-computer interfacing," *Medical & Biological Engineering & Computing*, vol. 37, no. 1, pp. 93–98, Jan. 1999, ISSN: 1741-0444. DOI: 10.1007/BF02513272. [Online]. Available: <https://doi.org/10.1007/BF02513272>.
- [74] A. Kharbouch, A. Shueb, J. Guttag, and S. S. Cash, "An algorithm for seizure onset detection using intracranial eeg," *Epilepsy & Behavior*, vol. 22, S29–S35, 2011, The Future of Automated Seizure Detection and Prediction, ISSN: 1525-5050. DOI: <https://doi.org/10.1016/j.yebeh.2011.08.031>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1525505011004951>.
- [75] F. Mormann, R. G. Andrzejak, C. E. Elger, and K. Lehnertz, "Seizure prediction: The long and winding road," *Brain*, vol. 130, no. 2, pp. 314–333, Sep. 2006, ISSN: 0006-8950. DOI: 10.1093/brain/awl241. eprint: <https://academic.oup.com/brain/article-pdf/130/2/314/798368/awl241.pdf>. [Online]. Available: <https://doi.org/10.1093/brain/awl241>.
- [76] C. Demanuele, C. J. James, and E. J. Sonuga-Barke, "Distinguishing low frequency oscillations within the 1/f spectral behaviour of electromagnetic brain signals," *Behavioral and Brain Functions*, vol. 3, no. 1, p. 62, Dec. 2007, ISSN: 1744-9081. DOI: 10.1186/1744-9081-3-62. [Online]. Available: <https://doi.org/10.1186/1744-9081-3-62>.
- [77] T. Inouye, K. Shinosaki, H. Sakamoto, *et al.*, "Quantification of eeg irregularity by use of the entropy of the power spectrum," *Electroencephalography and Clinical Neurophysiology*, vol. 79, no. 3, pp. 204–210, 1991, ISSN: 0013-4694. DOI: [https://doi.org/10.1016/0013-4694\(91\)90001-0](https://doi.org/10.1016/0013-4694(91)90001-0).

- doi.org/10.1016/0013-4694(91)90138-T. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/001346949190138T>.
- [78] C. M. Köllöd, “Development of brain-computer interfaces by using deep learning technologies,” Ph.D. dissertation, Roska Tamás Műszaki és Természettudományi Doktori Iskola, 2024. DOI: [10.15774/PPKE.ITK.2024.001](https://doi.org/10.15774/PPKE.ITK.2024.001).
- [79] Martín Abadi, Ashish Agarwal, Paul Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: <https://www.tensorflow.org/>.
- [80] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: [1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG]. [Online]. Available: <https://arxiv.org/abs/1412.6980>.
- [81] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, Atlanta, GA, vol. 30, 2013, p. 3.
- [82] R. Vallat, “Pingouin: Statistics in python,” *Journal of Open Source Software*, vol. 3, no. 31, p. 1026, Nov. 2018. DOI: [10.21105/joss.01026](https://doi.org/10.21105/joss.01026).
- [83] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965, ISSN: 00063444, 14643510. [Online]. Available: <http://www.jstor.org/stable/2333709> (visited on 11/08/2024).
- [84] H. B. Mann and D. R. Whitney, “On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other,” *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947. DOI: [10.1214/aoms/1177730491](https://doi.org/10.1214/aoms/1177730491). [Online]. Available: <https://doi.org/10.1214/aoms/1177730491>.
- [85] F. T. Hvaring and A. H. Ulltveit-Moe, “A comparison of visual evoked potential (vep)-based methods for the low-cost emotiv epoc neuroheadset,” 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:60143104>.
- [86] Student, “The probable error of a mean,” *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908, ISSN: 00063444, 14643510. [Online]. Available: <http://www.jstor.org/stable/2331554> (visited on 11/08/2024).