computers

MDPI

Article

Personalized Classifier Selection for EEG-Based BCIs

Javad Rahimipour Anaraki [1,2,*], Antonina Kolokolova [3] and Tom Chau [1,2,*]

[1] Institute of Biomedical Engineering, University of Toronto, Toronto, ON M5S 3G9, Canada
[2] Bloorview Research Institute, Holland Bloorview Kids Rehabilitation Hospital, Toronto, ON M4G 1R8, Canada
[3] Department of Computer Science, Memorial University of Newfoundland, St. John's, NL A1B 3X5, Canada; kol@mun.ca
* Correspondence: j.rahimipour@utoronto.ca (J.R.A.); tom.chau@utoronto.ca (T.C.)

Abstract: The most important component of an Electroencephalogram (EEG) Brain–Computer Interface (BCI) is its classifier, which translates EEG signals in real time into meaningful commands. The accuracy and speed of the classifier determine the utility of the BCI. However, there is significant intra- and inter-subject variability in EEG data, complicating the choice of the best classifier for different individuals over time. There is a keen need for an automatic approach to selecting a personalized classifier suited to an individual's current needs. To this end, we have developed a systematic methodology for individual classifier selection, wherein the structural characteristics of an EEG dataset are used to predict a classifier that will perform with high accuracy. The method was evaluated using motor imagery EEG data from Physionet. We confirmed that our approach could consistently predict a classifier whose performance was no worse than the single-best-performing classifier across the participants. Furthermore, Kullback–Leibler divergences between reference distributions and signal amplitude and class label distributions emerged as the most important characteristics for classifier prediction, suggesting that classifier choice depends heavily on the morphology of signal amplitude densities and the degree of class imbalance in an EEG dataset.

Keywords: EEG data; classification; algorithm portfolio; brain–computer interface

check for updates

Citation: Rahimipour Anaraki, J.; Kolokolova, A.; Chau, T. Personalized Classifier Selection for EEG-Based BCIs. *Computers* **2024**, *13*, 158. https://doi.org/10.3390/computers13070158

Academic Editor: Hersh Sagreiya Sagreiya

Received: 7 May 2024
Revised: 16 June 2024
Accepted: 20 June 2024
Published: 21 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Brain–Computer Interfaces (BCIs) are a type of assistive technology that allows individuals with profound motor impairments to directly use mental activity to control external devices and interact with their world [1]. BCIs infer movement or communication intentions directly from brain signals, typically Electroencephalogram (EEG), in real time [2]. EEG signals are measured from the scalp of the individual via electrodes. EEG signals are the spatial and temporal summations of thousands of synchronous, excitatory, and inhibitory post-synaptic potentials, mostly due to extra-cellular currents associated with the synaptic activity of pyramidal neurons [3]. Furthermore, EEG signals are susceptible to the blurring effects of volume conduction as the electrodes are far away from the signal sources [4]. Thus, EEG signals are inherently noisy and non-stationary (the statistics changing over time), with time-varying spectra and spatial distributions, making their classification challenging [1].

At the heart of the BCI is the classifier that translates the incoming stream of EEG data into functional commands (e.g., selection of words or control of an external device). Numerous classifiers have been deployed in BCIs [5], as the classification of EEG data is generally a difficult task, where no single classifier works well across all users [6]. Specifically, BCI performance for a given classifier and task often varies greatly across individuals [7]. This widespread inter-subject variability is manifested in the spectral characteristics of task-related EEG signals [8], the temporal features of evoked potentials [9], and the spatial distribution of sensorimotor-related activations [10]. Within-individuals, within-day, and

between-day EEG signals are also inherently non-stationary [11]. For example, children undergo developmental brain changes such as neurogenesis, neural migration, pruning, and myelin formation [12], while adults experience widespread regional brain volume reductions with aging [13]. Deciding on the best classifier is thus a particularly challenging and time-consuming problem for EEG BCIs.

To deal with this rampant intra- and inter-subject variability, some BCI studies have exploited transfer learning [14]. Chen and colleagues investigated the cross-subject distribution shift problem and proposed a solution based on deep adaptation networks [15], using a custom loss function to decrease both classification and adaptation losses concurrently [16]. In another work, by George et al., transfer learning in online and offline fashions was utilized to improve the classification accuracies of three deep neural networks—namely, BiGRU, Deep Net [17], and Multibranch CNN [18]—to tackle the non-stationary nature of motor imagery tasks within and across sessions and subjects, respectively [19]. Alternatively, others have proposed between-session updates of the trained classifier [11,20]. However, the choice of classifier remains unaddressed in such schemes. User-dependent classifiers generally tend to outperform user-independent classifiers [21], necessitating a personalized classifier selection. To this end, a scheme for expediently predicting the most accurate classifier for a given user at a given time of day would be valuable. For this paper, we leveraged empirical algorithmics and algorithm portfolio methods to design a framework that can automatically decide on the most accurate classifier (see Figure 1) for the BCI dataset at hand, on the basis of the structural characteristics of the dataset. The main contributions of this work are twofold:

- A systematic approach for classifier selection using structural characteristics of data is proposed;
- The applicability of our classifier selection method against 109 BCI2000 EEG datasets is evaluated.

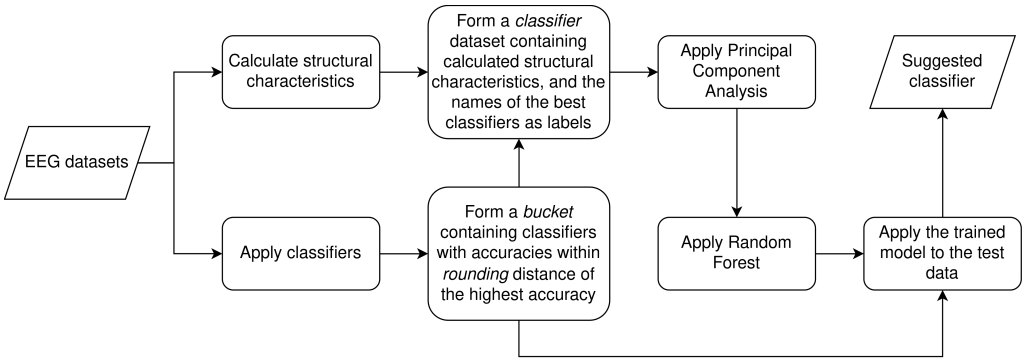


Figure 1. Flowchart of the proposed methodology.

2. Related Work

2.1. Algorithm Portfolios

One promising strategy for classifier selection is to use machine learning methods to learn from data and subsequently advise users on the most suitable algorithm without having to actually apply different methods to the data. One of the early works to address algorithm selection was done by Rice [22], who applied approximation theory to the algorithm selection problem, based on problem space, algorithm space, and performance-measures space. Later, Gomez and Selman [23] employed an algorithm portfolio for hard combinatorial search problems and showed that algorithms with higher variance in running time, such as stochastic algorithms, are advantageous over best-bound approaches. Leyton-Brown et al. also proposed an algorithm portfolio approach, reporting that a set of algorithms with a selection heuristic collectively outperforms its constituent algorithms on the combinatorial auction Winner Determination Problem (WDP) [24]. The algorithm portfolio approach has been successfully applied to solving various problems,

including propositional satisfiability [25,26], automated mission planning [27], university timetabling [28], traveling salesman [29], subgraph isomorphisms [30], collaborative filtering [31], dynamic maximal covering location [32], and human behavior and syllogistic reasoning [33]. The main advantage of combining algorithms in a portfolio is the reduction of computational cost while boosting classification accuracy [23].

Empirical or experimental algorithmics entail the use of empirical methods, such as statistics, to investigate the behaviors of algorithms [34–38]. Empirical algorithmics have found widespread application across various problem domains, including, for example, automating performance bottle-neck detection [39], input-sensitive profiling [40], and computational phylogenetics [41]. By statistically analyzing the behavior of different classifiers, one can distill a subset of classifiers, i.e., a portfolio, from which the most appropriate classifier for a given problem and dataset can be selected.

2.2. Automated Machine Learning

Automated Machine Learning (Auto-ML) is a field where the selection of machine learning tasks, such as feature selection and classification/regression and hyper-parameter tuning, are achieved using optimization techniques, thereby obviating the need for human expert input [42]. Auto-ML finds the best combination of machine learning tasks to maximize classification/regression accuracy. However, it is computationally expensive, which makes it less applicable to processing EEG data on the fly. Although imposing a time limit for Auto-ML is an option to bound the computational requirements, this comes at the expense of inferior accuracy.

Rooted in the Bayesian optimization methods of [43], Auto-WEKA [42] simultaneously addresses model selection and parameter optimization. Auto-WEKA started a trend of optimizing as many aspects of the machine learning pipeline as possible, from data preprocessing to architecture selection to hyper-parameter tuning. Later, in Auto-WEKA 2.0, the authors added regression methods, different performance metrics, and parallelism [44]. Auto-sklearn is an improvement of Auto-WEKA 2.0, incorporating meta-learning to boost Bayesian optimization [45]. For example, a subset of methods working well on a new dataset, based on previously computed results, is determined and passed to the optimization step. In the end, an ensemble is created automatically to provide more robust results compared to Auto-WEKA. Recently, Auto-ML methods have also been used to boost the performance of Deep Learning (DL) systems for applications such as image classification and natural language processing [46]. A benchmark of Auto-ML methods can be found in [47].

Auto-ML methods are well-suited to non-experts in ML and to problems where a streamlined end-to-end pipeline of pre-processing and high-accuracy classification/regression are needed. However, with EEG data processing, pre-processing is uniquely designed for each study based on the mental tasks, a challenge beyond AutoML. Considering the surveyed works above, there is a paucity of research on data-driven classifier selection, particularly for EEG processing for BCIs. We thus propose a potential solution using empirical algorithmics.

2.3. Algorithmic Fairness

While the motivation for finding personalized best classifiers for EEG data is very much in the spirit of algorithmic fairness [48,49] and the multicalibration/multiaccuracy line of work [50], our setting is somewhat different. Rather than samples corresponding to individuals and classes to protected groups, here we are dealing with a series of recordings for each person, which are samples that are not necessarily computationally identifiable as belonging to a specified group/person. Instead, we look at the aggregate features of the data belonging to a specific person and use that information to estimate which classifier is likely to give the best accuracy over a set of unseen samples from a distribution with those features. In our experiments, this method outperforms using the best overall classifier, as shown in Figure 2:

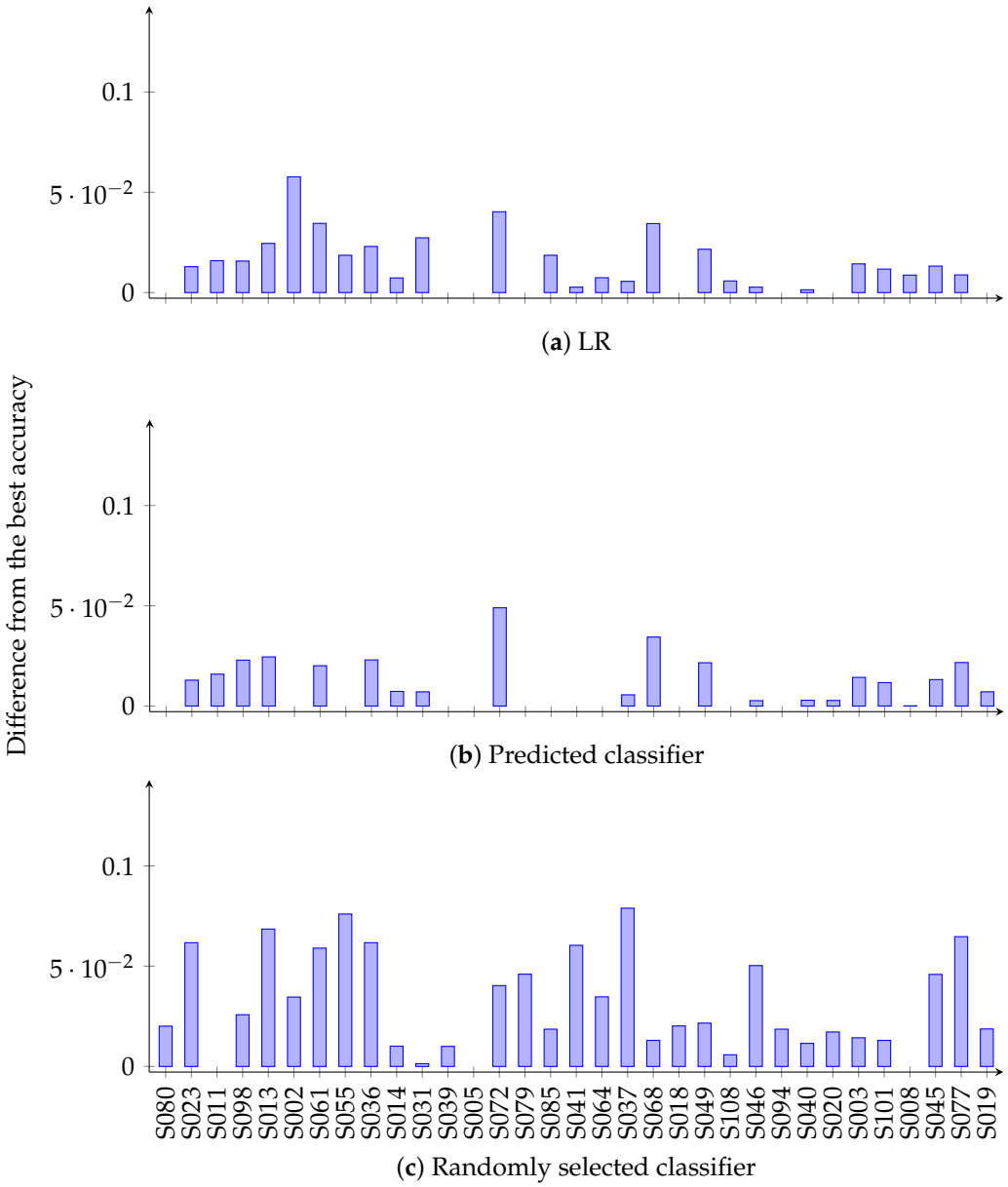


Figure 2. The difference in accuracy from that of the best classifier: (**a**) Logistic Regression (LR). (**b**) Predicted (*rounding* = 0.01). (**c**) Randomly selected classifiers for each dataset in the test set.

3. Methodology

3.1. EEG Data

For the analyses described herein, we used data from the BCI2000 [51] dataset. Briefly, this dataset consists of 64-channel Electroencephalogram (EEG) recordings sampled at 160 Hz from 109 subjects, performing three 2 min runs of hand and feet motor execution and imagery tasks. Here, we only consider the hand imagery tasks (imagining opening and closing of the left or right fist) and, for each participant, we merged their data from the respective runs (tasks 4, 8, and 12, in terms of the BCI2000 vernacular). In most EEG datasets considered, classes were imbalanced according to the ratio 1.8:1.0:1.05 (number of samples in class 1 to class 2 to class 3). We then applied independent component analysis (ICA), retaining enough components to attain a cumulative explained variance just below 99%. Subsequently, we down-sampled the signals to 10 Hz. The resulting data were merged into one data file for each participant. The EEG data for one participant is herein referred to as an EEG dataset. This is to be distinguished from a *classifier* dataset, explained in Section 3.2.3. Incidentally, given the EEG datasets described above, the EEG ternary

classification problem was that of distinguishing among rest, left-hand, and right-hand motor imagery.

3.2. Proposed Classifier Selection Method

To the best of our knowledge, there have been no investigations of classifier selection for EEG datasets that specifically leverage the characteristics of the data. Several efforts have sought suitable pairings of algorithms to datasets [52–54] but none of them took an empirical and automated approach to finding a match between an algorithm and an EEG dataset's characteristics. Here, we propose such a framework, in which each EEG dataset is characterized by its structural properties.

3.2.1. EEG Dataset Characteristics

First, we generated 41 structural characteristics of the dataset (see Table 1), agnostic to the source of the data, which is a necessary step for any arbitrary dataset. These dataset descriptors fall into three categories: learnability of the dataset, properties of signal features, and informativeness of class labels.

Table 1. Generated structural characteristics for each EEG dataset.

Characteristic	Description
m	Number of samples
n	Number of features
$\frac{m}{n}$	Ratio of number of samples to the number of features
`nClass`	Number of unique class labels
`meanAll`	Average of the values of all features
`avgSTDAll`	Average standard deviation of each feature
`entropyClass`	Entropy of class labels
`Q75`	Upper quantile (75%) of class labels
`Q25`	Lower quantile (25%) of class labels
`Q75-Q25`	Difference between upper and lower quantiles of class
`ChiClass`	Chi-square value of class label
`avgChiAll`	Average chi-square of all features
`medChiAll`	Median chi-square of all features
`stdChiAll`	Standard deviation of chi-square of all features
`minChiAll`	Minimum chi-square of all features
`maxChiAll`	Maximum chi-square of all features
`avgPearCorrClass`	Average Pearson correlation of all features to class
`avgPearCorrAll`	Average of Pearson inter-correlation of all features
`avgKendCorrClass`	Average of Kendall correlation of all features to class
`avgKendCorrAll`	Average of Kendall inter-correlation of all features
`avgSpeaCorrClass`	Average of Spearman correlation of all features to class
`avgSpeaCorrAll`	Average of Spearman inter-correlation of all features
`avgCovClass`	Average covariance of all features to class
`avgCovAll`	Average inter-covariance of all features
`avgKLNormAll`	Average KL of all features to Normal distribution
`avgKLUnifAll`	Average KL of all features to Uniform distribution
`avgKLLogiAll`	Average KL of all features to Logistic distribution
`avgKLExpoAll`	Average KL of all features to Exponential distribution
`avgKLChiAll`	Average KL of all features to Chi-square distribution
`avgKLRaylAll`	Average KL of all features to Rayleigh distribution
`avgKLParetAll`	Average KL of all features to Pareto distribution
`avgKLZipfAll`	Average KL of all features to Zipf distribution
`KLNormClass`	KL of class to Normal distribution
`KLUnifClass`	KL of class to Uniform distribution
`KLLogiClass`	KL of class to Logistic distribution
`KLExpoClass`	KL of class to Exponential distribution

Table 1. *Cont.*

Characteristic	Description
KLChiClass	KL of class to Chi-square distribution
KLRaylClass	KL of class to Rayleigh distribution
KLParetClass	KL of class to Pareto distribution
KLParetClass	KL of class to Zipf distribution
radComClass	Rademacher complexity of class labels

Learnability

In the first category, we included the number of samples, signal features, and classes, as well as the ratio of the number of samples to signal features. Learnable concepts can be defined using the Probably Approximately Correct (PAC) framework, in terms of sample complexity and time and space complexity, which depend on the cost of the representation of the concepts [55].

Properties of EEG Signal Features

In the second category, we focused on the properties of the signal features included in the dataset. We calculated the average of all the features and the average standard deviation and covariance of all the features. To evaluate the intra-correlation and redundancy of the features, we calculated the average chi-square and the inter-feature Pearson, Kendall, and Spearman correlations. The feature-to-class correlations were also included, to gauge the importance of features in predicting class labels. To render our approach agnostic to the data source, we also included the average Kullback–Leibler (KL) divergence [56] between all feature distributions and normal, uniform, logistic, exponential, chi-square, Rayleigh, Pareto, and Zipf distributions.

Informativeness of Class Labels

The last category focused on the properties of class labels. We calculated the entropy of class labels and characterized their skewness via upper and lower quantiles and chi-square values. Additionally, we included the average covariance of all features in the class and the Kullback–Leibler divergence between class label distributions and various distributions. The Rademacher complexity of class labels was also considered as a measure of the richness of class labels and their similarity to a randomly generated vector. After generating 41 characteristics for each dataset, we applied 22 classifiers to each dataset, to determine the best classifier, in terms of classification accuracy, as described next.

3.2.2. EEG Classifiers

We selected 22 classifiers (see Table 2) from ensemble, linear, Naive Bayes, nearest neighbors, neural networks, and tree-based classifiers, representing the most commonly used classifiers in EEG data processing. For the ensemble methods, we included Ada Boost (AB), Extra Trees (ET), Random Forest (RF), and Gradient Boosting (GB with two criteria). For the linear methods, which are very common in EEG data processing [5], we employed Linear Discriminant Analysis (LDA; 2 variants, one with singular value decomposition and the other with least-squares solver), Logistic Regression (LR; 2 variants, one with L2 and the other with no penalty), Ridge Classifier (RC), and the regularized linear model with Stochastic Gradient Descent Learning (SGD). As surveyed by multiple review papers, Naive Bayes, nearest neighbors, neural networks, and tree-based classifiers have been extensively used in emotion recognition [57,58], steady-state visual evoked potential [59], and motor imagery EEG classification [6]. Therefore, we included Bernoulli Naive Bayes (BNB), Complement Naive Bayes (CNB), Gaussian Naive Bayes (GNB), and Multinomial Naive Bayes (MNB) from the Naive Bayes family of methods; K-Neighbors Classifier (KN with K = 5 and BallTree, KDTree, and brute-force search algorithms) and Nearest Centroid (NC) from the nearest neighbor-based classifiers; Multi-Layer Perceptron (MLP) from the

neural networks classifiers; and Decision Tree (DT; 2 variants, one with Gini index and the other with entropy as impurity measures) from the tree-based classifiers.

Table 2. Scikit-learn [60] classifiers and their parameters used for the experiments.

Category	Classifier	Parameters
Ensemble	AB	algorithm = SAMME.R
	ET	n_estimators = 100, criterion = gini
	RF	n_estimators = 100, criterion = gini, bootstrap = True
	GB	loss = deviance, friedman_mse
	GB	loss = deviance, criterion = mse
Linear	LDA	solver=lsqr, tol = 1.0×10^{-4}
	LDA	solver = svd, tol = 1.0×10^{-4}
	LR	penalty = none, max_iter = 100
	LR	penalty = l2, max_iter = 100
	RC	alpha = 1.0, normalize = F, solver = auto
	SGD	loss = hinge, penalty = l2, alpha = 0.0001, l1_ratio = 0.15
Naive Bayes	BNB	alpha = 1.0, binarize = 0.0, fit_prior = T
	CNB	alpha = 1.0, fit_prior = T, norm = F
	GNB	var_smoothing = 1×10^{-9}
	MNB	alpha = 1.0, fit_prior = T
Nearest Neighbors	KN	algorithm = ball_tree
	KN	algorithm = kd_tree
	KN	algorithm = brute
	NC	–
Neural Networks	MLP	hidden_layer_sizes = 100, activation = relu, solver = adam, learning_rate_init = 0.001, max_iterint = 500
Tree	DT	criterion = gini, splitter = best
	DT	criterion = entropy, splitter = best

As the present study focused on classifier prediction, the downsampled (to 10 Hz) raw EEG signals were classified. Signals, each comprising 696 samples from 64 channels, were concatenated to form a 696×64 dimensional input vector for classification.

3.2.3. Classifier Dataset

All 22 classifiers were applied to each dataset, using 10-fold cross-validation (10-CV), and the name of the classifier achieving the highest accuracy was used as the label for the corresponding input EEG dataset from one participant. In other words, upon completion of this exercise, we obtained a *classifier* dataset, wherein a single instance comprised a vector of the 41 structural characteristics of a single participant's EEG dataset, paired with the name of the highest-accuracy classifier (as a categorical variable) for these EEG data. This *classifier* dataset contained 109 instances.

3.2.4. Predicting the Best Personalized Classifier

The classifier dataset was subjected to Principal Component Analysis (PCA) to extract the most informative structural characteristics of the EEG datasets. A Random Forest (RF) was implemented to classify the instances. In practice, predicting a classifier that performs almost as well as the best classifier is just as valuable as predicting the very best one. To account for that, we introduced a *rounding* variable, denoting the precision of the target accuracy. We formed a "bucket" for each instance and included any number of EEG classifiers whose accuracy fell within the *rounding* distance of that of the best classifier. When *rounding* was zero, only the actual label (i.e., the name of the single-best-performing EEG classifier) for a given EEG dataset was used, whether in the training or testing phase.

If *rounding* = t, where $t > 0$, then for the training phase we used the actual labels, but for the testing phase the predicted label for each instance was compared with the labels in the corresponding "bucket", which included the best classifier and those with accuracies at most t below the highest accuracy during the training phase. If the predicted classifier was among those in the "bucket", then the prediction was considered correct. We repeated this process 10 times for all the possible values of the number of extracted structural characteristics (i.e., 2 to 41), using PCA and a 70–30% split (76 samples for training and 33 samples for testing) of the classifier dataset. The proposed personalized classifier selection method (the code is available at https://github.com/jranaraki/PersonalizedClassifierSelection (accessed on 19 June 2024)) is summarized in Figure 1.

3.3. Environment

All the experiments were conducted using a machine with Ubuntu 22.04.4 LTS, Intel®Core™ i7-8750H, and 24GB of RAM. The proposed method was implemented in Python 3.12.2, and no proprietary library was used to run the experiments.

4. Experimental Results

Table 3 summarizes the frequency at which the classifiers were selected as the best, in terms of accuracy. The classifiers that were the least frequently selected as the best were Gradient Boosting (GB), Bernoulli NB (BNB), Multinomial NB (MNB), and Linear Discriminant Analysis (LDA). When GB, BNB, and MNB were selected as the best classifier, we noted that their accuracies were negligibly higher than that of the corresponding second-best classifier. As such, we discarded these classifiers from further consideration. On the other hand, LDA was retained, as its accuracy tended to be dramatically higher than that of the cognate second-best classifier. The revised counts of the number of times each of the top six classifiers were the most accurate are shown in Table 3. The average number of Floating Point Operations Per Second (FLOPS) for each method across all datasets is also reported.

Table 3. Frequency at which each classifier was selected as the most accurate, and the average of the FLOPS across all BCI2000 EEG datasets [51].

Classifier	Frequency		FLOPs
	All	Top 6	
Logistic Regression (LR)	28	31	7.29×10^5
Ridge Classifier (RC)	24	24	3.62×10^7
Multi-Layer Perceptron (MLP)	18	21	1.54×10^{10}
Random Forest (RF)	17	18	1.23×10^4
Extra Trees (ET)	11	12	1.22×10^4
Gradient Boosting (GB)	3	-	4.88×10^9
Bernoulli Naive Bayes (BNB)	3	-	3.75×10^6
Multinomial Naive Bayes (MNB)	3	-	3.62×10^6
Linear Discriminant Analysis (LDA; solver = lsqr)	2	2	4.71×10^7

The LR classifier was the best overall classifier for all participants. Using our method for each participant with *rounding* = 0.01, the accuracy of the predicted classifier exceeded that of LR by 0.0035 ± 0.0120, on average, with an average of 24.20 extracted features. Figure 2 illustrates the difference in classification accuracies from that of the best classifier for each dataset in the test set, sorted in terms of bucket size from smallest to largest.

Table 4 provides more details by presenting the classifiers included in each bucket for each participant, the best, randomly selected, and the predicted classifiers and their corresponding accuracies.

We performed an $N \times N$ Friedman test [61] to evaluate the differences between the accuracies of each approach on the test data (see Table 5 for the mean rank of each approach). Subsequently, Holm's post hoc pairwise comparisons were conducted (see Table 6) to account for multiple comparisons. Our proposed method (i.e., labeled as 'Predicted' in

Table 5) ranked higher than both the best overall classifier (i.e., LR) and the randomly selected classifiers. Based on the pairwise comparisons, all pairs were significantly different except for Predicted vs. LR, which confirmed that our method performed on a par with the best overall classifier.

Table 4. Predicted classifiers for each dataset using *rounding* = 0.01 and 24 extracted features using PCA, and accuracies of the best, randomly selected, predicted, and Logistic Regression (LR) classifiers sub-scripted. The bucket for each dataset could contain the Extra Trees classifier (ET), the Random Forest classifier (RF), the Linear Discriminant Analysis (LDA) classifier, the Logistic Regression (LR) classifier, the Ridge Classifier (RC), and the MLP classifier.

Dataset	Bucket	Best	Random	Predicted	LR
S005	LR, RC	$LR_{0.5000}$	$LR_{0.5000}$	$LR_{0.5000}$	0.5000
S045	RF, MLP, ET, RC	$RF_{0.4887}$	$LDA_{0.4428}$	$LR_{0.4755}$	0.4755
S072	MLP, ET	$MLP_{0.5115}$	$LR_{0.4712}$	$RC_{0.4625}$	0.4712
S018	LR, MLP	$LR_{0.6351}$	$RF_{0.6149}$	$LR_{0.6351}$	0.6351
S014	RF, LR	$RF_{0.4942}$	$MLP_{0.4841}$	$LR_{0.4869}$	0.4869
S037	RC, LR	$RC_{0.5342}$	$RF_{0.4553}$	$LR_{0.5286}$	0.5286
S108	MLP, RF, LR	$MLP_{0.4942}$	$LR_{0.4884}$	$MLP_{0.4942}$	0.4884
S002	ET	$ET_{0.5606}$	$MLP_{0.5260}$	$ET_{0.5606}$	0.5029
S040	ET, RF, LR	$ET_{0.5430}$	$RC_{0.5315}$	$RF_{0.5401}$	0.5416
S020	LR, RC, ET	$LR_{0.5446}$	$RF_{0.5274}$	$ET_{0.5418}$	0.5446
S077	RF, RC, ET, LR	$RF_{0.5147}$	$LDA_{0.4500}$	$MLP_{0.4930}$	0.5059
S011	MLP	$MLP_{0.5203}$	$MLP_{0.5203}$	$LR_{0.5044}$	0.5044
S098	RF	$RF_{0.5272}$	$ET_{0.5014}$	$RC_{0.5043}$	0.5115
S064	RC, LR	$RC_{0.5303}$	$ET_{0.4956}$	$RC_{0.5303}$	0.5229
S023	RC	$RC_{0.4900}$	$ET_{0.4283}$	$LR_{0.4771}$	0.4771
S013	RC	$RC_{0.5286}$	$LDA_{0.4601}$	$LR_{0.5041}$	0.5041
S019	LR, RC, MLP, RF	$LR_{0.5229}$	$RF_{0.5042}$	$RC_{0.5158}$	0.5229
S046	RC, MLP, LR	$RC_{0.5244}$	$ET_{0.4741}$	$LR_{0.5217}$	0.5217
S101	RF, MLP, ET	$RF_{0.5707}$	$MLP_{0.5577}$	$LR_{0.5590}$	0.5590
S061	MLP	$MLP_{0.5633}$	$RF_{0.5043}$	$RC_{0.5432}$	0.5288
S079	LR, RC	$LR_{0.5118}$	$RF_{0.4658}$	$LR_{0.5118}$	0.5118
S008	MLP, RC, LR	$MLP_{0.5935}$	$MLP_{0.5935}$	$RC_{0.5934}$	0.5848
S080	LR	$LR_{0.4741}$	$RC_{0.4540}$	$LR_{0.4741}$	0.4741
S094	LR, RF, ET	$LR_{0.5403}$	$MLP_{0.5217}$	$LR_{0.5403}$	0.5403
S055	RC	$RC_{0.5716}$	$ET_{0.4956}$	$RC_{0.5716}$	0.5530
S049	LR, RC, RF	$LR_{0.4885}$	$LR_{0.4885}$	$LR_{0.4885}$	0.4885
S041	RC, LR	$RC_{0.5260}$	$LDA_{0.4656}$	$RC_{0.5260}$	0.5233
S085	RF, ET	$RF_{0.5260}$	$LR_{0.5074}$	$RF_{0.5260}$	0.5074
S036	LDA, RC	$LDA_{0.4871}$	$ET_{0.4254}$	$LR_{0.4641}$	0.4641
S068	ET, RF	$ET_{0.5359}$	$RF_{0.5229}$	$LR_{0.5015}$	0.5015
S039	LR, RF	$LR_{0.4944}$	$RC_{0.4844}$	$LR_{0.4944}$	0.4944
S031	ET, RC	$ET_{0.4943}$	$MLP_{0.4929}$	$RC_{0.4872}$	0.4670
S003	MLP, RF, ET	$MLP_{0.5457}$	$LR_{0.5314}$	$LR_{0.5314}$	0.5314

Table 5. Average ranking of the algorithms, using the Friedman test.

Algorithm	Ranking
Best	1.4091
Predicted	2.4697
LR	2.6667
Random	3.4545

With *rounding* = 0.01 for most of the datasets, more than one classifier achieved the highest accuracy. By increasing the *rounding* from 0.00 to 0.04, the best, average and worst accuracies using RF are shown in Table 7. The rankings of the 41 structural characteristics (Table 1) of the EEG datasets are shown in Figure 3, based on the RF classification of the *classifier* dataset.

Table 6. Holm's post hoc comparisons. All pairwise comparisons were significantly different except for Predicted vs. LR, where $p > 0.05$.

i	Approaches Compared	$z = (R_0 - R_i)/SE$	p	Holm
6	Best vs. Random	6.4359	0	0.0083
5	Best vs. LR	3.9569	0.0001	0.0100
4	Best vs. Predicted	3.3371	0.0009	0.0125
3	Random vs. Predicted	3.0988	0.0019	0.01667
2	Random vs. LR	2.4790	0.0132	0.0250
1	Predicted vs. LR	0.6198	0.5354	0.0500

Table 7. The best, average, and worst accuracies of the Random Forest (RF) classifier on the *classifier* dataset with *rounding* ranging from 0.00 to 0.04. Structural characteristics are those from Table 1.

Rounding	# Structural Characteristics	Best	Average	Worst
0.00	11	0.4848	0.3121 ± 0.0791	0.2424
0.01	9	0.7576	0.5576 ± 0.1167	0.3939
0.02	9	0.9091	0.7515 ± 0.0776	0.6061
0.03	8	0.9697	0.8394 ± 0.0651	0.7576
0.04	9	1.0000	0.9121 ± 0.0497	0.8182

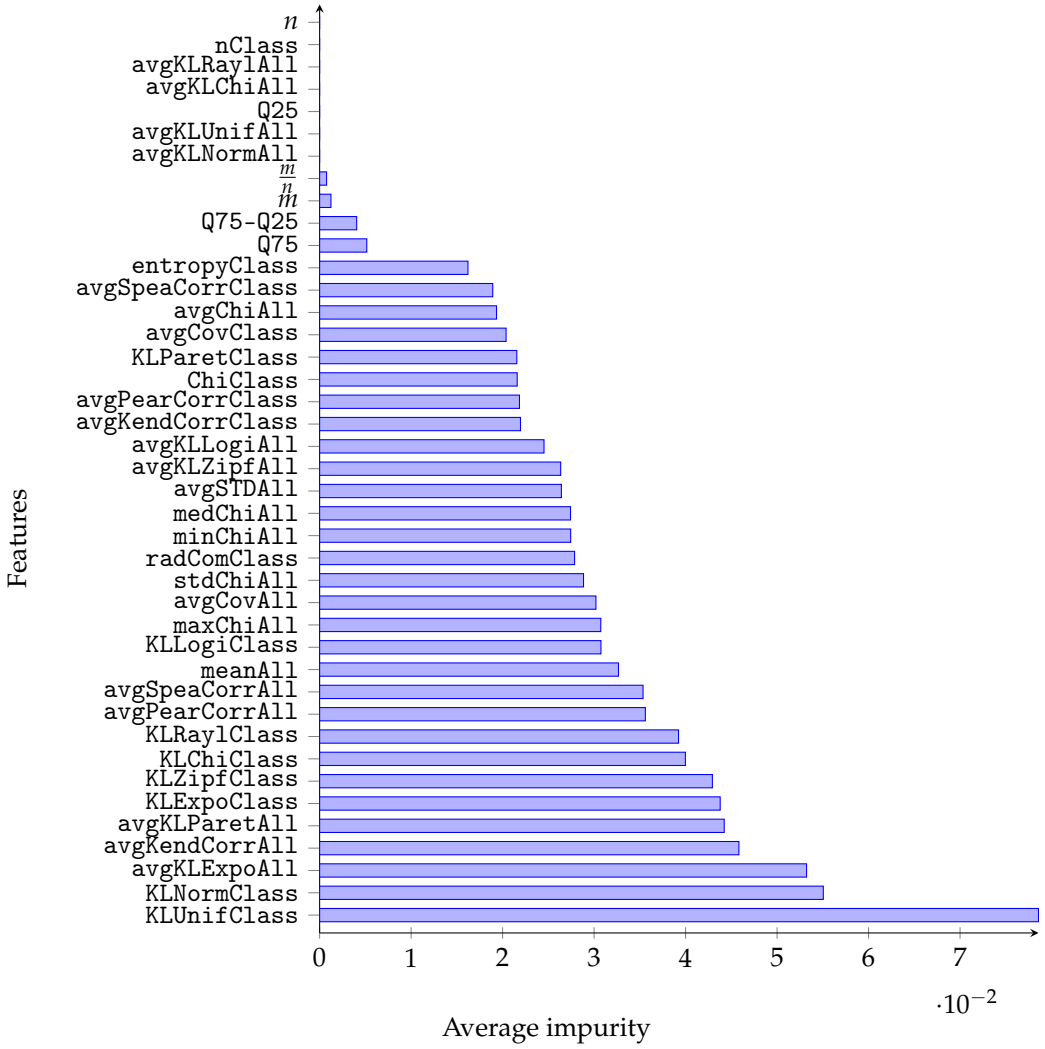


Figure 3. Average impurity of each structural characteristic based on the accuracy of the best classifier for each dataset from the BCI2000 repository [51].

5. Discussion

Finding the best classifier for the dataset at hand is a laborious task. Unsurprisingly, researchers often simply deploy the classifiers previously used for similar problems. To date, no empirical approaches systematically suggest a classifier based on the structural properties of EEG datasets. As a solution to this problem, we formed a *classifier* dataset of instances, each comprising a set of 41 structural characteristics of the EEG dataset and a target label (i.e., the best classifier for this dataset). Then, we applied feature extraction using PCA and introduced a *rounding* variable to account for variability in classification accuracies. By increasing *rounding* value, we allowed for more than one classifier to join the "bucket" (i.e., correct answers). We trained a Random Forest over the generated *classifier* dataset and compared the predicted classifiers with those in the "bucket". We evaluated our method on EEG datasets from BCI2000 [51].

5.1. Predicting a Classifier for a New User

Our findings suggest that it is indeed feasible to predict a classifier for a new EEG BCI user, strictly on the basis of the structural characteristics of their offline (i.e., training) EEG dataset (Figure 2 and Table 4). In other words, one could identify a person-specific classifier without the need for time-consuming experimentation (i.e., training and testing different classifiers). In fact, Tables 5 and 6 confirm that the proposed framework can predict a classifier that will perform no worse than the single-best-performing classifier across the participants. This is an important finding because it suggests that the proposed approach could allow BCI practitioners to quickly choose a subject-specific classifier once a cognate training dataset has been acquired, potentially accelerating the path to same-session online testing.

5.2. Most Predictive Structural Characteristics

From Figure 3, Kullback–Leibler measures feature prominently among the most important structural characteristics of the EEG dataset. The `KLUnifClass` and `KLNormClass` characteristics reflect the differences between the distribution of class labels (represented as integers 1, 2, and 3) and reference uniform and normal distributions, respectively. These characteristics can be interpreted as representing the degree of balance of samples across classes (i.e., if a dataset were completely balanced the distribution of class labels would be uniform). Our analyses thus seem to suggest that certain classifiers are preferred in the presence of class imbalances.

The `avgKLExpoAll` and `avgKLParetAll` characteristics represent how the different (across all classes) EEG amplitude distributions resemble exponential and Pareto distributions. Both distributions have one-sided, right-tailed densities that fall off as the distance from the mean increases. However, the Pareto density, $Pareto(x)$, has a heavy tail compared to an exponential density $exp(x)$ with the same mean and, thus, higher probabilities for large values of x. Our findings suggest that classifier choice hinges, in large part, on the shape of the EEG amplitude density, namely, where it lies between power law and exponential decay. The positive skewness of the amplitude density is associated with nonlinear temporal dynamics in the signal [62].

In sum, class balance and the morphology of signal amplitude distributions appear to be critical structural characteristics of an EEG dataset for classifier prediction.

5.3. The Elusive Best Classifier

The logistic regression classifier was the single-best classifier across the motor imagery EEG datasets. This finding corroborates previous motor imagery BCI research, which identified the logistic regression classifier as yielding the highest accuracy [63] and greatest receiver operating characteristic area [64] among other motor imagery classifiers. With the BCI2000 dataset, the choice of best classifier was seemingly not unique in many instances; more than one preferred classifier could be selected with little difference in accuracy. This could, in part, be attributable to the well-documented, clearly lateralized, and machine-

discernible event-related desynchronization and synchronization reflected in EEG signals accompanying motor imagery in adults [65]. For other BCI classification challenges, such as emotion recognition [66] or speech decoding [11], where common topographical patterns across participants are less probable, the performance difference among classifiers may be more evident.

5.4. Limitations and Future Work

We only considered a homogeneous dataset (i.e., BCI2000 [51]), where the same protocol and instrumentation were implemented across participants. As such, certain structural characteristics—namely, the number of features (n), the number of classes (`nClass`), the ratio of the number of samples to the number of classes ($\frac{m}{n}$), and the number of samples (m)—contributed negligibly to classifier prediction (Figure 3). The value of the proposed method would be more evident with heterogeneous datasets, comprising data from different subjects, dissimilar protocols, and varied instrumentation. Furthermore, the performance among classifiers would likely be more dramatic with heterogeneous datasets, rendering the choice of classifier even more critical.

We were able to predict a classifier that performed on a par with the single-best classifier across the participants. However, this classifier may not be the best classifier for an individual user. Future research ought to investigate the prediction of the highest accuracy classifier for a new user (i.e., with $rounding = 0$), as well as validating the proposed method on the data collected on a different day.

We only predicted the classifier but did not optimize other parts of the signal-processing pipeline on a per-user basis. The predicted classifier itself and the preceding filtering and feature extraction could be optimized via an AutoML method without the need for further data. This could be followed by studying metrics specific to evaluating imbalanced datasets. In this way, the proposed method could be applied to other challenging classification problems, such as MRI and genomic data classification, where additional data collection is costly or logistically challenging.

6. Conclusions

We showed that it is feasible to automatically predict a classifier based on the structural characteristics of an EEG dataset. Our proposed approach can recommend a subject-specific classifier, the average accuracy of which can surpass the average classification accuracy of the single-best classifier across the participants. Personalized classifier selection has the potential to reduce the time and effort required to optimize BCIs for specific individuals.

Author Contributions: Conceptualization, J.R.A., A.K. and T.C.; methodology, J.R.A. and A.K.; software, J.R.A.; validation, J.R.A., A.K. and T.C.; formal analysis, J.R.A., A.K. and T.C.; investigation, J.R.A., A.K. and T.C.; resources, T.C.; data curation, J.R.A.; writing—original draft preparation, J.R.A., A.K. and T.C.; writing—review and editing, J.R.A., A.K. and T.C.; visualization, J.R.A. and A.K.; supervision, A.K. and T.C.; project administration, T.C.; funding acquisition, J.R.A. and T.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded partially by Mitacs through the Mitacs Elevate program and the Holland Bloorview Kids Rehabilitation Hospital Foundation.

Data Availability Statement: The data presented in this study are available in PhysioNet at https://doi.org/10.13026/C28G6P (accessed on 19 June 2024). These data were derived from the following resource available in the public domain: https://physionet.org/content/eegmmidb/1.0.0/ (accessed on 19 June 2024).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Moghimi, S.; Kushki, A.; Marie Guerguerian, A.; Chau, T. A review of EEG-based brain-computer interfaces as access pathways for individuals with severe disabilities. *Assist. Technol.* **2013**, *25*, 99–110. [CrossRef] [PubMed]
2. Orlandi, S.; House, S.C.; Karlsson, P.; Saab, R.; Chau, T. Brain-computer interfaces for children with complex communication needs and limited mobility: A systematic review. *Front. Hum. Neurosci.* **2021**, *15*, 643294. [CrossRef] [PubMed]
3. Kirschstein, T.; Köhling, R. What is the source of the EEG? *Clin. EEG Neurosci.* **2009**, *40*, 146–149. [CrossRef] [PubMed]
4. Rutkove, S.B. Introduction to volume conduction. In *The Clinical Neurophysiology Primer*; Humana Press Inc.: Totowa, NJ, USA, 2007; pp. 43–53.
5. Lotte, F.; Bougrain, L.; Cichocki, A.; Clerc, M.; Congedo, M.; Rakotomamonjy, A.; Yger, F. A review of classification algorithms for EEG-based brain–computer interfaces: A 10 year update. *J. Neural Eng.* **2018**, *15*, 031005. [CrossRef] [PubMed]
6. Padfield, N.; Zabalza, J.; Zhao, H.; Masero, V.; Ren, J. EEG-based brain-computer interfaces using motor-imagery: Techniques and challenges. *Sensors* **2019**, *19*, 1423. [CrossRef] [PubMed]
7. Saha, S.; Ahmed, K.I.U.; Mostafa, R.; Hadjileontiadis, L.; Khandoker, A. Evidence of variabilities in EEG dynamics during motor imagery-based multiclass brain–computer interface. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2017**, *26*, 371–382. [CrossRef] [PubMed]
8. Myrden, A.; Chau, T. Feature clustering for robust frequency-domain classification of EEG activity. *J. Neurosci. Methods* **2016**, *262*, 77–84. [CrossRef] [PubMed]
9. Dandekar, S.; Ales, J.; Carney, T.; Klein, S.A. Methods for quantifying intra-and inter-subject variability of evoked potential data applied to the multifocal visual evoked potential. *J. Neurosci. Methods* **2007**, *165*, 270–286. [CrossRef] [PubMed]
10. Saha, S.; Baumert, M. Intra-and inter-subject variability in EEG-based sensorimotor brain computer interface: A review. *Front. Comput. Neurosci.* **2020**, *13*, 87. [CrossRef]
11. Sereshkeh, A.R.; Trott, R.; Bricout, A.; Chau, T. EEG classification of covert speech using regularized neural networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *25*, 2292–2300. [CrossRef]
12. Kolb, B.; Gibb, R. Brain plasticity and behaviour in the developing brain. *J. Can. Acad. Child Adolesc. Psychiatry* **2011**, *20*, 265.
13. Raz, N.; Lindenberger, U.; Rodrigue, K.M.; Kennedy, K.M.; Head, D.; Williamson, A.; Dahle, C.; Gerstorf, D.; Acker, J.D. Regional brain changes in aging healthy adults: General trends, individual differences and modifiers. *Cereb. Cortex* **2005**, *15*, 1676–1689. [CrossRef]
14. Zheng, M.; Yang, B.; Xie, Y. EEG classification across sessions and across subjects through transfer learning in motor imagery-based brain-machine interface system. *Med. Biol. Eng. Comput.* **2020**, *58*, 1515–1528. [CrossRef]
15. Long, M.; Cao, Y.; Wang, J.; Jordan, M. Learning transferable features with deep adaptation networks. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 97–105.
16. Chen, Y.; Yang, R.; Huang, M.; Wang, Z.; Liu, X. Single-source to single-target cross-subject motor imagery classification based on multisubdomain adaptation network. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 1992–2002. [CrossRef] [PubMed]
17. Lawhern, V.J.; Solon, A.J.; Waytowich, N.R.; Gordon, S.M.; Hung, C.P.; Lance, B.J. EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces. *J. Neural Eng.* **2018**, *15*, 056013. [CrossRef] [PubMed]
18. Schirrmeister, R.T.; Springenberg, J.T.; Fiederer, L.D.J.; Glasstetter, M.; Eggensperger, K.; Tangermann, M.; Hutter, F.; Burgard, W.; Ball, T. Deep learning with convolutional neural networks for EEG decoding and visualization. *Hum. Brain Mapp.* **2017**, *38*, 5391–5420. [CrossRef] [PubMed]
19. George, O.; Dabas, S.; Sikder, A.; Smith, R.; Madiraju, P.; Yahyasoltani, N.; Ahamed, S.I. Enhancing motor imagery decoding via transfer learning. *Smart Health* **2022**, *26*, 100339. [CrossRef]
20. Power, S.D.; Kushki, A.; Chau, T. Intersession consistency of single-trial classification of the prefrontal response to mental arithmetic and the no-control state by NIRS. *PLoS ONE* **2012**, *7*, e37791. [CrossRef]
21. Ravi, A.; Beni, N.H.; Manuel, J.; Jiang, N. Comparing user-dependent and user-independent training of CNN for SSVEP BCI. *J. Neural Eng.* **2020**, *17*, 026028. [CrossRef]
22. Rice, J.R. The algorithm selection problem. In *Advances in Computers*; Elsevier: Burlington, MA, USA, 1976; Volume 15, pp. 65–118.
23. Gomes, C.P.; Selman, B. Algorithm portfolios. *Artif. Intell.* **2001**, *126*, 43–62. [CrossRef]
24. Leyton-Brown, K.; Nudelman, E.; Andrew, G.; McFadden, J.; Shoham, Y. Boosting as a metaphor for algorithm design. In Proceedings of the International Conference on Principles and Practice of Constraint Programming, Kinsale, Ireland, 29 September–3 October 2003; pp. 899–903.
25. Nudelman, E.; Leyton-Brown, K.; Devkar, A.; Shoham, Y.; Hoos, H. Satzilla: An algorithm portfolio for SAT. *Solver Descr. SAT Compet.* **2004**, *2004*, 1–2.
26. Saleh, A.M.E.; Arashi, M.; Kibria, B.G. *Theory of Ridge Regression Estimation with Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2019; Volume 285.
27. Chien, S.; Cichy, B.; Davies, A.; Tran, D.; Rabideau, G.; Castano, R.; Sherwood, R.; Mandl, D.; Frye, S.; Shulman, S.; et al. An autonomous earth-observing sensorWeb. *IEEE Intell. Syst.* **2005**, *20*, 16–24. [CrossRef]
28. Qu, R.; Burke, E.K. Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems. *J. Oper. Res. Soc.* **2009**, *60*, 1273–1285. [CrossRef]
29. Xie, X.F.; Liu, J. Multiagent optimization system for solving the traveling salesman problem (TSP). *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2008**, *39*, 489–502.

30. Kotthoff, L.; McCreesh, C.; Solnon, C. Portfolios of subgraph isomorphism algorithms. In Proceedings of the International Conference on Learning and Intelligent Optimization, Ischia, Italy, 39 May–1 June 2016; pp. 107–122.
31. Mısır, M.; Sebag, M. *Algorithm Selection as a Collaborative Filtering Problem*; Technical Report; INRIA: Paris, France, 2013.
32. Calderín, J.F.; Masegosa, A.D.; Pelta, D.A. An algorithm portfolio for the dynamic maximal covering location problem. *Memetic Comput.* **2017**, *9*, 141–151. [CrossRef]
33. Riesterer, N.; Brand, D.; Ragni, M. The predictive power of heuristic portfolios in human syllogistic reasoning. In Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz), Berlin, Germany, 24–28 September 2018; pp. 415–421.
34. Zaparanuks, D.; Hauswirth, M. Algorithmic profiling. In Proceedings of the Programming Language Design and Implementation (PLDI), Beijing, China, 11–16 June 2012; pp. 67–76.
35. Jaśkowski, W.; Liskowski, P.; Szubert, M.; Krawiec, K. The performance profile: A multi–Criteria performance evaluation method for test–Based problems. *Int. J. Appl. Math. Comput. Sci.* **2016**, *26*, 215–229. [CrossRef]
36. Fleischer, R.; Moret, B.; Schmidt, E.M. *Experimental Algorithmics: From Algorithm Design to Robust and Efficient Software*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2547.
37. Moret, B.M. Towards a discipline of experimental algorithmics. In *Data Structures, Near Neighbor Searches, and Methodology*; American Mathematical Society: Providence, RI, USA, 2002; Volume 59, pp. 197–213.
38. Hromkovic, J. *Algorithmics for Hard Problems*; Texts in Theoretical Computer Science; Springer: Berlin/Heidelberg, Germany, 2002.
39. Shen, D.; Luo, Q.; Poshyvanyk, D.; Grechanik, M. Automating performance bottleneck detection using search-based application profiling. In Proceedings of the 2015 International Symposium on Software Testing and Analysis, Baltimore, MD, USA, 13–17 July 2015; pp. 270–281.
40. Coppa, E.; Demetrescu, C.; Finocchi, I. Input-sensitive profiling. *IEEE Trans. Softw. Eng.* **2014**, *40*, 1185–1205. [CrossRef]
41. Moret, B.M.; Bader, D.A.; Warnow, T. High-performance algorithm engineering for computational phylogenetics. *J. Supercomput.* **2002**, *22*, 99–111. [CrossRef]
42. Waring, J.; Lindvall, C.; Umeton, R. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artif. Intell. Med.* **2020**, *104*, 101822. [CrossRef]
43. Thornton, C.; Hutter, F.; Hoos, H.H.; Leyton-Brown, K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 847–855.
44. Kotthoff, L.; Thornton, C.; Hoos, H.H.; Hutter, F.; Leyton-Brown, K. Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA. In *Automated Machine Learning*; Springer: Cham, Switzerland, 2019; pp. 81–95.
45. Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.T.; Blum, M.; Hutter, F. Auto-sklearn: Efficient and robust automated machine learning. In *Automated Machine Learning*; Springer: Cham, Switzerland, 2019; pp. 113–134.
46. He, X.; Zhao, K.; Chu, X. AutoML: A Survey of the State-of-the-Art. *Knowl.-Based Syst.* **2021**, *212*, 106622. [CrossRef]
47. Zöller, M.A.; Huber, M.F. Benchmark and survey of automated machine learning frameworks. *J. Artif. Intell. Res.* **2021**, *70*, 409–472. [CrossRef]
48. Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; Zemel, R. Fairness through awareness. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, MA, USA, 8–10 January 2012; pp. 214–226.
49. Dwork, C.; Kim, M.P.; Reingold, O.; Rothblum, G.N.; Yona, G. Outcome indistinguishability. In Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, 21–25 June 2021; pp. 1095–1108.
50. Hébert-Johnson, U.; Kim, M.; Reingold, O.; Rothblum, G. Multicalibration: Calibration for the (computationally-identifiable) masses. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1939–1948.
51. Schalk, G.; McFarland, D.J.; Hinterberger, T.; Birbaumer, N.; Wolpaw, J.R. BCI2000: A general-purpose brain-computer interface (BCI) system. *IEEE Trans. Biomed. Eng.* **2004**, *51*, 1034–1043. [CrossRef] [PubMed]
52. King, R.D.; Feng, C.; Sutherland, A. Statlog: Comparison of classification algorithms on large real-world problems. *Appl. Artif. Intell. Int. J.* **1995**, *9*, 289–333. [CrossRef]
53. Zhang, C.; Liu, C.; Zhang, X.; Almpanidis, G. An up-to-date comparison of state-of-the-art classification algorithms. *Expert Syst. Appl.* **2017**, *82*, 128 – 150. [CrossRef]
54. Sen, P.C.; Hajra, M.; Ghosh, M. Supervised Classification Algorithms in Machine Learning: A Survey and Review. In *Emerging Technology in Modelling and Graphics*; Mandal, J.K., Bhattacharya, D., Eds.; Springer: Singapore, 2020; pp. 99–111.
55. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*; MIT Press: Cambridge, MA, USA, 2018.
56. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
57. Al-Nafjan, A.; Hosny, M.; Al-Ohali, Y.; Al-Wabil, A. Review and classification of emotion recognition based on EEG brain-computer interface system research: A systematic review. *Appl. Sci.* **2017**, *7*, 1239. [CrossRef]
58. Hamada, M.; Zaidan, B.; Zaidan, A. A systematic review for human EEG brain signals based emotion classification, feature extraction, brain condition, group comparison. *J. Med. Syst.* **2018**, *42*, 1–25. [CrossRef] [PubMed]
59. Abiri, R.; Borhani, S.; Sellers, E.W.; Jiang, Y.; Zhao, X. A comprehensive review of EEG-based brain–computer interface paradigms. *J. Neural Eng.* **2019**, *16*, 011001. [CrossRef] [PubMed]
60. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

61. Daniel, W.W. *Applied Nonparametric Statistics*; PWS-KENT Pub.: Boston, UK, 1978.
62. Chau, T.; Rizvi, S. Automatic stride interval extraction from long, highly variable and noisy gait timing signals. *Hum. Mov. Sci.* **2002**, *21*, 495–514. [CrossRef]
63. Siuly; Li, Y.; Wu, J.; Yang, J. Developing a logistic regression model with cross-correlation for motor imagery signal recognition. In Proceedings of the 2011 IEEE/ICME International Conference on Complex Medical Engineering, Harbin, China, 22–25 May 2011; pp. 502–507.
64. Chatterjee, R.; Bandyopadhyay, T.; Sanyal, D.K.; Guha, D. Comparative analysis of feature extraction techniques in motor imagery EEG signal classification. In *Proceedings of the First International Conference on Smart System, Innovations and Computing, SSIC 2017, Jaipur, India, 14–16 April 2017*; Springer: Singapore, 2018; pp. 73–83.
65. Jeon, Y.; Nam, C.S.; Kim, Y.J.; Whang, M.C. Event-related (De) synchronization (ERD/ERS) during motor imagery tasks: Implications for brain–computer interfaces. *Int. J. Ind. Ergon.* **2011**, *41*, 428–436. [CrossRef]
66. Torres, E.P.; Torres, E.A.; Hernández-Álvarez, M.; Yoo, S.G. EEG-based BCI emotion recognition: A survey. *Sensors* **2020**, *20*, 5083. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.