Time Majority Voting, a PC-based EEG Classifier for Non-expert Users

Guangyao Dou[1][0000−0001−8011−9658], Zheng Zhou[1][0000−0001−9313−2106], and Xiaodong Qu[1,2][0000−0001−7610−6475]

[1] Brandeis University, Waltham MA 02453, USA
{guangyaodou,zhengzhou}@brandeis.edu
[2] Swarthmore College, Swarthmore PA 19081, USA
xqu1@swarthmore.edu

Abstract. Using Machine Learning and Deep Learning to predict cognitive tasks from electroencephalography (EEG) signals is a rapidly advancing field in Brain-Computer Interfaces (BCI). In contrast to the fields of computer vision and natural language processing, the data amount of these trials is still rather tiny. Developing a PC-based machine learning technique to increase the participation of non-expert end-users could help solve this data collection issue. We created a novel algorithm for machine learning called Time Majority Voting (TMV). In our experiment, TMV performed better than cutting-edge algorithms. It can operate efficiently on personal computers for classification tasks involving the BCI. These interpretable data also assisted end-users and researchers in comprehending EEG tests better.

Keywords: Brain-Machine Interface · Machine Learning · Ensemble Methods · Voting · Time Series · Interpretable AI

1 Introduction

Researchers from Computer Science, Neuroscience, and Medical fields have applied EEG-based Brain-Computer Interaction (BCI) techniques in many different ways [2, 15, 19, 22, 24, 26, 34], such as diagnosis of abnormal states, evaluating the effect of the treatments, seizure detection, motor imagery tasks [4, 5, 6, 17, 23, 27], and developing BCI-based games [14]. Previous studies have demonstrated the great potential of machine learning, deep learning, and transfer learning algorithms [1, 3, 7, 8, 12, 16, 18, 20, 21, 25, 28, 29, 37, 38, 39, 40, 41, 42] in such clinical and non-clinical data analysis.

However, the data size of such experiments is still relatively small compared to the areas of computer vision or natural language processing. Thus, some deep learning or big data approaches still struggling with the limitation of small dataset size. Also, EEG signals have noise issues, partly because of the contact of sensors and skin for several current non-invasive consumer-grade devices. The outlier issue is also a concern for the EEG data because of the difficulties subjects have in concentrating on the experimental tasks during the entire session. Current machine learning and deep learning algorithms are more for clinical experiments and less for the possible experiment for non-expert user to conduct at home.

Our research questions are: Can we develop a PC-based machine learning algorithm for non-expert end-users to do EEG classification at home? Can we achieve reasonably high accuracy while keeping the run time in an acceptable range? Can we make the machine learning classification results explainable to the end-users? To answer these questions, we proposed a new machine learning classification algorithm, Time Majority Voting (TMV). We found TMV outperformed other state-of-the-art classifiers. Also, its run time on a PC is still acceptable compared to the deep learning algorithms. The classification results are adequately interpretable to the end-users.

The paper is organized as follows: section two discusses several most frequently used classification algorithms for BCI research, then present our new algorithm. Section three presents our experiment conducted to test the new algorithm. Section four elaborates our result followed by sedition five, which discusses the limitation and future work. Lastly, section six concludes the study and summarized our answers to the research questions.

2 Algorithms

All of the code was run on a 2018 Macbook Pro with a 2.2GHz 6-core Intel Core i7 processor and with 16 GB of memory. The Python version is 3.8. The scikit-learn [31] version is 0.24.1. The PyTorch [30] version is 1.10. The code discussed in this paper is available online (`https://github.com/GuangyaoDou/Time_Majority_Voting`).

2.1 Existing Algorithms

u We reviewed and implemented several machine learning algorithms commonly used in the field [9, 10, 11, 13]. For examples, Linear Classifiers, Nearest Neighbors, Decision Trees, and Ensemble Methods.

Linear Classifiers: The Shrinkage Linear Discriminant Analysis (Shrinkage LDA) performed adequately on EEG datasets with simple tasks. The Support Vector Machine (SVM), effective in high dimensional spaces, performed reasonably well based on the previous research. These algorithms are simple to implement and are computationally efficient.

Nearest Neighbor: Such a classifier implements the K-Nearest Neighbor(KNN). KNN performs voting to determine an unseen dataset to one of the k nearest neighbors. The KNN performed pretty well compared to most other classifiers on the EEG dataset.

Decision Tree: The Decision Trees classifier is easy to understand, implement, and interpret. The decision tree creates a model that predicts the outcome of a data point based on decision rules. The computational cost is low and can handle both numerical and categorical data. However, it might overfit as trees are too complex.

Ensemble Methods: We used Random Forest and boosting. These are Ensemble Machine Learning algorithms that combine the predictions of several weaker learners and form more robust and more accurate predictions. These have been widely used in EEG-based experiments, and research [24, 33, 35].

Deep Learning: We implemented CNN with ReLu, and RNN, especially LSTM, mainly using toolsets from the PyTorch platform. [15, 34, 37]. These DLs performed very well on EEG datasets. However, we excluded these algorithms in this paper due to their high runtime.

2.2 Our New Algorithms

This paper proposed a new voting approach based on the top two individual classifiers. Ensemble methods, especially boosting, bagging, and voting, have demonstrated excellent performance in previous research. [15, 24, 34, 36, 37] In EEG-based BCI classification research, the following voting methods have been investigated in several experiments [24, 33]: majority voting, weighted voting, and time continuity voting. Here we considered the advantages of both majority voting and time continuity voting and developed our new Time Majority Voting (TMV) algorithm.

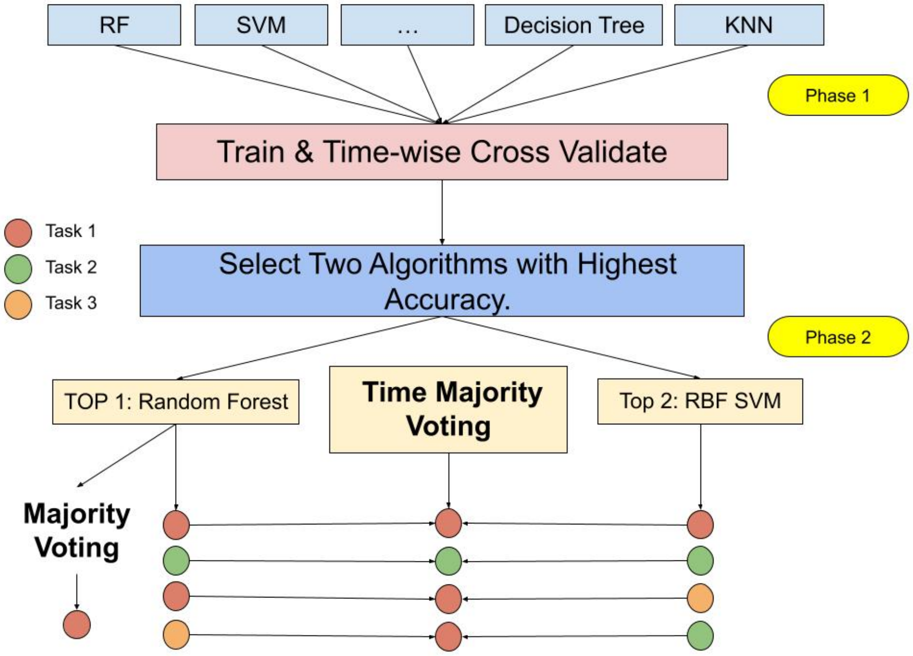

Fig. 1. Our New Algorithm: Time Majority Voting

Time Majority Voting (TMV).
Figure 1 demonstrated the concept of the new algorithm. More details are in the Experiment section and the Result section. There are two phases in the new algorithm. First, we investigated the state-of-the-art machine learning algorithms [15, 24, 34, 37] and found the top two performers on average. In our experiments during phase 1, We tested Random Forest (RF), RBF and linear SVM, kNN, Decision Tree, and several boosting algorithms. We found Random Forest performed the best, and RBF SVM performed the second on average. Next, we entered phase two. For each subject, we picked the majority task predicted by the best performing classifier, the Random Forest classifier, for each time interval of each task from each session.

The next step is voting. We used the Random Forest and the RBF SVM to conduct the voting process. The best algorithm is the Random Forest, and the second algorithm is RBF SVM. The voting details are shown in the six examples at the bottom of Figure 1. If both the Random Forest and the RBF SVM algorithms agree with the results, as shown in the first two rows, the results reflect both algorithms' results. For example, as the second row shows, if both the classifiers predicted the task 2, then the Time Majority Voting will yield task 2 as the result. If the two algorithms do not agree on the prediction results, the results will be labeled as the majority of tasks already determined by the Random Forest classifier, as shown in the last two rows. No matter what the two algorithms predict out of the five tasks, even if they predicted it as two different tasks other than the majority task, as the last row shows, Task 2 and Task 3, the TMV result is still set to be Task one. This concept is based on the temporal dependence time-series features in the previous research [25, 36].

3 Experiments

Several EEG experiments focus on the high-level cognitive tasks that college students frequently conduct, as mentioned in Table 1. In this experiment, we used the dataset from the Think-Count-Recall (TCR) paper [33]. Scalp-EEG signals were recorded from seventeen subjects. Each one was tested in six sessions. Each session is five minutes long, with five tasks, each task is one minute. Tasks were selected by the subjects together with the researchers based on frequent tasks in study environments for students in their everyday lives. Each subject completed six sessions over several weeks. The five tasks are Think(T), Count(C), Recall(R), Breathe(B), and Draw(D).

3.1 Data Preprocess

Data cleaning: As mentioned in [33, 35], for each task during each session, the first 30% of the data, which is the first 18 seconds of each task and during the transition phase, will be removed. Thus, each one-minute task only had 42 seconds left. This has been proved reasonable during the data cleaning phase. Some electrodes may have temporarily lost contact with the subjects' scalp during

(E) T	1	2	3	4	5
[32]	Math	Close-eye Relax	Read	Open-eye Relax	None
[36]	Python Passive	Math Passive	Python Active	Math Active	None
[35]	Read	Write Copy	Write Answer	Type Copy	Type Answer
[33]	Think	Count	Recall	Breathe	Draw

Table 1. Tasks (T) in Experiments (E)

the EEG recording. The result was that multiple sequential spectral snapshots from one or more electrodes had the same value. In this paper, we decide to remove such anomaly when detected for a consecutive 1.4 seconds. Such a cleaning action caused a different level of loss of the data for each subject.

Subjects: We had a total of seventeen subjects. After the data cleaning actions, subjects who lost more than 65% of the total data will be excluded from the subsequent analysis. In the end, there were twelve subjects left to continue the analysis. Moreover, for the six sessions of each subject, if a session lost more than 65% of the data, then that session will also be excluded for further analysis.

Time-wise Cross-Validation: We adopted time-wise cross-validation. We divided each task into seven subsets, meaning each subset had six seconds, evenly and continuously. Then, we created a total of seven folds. Each fold contained six seconds of data for each task for each session. We checked any folds that lost more than 65% of the original data in each fold and discarded these folds for future analysis. Next, we used one fold for testing and the remaining non-discarded folds for training, and we cross-validated them.

4 Results

4.1 Existing Algorithms

We reported the average accuracy for all subjects and the runtime of each classifier for each subject we trained and tested during phase 1 in table 2. As we can

Algorithms	Average Accuracy	Average code run-time (s)
Random Forest Phase 1	**0.55**	42.0
RBF SVM	**0.53**	30.5
Nearest Neighbors	0.48	**1.9**
Decision Tree	0.44	0.9
Linear SVM	0.42	23.2
Shrinkage LDA	0.42	0.2
Adaboost Classifier	0.39	47.8
RUSBoost	0.39	28.2
GradientBoost	0.31	24.0

Table 2. State-of-the-art Algorithms with Accuracy and Run-time

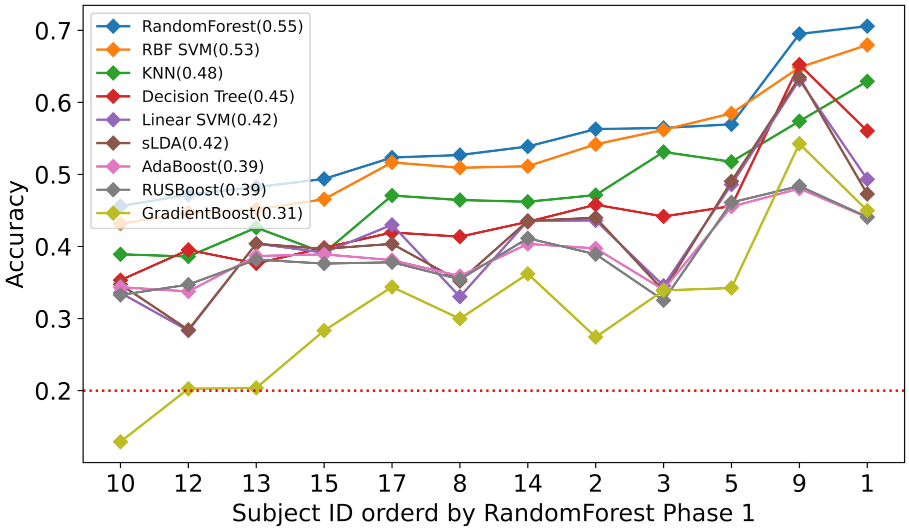

Fig. 2. Accuracy for Different Algorithms

see, Random Forest had the highest accuracy of 0.55 in our experiment, and the SVM with RBF kernel performed adequately on the TCR dataset with an average accuracy of 0.53. Though Nearest Neighbors did not perform as well as the Random Forest and the RBF SVM, it was one of the fastest algorithms on personal computers. Other ensemble methods such as Adaboost, RusBoost, and GradientBoosting performed relatively lower than these top three algorithms.

The individual difference may impact the accuracy of each subject. But we can still recognize a general pattern from Fig 2. We ordered all twelve subjects by prediction accuracy using Random Forest. Most of the algorithms demonstrated consistent patterns for the different algorithms. Random Forest and RBF SVM are above most of the other algorithms. We kept the threshold of maintaining the subject to 35% of the remaining data, as we believed that when a subject has little data left, the high accuracy from that subject contributes little to our research.

4.2 Identify Noisy Sessions

Figure 3 shows what the Random Forest and the RBF SVM in phase 1 predicted during the 42 seconds of subject 3's task 1 for all six sessions. As we can see, both the Random Forest and the RBF SVM mainly produced results with relatively high accuracy in sessions one, four, and six. On the other hand, session two and three had much noise, and session five predicted task 3 for the majority of the time. To minimize the impact of noisy datasets, we calculated the accuracy compared to the ground truth based on the output of Random Forest for each session for each task in phase 1.

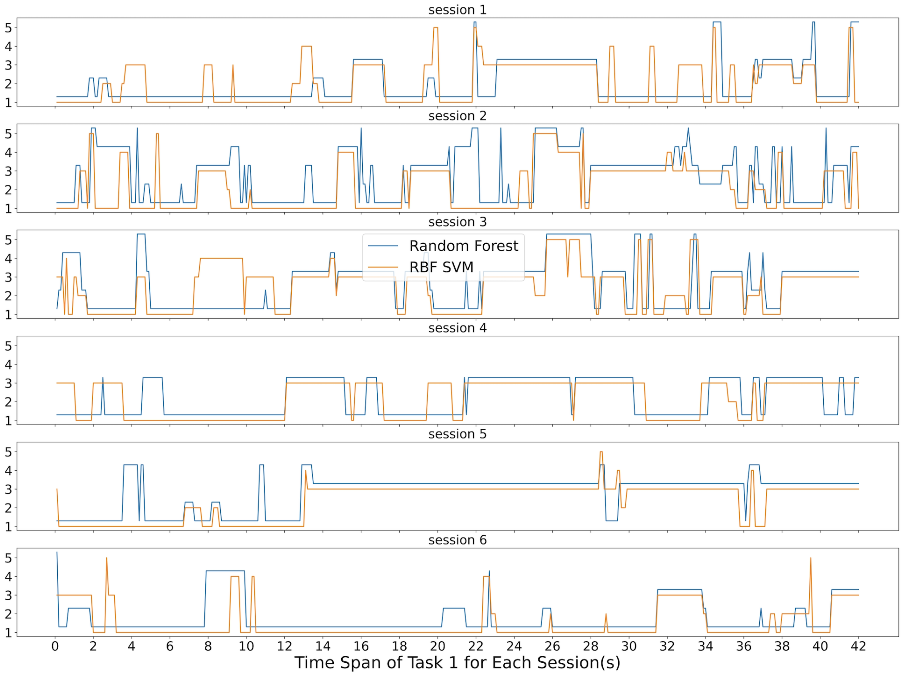

Fig. 3. subject 3's task 1's RF and SVM in Phase 1

We excluded any sessions that yielded an accuracy of less than 50%. In the case of Figure 3, we excluded sessions two, three, and five for further machine learning analysis. We reported to subject three and started discussing what may have happened in these sessions. With the exclusion of noisy sessions, the new accuracy for the Random Forest in phase 2 is referred to as "Random Forests Phase 2" later in the paper. We also performed Time Majority Voting on this cleaner dataset.

4.3 Time Majority Voting

As shown in Table 3, the Time Majority Voting (TMV) has achieved a higher accuracy for subject three, all the six sessions. A value of -1 means that we excluded that session for that task, as we discussed in the previous section. The Random Forest classifier also reached a higher accuracy after cleaning the noisy sessions. As Figure 4 and Figure 5 shows, the pattern is consistent across all the subjects. Figure 5 also shows a clear pattern that not only the Random Forest but also the RBF SVM also increased accuracy in phase 2 across all subjects. Table 4 shows the TMV achieved an 80% average accuracy with an average 74.3 seconds run time. The runtime consists of 39.1 seconds of running the Random Forest and 29.5 seconds of running the RBF SVM. The time for the actual voting

S/T	t1 RF	**t1 T**	t2 RF	**t2 T**	t3 RF	**t3 T**	t4 RF	**t4 T**	t5 RF	**t5**
s1	0.686	0.745	0.669	0.683	**-1**	**-1**	0.569	0.781	0.798	0.898
s2	**-1**	**-1**	**-1**	**-1**	**-1**	**-1**	0.693	0.838	0.633	0.707
s3	**-1**	**-1**	**-1**	**-1**	**-1**	**-1**	0.731	0.824	0.938	0.969
s4	0.543	0.59	0.743	0.869	0.74	0.788	0.521	0.671	**-1**	**-1**
s5	**-1**	**-1**	0.645	0.738	0.588	0.8	0.593	0.743	0.662	0.829
s6	0.762	0.871	**-1**	**-1**	0.507	0.714	0.812	0.917	0.681	0.845
Average	0.663	0.736	0.686	0.763	0.612	0.767	0.653	0.796	0.742	0.85

Table 3. TCR, Accuracy of Top 1 (RF) Phase 2, and TMV(T) (subject 3, by Session(S)/Task(T))

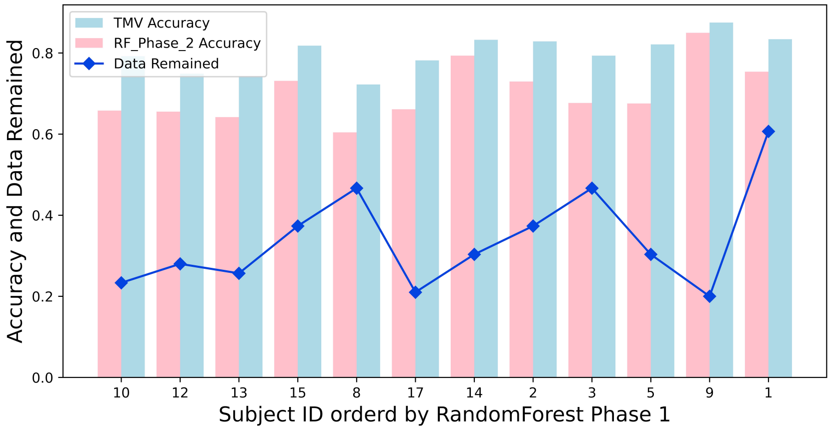

Fig. 4. TMV and RF Phase 2

Algorithms	Average Accuracy	Average code run-time (s)
Time Majority Voting	**0.80**	39.1 + 29.5 + 5.7 = 74.3
Random Forest Phase 2	0.7	39.1
RBF SVM Phase 2	0.66	29.5
Random Forest Phase 1	0.55	39.1
RBF SVM Phase 1	0.53	29.5

Table 4. TMV and Random Forest with Accuracy and Run-time

is, on average, 5.7 seconds. The training process is the most time-consuming part of this analysis.

Figure 6 shows what the Time Majority Voting(TMV) predicted during the 42 seconds of subject 3's task 1 for all six sessions. As you can see, sessions 2, 3, and 5 have values of -1, which means that they have been excluded. Sessions 1, 4, and 6 have less noise than sessions 2, 3, and 6, and the accuracy is relatively high.

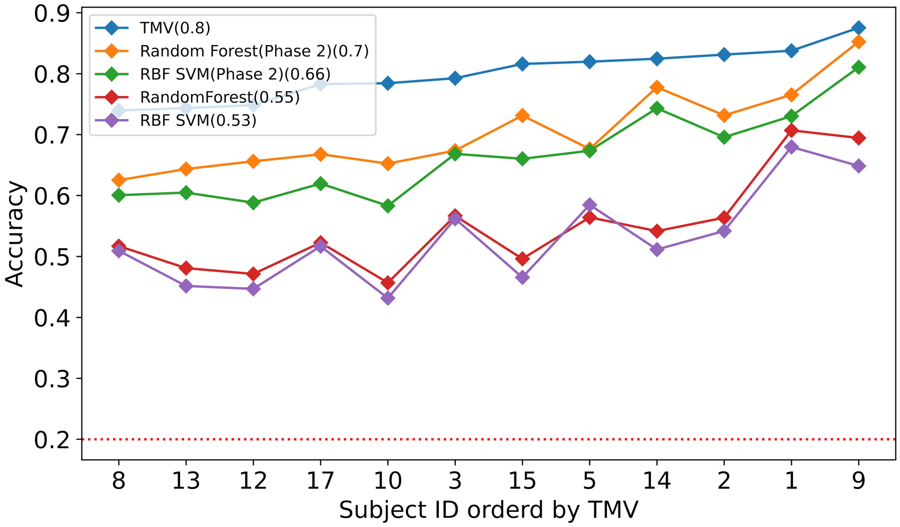

Fig. 5. TMV, RF Phase 2, RBF SVM Phase 2, RF phase 1, and RBF SVM Phase 1

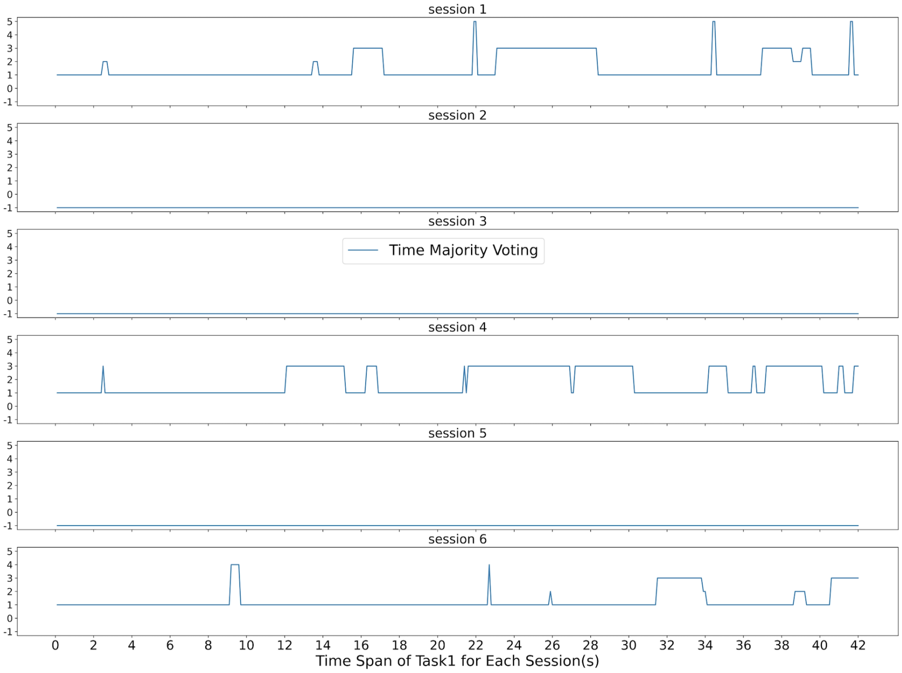

Fig. 6. subject 3's task 1's TMV

5 Discussion

5.1 Accuracy and Data Remain

The innovation of this method is mainly about temporal dependency. As [15, 24, 34, 36, 37] mentioned, EEG data has a significant temporal dependency. The signal of the same task takes about 12 to 18 seconds to switch to the next task. Using majority voting can catch this type of time continuity effect. If both classifiers recognize the same pattern, it is more likely to assure the results. If both of the classifiers recognize the same pattern that is different from the majority result, it is possible that the participants were doing other tasks during the data collection. If only one classifier detects some unusual behaviors, we label it as the majority task of the session. Thus we highlight the noise and keep the remaining data to reflect more on the time continuity nature of the EEG signals.

5.2 Runtime and Training Data

As Figure 7 shows, the run time directly correlated to the training data size. After the data pre-processing, we cleaned up the noise with a plateau longer than a threshold, as mentioned in [24, 32, 34]. We first identified more noisy sessions during the Time Majority Voting process based on the top two classifiers. During this step, more sessions were excluded for further analysis. The training process was the most time-consuming step during the coding running process. Thus the runtime changed together with the size of the training data.

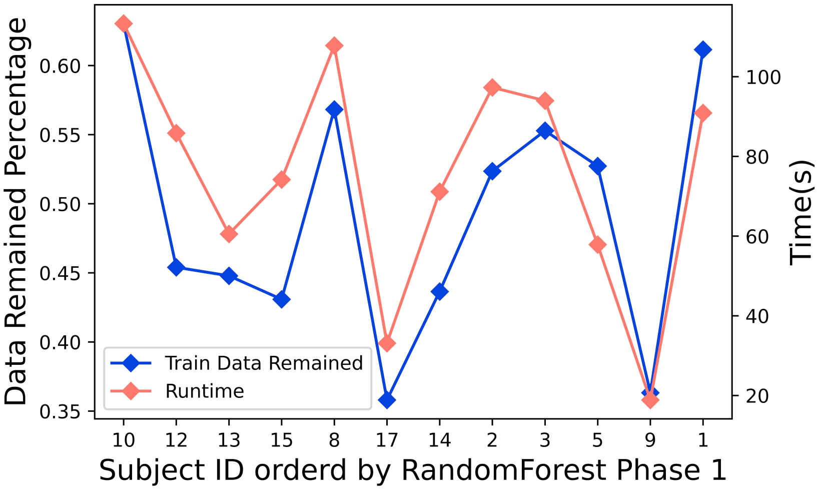

Fig. 7. Data Remained for Training and Run-time

5.3 Interpretability

Figure 3 and Figure 6 shows two examples of the feedback results we present to the end-users. In this figure 3, the six sessions of subject three, task one is listed as six horizontal charts. The first chart represents session one. Both the Random Forest and the RBF SVM identified the majority task as task one. And for the results of the different predictions, both of the classifiers agreed at some time spots, but not all of them. Our Time Majority Voting algorithm favorite the majority voting results.

We started with the sessions with good prediction results when demonstrating these figures to each subject. For example, in this figure 3, sessions one, four, and six show pretty consistent patterns. The majority of task prediction results were the designed task one. That implies that the subject may spend more time on task one as planned during these sessions. Session two and three had a lot of different prediction results from both classifiers. Thus we suspected some unexpected reason might cause this situation. We referred back to the experiment notes and reached out to subject one. After discussing with the end-user, we figured out that he had many issues with the sensor signals and was adjusting the EEG headset most of the time during the sessions 2 and 3. Thus we had more information to exclude these sessions from further data analysis. Session five is another situation. The data implied that the subject was doing task three, but the experiment notes were missing for that session, and the subject did not remember the details about that session. Thus, we left a question mark for that session, excluded the session for now, and came up with an improvement plan to keep better experiment notes. This type of machine learning result is explainable to the end-user.

Such interpretability could contribute to a better understanding of the results and better design for future experiments.

6 Conclusion

This paper investigated the state-of-the-art machine learning algorithms that can run on mainstream personal computers for EEG-based BCI. We then proposed a new algorithm, Time Majority Voting (TMV). The results demonstrated that TMV outperformed other existing classifiers. The run time for TMV is still within the acceptable range on a PC. The interpretability of TMV can contribute to a better understanding of the machine learning analysis and an improved design for future experiments.

Bibliography

1. An, S., Ogras, U.Y.: Mars: mmwave-based assistive rehabilitation system for smart healthcare. ACM Transactions on Embedded Computing Systems (TECS) **20**(5s), 1–22 (2021)
2. Appriou, A., Cichocki, A., Lotte, F.: Modern machine-learning algorithms: for classifying cognitive and affective states from electroencephalography signals. IEEE Systems, Man, and Cybernetics Magazine **6**(3), 29–38 (2020)
3. Basaklar, T., Tuncel, Y., An, S., Ogras, U.: Wearable devices and low-power design for smart health applications: challenges and opportunities. In: 2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). pp. 1–1. IEEE (2021)
4. Bashivan, P., Bidelman, G.M., Yeasin, M.: Spectrotemporal dynamics of the EEG during working memory encoding and maintenance predicts individual behavioral capacity. European Journal of Neuroscience **40**(12), 3774–3784 (2014)
5. Bashivan, P., Rish, I., Heisig, S.: Mental state recognition via wearable eeg. arXiv preprint arXiv:1602.00985 (2016)
6. Bashivan, P., Rish, I., Yeasin, M., Codella, N.: Learning representations from EEG with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448 (2015)
7. Bhat, G., Tuncel, Y., An, S., Ogras, U.Y.: Wearable IoT Devices for Health Monitoring. In: TechConnect Briefs 2019. pp. 357–360 (2019)
8. Bird, J.J., Manso, L.J., Ribeiro, E.P., Ekart, A., Faria, D.R.: A study on mental state classification using eeg-based brain-machine interface. In: 2018 International Conference on Intelligent Systems (IS). pp. 795–800. IEEE (2018)
9. Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
10. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
11. Breiman, L.: Classification and regression trees. Routledge (2017)
12. Chen, L., Lu, Y., Wu, C.T., Clarke, R., Yu, G., Van Eyk, J.E., Herrington, D.M., Wang, Y.: Data-driven detection of subtype-specific differentially expressed genes. Scientific reports **11**(1), 1–12 (2021)
13. Chevalier, J.A., Gramfort, A., Salmon, J., Thirion, B.: Statistical control for spatio-temporal meg/eeg source imaging with desparsified multi-task lasso. arXiv preprint arXiv:2009.14310 (2020)
14. Coyle, D., Principe, J., Lotte, F., Nijholt, A.: Guest editorial: Brain/neuronal-computer game interfaces and interaction. IEEE Transactions on Computational Intelligence and AI in games **5**(2), 77–81 (2013)
15. Craik, A., He, Y., Contreras-Vidal, J.L.: Deep learning for electroencephalogram (eeg) classification tasks: a review. Journal of neural engineering **16**(3), 031001 (2019)
16. Derby, J.J., Zhang, C., Seebeck, J., Peterson, J.H., Tremsin, A.S., Perrodin, D., Bizarri, G.A., Bourret, E.D., Losko, A.S., Vogel, S.C.: Computational

modeling and neutron imaging to understand interface shape and solute segregation during the vertical gradient freeze growth of babrcl: Eu. Journal of Crystal Growth **536**, 125572 (2020)

17. Devlaminck, D., Waegeman, W., Bauwens, B., Wyns, B., Santens, P., Otte, G.: From circular ordinal regression to multilabel classification. In: Proceedings of the 2010 Workshop on Preference Learning (European Conference on Machine Learning, ECML). p. 15 (2010)
18. Gu, J., Zhao, Z., Zeng, Z., Wang, Y., Qiu, Z., Veeravalli, B., Goh, B.K.P., Bonney, G.K., Madhavan, K., Ying, C.W., et al.: 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). pp. 5814–5817. IEEE (2020)
19. Kastrati, A., Płomecka, M.M.B., Pascual, D., Wolf, L., Gillioz, V., Wattenhofer, R., Langer, N.: Eegeyenet: a simultaneous electroencephalography and eye-tracking dataset and benchmark for eye movement prediction. arXiv preprint arXiv:2111.05100 (2021)
20. Kaya, M., Binli, M.K., Ozbay, E., Yanar, H., Mishchenko, Y.: A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces. Scientific data **5**(1), 1–16 (2018)
21. Li, S., Zhao, Z., Xu, K., Zeng, Z., Guan, C.: Hierarchical consistency regularized mean teacher for semi-supervised 3d left atrium segmentation. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). pp. 3395–3398. IEEE (2021)
22. Lotte, F.: A tutorial on EEG signal-processing techniques for mental-state recognition in brain–computer interfaces. In: Guide to Brain-Computer Music Interfacing, pp. 133–161. Springer (2014)
23. Lotte, F.: Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces. Proceedings of the IEEE **103**(6), 871–890 (2015)
24. 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. Journal of neural engineering **15**(3), 031005 (2018)
25. Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B.: A review of classification algorithms for EEG-based brain–computer interfaces. Journal of neural engineering **4**(2), R1 (2007)
26. Lotte, F., Guan, C.: Regularizing common spatial patterns to improve bci designs: unified theory and new algorithms. IEEE Transactions on biomedical Engineering **58**(2), 355–362 (2010)
27. Lotte, F., Jeunet, C.: Towards improved bci based on human learning principles. In: The 3rd International Winter Conference on Brain-Computer Interface. pp. 1–4. IEEE (2015)
28. Lotte, F., Jeunet, C., Mladenović, J., N'Kaoua, B., Pillette, L.: A bci challenge for the signal processing community: considering the user in the loop (2018)
29. Miller, K.J.: A library of human electrocorticographic data and analyses. Nature human behaviour **3**(11), 1225–1235 (2019)

30. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. NeurIPS (2017)
31. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
32. Qu, X., Hall, M., Sun, Y., Sekuler, R., Hickey, T.J.: A personalized reading coach using wearable EEG sensors-a pilot study of brainwave learning analytics. In: CSEDU (2). pp. 501–507 (2018)
33. Qu, X., Liu, P., Li, Z., Hickey, T.: Multi-class time continuity voting for eeg classification. In: International Conference on Brain Function Assessment in Learning. pp. 24–33. Springer (2020)
34. Qu, X., Liukasemsarn, S., Tu, J., Higgins, A., Hickey, T.J., Hall, M.H.: Identifying clinically and functionally distinct groups among healthy controls and first episode psychosis patients by clustering on eeg patterns. Frontiers in psychiatry p. 938 (2020)
35. Qu, X., Mei, Q., Liu, P., Hickey, T.: Using eeg to distinguish between writing and typing for the same cognitive task. In: International Conference on Brain Function Assessment in Learning. pp. 66–74. Springer (2020)
36. Qu, X., Sun, Y., Sekuler, R., Hickey, T.: EEG markers of stem learning. In: 2018 IEEE Frontiers in Education Conference (FIE). pp. 1–9. IEEE (2018)
37. Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T.H., Faubert, J.: Deep learning-based electroencephalography analysis: a systematic review. Journal of neural engineering **16**(5), 051001 (2019)
38. Zeng, Z., Zhao, W., Qian, P., Zhou, Y., Zhao, Z., Chen, C., Guan, C.: Robust traffic prediction from spatial-temporal data based on conditional distribution learning. IEEE Transactions on Cybernetics (2021)
39. Zhang, X., Yao, L., Wang, X., Monaghan, J.J., Mcalpine, D., Zhang, Y.: A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers. Journal of Neural Engineering (2020)
40. Zhao, Z., Chopra, K., Zeng, Z., Li, X.: Sea-net: Squeeze-and-excitation attention net for diabetic retinopathy grading. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 2496–2500. IEEE (2020)
41. Zhao, Z., Qian, P., Hou, Y., Zeng, Z.: Adaptive mean-residue loss for robust facial age estimation. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). IEEE (2022)
42. Zhao, Z., Zhang, K., Hao, X., Tian, J., Chua, M.C.H., Chen, L., Xu, X.: Bira-net: Bilinear attention net for diabetic retinopathy grading. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 1385–1389. IEEE (2019)