# Brainwave classification

Arshad Mehtiyev
*Systems and Computer Engineering*
*Carleton University*
Ottawa, Canada
arshadmehtiyev@cmail.carleton.ca

Aziz Al-Najjar
*Systems and Computer Engineering*
*Carleton University*
Ottawa, Canada
azizalnajjar@cmail.carleton.ca

***Abstract*— A classification model to classify motor imagery (MI) with high accuracy can be implemented to help people with disabilities or minimize the gap between human-computer interactions in the game industry. In this work, various deep learning (DL) models, such as Multi-Layer Perceptron (MLP), Transformer, Convolutional Neural Network (CNN), and Hybrid, were explored and combined in an Ensemble Learning Soft Voting method to verify the possibility of exceeding the performance results of the reference paper which was used as a baseline for the project. Augmentation methods such as random horizontal flip and noise addition were applied to the given preprocessed dataset. Then the models were created and fine-tuned using Grid Search with 5-fold cross-validation on the training set. After finding the best hyperparameters for each model performance, the results were retrieved by evaluating the test set. In the end, the performance results were compared to the baseline paper results.**

***Keywords*— *EEG signals, binary classification, convolutional neural network, multilayer perceptron, hybrid, Transformer, deep learning***

## I. Introduction

Due to its broad and creative applications, transmitting and receiving information between the human brain and computers has been a hot research area. Its usefulness can vary from helping people with disabilities live a more leisurely life and perform daily activities, diagnosing different potential brain diseases, to using the brain as a gaming controller. However, the Electroencephalography (EEG) brain signals can be hard to understand by the computer due to their non-stationary and distinctive nature. Therefore, this project was focused on developing DL models that could translate the imagery of brain signals to perform specific body movements, such as right hand and right foot movements. Dataset was taken from publicly available BCI Competition III (Data set IVa)[1], for which correct test labels were provided for performance evaluation. The brainwave signals for these movements have already been collected and preprocessed using a non-invasive sensor from five different subjects – AA, AL, AV, AW, and AY. Dataset consists of a 6x201 matrix with 280 trials split into training and test sets separately for each subject, as shown in Table 1.

TABLE I. Training and Test Split

| Subjects | AA | AL | AV | AW | AY |
|---|---|---|---|---|---|
| Trials | 280 | 280 | 280 | 280 | 280 |
| Training set (Tr) | 168 | 224 | 84 | 56 | 28 |
| Testing set (Ts) | 112 | 56 | 196 | 224 | 252 |

Several research works were done to improve the efficiency and accuracy of the model implementation. One of the respected works is proposed in [2], where authors propose a new EEG signal decoding algorithm using a modified common spatial pattern and capsule network (CapsNet). The proposed method is done by extracting the spectral-temporal common spatial pattern features from the EEG signals while the time resolution of the signal is preserved. Then the extracted features were fed into the capsule network for automatic feature selection and classification. The authors suggest that the proposed new algorithm is better than regular CNN implementation requiring less training data. The results of this work are the baseline for the project, as the aim was to explore various DL models and combine them into an Ensemble Learning with Soft Voting to overperform the results of [2].

Another related work was [3], which was an inspiration for implementing the Hybrid model in this project. In [3], the authors implement a hybrid model consisting of CNN with an evolutionary algorithm (EA) to classify brainwave signals. They utilize a single Discrete Wavelet Transform (DWT) Coiflet filter instead of several pooling layers to extract features.

In [4], the authors explored, constructed, and implemented multiple Transformer-based models for MI EEG classification. They compared their results with previous state-of-art model performances, where proposed models produced better results.

In this project, four main - MLP, Transformer, CNN, and Hybrid methods were implemented, which then will be fed into a Soft Voting Ensemble Learning model, which combines classifiers to outperform each classifier's result. Data Augmentation was implemented to increase the training set and avoid overfitting. Hyperparameters were searched for each main model by implementing the GridSearchCV function from Scikit-Learn. Within GridSeachCV 5-fold cross-validation was implemented. A summary of the overall process is shown in Fig. 1. Section II describes the details of each method's overall process and implementation.

## II. Methodology

### A. Pre-processing of Dataset

The provided dataset was already cleaned and processed. But further processing was required to make it compatible with implemented DL models. First data was transposed from the shape of (6,201,280) to (280,201,6). Then stratified split was implemented to split the dataset for training and testing as per the given ratio in Table 1. After the split, training data for each subject were combined. In addition, data augmentation, such as horizontal flip and Gaussian noise addition, was implemented with randomly selected samples. This allowed us to increase the training set and avoid overfitting. One-Hot Encoding was applied to the labels to improve the effectiveness of the DL model. As a final step, the StandardScaler function from the Scikit-learn library is devoted to standardizing augmentation features (1), where *x* is a given sample, *u* is the mean of the training sample, and s is the standard deviation.

$$z = (x - u) / s \qquad (1)$$

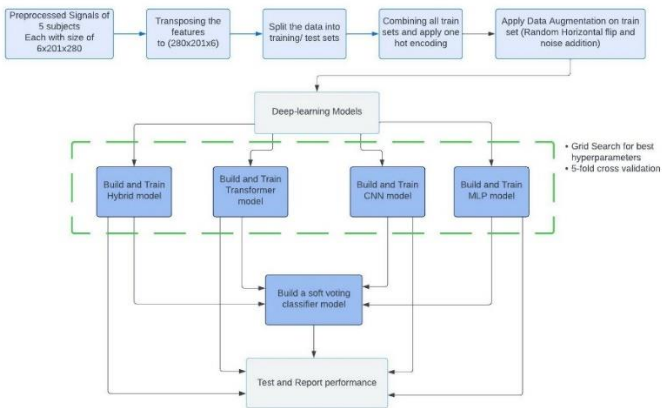


Fig. 1. Overall Project flowchart

## B. MLP model

The first model to be trained for the brainwave classification problem is the fully connected feedforward MLP model, which is considered one of the most common Artificial Neural Networks (ANN) types. The layers of our MLP model are divided into three levels, the input layer, hidden layers, and the output layer. The input layer consists of flattened 201*6 features, i.e., 1206 nodes. These features are fed into a network of 20 hidden layers. Each hidden layer contains 100 neuron units, where each neuron contains summation and activation functions [5]. The summation function sums up the result of the multiplication of weights with inputs and a bias, as shown in (2). The summation result will be used as an input to the activation function. With Grid-search and trial and error, Rectified Linear Unit (ReLU) has been chosen to perform the linear activation for hidden layers' neurons. The last layer of the MLP is the binary output layer with sigmoid activation. The MLP network can be seen in Fig.2

$$S_j = \sum_{i=1}^{n} w_{ij} I_i + \beta_j \qquad (2)$$

## C. CNN Model

Compared to MLP, CNNs can accept input in 2D shape, which primarily allows processing and learning of images. In CNN architecture, fully connected layers are replaced with convolution layers, in which every next Conv layer extracts the learnable features from the previous layer [6]. The input data has a 2D shape in our case, which allows us to implement this architecture. Although our dataset is not image type as it doesn't have color depth, to overcome this problem, we just included an additional single depth dimension to the dataset before feeding it to the created model.

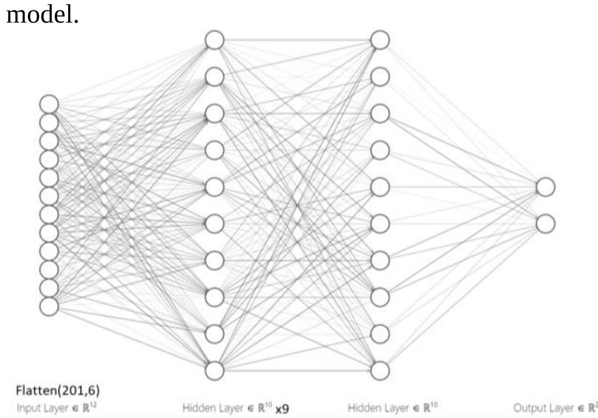


Fig. 2. Multi-Layer Perceptron Model Architecture

The CNN model architecture has 3xConv2D, Maxpooling2D, Flatten, 2xFully connected, and a binary output layer stacked in sequence. The first Conv2D layer has 32 filters with a 3x3 window size. The second Conv2D layer has 64 filters with a 1x1 window size. The third Conv2D layer has 128 filters with a 3x3 window size. The max-pooling layer has a 1x1 pool size. The first Dense layer has 128 nodes, and the second one has 64 nodes. Dropout is applied after Max pooling and Dense layers with rates of 0.25, 0.5, and 0.5, respectively. All the layers have ReLu activation in the output, except for the binary output layer, which has SoftMax activation.

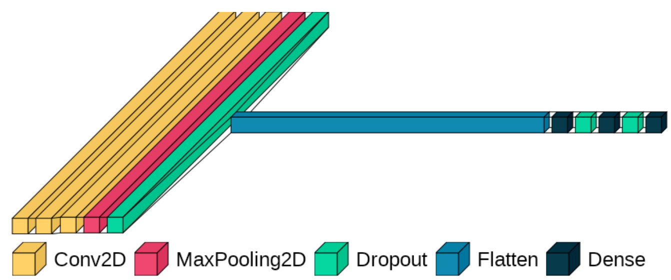


Fig. 3. CNN model architecture

## D. Hybrid (CNN+XGBoost)

The hybrid model combines the previous CNN model with the XGBoost algorithm. XGBoost is a machine learning(ML) algorithm that is based on scalability(computationally), and distributed gradient boosted trees [7]. This algorithm gave promising results in several ML challenges, which is why it was decided to incorporate it with the CNN model. To implement the Hybrid model, we first fully train the CNN model, then pass the training set through the trained CNN model and pick the output from the first Dense (aka middle) layer. Then we pass this output set as an input to XGBClassifier by providing the same label set in the output. By doing this, we are training the XGBoost classifier. When training is done, we pass the test set through CNN, then from the middle layer transfer results as an input to the XGBoost model to make the predictions. The architecture of the Hybrid model is shown in Fig. 4.

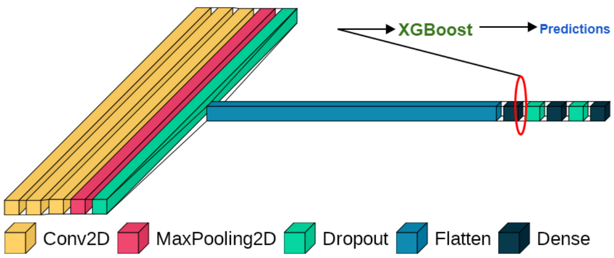


Fig. 4. Hybrid model architecture

## E. Vision Transformer Model (ViT)

The first Self-attention-based Transformer was proposed in [8]. This Transformer dominates the natural language processing (NLP) and machine translation applications. Moreover, in [9] authors proposed a propagation of the NLP Transformer to a Vision Transformer (ViT). The approach proposed in [9] was used for this part of the project, which is described in Fig. 5. First, brainwave signals were converted into three-channel images, as shown in Fig. 6a). These images were then reshaped into a sequence of flattened 2D patches. The total number of patches is *N* = image resolution / (single patch size)^2. The number of patches is the input sequence length for the Transformer, as shown in Fig. 6b). Next, the patches were flattened and mapped with a trainable linear projection into a latent vector of size *D*, which exists through all the transformer layers. This projection step is referred to as patch embeddings. Also, learnable embedding was appended to the sequence of patches to help represent the signal after the Transformer encoder. Before sending the sequence of embedding the encoder, Position embeddings were added to retain the positional information. An MLP model with two layers and GELU activation is attached to the output of the Transformer encoder. The transformer encoder is similar to the one proposed in [9]. It consists of alternating layers of multiheaded self-attention and MLP blocks, layer normalization before every block, and residual connections after each block.

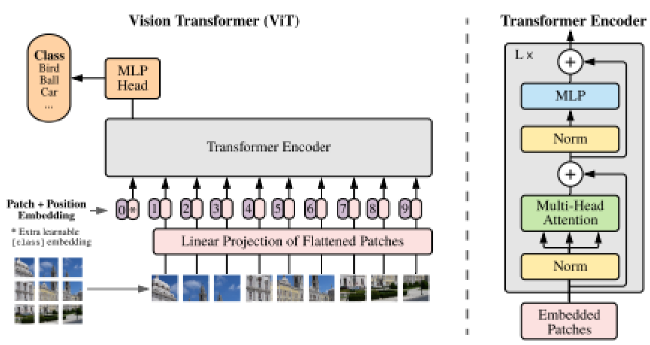


Fig. 5. Vision Transformer Model Overview

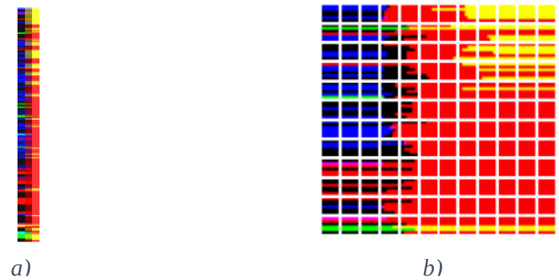


Fig. 6. a) (201,2,3) signal as an image
b) Resizing the image to 72x72 and dividing the signal image into 144 patches of size 6x6

## F. Ensemble Learning with Soft Voting

Ensemble Learning with Soft Voting was implemented to combine the probabilistic predictions of each DL classifier to make a final prediction. This custom soft voting algorithm was written to sum all the predicted class probabilities. Then the class with the highest score was assigned either 0 or 1. This method usually outperforms each classifier.

# III. Results

To report the performance of the models, it was decided to report the following metrics: Training performance, accuracy, F1 score, precision, recall, AUC value, confusion matrix, and ROC curve for each model for each subject. Performance results are provided below.

## A. MLP Model Results

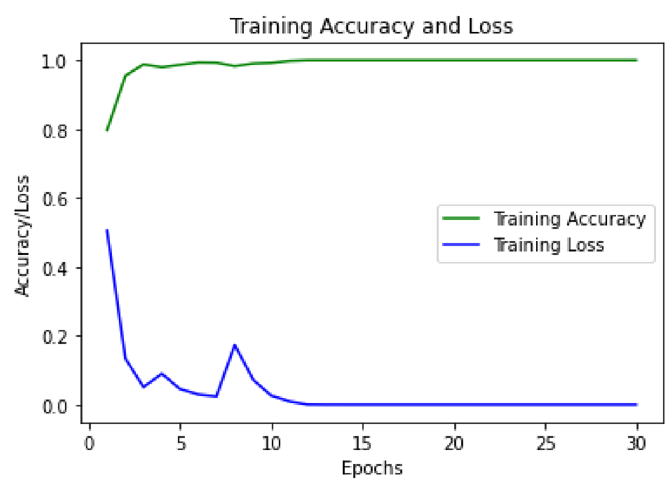


Fig. 7. Training Performance of MLP

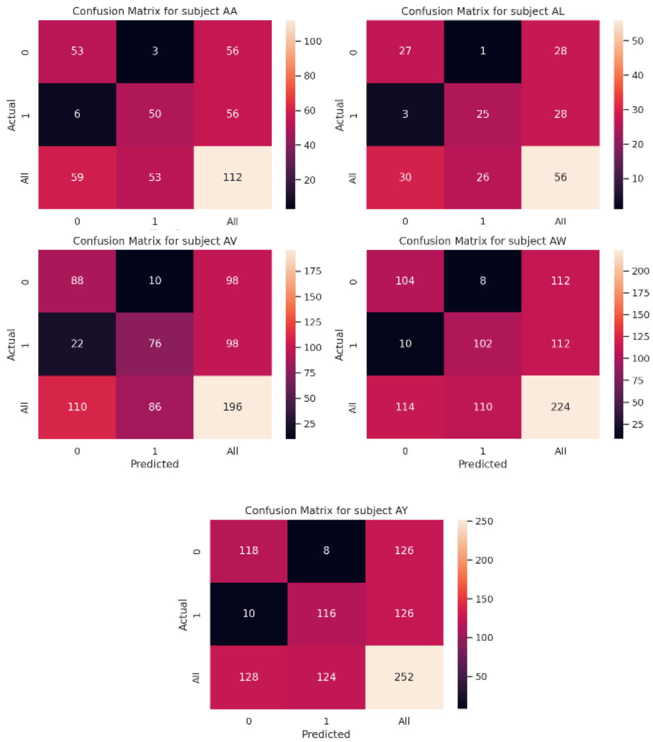


Fig. 8. Confusion matrices of MLP model on test set for each subject

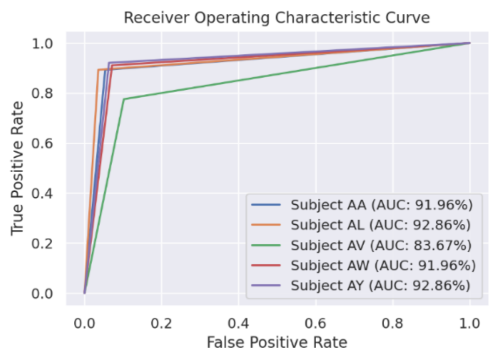


Fig. 9. ROC curve of MLP model on test set for each subject

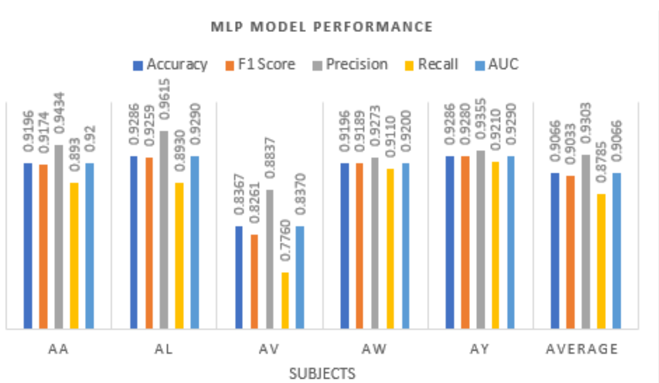


Fig. 10. Performance summary of MLP model on test set for each subject

## *B. CNN model Results*

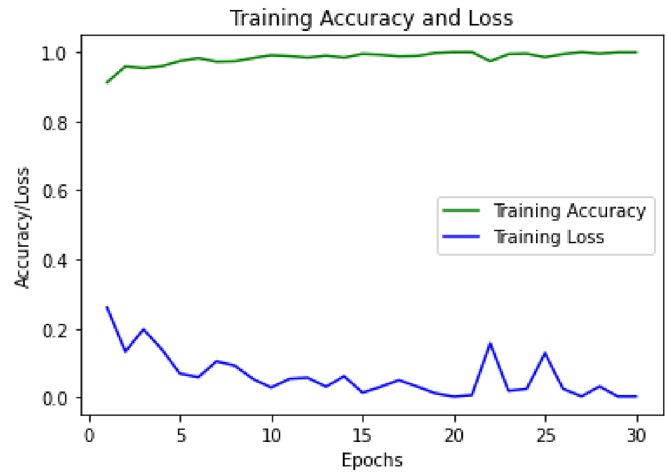


Fig. 11. Training Performance of CNN model

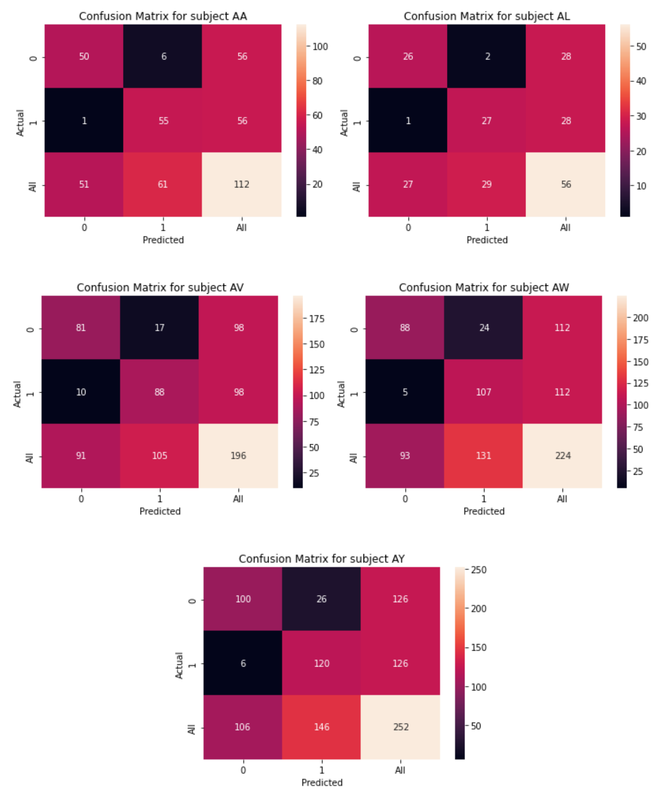


Fig. 12. Confusion matrices of CNN model on test set for each subject

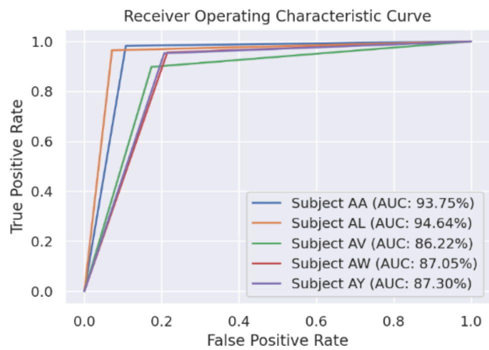


Fig. 13. ROC curve of CNN model on test set for each subject

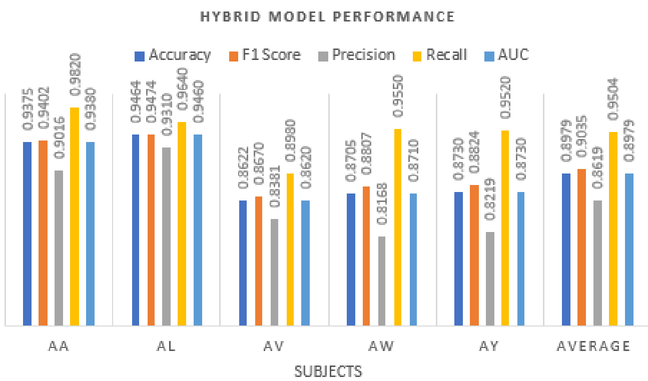


Fig. 14. Performance summary of CNN model on test set

## C. Hybrid Model Results

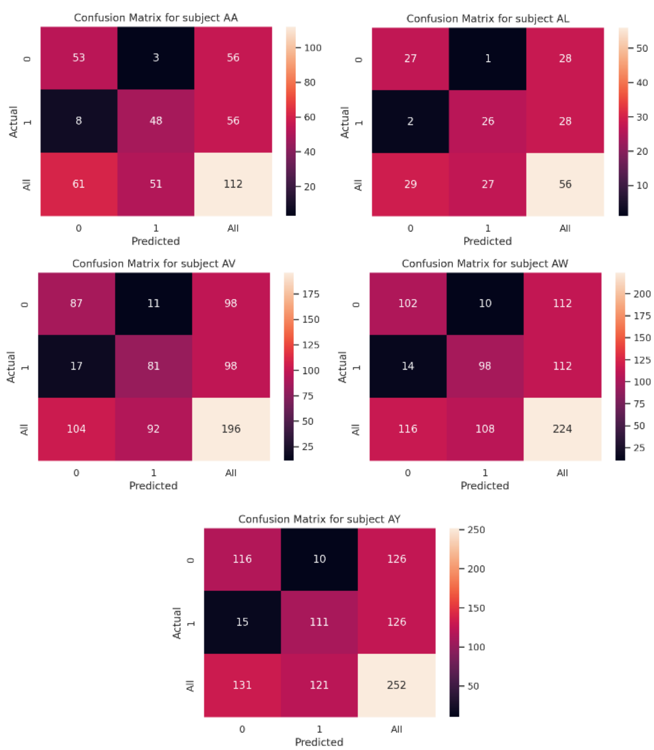


Fig. 15. Confusion matrices of Hybrid model on test set for each subject

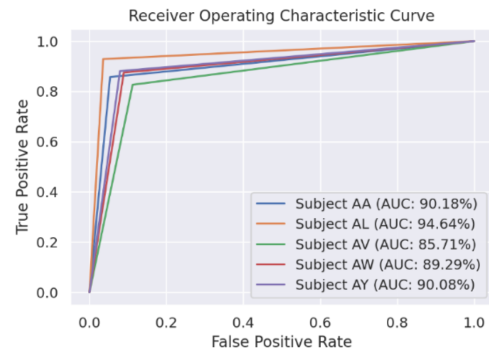


Fig. 16. ROC curve of Hybrid model on test set for each subject

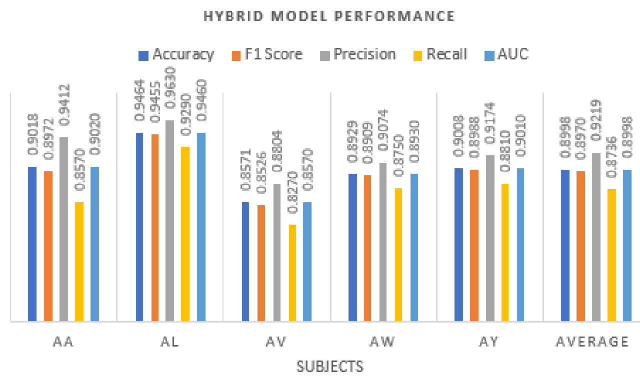


Fig. 17. Performance summary of Hybrid model on test set

## D. ViT Results

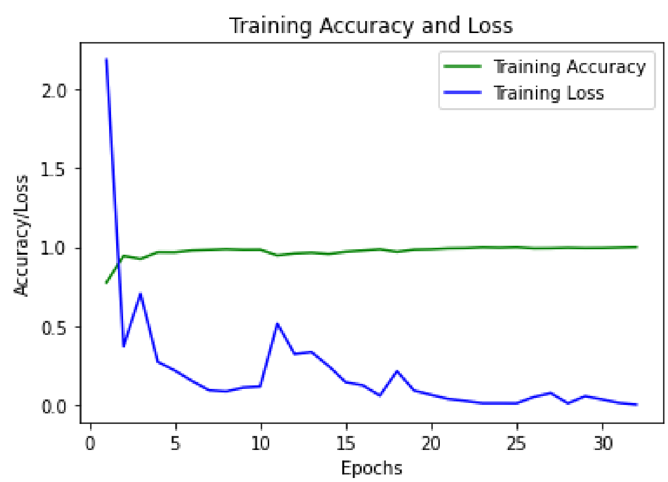


Fig. 18. Training Performance of ViT (Transformer) model

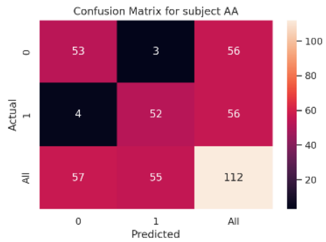


Fig. 19.

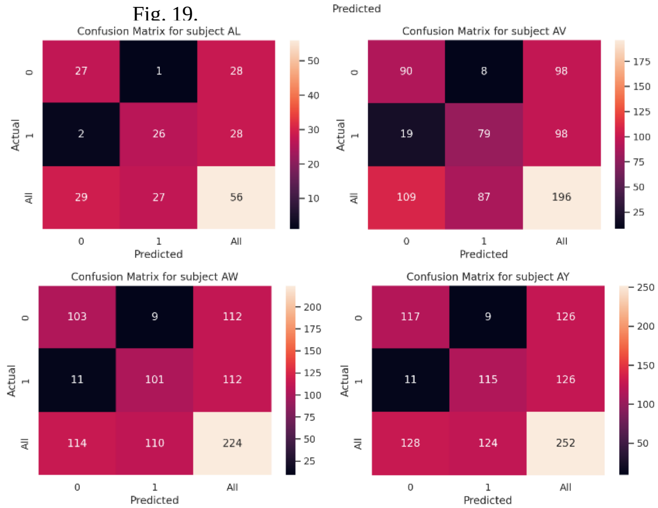


Fig. 20. Confusion matrices of ViT model on test set for each subject

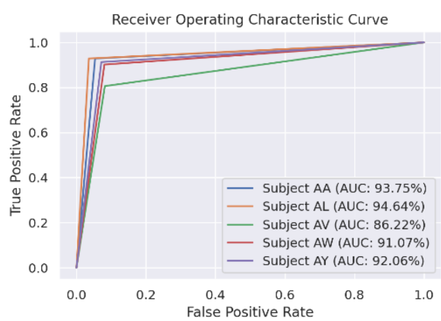


Fig. 21. ROC curve of ViT model on test set for each subject

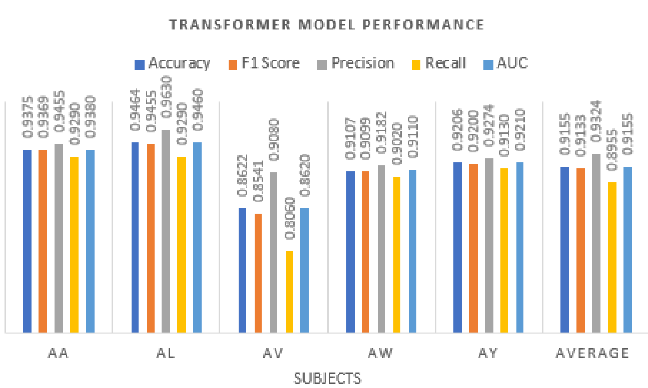


Fig. 22. Performance summary of Transformer model on test set

## *E. Ensemble model with Soft-voting*

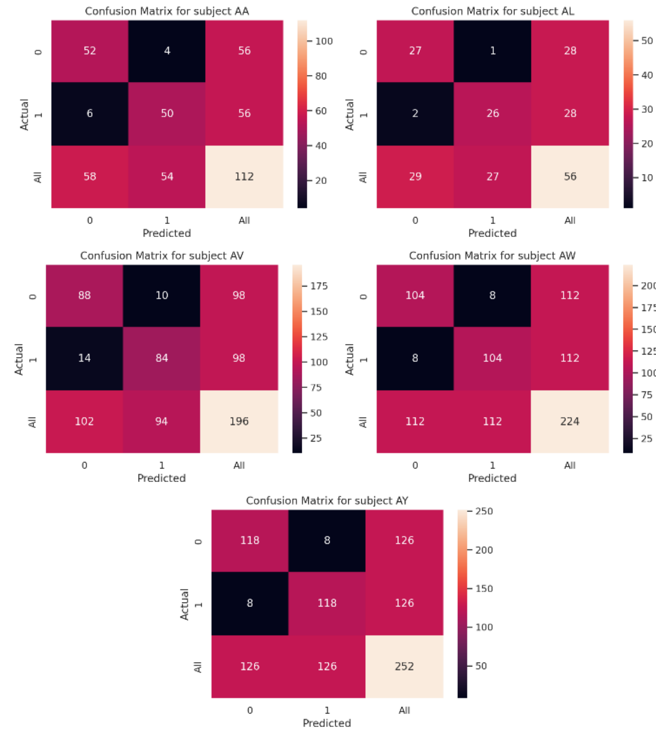


Fig. 23. Confusion matrices of enemble model on test set for each subject

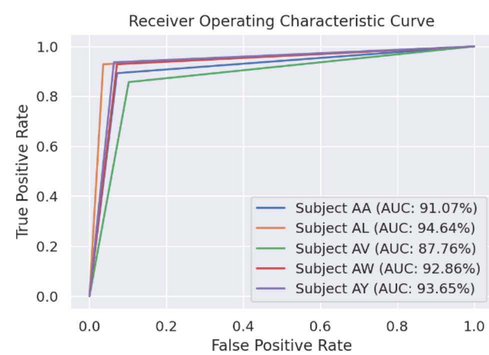


Fig. 24. ROC curve of Ensemble model on test set for each subject

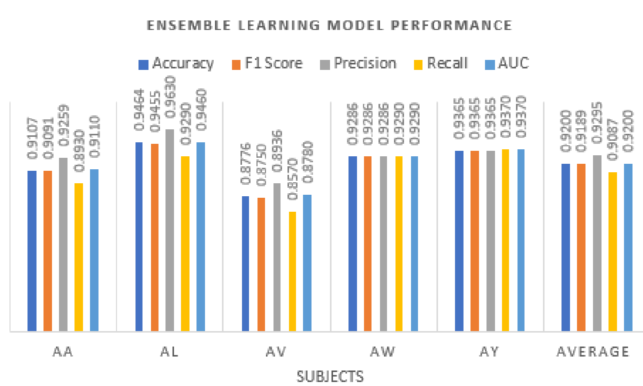


Fig. 25. Performance summary of Ensemble model on test set

## IV. RESULTS DISCUSSION

The results above show that with the correct parameters, the ViT transformer can outperform all other models, with CNN coming with the lowest performance, and with the help of XGBoost, the CNN performance boosted a little. Moreover, combining all models in a soft model ensemble allowed us to increase the performance and comparable to base paper [2]. Table III below shows that our ensemble model outperforms the EEGCAPS for AA, AV, and AY subjects from Table II from [2]. We believe our results could be improved even further with more research work to develop a better architecture for each model and more hyperparameter trials.

TABLE II. PERFORMANCE RESULT OF PROPOSED EEGCAPS METHOD FROM [2] COMPARED TO PREVIOUS RESULTS

CLASSIFICATION ACCURACY OBTAINED USING THE PROPOSED METHOD AS WELL AS THOSE YIELDED BY THE OTHER METHODS.

| | Dataset 4a, BCI III | | | | |
|---|---|---|---|---|---|
| | AA | AL | AV | AW | AY |
| CSP [5] | 66.07 | 96.43 | 43.47 | 71.88 | 49.60 |
| R-CSP [2] | 77.7 | 96.4 | 58.7 | 92.0 | 68.3 |
| MSRCSP [21] | 69.64 | 96.43 | 59.18 | 71.88 | 52.78 |
| SBCSP [9] | 83.03 | 98.21 | 52.04 | 89.05 | 58.33 |
| FBCSP [10] | 83.93 | 96.43 | 63.26 | 72.32 | 54.37 |
| SSRCSP [6] | 70.54 | 96.43 | 53.57 | 71.88 | 75.39 |
| LRDS [8] | 80.4 | 94.6 | 50.0 | 90.6 | 83.3 |
| FBRCSP [11] | 84.82 | 96.43 | 63.78 | 74.55 | 73.81 |
| **EEGCAPS** | **85.50** | **97.52** | **62.15** | **94.70** | **83.57** |

TABLE III. RESULTS OF ENSEMBLE LEARNING

| | **AA** | **AL** | **AV** | **AW** | **AY** |
|---|---|---|---|---|---|
| **Proposed Ensemble method** | 91.07 | 94.64 | 87.76 | 92.86 | 93.65 |

## V. CONCLUSION

In this report, we started by talking about the methodology followed for brainwave classification. We used MLP, CNN, Hybrid, and ViT (Transformer). models The result shows that each model performed slightly differently from the other. Also, the results for each subject are somewhat different for each model. Considering the difference in train/test split ratio, this was expected. If we compare the results of the CNN and Hybrid models, it is visible that the Hybrid model slightly overperformed the CNN model. This shows that even minor stacking methods can improve the model's performance. Also, we can confirm the ViT model could be applied to signals with apparent features and outperform the CNN model.

## VI. GROUP MEMBER CONTRIBUTION

Both - Arshad and Aziz efficiently distributed the workload among each other. Arshad primarily focused on implementing CNN and Hybrid models, whereas Aziz focused on MLP and Transformer models. Both worked on code implementation in Python programming language, visualization in Python and Excel, presentation, and reporting in coordination.

## REFERENCES


[1] BCI competition III, Data set IVa *‹motor imagery, small training sets›*, https://bbci.de/competition/iii/desc_IVa.html

[2] H. Sadreazami and G. D. Mitsis, Motor task learning in brain computer interfaces using time-dependent regularized common spatial patterns and residual networks, IEEE 18th International NEWCAS Conference, pp. 190- 193, 2020

[3] Rostam, Z. R. K., & Mahmood, S. A. (2019, December 8). Classification of Brainwave Signals based on hybrid deep learning and an evolutionary algorithm. arXiv.org. Retrieved January 27, 2022, from https://arxiv.org/abs/1912.07361

[4] Sun, Xie, J., & Zhou, H. (2021). EEG Classification with Transformer-Based Models. 2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech), 92–93.

https://doi.org/10.1109/LifeTech52111.2021.9391844

[5] Faris, H., Aljarah, I. & Mirjalili, S. Training feedforward neural networks using multi-verse optimizer for binary classification problems. Appl Intell 45, 322–332 (2016). https://doi.org/10.1007/s10489-016-0767-1.

[6] Notes of the Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition.(2022)

https://cs231n.github.io/convolutional-networks/

[7] T. Chen, C. Guestrin, "XGBoost: A Scalable Tree Boosting System", 2016

[8] A. Vaswani, et al.. Attention is all you need. In NIPS, 2017. https://arxiv.org/pdf/1706.03762.pdf

[9] A. Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In ICLR 2021. 2010.11929.pdf (arxiv.org)