Machine Learning Algorithms Performance on BCI Competition 4 Dataset 4

Gauttam Jangir[1], Gaurav Purohit[2], Nisheeth Joshi[1]

Department of Computer Science, Banasthali Vidyapith, Rajasthan, India

Senior Scientist, CSIR-CEERI Pilani, Rajasthan, India[2]

Corresponding author: Gauttam Jangir, Email: gauttamj49@gmail.com

Brain computer interface (BCI) is evolving rapidly, as machine learning algorithms are getting better. Algorithms performance plays central role while selecting for real time use cases. One such case is robotic fingers movement, as they play important role in BCI due to their involvement in day-to-day task. In this paper one such standard dataset “BCI competition 4 dataset iv” is considered to evaluate the performance of various machine learning algorithms like Linear Regression (LR),Bayesian Ridge (BR), Light Gradient Boosting (LGB) and decision Tree (DT). Algorithms performance also evaluated on pre-processed dataset. The final result is presented with highest achieved correlation value as 0.76 (noisy dataset) and 0.80 (cleaned dataset) by using the DT.

Keywords: Brain computer interface(BCI), Algorithms, robotic fingers, performance, Linear Regression (LR), Bayesian Ridge (BR), Light Gradient Boosting (LGB) and decision Tree (DT).

2025. In Puneet Kumar Gupta (ed.), *Computational Models for Intelligence and Automation*, 70–84. Computing & Intelligent Systems, SCRS, India. DOI: https://doi.org/10.56155/978-81-975670-1-8-7

1. Introduction

Brain-Computer Interfaces (BCIs) represent a new dive in technology, fundamentally transforming how humans interact with external devices. At the heart of this innovation lies the ability to bridge the gap between the human brain and various forms of machinery, an achievement made possible through the use of machine learning algorithms [1]. These algorithms are essential for decoding the complex signals generated by the brain and converting them into actionable commands in real-time [3][4] [10]. This capability enables a diverse range of applications, from medical rehabilitation to enhancing human capabilities in numerous domains. The continuous improvement of machine learning techniques has significantly accelerated the progress in the field of BCIs. These algorithms are meticulously designed to process and interpret the electrical activity produced by the brain, known as electroencephalogram (EEG) signals, with a high degree of accuracy and speed. Accurate decoding of these signals is crucial for real-time BCI applications, where the timing and precision of interpreting brain activity directly influence the usability and effectiveness of the system.

One of the most promising applications of BCI technology lies in the control of robotic fingers. The human hand, renowned for its agile and precise movements, plays a critical role in our daily interactions with the environment. For individuals with motor impairments, the ability to control robotic fingers via BCIs can significantly enhance their quality of life, granting them a degree of independence and functionality previously estimated unattainable. The control of robotic fingers using BCIs involves the detection and interpretation of specific brain signals associated with finger movements. This process demands not only the accurate identification of these signals but also the ability to translate them into precise and timely commands that the robotic fingers can execute. This application highlights the potential of BCIs to provide practical solutions to real-world problems, particularly in the domain of assistive technology.

Selecting the most appropriate machine learning algorithm for BCI applications is a pivotal step in developing effective and reliable systems. Each algorithm possesses unique strengths and weaknesses in terms of accuracy, computational efficiency, and interpretability. Consequently, a comprehensive evaluation of different algorithms is imperative to determine the best approach for specific BCI tasks. This research focus to evaluate the performance (correlation) of several well-established machine learning procedures (algorithms) in the context of BCI-based robotic finger movement control. We focused on algorithms that are Linear Regression (LR), Decision Tree (DT), Bayesian Ridge (BR), Light Gradient Boosting (LGB). These algorithms are considered due to their diverse characteristics and wide uses in various predictive modeling tasks. The primary focus of this applied research work is to identify the algorithm that achieve the most accurate finger movement predictions. We measure the performance of each algorithm using the association (correlation) coefficient metric, which quantifies the strength and direction of the relationship among the predicted and actual finger movements. A higher correlation coefficient indicates better predictive performance.

2. Literature Review

Brain-Computer Interfaces (BCIs) have added significant attention in recent years due to their potential to revolutionize how humans interact with machines or devices, particularly for individuals with motor impairments [11]. An important aspect of BCI development is the application of machine learning algorithms to accurately interpret electroencephalogram (EEG) signals and translate them into actionable commands [7] [8] [9]. Linear Regression (LR) is one of the simplest algorithm used by Bashashati et al. [12], who noted that linear models might not capture the complexity of brain signals effectively. Decision Trees (DT) are intuitive models that split data based on feature values to predict, used by Lotte et al. [18] demonstrated that DTs could effectively classify EEG signals, although they are prone to overfitting, especially with noisy data. Bayesian Ridge Regression (BR) introduces Bayesian inference to the regression model, used by Tipping [19] emphasized its regularization capabilities, making it robust against overfitting. LightGBM is designed for speed and efficiency, using a leaf-wise growth strategy and histogram-based decision tree learning. Ke et al. [17] highlighted LightGBM's ability to handle large datasets with low memory usage, making it ideal for real-time BCI applications.

Table 1. Algorithms Comparison

Aspect	Linear Regression (LR)	Decision Tree (DT)	Light Gradient Boosting (LGB)	Bayesian Ridge (BR)
Type	Parametric, Regression	Non-parametric, Tree-based	Non-parametric, Ensemble, Boosting	Parametric, Bayesian, Regression
Complexity	Low	Moderate	High	Moderate
Scalability	Medium	Medium	High	Medium
Handling Non-linearity	Poor	Good	Excellent	Poor
Robustness to Outliers	Poor	Poor	High	Moderate
Interpretability	High	High	Low	High
Training Speed	Fast	Moderate (depends on tree depth and dataset size)	Fast (optimized for large datasets)	Moderate
Overfitting Tendency	High (without regularization)	High	Moderate (requires tuning)	Low
Feature Engineering	Requires scaling and transformation	No scaling needed	Handles categorical features natively	Requires scaling and transformation
Ensemble Parameters	Not part of an ensemble	Can be part of an ensemble (e.g., Random Forest)	Ensemble method (Boosting, multiple trees)	Not part of an ensemble

Algorithms comparison is listed in table 1. LR have the lowest complexity and LGB have the highest due to its ensemble approach. DT performs poor while handling outliers but LGB handles outliers in good manner due to its ensemble approach and on the same concept DT suffers from the overfitting tendency but LGB not. LR also suffers from overfitting depending on the data distribution patterns. BR uses the prior information of data that is used to handle the overfitting scenario.

3. Experimental Setup

In this machine learning experimental setup on BCI competition 4 dataset iv, understating dataset is important before applying algorithms over it. This dataset is used by many researchers to evaluate the performance.

Dataset

The "BCI competition 4 dataset iv" dataset is recoded from three epileptic patients by using the Synamps2 amplifiers (48-64 electrodes)at university of Washington. Objective behind the setup is to develop the model to predict the finger movement. To generate the visual stimuli in the brain, visual cues were presented to subjects on time frame that includes rest period also [20]. This process is repeated for 6 minutes that include train and test part of the dataset. Fingers movement collected using the data glove.

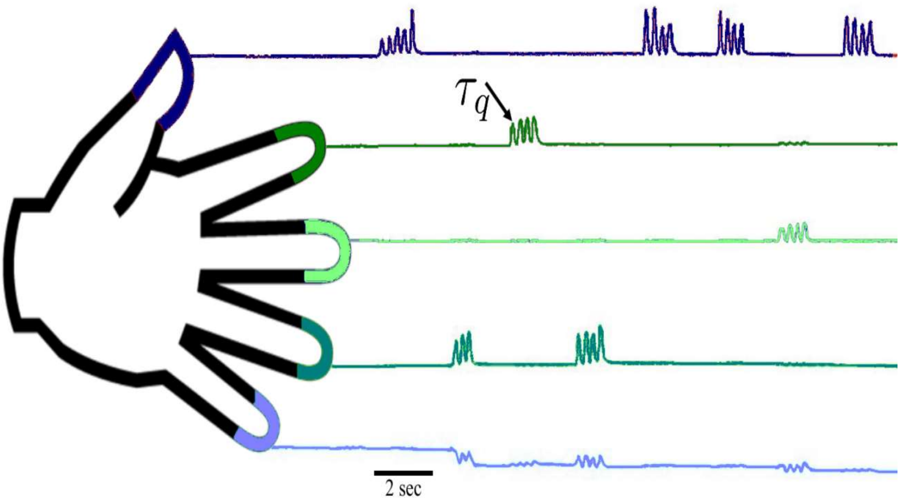

Figure 1. Capturing individual finger flexion: [13]

Dataset compiled using ECoG signals and fingers movement data. Histogram is one of the tool to understand the nature of data [6]. Subject 1 fingers movement is presented using the histogram in figure 2. Thumb finger movement near to zero is more in place of other values, similar kind of pattern is also visible for remaining fingers too.

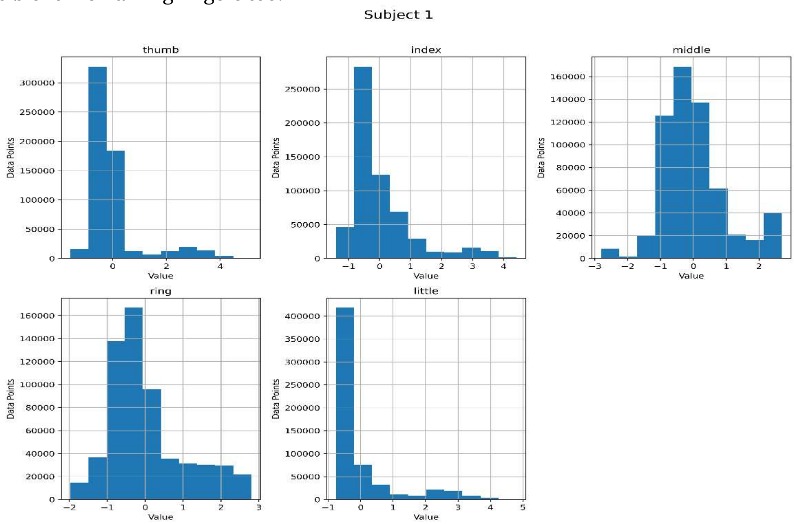

Figure 2. Subject 1 Histogram

Figure 3 histogram presents the subject 2. Most of the values are zero or nearby to zero, other values are less like 2, 4 or 6 and rarely 8. This pattern is similar to subject 1 pattern of histogram.

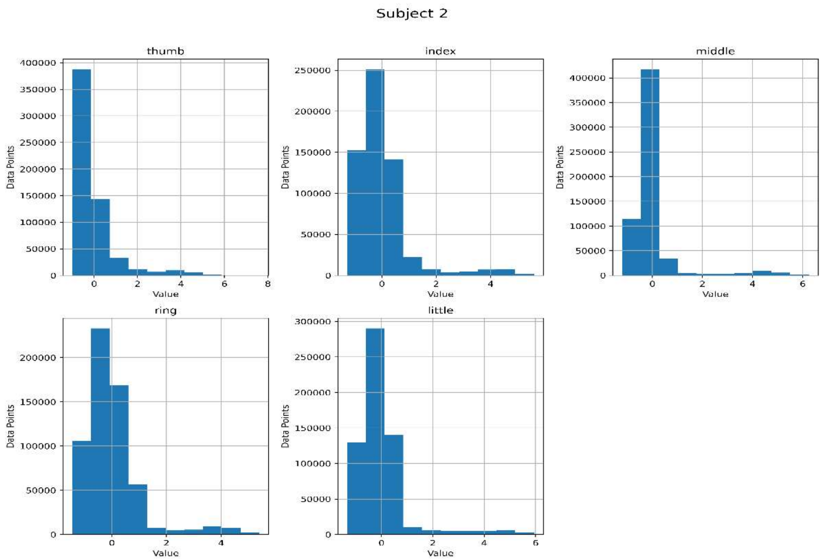

Figure 3. Subject 2 Histogram

Figure 4 histogram shows the data distribution of subject 3. Most of the values are in the range of zero and some of the values are out of the range of zero that are near to 2, 4 or 6. After observation of all histograms, outliers are visible. Outliers degrade the accuracy of algorithms.

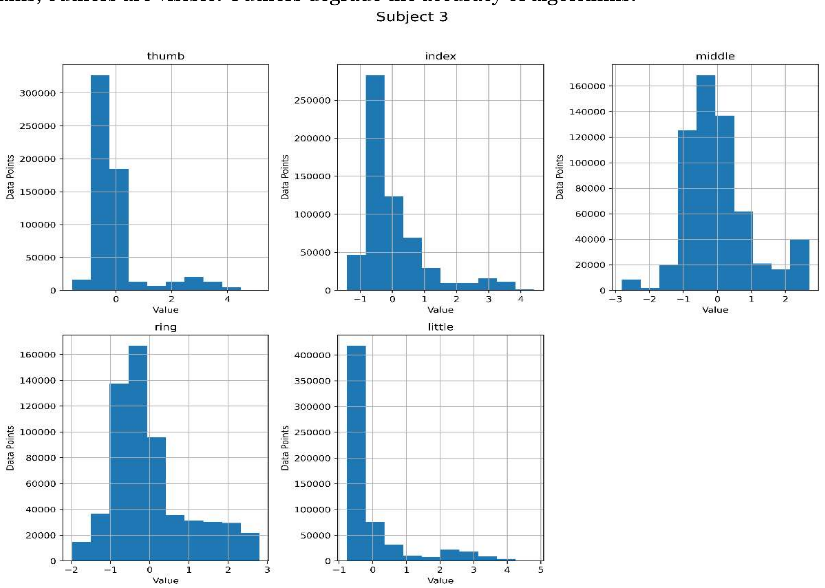

Figure 4. Subject 3 Histogram

Dataset Preprocessing

Dataset pre-processing remove the outliers and increase the contribution of quality data. Isolation forest [21] is the one of the algorithm that efficiently remove the outliers. It operates on the principle that anomalies are both rare and different from normal data points, making them easier to isolate. The algorithm builds multiple random decision trees, where each tree is constructed by randomly selecting a feature and splitting it at a random value within its range. This recursive splitting continues until each data point is isolated in a tree. Anomalies, being sparse and located in extreme regions of the dataset, tend to be isolated closer to the root of the tree, requiring fewer splits compared to normal points as shown in figure 5.

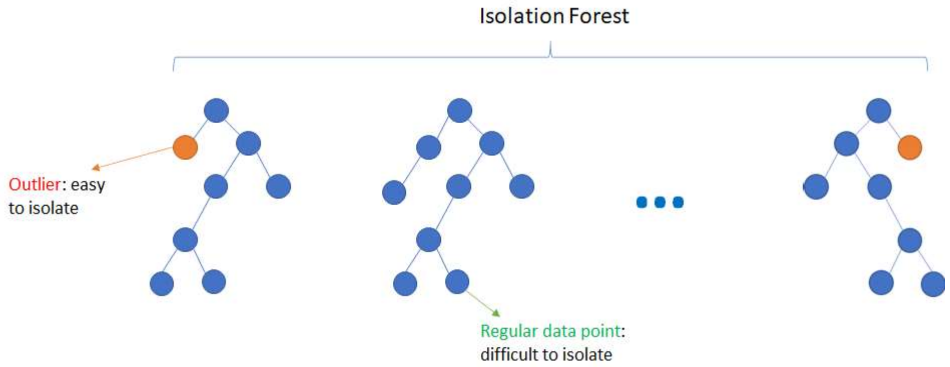

Figure 5. Isolation Forest [22]

The algorithm aggregate results from all trees to calculate an anomaly score for each point, with higher scores indicating a higher likelihood of being an outlier. Isolation Forest is efficient, scalable to large datasets, and works well with high-dimensional data. The "BCI competition 4 dataset iv"is processed through the IF algorithm and data quality increased as show in figure 6,7 and 8 for subject 1,2 and 3.

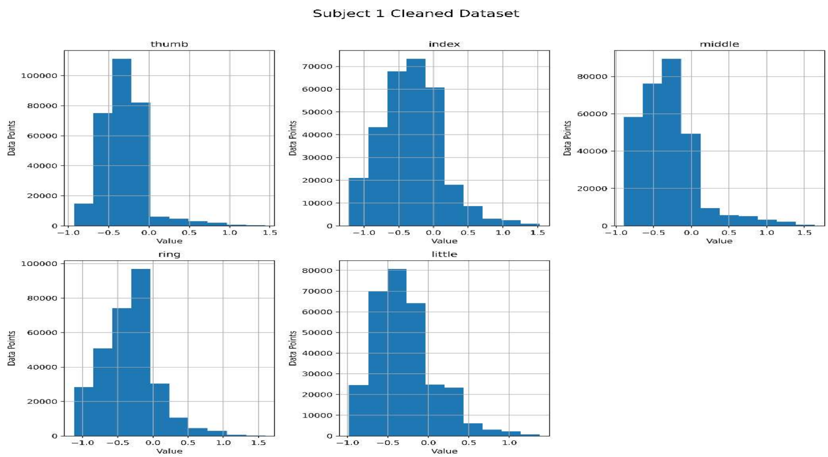

Figure 6. Subject 1 Histogram (Cleaned Dataset)

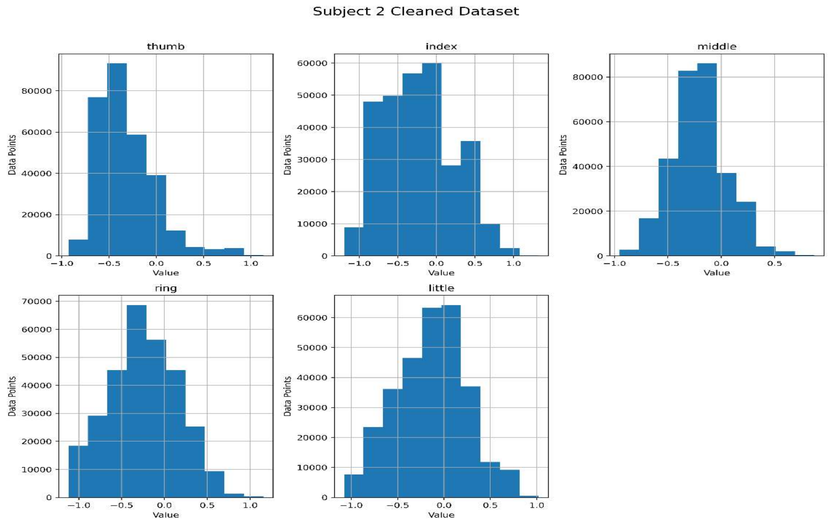

Figure 7. Subject 2 Histogram (Cleaned Dataset)

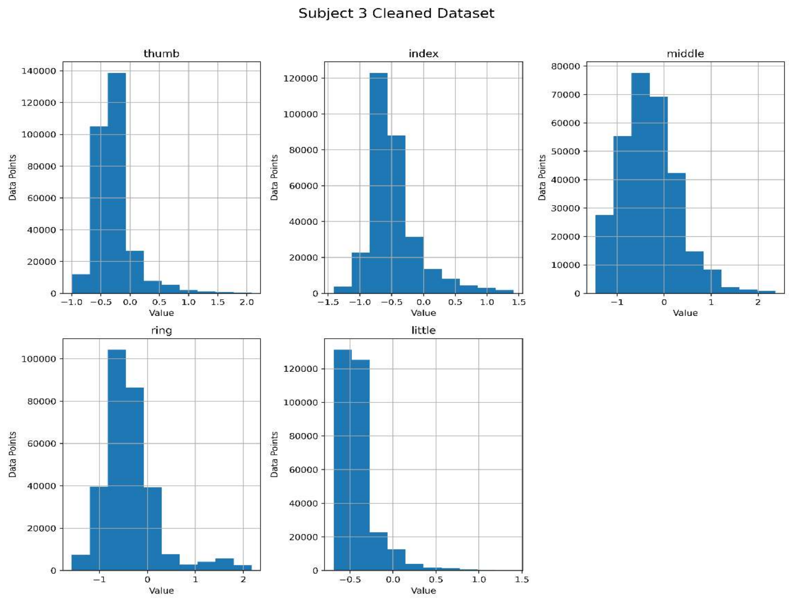

Figure 8. Subject 3 Histogram (Cleaned Dataset)

Subject 1 noisy data has distribution range from 0 to 6 but mostly data lies in the range -1 to 1 for the thumb. After pre-processing data range is -1 to 1.5 (figure 5) that show that outliers are removed but not completely. Similar pattern is visible for other fingers and subjects.

4. Methodology

Machine learning algorithms are pivotal in Brain-Computer Interfaces (BCIs) due to their ability to decode complex brain signals and convert them into actionable commands. In the context of BCI-based robotic finger movement control, various algorithms can be utilized to enhance accuracy, efficiency, and reliability. Each algorithm has unique characteristics and applications, making them appropriate for different aspects of modelling. Mentioned algorithms are applied on the dataset one by one. Dataset is divided into train and test in the part by splitting in the portion of 75% and 25%. Algorithms are applied as shown in figure 9.

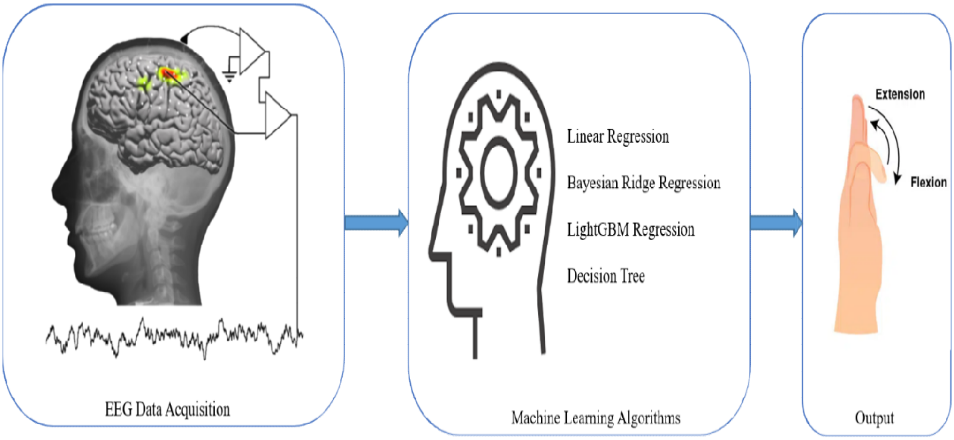

Figure 9. Brain Computer Interface Methodology

LR is one of the earliest and mostly used algorithms in machine learning. It iterates model to find the relationship between a dependent variable (y) and one or more independent variables (X) by finding a linear equation to observed data [15] [16].BR Regression is a linear regression procedure that integrates bayesian ideologies to provide a probabilistic approach to parameter estimation. It applies regularization by imposing a prior distribution on the coefficients. LGBM is a slope (gradient) boosting framework that uses a histogram dependent approach to speed up training phase and shrink memory usage [2][5] [14]. It is considered for effectiveness and scale-ability, making it suitable for large datasets. DT are non-parametric models that fragment the data into sub-sets according to the input features. Each internal node element presents a decision based on a certain feature, individually branch represents the result of the decision, and each leaf (last)node represents a continuous value or class label.

5. Results

BCI Competition 4 dataset iv is the standard dataset used by many researchers related to predict the finger movement of three epileptic patients. Linear regression is the first algorithm applied on this dataset and 0.21 is average correlation value achieved as shown in table 1 and figure 10. Table 2 and figure 11 shown the performance of LR on cleaned dataset but it reduces to 0.18 from 0.21 on noisy dataset. BR performance is shown in Table 3 and figure 12 that achieve 0.21 correlation value. BR applied on cleaned dataset and correlation value is 0.18 as shown in table 4 and figure 13. Both LR and BR performance goes down as applied on the cleaned dataset.

Table 1. Linear Regression (Noisy Dataset)

	Thumb	Index	Middle	Ring	Little	AVG
Sub1	0.14	0.14	0.17	0.17	0.14	0.15
Sub2	0.17	0.17	0.20	0.20	0.14	0.18
Sub3	0.26	0.33	0.26	0.33	0.24	0.29
AVG	0.19	0.22	0.21	0.23	0.18	0.21

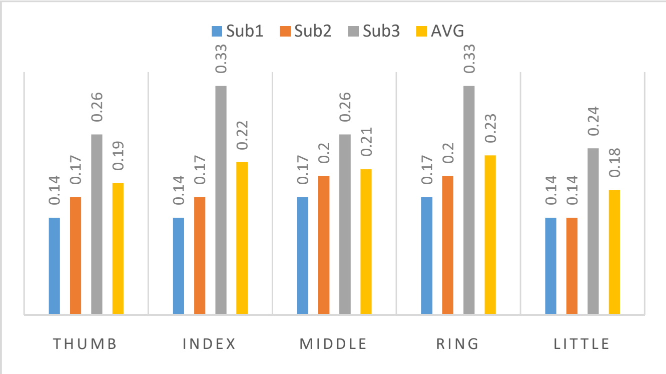

Figure 10. Linear Regression Performance on Noisy Dataset

Table 2. Linear Regression (Cleaned Dataset)

	Thumb	Index	Middle	Ring	Little	AVG
Sub1	0.14	0.14	0.17	0.17	0.17	0.16
Sub2	0.14	0.17	0.17	0.14	0.14	0.15
Sub3	0.17	0.22	0.22	0.26	0.22	0.22
AVG	0.15	0.18	0.19	0.19	0.18	0.18

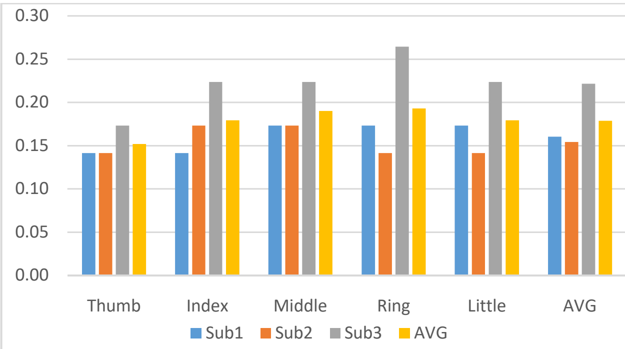

Figure 11. Linear Regression Performance on Cleaned Dataset

Table 3. Bayesian Ridge (Noisy Dataset)

	Thumb	Index	Middle	Ring	Little	AVG
Sub1	0.14	0.14	0.17	0.17	0.14	0.15
Sub2	0.20	0.17	0.17	0.20	0.14	0.18
Sub3	0.28	0.33	0.26	0.33	0.24	0.29
AVG	0.21	0.22	0.20	0.23	0.18	0.21

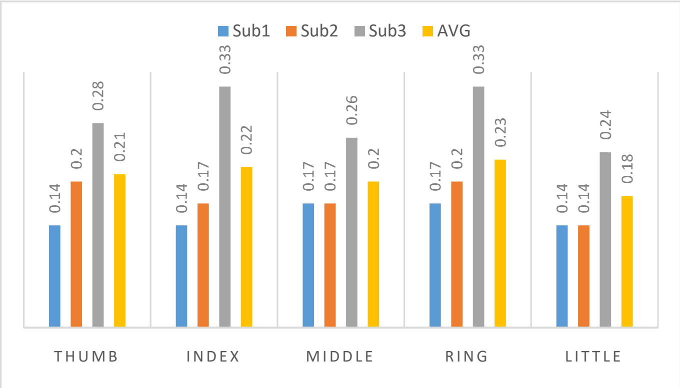

Figure 12. Bayesian Ridge Performance on (Noisy Dataset)

Table 4. Bayesian Ridge (Cleaned Dataset)

	Thumb	Index	Middle	Ring	Little	AVG
Sub1	0.17	0.14	0.17	0.17	0.17	0.17
Sub2	0.14	0.17	0.17	0.14	0.14	0.15
Sub3	0.17	0.22	0.22	0.28	0.22	0.23
AVG	0.16	0.18	0.19	0.20	0.18	0.18

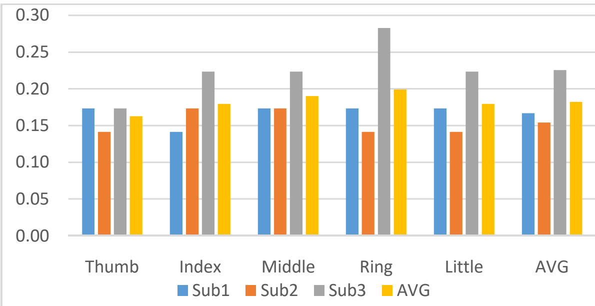

Figure 13. Bayesian Ridge Performance on (Cleaned Dataset)

Table 5. LGB Regression (Noisy Dataset)

	Thumb	Index	Middle	Ring	Little	AVG
Sub1	0.45	0.42	0.47	0.46	0.44	0.45
Sub2	0.66	0.63	0.67	0.58	0.62	0.63
Sub3	0.59	0.57	0.51	0.57	0.54	0.55
AVG	0.56	0.54	0.55	0.54	0.53	0.54

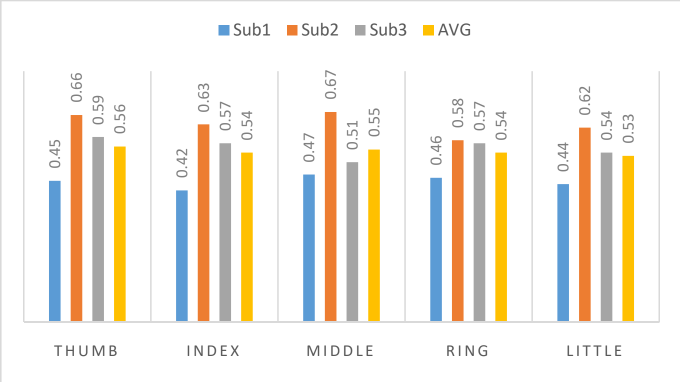

Figure 14. LGB Performance on (Noisy Dataset)

Table 6. LGB Regression (Cleaned Dataset)

	Thumb	Index	Middle	Ring	Little	AVG
Sub1	0.48	0.48	0.50	0.49	0.49	0.49
Sub2	0.70	0.64	0.63	0.62	0.63	0.65
Sub3	0.57	0.57	0.55	0.58	0.58	0.57
AVG	0.58	0.56	0.56	0.57	0.57	0.57

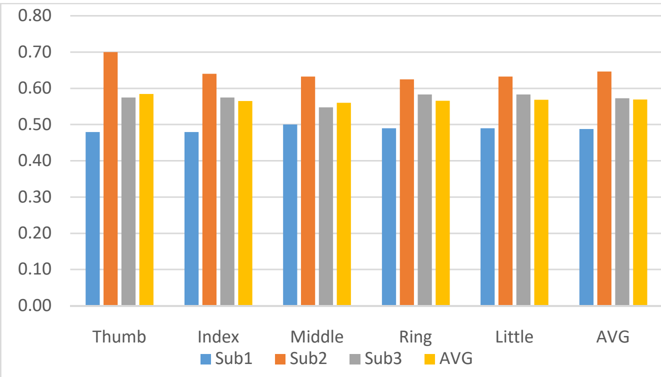

Figure 15. LGB Performance on (Cleaned Dataset)

Table 7. Decision Tree (Noisy Dataset)

	Thumb	Index	Middle	Ring	Little	AVG
Sub1	0.69	0.72	0.75	0.73	0.73	0.73
Sub2	0.86	0.89	0.87	0.84	0.87	0.87
Sub3	0.69	0.65	0.66	0.69	0.67	0.67
AVG	0.75	0.75	0.76	0.75	0.76	0.75

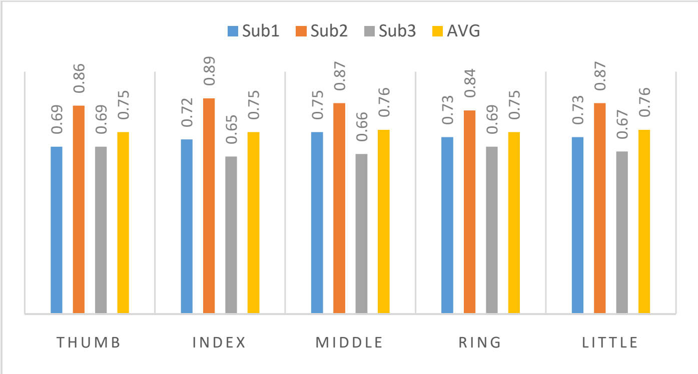

Figure 16. DT Performance on (Noisy Dataset)

Table 8. Decision Tree (Cleaned Dataset)

	Thumb	Index	Middle	Ring	Little	AVG
Sub1	0.76	0.79	0.75	0.79	0.78	0.78
Sub2	0.91	0.92	0.92	0.91	0.91	0.91
Sub3	0.73	0.69	0.71	0.71	0.70	0.71
AVG	0.80	0.80	0.80	0.80	0.80	0.80

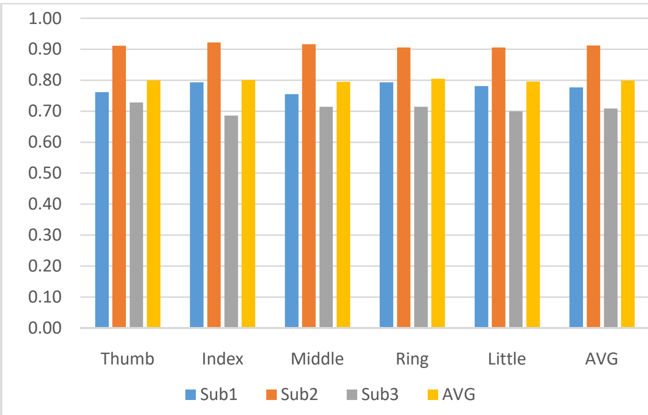

Figure 17. DT Performance on (Cleaned Dataset)

Table 9. Performance Comparison (Noisy Dataset)

S. No.	Algo.	Sub1	Sub2	Sub3	AVG
1	LR	0.15	0.18	0.29	0.21
2	BR	0.15	0.18	0.29	0.21
3	LGBR	0.45	0.63	0.55	0.54
5	DT	0.73	0.87	0.67	0.76

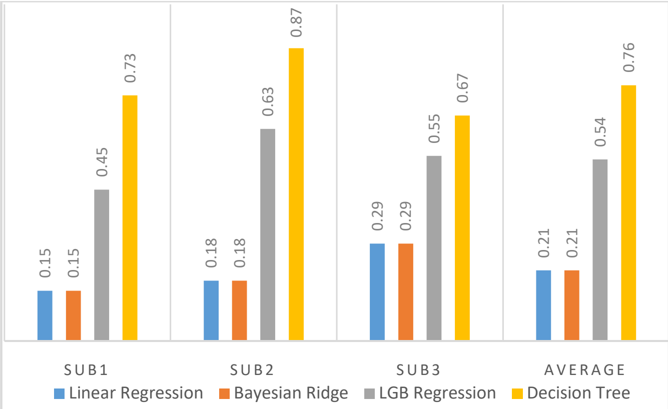

Figure 18. Performance Graph on (Noisy Dataset)

Table 10. Performance Comparison (Cleaned Dataset)

S. No.	Algo	Sub1	Sub2	Sub3	AVG.
1	LR	0.16	0.15	0.22	0.18
2	BR	0.17	0.15	0.23	0.18
3	LGB	0.49	0.65	0.57	0.57
4	DT	0.78	0.91	0.71	0.80

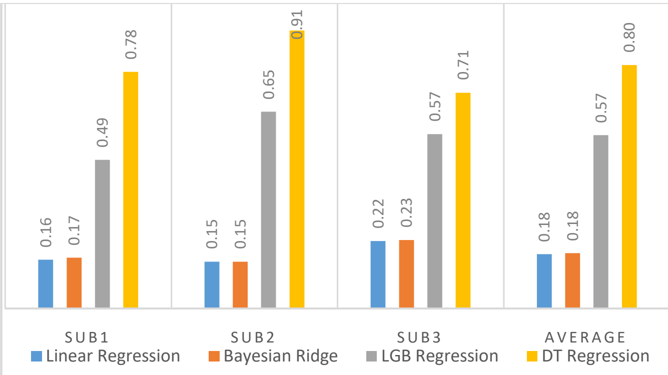

Figure 19. Performance Graph on (Cleaned Dataset)

LGB Regression showed a marked improvement with an average correlation value of 0.54 as shown in table 5 and figure 13. This indicates that LGB's ability to handle large datasets and complex patterns significantly enhanced performance compared to linear methods. Performance improvement is there as applied on the cleaned dataset and achieved correlation value 0.57 as shown in table 6 and figure 14. Decision Tree Regression achieved an average correlation value of 0.75 as shown in table 7 and figure 16. Decision trees are capable of capturing non-linear relationships in the data, which likely contributed to this higher performance, further performance improvement is there as applied on cleaned dataset and achieved 0.80 correlation value as shown in table 8 and figure 17.

6. Discussion and Challenges

Pre-processing effect is observed when LGB and DT are applied by 0.03 and 0.04 respectively as shown in table 9, 10 and figure 18 and 19. Negative impact is also observed when LR and BR algorithm applied on the cleaned dataset by reduction in the correlation value from 0.21 to 0.18 by both the algorithm. These linear methods, while simpler and faster but fall short in performance compared to these more complex algorithms. While ensemble methods like DT and LGB which can capture complex patterns are particularly effective for this type of high-dimensional EEG data. DT performed well on this dataset but have several drawbacks. One of the key drawbacks is their tendency to overfit, especially when the tree becomes deep. This means DT can become overly specific to the training data and fail to generalize well to unseen data. Additionally, DT is unstable—small variations in the data can result in a completely different tree structure, which can be problematic, particularly in noisy datasets. This also tend to be biased towards attributes with more levels, potentially favoring certain features over others, which can introduce bias into the model. DT also face difficulties when it comes to capturing interactions between multiple features, often requiring a large and deep tree to account for such dependencies. When dealing with large datasets, DT can become computationally expensive and slow to train. The greedy approach use for feature selection can also lead to suboptimal results, as it makes decisions based on local criteria rather than a global optimum. To mitigate these disadvantages, ensemble methods like Random Forests or Gradient Boosting Trees are often used to improve performance and stability.

At higher end other ensemble methods like RFR and KNN regression offer high accuracy in offline evaluations. their computational complexity, memory usage, and energy consumption make them unsuitable for real-time applications, especially in resource-constrained environments like BCIs. Simpler models often strike a better balance between performance and practicality, enabling them to meet the stringent requirements of real-time systems more effectively.

7. Conclusion

BCI competition 4 dataset iv is the standard dataset used and machine learning algorithm are applied on this dataset. Linear regression algorithm achieved only 0.21 correlation value. DT regression algorithm achieved 0.80 that is highest correlation value among all the algorithms, as ensemble approach based algorithms take a lot of time during the training phase along with test phase too. So, time is critical point to consider while using the higher accuracy algorithms.

References:

[1] Wolpaw, N., Birbaumer, D., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. Communications of the ACM, 44(7), 76-82.

[2] Lotte, F., Congedo, M., Mukovskiy, A., & Lécuyer, A. (2007). Classification of imagined movements with high-dimensional electroencephalographic data by random forests: a comparison with svm. Biomedical Signal Processing and Control, 2(1), 14-25.

[3] Wang, W., Xu, D., Degenaar, J. P., & Van Putten, M. J. (2016). A survey on user-oriented performance metrics in natural language processing for spoken language understanding. Artificial Intelligence, 238, 1-28.
[4] Daly, J. J., Wolpaw, D. J., & Willoughby, J. A. (2009). Brain-computer interface (BCI) control of a robotic arm for rehabilitation. Journal of Neural Engineering, 6(4), 046001.
[5] Lotte, F., & Michel, C. (2005). Overview of classification algorithms for EEG-based brain-computer interfaces (BCI). Journal of Neural Engineering, 4(4), R121.
[6] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning with applications in R (Vol. 112). Springer.
[7] Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and regression trees. Wadsworth international group.
[8] Mackay, D. J. (2003). Information theory, inference, and learning algorithms. Cambridge university press.
[9] Geurts, P., Brichet, F., & Dräxler, A. (2006). Extra-trees for modeling in regression and classification. Journal of Machine Learning Research, 6, 1071-1099.
[10] Gauttam Jangir; Rakesh Kumar. (2017). "Windowing-based threshold technique to play the simple breakout game at neutral attention level" International Journal of System of Systems Engineering, 2017 Vol.8 No.2, pp.147 - 173.
[11] Schalk, G., Kubanek, J., Miller, K.J., Anderson, N.R., Leuthardt, E.C., Ojemann, J.G., Limbrick, D., Moran, D.W., Gerhardt, L.A., and Wolpaw, J.R. Decoding Two-Dimensional Movement Trajectories Using Electrocortico-graphic Signals in Humans, J Neural Eng, 4: 264-275,2007.
[12] Bashashati, A., Fatourechi, M., Ward, R. K., & Birch, G. E. (2007). A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. Journal of Neural Engineering, 4(2), R32.
[13] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
[14] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).
[15] Cunningham, P., & Delany, S. J. (2007). k-Nearest neighbour classifiers. Multiple Classifier Systems, 34, 1-17.
[16] Grosse-Wentrup, M., & Buss, M. (2008). Multiclass common spatial patterns and information theoretic feature extraction. IEEE Transactions on Biomedical Engineering, 55(8), 1991-2000.
[17] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems (pp. 3146-3154).
[18] Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., & Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain–computer interfaces. Journal of Neural Engineering, 4(2), R1.
[19] Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research, 1(Jun), 211-244.
[20] Wang, Y., Gao, X., Hong, B., Jia, C., & Gao, S. (2012). Brain-computer interfaces based on visual evoked potentials. IEEE Engineering in Medicine and Biology Magazine, 27(5), 64-71.
[21] Ski Celi (2024). An Overview of Outlier Detection Methods. London Journal of Engineering Research, 37-79.
[22] https://machinelearninggeek.com/outlier-detection-using-isolation-forests/ [as last accessed on 04/01/2025]