Too Fine or Too Coarse? The Goldilocks Composition of Data Complexity for Robust Left-Right Eye-Tracking Classifiers

Brian Xiang
bxiang1@swarthmore.edu
Swarthmore College
Swarthmore, Pennsylvania, USA

Abdelrahman Abdelmonsef
ayahia1@swarthmore.edu
Swarthmore College
Swarthmore, Pennsylvania, USA

ABSTRACT
The differences in distributional patterns between benchmark data and real-world data have been one of the main challenges of using electroencephalogram (EEG) signals for eye-tracking (ET) classification. Therefore, increasing the robustness of machine learning models in predicting eye-tracking positions from EEG data is integral for both research and consumer use. Previously, we compared the performance of classifiers trained solely on finer-grain data to those trained solely on coarse-grain. Results indicated that despite the overall improvement in robustness, the performance of the fine-grain trained models decreased, compared to coarse-grain trained models, when the testing and training set contained the same distributional patterns [35]. This paper aims to address this case by training models using datasets of mixed data complexity to determine the ideal distribution of fine- and coarse-grain data. We train machine learning models utilizing a mixed dataset composed of both fine- and coarse-grain data and then compare the accuracies to models trained using solely fine- or coarse-grain data. For our purposes, finer-grain data refers to data collected using more complex methods whereas coarser-grain data refers to data collected using more simple methods. We apply covariate distributional shifts to test for the susceptibility of each training set. Our results indicated that the optimal training dataset for EEG-ET classification is not composed of solely fine- or coarse-grain data, but rather a mix of the two, leaning towards finer-grain.

CCS CONCEPTS
• General and reference → Evaluation; Performance; Reliability;

KEYWORDS
Data Transformation, Machine Learning, Covariate Distributional Shift, XGBoost, Gradient Boost, Ada Boost, RUSBoost, Random Forest, Decision Tree, MLP, LDA, sLDA, RBF SVC, Linear SVC, Gaussian NB, EEG-ET

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Brian Xiang and Abdelrahman Abdelmonsef. 2018. Too Fine or Too Coarse? The Goldilocks Composition of Data Complexity for Robust Left-Right Eye-Tracking Classifiers. In Proceedings of 28th ACM SIGKDD Conference On Knowledge Discovery and Data Mining (KDD ’18). ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION
Recently, machine-learning classifiers have found more and more consumer applications as well as "on the field" usage. For example, machine learning models can be used to identify shopping motives relatively early in the search process [19], to detect work-related strain in truck drivers [13], and to assess the diagnosis of many neurological diseases such as Autism Spectrum Disorder and Alzheimer’s [4, 11, 30, 31]. Machine learning approaches have also shown adequate performance on computer vision bioinformatics applications, medical image analysis, Eye-Tracking (ET) analysis, and Electroencephalography (EEG) analysis [1, 3, 8, 15, 22, 23, 25–27, 29, 36, 37, 40].

As machine learning classification is used in more and more unfamiliar environments, it is increasingly important for these classifiers to be robust. In the EEG-ET research, recent work in this area has focused on determining what machine learning models are best equipped to predict eye position from EEG signals [15] as well as eliminating the noise associated with EEG data collection automatically [17, 21, 24, 28, 33, 39].

These approaches all examine robustness across a variety of underlying factors of data analysis. In this paper, We focus on the inherent distributional patterns and the underlying differences between finer-grain and coarser-grain data. For this paper, finer-grain data refers to data collected in a more complicated framework in an environment with more uncontrolled conditions. On the contrary, coarser-grain data refers to data collected in a more simplified collection format with more restrictions on the experiment’s environment for the same task.

In medical research, fine-grain data collection methods have been explored with great success [9, 20]. That is the use of data from more complex, or finer-grained, data collection methods to test for simpler tasks. Oftentimes, systems developed in a coarse-grained "lab conditions" only work in controlled environments, causing difficulties when utilized in uncontrolled conditions [16, 34].

Recently, fine-grain data approaches have shown successful results for Covid-19 contact tracing machine learning classifiers [7]. In this paper, We want to emphasize the impact of data granularity as well as the beneficial effects of using fine-grain data in machine learning classification.

Previously, we showed that Machine Learning models trained on fine-grained data are more robust, meaning that they maintain more
consistent accuracies, than models trained on coarse-grained data when tested on data of different complexities [35]. We extend upon our previous findings by examining the idea of training on mixed fine- and coarse-grain datasets. To our knowledge, this approach was not tested with machine learning classifiers before. The goal of mixing datasets of varying data granularity is to improve upon the draw-back we previously found with training on purely fine-grain data. That is the model’s performance when tested on data of similar complexity drops a significant amount when compared to purely coarse-grain data. We suspect that a mixed complexity training set may perform adequately across a broader range of testing distributional patterns, including different and similar complexities.

In this study, we train machine learning models for left-right eye-tracking classification using data from a mix of binary-classified (coarse-grain) and vector-based (fine-grain) collection frameworks. We then compare the results to models trained exclusively on either binary-classified or vector-based data. The goal is to expand upon our previous conclusions regarding the optimal method to increase both accuracy and robustness of machine learning classifiers using fine- and coarse-grain data. Robustness is determined by the accuracy after a covariate distributional shift. The distributional shift in combination with the different mixes of data complexity attempts to mimic realistic data which often contains varying distributional patterns. Since we had previously determined the superiority of fine-grain data in terms of robustness, we hypothesize that classifiers trained using more fine-grain data will attain higher accuracies after covariate distributional shifts are applied.

The purpose is to determine the “Goldilocks” or optimal training set composition of fine- and coarse-grain data for left-right gaze classification in terms of both accuracy and robustness as well as verify that fine-grain data performs better than coarse-grain data using a more encompassing experimental design.

2 DATA

We used the EEGEyeNet dataset due to its large size [12]. The dataset contains simultaneously-recorded EEG signals and Eye-Tracking data from three different experimental tasks. A summary table for the three tasks’ metadata is in Appendix D. In this study, we used data from two experiments: pro-antisaccade (PA) and Large Grid (LG). In the PA trials, participants were asked to focus on a cue that appears on either the screen’s center, horizontally left, or horizontally right, as shown in the left screen in Figure 1. Gaze positions in PA were restricted to the horizontal axis and were binary-classified either left or right, relative to the screen’s center. In the LG trials, participants were asked to fixate on dots presented one at a time at 26 different screen positions, as shown in the right screen in Figure 1. Therefore, LG’s framework enabled finer-grain data collection since it allowed more freedom for eye movements, and gaze positions were encoded in a two-dimensional format including both angle and amplitude, compared to the one-liner binary-encoded eye movements in PA.

3 EXPERIMENT DESIGN

The learning objective of the models trained in our experiment was to use EEG brain signals to predict the direction of a subject’s gaze along the horizontal axis (whether they are looking to the left or the right). Although predictions for this task using the same dataset were previously made, they were performed exclusively using data from the PA tasks for training and testing [12].

In this paper, we train 13 classifiers using data that is composed of a mixture of both PA and LG data and perform a covariate distributional shift by comparing their performance based on their accuracy when tested on PA and LG data. The different compositions are described in Figure 2

3.1 Data Processing

Given the classification nature of our learning problem, data from the LG task, encoded as Angle and Amplitude, should be transformed and relabeled into the expected format as left or right for training left-right classification models. The convention we used for the relabelling process was that for angle $\alpha$; when $|\alpha| < \frac{\pi}{2}$ the data is classified as right, otherwise the data is classified as left. This logic was confirmed by the dataset’s authors in Appendix A and is shown in Figure 3.

3.2 Models Training

As per EEGEyeNet authors’ recommendation, we used the minimally preprocessed EEG data. Data processing was done using the NumPy library, and the model implementations were installed from the SKLearn library [18]. From a broader perspective, models were
Table 1: The 20 different possible combinations of training and testing sets

| Training Set | Testing Set |
|--------------|-------------|
| LG           | LG          | Mixed (20-80) | Mixed (40-60) | Mixed (60-40) | Mixed (80-20) | PA         |
| PA           | LG          | Mixed (20-80) | Mixed (40-60) | Mixed (60-40) | Mixed (80-20) | PA         |
| Mixed (20-80)| LG          | PA            | LG            | Mixed (40-60) | Mixed (60-40) | Mixed (80-20) | PA         |
| Mixed (40-60)| LG          | PA            | LG            | PA            | Mixed (60-40) | Mixed (80-20) | PA         |
| Mixed (60-40)| LG          | PA            | LG            | PA            | PA            | Mixed (60-40) | Mixed (80-20) | PA         |
| Mixed (80-20)| LG          | PA            | LG            | PA            | PA            | PA          | Mixed (60-40) | Mixed (80-20) | PA         |

4 MODELS

4.1 Machine Learning Models

In this study, machine learning models operate on features extracted from the data rather than the data itself. Feature extraction has been applied in two steps. First, [12] applied a band-pass filter in frequencies in the range [8 - 13 HZ] on the acquired signals through all trials. This choice of frequencies is based on suggestions from [6]. Following the filtering step, the Hilbert transform was applied, resulting in a complex time series from which targeted features were extracted for learning models. Since we are considering a classification problem, we experimented with classification-only models and models that can be applied to both classification and regression problems.

4.1.1 Linear Classifiers: Linear classifiers gather discriminant classifiers that use linear decision boundaries between the feature vectors of each class [15]. In this paper, we will use Linear and Radial-Basis-function Support Vector Classifiers (SVC) and Normal and Shrinkage Linear Discriminant Analysis (LDA) classifiers. All four algorithms still perform well after several decades since they were first implemented in this field, with SVC outperforming other classifiers, especially for two-classes problems [2, 14, 15].

4.1.2 Ensemble Classifiers. Ensemble (or voting) classifier is a machine learning classification algorithm that trains with different classification models and makes predictions through ensembling their predictions to make a stronger classification. These algorithms have been the gold standard for several EEG-based classification experiments [25]. In our study, we used Random Forest, XGBoost, GradientBoost, RUSBoost, and AdaBoost.

4.1.3 Naive Bayes and Decision Tree Classifiers. Naive Bayes (NB) Classifier is the statistical Bayesian classifier [5]. It assumes that all variables are mutually correlated and contribute to some degree towards classification. It is based on the Bayes’ Theorem and is commonly used with high dimensional inputs. On the other hand, a decision tree is not a statistically based one; rather, it is a data mining induction technique that recursively partitions the dataset using a depth-first greedy algorithm until all data is classified with a particular class. Both NB and the decision tree are relatively fast and well suited for large data. Furthermore, they can deal with noisy data, which makes them well suited for EEG classification applications [10].

4.2 Deep Learning Models

Deep learning is a subfield of machine learning algorithms in which computational models learn features from hierarchical representations of input data through successive non-linear transformations [29]. Deep learning methods, especially Convolutional Neural Network (CNN), performed well in several previous EEG band power (feature) based research [25]. Still, these methods have not demonstrated convincing and consistent improvements [15]. Given so and the expected high run time for such algorithms due to the dataset’s large size, we only included a simple and relatively fast multi-layer perception neural network (MLP) in our experiment.

5 RESULTS

We trained and tested the machine learning models using datasets composed of both fine- and coarse-grain data (pro-antisaccade and “transformed” large grid) and compared the results. Thus, we had...
### Table 2: Models Trained on Mixed Data and Tested on Pro-Antisaccade Data

| Model       | PA  | Mixed (80-20) | Mixed (60-40) | Mixed (40-60) | Mixed (20-80) | LG  |
|-------------|-----|---------------|---------------|---------------|---------------|-----|
| XGBoost     | 96.7% | 96.7% | 95.7% | 96.3% | 95.5% | 94.5% |
| GradientBoost | 96.4% | 96.3% | 95.7% | 95.7% | 95.8% | 94.2% |
| RandomForest | 95.9% | 95.7% | 95.5% | 95.1% | 94.3% | 93.3% |
| AdaBoost     | 95.4% | 94.8% | 94.7% | 94.2% | 94.0% | 93.1% |
| RUSBoost     | 95.3% | 95.2% | 94.3% | 93.7% | 93.4% | 93.4% |
| DecisionTree | 94.4% | 94.7% | 94.1% | 93.8% | 93.4% | 92.1% |
| MLP          | 92.8% | 93.9% | 93.3% | 92.4% | 92.1% | 91.3% |
| LinearSVC   | 91.1% | 90.4% | 90.6% | 90.5% | 89.9% | 89.9% |
| LDA          | 90.6% | 90.6% | 90.2% | 90.2% | 90.0% | 89.8% |
| sLDA         | 90.3% | 90.3% | 90.1% | 89.8% | 90.1% | 89.8% |
| KNN          | 90.3% | 89.3% | 89.3% | 88.1% | 88.4% | 87.9% |
| RBF SVC      | 89.2% | 89.2% | 88.0% | 87.3% | 88.3% | 88.1% |
| GaussianNB   | 86.0% | 85.4% | 84.6% | 83.9% | 84.3% | 82.8% |
| Average      | 92.6% | 92.5% | 92.0% | 91.6% | 91.5% | 90.8% |

### Table 3: Models Trained on Mixed Data and Tested on Large Grid Data

| Model       | PA  | Mixed (80-20) | Mixed (60-40) | Mixed (40-60) | Mixed (20-80) | LG  |
|-------------|-----|---------------|---------------|---------------|---------------|-----|
| XGBoost     | 92.7% | 94.1% | 94.3% | 95.4% | 95.2% | 95.4% |
| GradientBoost | 91.8% | 93.7% | 93.4% | 94.6% | 94.9% | 95.3% |
| RandomForest | 90.9% | 93.1% | 93.1% | 93.8% | 94.0% | 93.9% |
| AdaBoost     | 89.2% | 90.1% | 91.5% | 92.5% | 93.0% | 93.4% |
| MLP          | 89.5% | 91.2% | 90.4% | 91.2% | 92.3% | 92.9% |
| RUSBoost     | 86.0% | 90.4% | 90.8% | 92.5% | 92.3% | 92.7% |
| DecisionTree | 88.0% | 90.0% | 90.1% | 91.9% | 91.6% | 91.9% |
| sLDA         | 90.3% | 90.8% | 91.5% | 91.4% | 90.9% | 91.8% |
| LDA          | 89.9% | 90.6% | 91.2% | 90.9% | 90.7% | 91.6% |
| LinearSVC   | 89.8% | 91.2% | 91.5% | 91.5% | 91.2% | 91.1% |
| KNN          | 89.3% | 89.2% | 88.8% | 88.9% | 89.9% | 90.0% |
| RBF SVC      | 84.8% | 85.5% | 85.8% | 87.2% | 87.4% | 88.6% |
| GaussianNB   | 83.4% | 82.7% | 82.8% | 83.2% | 87.1% | 87.2% |
| Average      | 88.9% | 90.2% | 90.4% | 91.2% | 91.6% | 92.0% |

### Table 4: Models Trained on Pro-Antisaccade Data and Tested on Mixed

| Model       | PA  | Mixed (80-20) | Mixed (60-40) | Mixed (40-60) | Mixed (20-80) | LG  |
|-------------|-----|---------------|---------------|---------------|---------------|-----|
| XGBoost     | 96.7% | 95.4% | 93.7% | 93.3% | 93.2% | 92.7% |
| GradientBoost | 96.4% | 95.0% | 93.2% | 92.6% | 92.7% | 91.8% |
| RandomForest | 95.9% | 94.5% | 92.2% | 91.7% | 91.7% | 90.9% |
| AdaBoost     | 95.4% | 93.6% | 91.0% | 90.5% | 90.1% | 89.2% |
| RUSBoost     | 95.3% | 93.0% | 89.5% | 88.1% | 87.2% | 86.0% |
| DecisionTree | 94.4% | 92.2% | 90.0% | 88.6% | 89.2% | 88.0% |
| MLP          | 92.8% | 91.5% | 89.5% | 89.9% | 90.4% | 89.5% |
| LinearSVC   | 91.1% | 90.0% | 88.7% | 89.5% | 90.4% | 89.8% |
| LDA          | 90.6% | 89.9% | 88.7% | 89.2% | 90.9% | 89.9% |
| sLDA         | 90.3% | 89.6% | 88.6% | 89.2% | 91.1% | 90.3% |
| KNN          | 90.3% | 90.1% | 89.6% | 89.1% | 90.1% | 89.3% |
| RBF SVC      | 89.2% | 87.2% | 85.5% | 84.7% | 86.0% | 84.8% |
| GaussianNB   | 86.0% | 84.9% | 84.0% | 83.7% | 84.9% | 83.4% |
| Average      | 92.6% | 91.3% | 89.6% | 89.2% | 89.8% | 88.9% |
Table 5: Models Trained on Large Grid Data and Tested on Mixed

| Model       | PA     | Mixed (80-20) | Mixed (60-40) | Mixed (40-60) | Mixed (20-80) | LG     |
|-------------|--------|---------------|---------------|---------------|---------------|--------|
| XGBoost     | 94.5%  | 93.9%         | 94.1%         | 94.8%         | 95.9%         | 95.4%  |
| GradientBoost| 94.2%  | 93.7%         | 93.9%         | 94.5%         | 95.8%         | 95.3%  |
| RandomForest| 93.3%  | 93.0%         | 93.2%         | 93.1%         | 94.3%         | 93.9%  |
| AdaBoost    | 93.1%  | 93.0%         | 93.0%         | 92.9%         | 93.8%         | 93.4%  |
| MLP         | 91.3%  | 91.3%         | 92.0%         | 92.4%         | 93.6%         | 92.9%  |
| RUSBoost    | 93.4%  | 92.8%         | 92.5%         | 92.3%         | 93.2%         | 92.7%  |
| sLDA        | 89.8%  | 90.1%         | 90.1%         | 90.4%         | 92.6%         | 91.8%  |
| DecisionTree| 92.1%  | 91.9%         | 91.3%         | 91.1%         | 92.5%         | 91.9%  |
| LDA         | 89.8%  | 90.1%         | 89.9%         | 90.2%         | 92.3%         | 91.6%  |
| LinearSVC   | 89.9%  | 90.3%         | 89.9%         | 90.0%         | 92.0%         | 91.1%  |
| KNN         | 87.9%  | 87.8%         | 88.5%         | 88.8%         | 90.3%         | 90.0%  |
| RBF SVC     | 88.1%  | 87.4%         | 88.1%         | 88.5%         | 89.5%         | 88.6%  |
| GaussianNB  | 82.8%  | 83.4%         | 84.1%         | 84.9%         | 87.8%         | 87.2%  |
| Average     | 90.8%  | 90.7%         | 90.5%         | 91.1%         | 92.6%         | 92.0%  |

Table 6: Average Accuracies of Models Trained on Mixed Data and Tested on Pro-Antisaccade/LG Data

| Data Composition | PA   | LG   | Average |
|------------------|------|------|---------|
| PA               | 92.6%| 88.9%| 90.75%  |
| Mixed (80-20)    | 92.5%| 90.2%| 91.35%  |
| Mixed (60-40)    | 92.0%| 90.4%| 91.2%   |
| Mixed (40-60)    | 91.6%| 91.2%| 91.4%   |
| Mixed (20-80)    | 91.5%| 91.6%| 91.6%   |
| LG               | 90.8%| 92.0%| 91.4%   |

6 DISCUSSION

Table 6 describes the average accuracy from 13 different machine learning model for left-right ET classification trained on data of different data complexity compositions.

This paper utilizes the benchmark data from the EEGEyeNet dataset and 13 machine learning models to create left-right classifiers trained on data of varying compositions of fine- and coarse-grain data. The accuracies of the models are then tested using two testing sets with distinctly different distributional patterns (PA and LG). In this way, the effects of covariate distributional shifts are more apparent and provide insight on the optimal data complexity composition of fine- and coarse-grain data in order to reduce the impact of such phenomenon. The results also provide useful information on the types of machine learning classifiers that should be used for EEG-ET classification tasks.

6.1 Finding the Goldilocks Composition

Previously, we had concluded that machine learning classifiers trained purely on fine-grain data outperform machine learning classifiers trained purely on coarse-grain data in terms of robustness. This was confirmed by Tables 4 and 5. Therefore, we theorized that the ideal composition of data complexity would lean towards finer-grain data.

The hypothesis is confirmed by Table 6. The composition of 20% coarse-grain data and 80% fine-grain data produced the most consistently accurate results as shown by the average percentages. This suggests that the optimal training set for EEG-ET classification should not be created using data of solely one distributional complexity, but rather a mix of both fine- and coarse-grain data, leaning towards more fine-grain data.

6.2 Determining the Best Classifiers for EEG-ET Classification

The consistently most accurate classifiers across Tables 4, 5, 2, and 3 are XGBoost, GradientBoost, and RandomForest. Alternatively, the consistently worst performing classifiers were RBF SVC and
GaussianNB. Therefore regardless of distributional complexity, XGBoost, GradientBoost, and RandomForest should be used to classify EEG-ET data.

### 6.3 Future Recommendations and Improvements

To further advance the work provided in this study, four steps are highlighted for future exploration.

Although deep learning models were excluded due to inconsistent results [15], the main reason was due to restraints in time and resources. As deep learning models, especially CNN and Attention [32, 38], have shown promising results in regards to EEG-ET classification, it is important to thoroughly explore the effects of fine-versus coarse-grain data on the robustness of such models.

Additionally, the angle value is currently the only indicator used to determine whether the saccade is towards the left or the right. The amplitude was completely ignored in data processing. The amplitude could be incorporated by using it as a weighting/scaling factor to indicate the left-right extension and solve this as a regression problem. Based on the predictions made by the regression value, we can then classify it as either left or right utilizing finer-grain data.

Furthermore, although we determined an approximately optimal training set, we are almost certain this is not the Goldilocks composition for mixed data complexity. Perhaps an application of gradient descent would be able to more precisely compute such a composition for each machine learning model and specific machine learning task. Additionally due to time restraints, we were unable to test our machine learning models trained on mixed distributional complexity against a broader range of testing sets like we did for exclusively fine- or exclusively coarse-grain trained classifiers in Tables 4 and 5.

Although our work discusses EEG-ET classification, the results can be potentially relevant to other machine learning models in other applications. We highly encourage further investigation on training machine learning models on data composed of different granularities across an array of other research topics.

### 7 CONCLUSION

The motivation behind this work was to extend upon our previous work on determining whether training machine learning models on finer-grain data leads to more robust models as well as to analyze the effects of training machine learning classifiers on datasets composed of mixed data complexity. We verified our previous findings across a broader range of distributional patterns as well as determined that ensemble methods are the most suitable machine learning classifiers for EEG-ET classification tasks. Furthermore we identified that EEG-ET machine learning classifiers seem to produce the most consistent results when the training set contains a mix of distributional patterns, leaning towards finer-grain data. We hope that future applications of practical EEG-ET interfaces utilize data of varying distributional patterns to increase classification accuracy and robustness.

### 8 CITATIONS AND BIBLIOGRAPHIES

### REFERENCES

[1] Siyee An and Umit Y Ogras. 2021. MARS: mmWave-based Assistive Rehabilitation System for Smart Healthcare. ACM Transactions on Embedded Computing Systems (TECS) 20, 5s (2021), 1–22.

[2] Pouya Bashivan, Irina Rish, and Steve Heisig. 2016. Mental state recognition via wearable EEG. arXiv preprint arXiv:1602.08985 (2016).

[3] Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. 2019. Deep learning for electroencephalogram (EEG) classification tasks: a review. Journal of neural engineering 16, 3 (2019), 035001.

[4] Ranadeep Deb, Ganapati Bhat, Siyee An, Holly Shill, and Umit Y Ogras. 2021. Trends in technology usage for parkinson’s disease assessment: A systematic review. MedRxiv (2021).

[5] Richard O Duda, Peter E Hart, et al. 1973. Pattern classification and scene analysis. Vol. 3. Wiley New York.

[6] Joshua J Foster, David W Sutterer, John T Serences, Edward K Vogel, and Edward Awh. 2017. Alpha-band oscillations enable spatially and temporally resolved tracking of covert spatial attention. Psychological science 28, 7 (2017), 929–941.

[7] Carlos Gómez, Niambi Bellton, Brian Quach, Jack Nicholls, and Devanshu Anand. 2020. A simplistic machine learning approach to contact tracing. arXiv preprint arXiv:2012.05940 (2020).

[8] Jiajia Gu, Ziyuan Zhao, Zeng Zeng, Yuzhe Wang, Zhengyiren Qu, Bharadwaj Veeravalli, Brian Kim Poh Goh, Glenn Rummah Bonney, Krishnakumar Madhavan, Chan Wan Ying, et al. 2020. Multi-phase cross-modal learning for noninvasive gene mutation prediction in hepatocellular carcinoma. In 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 5814–5817.

[9] Christopher Higgisson, Peter A Arnett, and William D Voss. 2000. The ecological validity of clinical tests of memory and attention in multiple sclerosis. Archives of Clinical Neuropsychology 15, 3 (2000), 185–204.

[10] Sayali D Jadhav and HP Channe. 2016. Comparative study of K-NN, naive Bayes and decision tree classification techniques. International Journal of Science and Research (IJSR) 5, 1 (2016), 1842–1845.

[11] Jiannan Kang, Xiaoya Han, Jasja Song, Zikang Niu, and Xiaoli Li. 2020. The identification of children with autism spectrum disorder by SVM approach on EEG and eye-tracking data. Computers in biology and medicine 120 (2020), 103722.

[12] Ard Kastrati, Martyna Martyna Beata Plomecka, Damian Pascual, Lukas Wolf, Victor Gillioz, Roger Wattenhofer, and Nicolas Langer. 2021. EEGEyeNet: A Simultaneous Electroencephalography and Eye-tracking Dataset and Benchmark for Eye Movement Prediction. arXiv preprint arXiv:2111.05190 (2021).

[13] Jesus L Lobo, Javier Del Ser, Flavia De Simone, Roberta Presta, Simona Collina, and Zdenek Moravek. 2016. Cognitive workload classification using eye-tracking and EEG data. In Proceedings of the International Conference on Human-Computer Interaction in Aerospace. 1–8.

[14] Fabien Lotte. 2015. Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces. Proc. IEEE 103, 6 (2015), 871–890.

[15] Fabien Lotte, Laurent Bougrain, Andrezj Cichecki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. 2018. A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. Journal of neural engineering 15, 3 (2018), 031005.

[16] Thomas D Marcotte, J Cobb Scott, Ruiyi Khatam, and Robert K Heaton. 2010. Neuropsychology and the prediction of everyday functioning. The Guilford Press.

[17] Vangelis P Okonomou, Spiros Nikolopoulos, and Ioannis Kompatsiaris. 2020. Machine-learning techniques for EEG data. Signal Processing to Drive Human-Computer Interaction: EEG and eye-controlled interfaces (2020). 145.

[18] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. The journal of machine learning research 12 (2011), 2825–2830.

[19] Jella Pfeiffer, Thies Pfeiffer, Martin Meißner, and Elisa Weiß. 2020. Eye-tracking-based classification of information search behavior using machine learning: evidence from experiments in physical shops and virtual reality shopping environments. Information Systems Research 31, 3 (2020), 675–691.

[20] Gien Planche, A Tirard, V Gyselinck, S Nicolas, and P Pinoli. 2012. Using virtual reality to characterize episodic memory profiles in amnestic mild cognitive impairment and Alzheimer’s disease: influence of active and passive encoding. Neuropsychologia 50, 5 (2012), 592–602.

[21] Michael Plochl, José Pablo Osandón, and Peter König. 2012. Combining EEG and eye tracking: identification, characterization, and correction of eye movement artifacts in electroencephalographic data. Frontiers in human neuroscience 6 (2012), 278.

[22] Feihong Qian, Ziyuan Zhao, Cong Chen, Zeng Zeng, and Xiaoli Li. 2021. Two Eyes Are Better Than One: Exploiting Binocular Correlation for Diabetic Retinopathy Severity Grading. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2115–2118.
# Appendix A

After emailing Ard Kastrati, Ard verified that for angle $\alpha$ in radians:

- $\alpha = 0$ is right
- $\alpha = \frac{\pi}{2}$ is down
- $\alpha = \pi$ is left
- $\alpha = -\frac{\pi}{2}$ is up

# Appendix B

The models were trained and tested on this environment settings:

OS: Mac 12.2.1

Python: 3.9.0

cleverhans: 2.1.0

Keras: 2.2.4

tensorflow-gpu: 1.9.0

numpy: 1.22.1

keras: 2.2.4

scikit-learn: 1.0.2

scipy: 1.8.0

The total space occupied by the dataset on the device is 69.0574 GB, and the total time for training and testing was 30 mins on average.

## 9.3 Appendix C

The code for data processing and evaluation is provided here: https://github.com/ayahia1/KDD-ML

## 9.4 Appendix D

The main reason for using EEGEyeNet in our study was its relatively large size. The table below presents the number of subjects, the number of sample data points, and the length of the recording time for each of the three experimental tasks in the EEGEyeNet dataset.

The numbers below should prove the large size of this dataset.

### Notes:
- First, in the table, experimental tasks are referred to as “paradigms.”
- Second, although the total number of subjects in the table is 486, some performed more than one experiment and, thus, referenced twice. The total number of unique subjects, when the dataset paper was first published, was 356. Finally, the recording time column contains the numbers rounded to the nearest hour.

### Table 7: Metadata comparison of the 3 experimental paradigms

| Paradigm | # subjects | # samples | Recording time |
|----------|------------|-----------|----------------|
| PA       | 369        | 30842     | 38h            |
| Large Grid | 30        | 17830     | 8h             |
| VSS      | 87         | 31563     | 1h             |
| Total    | 486        | 80235     | 47h            |