Received February 21, 2022, accepted March 29, 2022, date of publication April 6, 2022, date of current version May 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3165197

State-of-the-Art Versus Deep Learning: A Comparative Study of Motor Imagery Decoding Techniques

OLAWUNMI GEORGE[1], SARTHAK DABAS[1], ABDUR SIKDER[1], (Member, IEEE), ROGER SMITH[2], PRAVEEN MADIRAJU[1], NASIM YAHYASOLTANI[1], AND SHEIKH IQBAL AHAMED[1], (Senior Member, IEEE)

[1]Computer Science Department, Marquette University, Milwaukee, WI 53233, USA
[2]Department of Rehabilitation Sciences and Technology, University of Wisconsin–Milwaukee, Occupational Therapy, Sciences and Technology Program, Milwaukee, WI 53211, USA

Corresponding authors: Olawunmi George (olawunmi.george@marquette.edu) and Sheikh Iqbal Ahamed (sheikh.ahamed@marquette.edu)

This work was supported in part by a number of grants of Ubicomp Lab, Marquette University, USA.

ABSTRACT State-of-the-art techniques (SOTA) for motor imagery decoding have largely involved the use of common spatial patterns (CSP) and power spectral density (PSD), for feature extraction. Other frequency transforms, such as wavelets and empirical mode decomposition (EMD) have also been used but the aforementioned two have been the most popular. For classification, linear discriminant analysis (LDA) and support vector machines (SVM) have been mostly used. It is, however, worth investigating other approaches, such as deep learning, which offer a potential for improvement, but are not yet mainstream. Deep learning techniques based on neural networks (NNs) have been underexplored in motor imagery processing. Considering their success in other fields, which speaks to their potential for obtaining improved results over the SOTA, they should be explored for motor imagery decoding. This study takes a comparative approach in the use of deep learning as compared with the SOTA. From our findings, we infer that neural networks are suitable for motor imagery decoding and might be preferable over the SOTA. The use of specific feature extraction is also not as necessary as seen with SOTA approaches, though it might offer some gains in performance. Our results show a statistically significant improvement in decoding accuracies, up to 20% increase, with the use of NNs as compared with the SOTA. Also, we conclude that the use of crops for data augmentation might yield better results with shallow architectures as against deeper ones and that there might be other factors affecting the effectiveness of crops, needing further investigation.

INDEX TERMS EEG, BCI, motor imagery, deep learning, machine learning.

I. INTRODUCTION

Motor imagery is a widely used paradigm in brain computer interface (BCI) experiments for communication and control [1]. Examples of such uses include the control of devices – assistive and non-assistive, vehicles and games [2], [3]. Also, very common is its use in neuro-rehabilitative studies. The central idea of motor imagery (MI) rests on the fact that the brain exhibits specific neuronal characteristics during imagined movements, similar to that of actual motor action [4]. It has been established over the years, that the event related desynchronization (ERD) and the event related synchronization (ERS) noticed over the sensorimotor cortex play a vital role in motor imagery decoding. The most relevant brain frequency bands used in electroencephalographic (EEG) motor imagery decoding have been the alpha (8–14 Hz) and beta (14–30 Hz) bands. These two have been stated as the most significant for distinguishing imagined actions [1], [4].

Traditional techniques used in motor imagery processing typically involve pre-processing, feature extraction and then classification. Pre-processing is geared towards cleaning the data, to rid it of noise and improve the signal-to-noise ratio. Such pre-processing procedures are usually necessary when using EEG, as EEG signals have been known for their susceptibility to varying types of noises [5]. The pre-processing procedures get rid of artifacts, which could be environmental, instrumental and/or biological. Other device types, apart from

The associate editor coordinating the review of this manuscript and approving it for publication was Essam A. Rashed.

EEG may not be as susceptible to all these artifact types. Feature extraction is performed to extract relevant features, which provide distinguishing characteristics between imagined tasks. Typical feature extraction techniques include the use of the CSP algorithm, well known for providing plausibly distinguishing features. So, also have other time-based, frequency-based and time-frequency transforms been applied for feature extraction, such as wavelets, auto-regressive modelling and empirical mode decomposition (EMD). For classification, support vector machines (SVM) and linear discriminant analysis (LDA) have been mostly used, as compared with other algorithms [6]–[9].

A consideration of deep learning techniques and their use in computer vision and natural language processing shows that deep learning techniques could be utilized in many fields, with the potential for improvements. More specifically, the use of convolutional neural networks (CNNs) and sequence models – recurrent neural networks (RNNs), long short-term memory (LSTM) networks and gated recurrent unit (GRU) networks - has shown significant improvements in these fields [10]–[12]. Deep learning techniques have some advantages over traditional approaches. These include little to no reliance on a priori knowledge and the elimination of specific feature engineering. Neural networks can learn features from the data and do not require manually crafted features for optimal decoding performances.

Considering their degree of success and use, this study takes a comparative approach of using these techniques against the SOTA. First, we take an exploratory approach in the use of these networks, determining the best-performing architectures and then finding optimal parameters for best architectures. In addition, we compare these results with SOTA techniques – CSP and spectrograms for feature extraction; SVM and LDA for classification - inferring based on the results that deep learning techniques can equally be applied in motor imagery decoding with significantly better results. One reason for the wide use of SVM and LDA has been the small amount of data acquired during motor imagery studies. This is typically a few hundred samples, per session, due to the nature of the experiments and the fatiguing effect they could have on subjects. However, the application of data augmentation techniques could enhance decoding when using deep learning techniques. The remainder of this manuscript is structured as follows: related works, methods, results and discussion and conclusion.

II. RELATED WORKS

Previous works exploring motor imagery for communication and control, have mostly used CSP and frequency transforms for feature extraction and SVM and LDA for classification. In this section, we present some of the works using these techniques. We also present other more recent works exploring neural networks or deep learning-based techniques for classification.

Steyrl *et al.* [13] made a comparative use of CSP features with regularized LDA and random forests. They demonstrated the online use of a random forests classifier with the discrete Fourier transform (DFT) of the motor imagery signals, in an earlier work [14]. Their comparative analyses, which was done afterwards, showed that the CSP features contained highly discriminatory information. The results showed good performances for both classifiers, with peak accuracies up to 100%, though, random forest was reported to have made better use of the CSP features. Overall, the authors reported significantly better results with the use of CSP, as compared with the DFT. Another work by Wu *et al.* used a combination of CSP and LDA. The authors used cross validation to determine the optimal time window and number of CSP features for each subject and reported a maximum classification accuracy of 80%, using only 9 channels [15].

In Jin *et al.*'s work [16], the authors made use of a correlation-based approach for channel selection. Their correlation-based channel selection (CCS) method was used to select channels that contained more relevant information. After channel selection through the CCS method, regularized CSP was applied for feature extraction and finally, an SVM classifier was used for classification. The authors validated their approach on the BCI competition datasets. On the BCI Competition IV dataset, their method achieved up to 94.5% accuracy on a subject and a mean accuracy of 81.6 ± 11.5% across all subjects. On the BCI Competition III dataset IVa, up to 96.8% accuracy was reported on a subject and a mean accuracy of 87.4 ± 10.6% across all subjects and on the IIIa dataset, up to 98.9% single-subject accuracy was achieved and 91.9 ± 10.3% mean accuracy across all subjects [16].

Yet another work by Feng *et al.* [17] made use of CSP and SVM for feature extraction and classification respectively, with 10-fold cross-validation. Their approach considered time in the feature extraction, with an argument that there exists a time latency in the performance of MI for subjects and that this latency could affect the performance of the BCI, if not accounted for. To that end, they proposed a correlation-based time window selection (CTWS) method, to determine optimal time windows for both training and testing samples, adjusting the windows by using correlation analysis. Afterwards, feature extraction and classification were performed. They validated their approach on two datasets – the BCI IV competition dataset I [18] and a primary data source of MI EEGs from stroke patients. They reported having the average classification accuracy improved by 16.72% on the dataset of healthy subjects (BCI Competition IV Dataset 1), and 5.24% on the dataset of stroke patients. These works described, so far, made use of SOTA processing techniques.

Some other works have explored neural networks or deep learning techniques, in MI classification. Though, there are some works that have used this approach, it still appears that deep learning techniques have been underexplored in the space. A look at such works, shows that more have used convolutional neural networks (CNNs). Others have also used RNNs, LSTMs and GRUs, as these sequence models have been known to model time series relationships well.

Sakhavi *et al.* [19], demonstrated the use of CNNs for MI classification. The authors made use of a modified version of the filter-bank CSP (FBCSP) and a CNN architecture tailored for the extracted features. The temporal representation used was the channel-relative instantaneous energy of the signal envelope, extracted using the Hilbert transform. Building upon one of their previous works [20], the authors made use of this temporal representation, as different from their earlier work. Their approach entailed the following:

1) First, FBCSP was performed on the signal.
2) Next, the envelope of each signal was extracted using Hilbert transform.
3) Afterwards, three possible representations were generated – the raw or smoothed version of the EEG envelope (R1); the power of the envelope (R2); and the ratio of the envelope to the total energy of each of the channels in each trial (R3).

The three generated representations (R1, R2 and R3) were used in determining the most effective representation. In constructing the CNN, the authors made use of three approaches: convolutions only across time with a common kernel shape for all channels, convolutions only across channels; and convolutions across both time and channels. Varying kernel and stride sizes for the first layers were used, to determine the optimal sizes. They excluded the use of pooling layers and had just two convolution layers, with a fully connected layer just before the output layer. They included batch normalization and dropout layers before and after the rectified linear unit (ReLU) activation layers respectively and used the Adam optimizer for learning. Their results showed that the first signal representation - R1 - was better, achieving better results compared to others. They also compared the results of their 2-dimensional CNN architecture, used for selecting the best representation, with a multi-layer perceptron (MLP) and support vector machine (SVM) and reported having better results with the CNN. Their approach was validated on the public BCI IV 2a competition dataset.

Other works using neural networks include the work by Dose *et al.* [21], where the authors used a CNN to classify MI signals in the Physionet movement and motor imagery database (MMIDB) [22], [23]. Their CNN architecture was based on the shallow ConvNet proposed by Schirrmeister *et al.* [24] and consisted of two convolutional layers with 40 kernels per layer and a fully connected layer. They utilized AveragePooling within the network, a ReLU activation and Adam optimizer for the learning. Their results showed better performances as compared with other works using the Physionet MMIDB. Wang *et al.* [25] also made use of a Long Short-Term Memory (LSTM) network, with an architecture inspired by classical CSP. The authors used a one-dimensional aggregate approximation (1dAX) for extracting a signal representation for the LSTM network. They validated their results on the BCI competition IV dataset 2a and as compared with other architectures, their results were consistently better for all but one subject.

Finally, Zhang *et al.*'s work [26] combined deep learning and data augmentation for EEG MI classification. They decomposed the signals into intrinsic mode functions (IMF) using EMD, combined the IMFs to create new signals and used wavelet transforms and a CNN for feature extraction and classification, respectively. The authors recorded 90.1% accuracy with the augmented approach, as compared with 89.3%, achieved by the winner of the BCI competition. They also used the pretrained ResNet-18 image classifier, but the results worsened, which is attributable to the large depth of the network. These show that deep learning approaches can perform as good as SOTA techniques and even surpass them. They should, therefore, be explored and adapted for use in MI BCIs.

III. METHODS

This section details our approach in our comparative study. All techniques and methods were applied on a public dataset. Details of the dataset and the methods are given in following subsections.

A. DATASET

The public dataset used in this study was provided by Kaya *et al.* [27]. The full dataset contains 60 hours of EEG recordings from 13 participants, collected over 75 recording sessions and yielding over 60 000 examples of motor imageries in 4 interaction paradigms. The dataset is currently one of the largest EEG BCI datasets made public. The data was collected at the University of Mersin, Turkey. 13 healthy participants (8 males and 5 females), who were students, within the ages of 20 and 35 were recruited for the study. Participants were labelled A-M, for anonymity. Necessary checks were done to ensure the health status of the participants. The EEG-1200 system, a standard medical EEG station used in many hospitals, was used for data collection. The sampling rate of the device was 200 Hz and 19 channels were used, being placed according to the 10-20 standard. Participants kept their gaze at the centre of a graphical user interface (GUI) window, containing icons representing the imageries to be performed. The original dataset contains 2-class and 6-class motor imageries; however, for this study, we made use of the 6-class dataset. The 6 motor imageries were the left hand, right hand, left leg, right leg, tongue and a passive mode. Each imagined action lasted a second. For all subjects except one (Subject D), there were one or more 6-class motor imagery sessions, resulting in the use of 12 subjects' data for this study.

B. PRE-PROCESSING

First, the data were pre-processed, to remove different types of artifacts, mostly oculographic. The data had already been notched at 50Hz, to remove line noise, with an inbuilt filter in the device. The following steps were taken:

1) Bandpass filtering for 1-40Hz.
2) Oculographic artifact correction, using independent component analysis (ICA). First, a bipolar channel was created using the two available pre-frontal

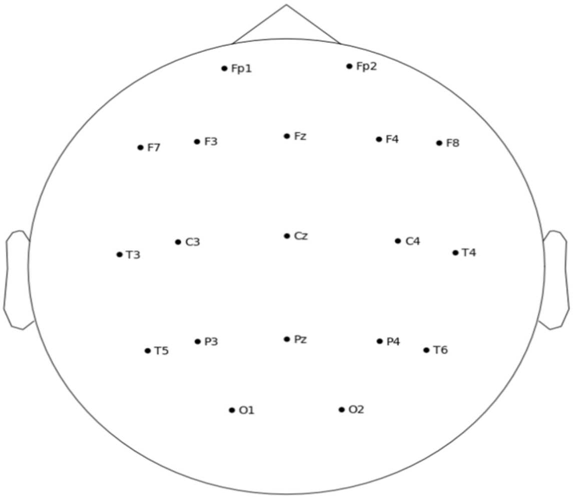

FIGURE 1. 10-20 Montage used for the study.

electrodes – Fp1 and Fp2. The montage used is depicted in Figure 1. The constructed channel was then used as a simulated electrooculographic (EOG) channel for oculographic artifact removal, using the MNE python package [28]. With ICA, eye artifact components are detected using the bipolar channel and are marked for rejection. The signal is then reconstructed using other non-artifact components. The use of the bipolar channel is based on the fact that the prefrontal electrodes typically capture eye artifacts, rather than activities related to the imagined action.

3) Re-referencing to average. The data was re-referenced to improve the signal-to-noise ratio.
4) Baseline correction on epochs, with 200ms pre-cue data.
5) Artifact rejection, using the auto-reject package [29]. Some subjects had more trials rejected, than others, due to more noise being present.

Figure 2 shows the stages in the decoding processing with the pre-processing, feature extraction and classification techniques used for the SOTA and deep learning approaches.

C. STATE-OF-THE-ART FEATURE EXTRACTION

For the SOTA techniques, we chose to use CSP in the time domain and power spectrum, computed as spectrograms, as features for the SOTA classifiers. We also used the Welch's periodogram, however, this yielded results which were not as comparable as those achieved with the spectrograms. We, therefore, used the spectrograms as the desired frequency representation.

In computing the features, first, optimal parameters were chosen to get the best feature representations possible. This involved using each option from the range of options available for each parameter to generate the CSP features, in turn. Afterwards, the features were used in training the untuned LDA classifier, with 10-fold cross-validation (CV). The number of components giving the highest CV score was then chosen as the optimal parameter. For CSP, the two most important parameters to be optimized were the number of CSP components and the covariance estimation technique, which could be done with or without shrinkage. The optimal number of components was found to be 19 (all), as chosen from a range of 4 to 19. Covariance estimation options included principal component analysis (PCA); Ledoit-Wolf shrinkage [30]; diagonal fixed regularization, and an empirical mode, using the estimated noise covariance matrix, without shrinkage. The optimal technique was the empirical mode.

For PSD, on the other hand, the parameters tuned were the number of samples per segment and the amount of overlap between segments. The optimal number of samples per segment from a range of values between 50 (250ms) and 200 (1000ms), was found to be 100, equivalent to 500ms of imagery period. The optimal overlap was found to be between 75% and 80%; however, we chose 80%, as that generally seemed to give better results. With the optimal parameters, CSP components and spectrograms were generated and finally fed into the SOTA classifiers.

D. STATE-OF-THE-ART CLASSIFIERS

For classification using SVM and LDA, we performed parameter searches for both classifiers to determine optimal parameters giving the best results. To this end, we ran a 10-fold cross-validated grid search on both models, varying the key parameters. The key parameters for SVM, were the C-value, gamma and polynomial degree. For LDA, the key parameters were the solver and shrinkage value. For the SVM classifier, the C-value ranged between 0.1 and 100, with steps in multiples of 10. The range of values for gamma was from 1 to 1e-8, with steps in multiples of 10e-2 and a scale value, which is the inverse of the product of the number of features and the variance in the data. The polynomial degree ranged from 1 to 3. The optimal C-value and polynomial degree were set to 10 and 1, respectively, and a radial basis function (RBF) kernel was used.The optimal gamma was set to the scale, which is the inverse of the product of the number of features and the variance in the data. This is represented by the formula:

$$gamma = \frac{1}{(N * \sigma^2)} \tag{1}$$

where N is the number of features and σ^2 is the variance [31].

For LDA, the eigen solver was chosen over the single value decomposition and least squares solvers. The shrinkage for LDA was placed at 0.6, chosen from a range of values between 0 and 1 with a step of 0.1. The classifiers were trained using 10-fold cross-validation on the training set and the best resulting classifier was used for evaluations on the test set.

To handle potential class imbalance resulting from rejection of trials, we oversampled from any class with trials less than the maximum number of class trials in any set. This was done differently for the SOTA and NNs, to carefully

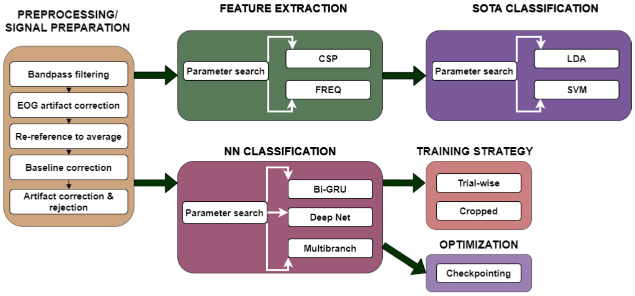

FIGURE 2. Schematic depicting processing flow of data from pre-processing to classification.

avoid data leakages. For the SOTA cross-validation, each fold from the training set was individually oversampled and for the NNs, each of the train, validation and test sets were individually oversampled. All of these were done after data split.

E. DEEP LEARNING TECHNIQUES

To compare the deep learning-based techniques, we made use of a variety of network architectures. CNNs, RNNs, LSTMs and GRUs were used either solely in a network or in a hybrid fashion, such that CNNs and sequence layers were both used within the networks. Of the various networks, we chose the 3 best performing for comparison, with the SOTA, namely: the bidirectional GRU (bi-GRU), the deep net and the multibranch network. We also experimented with pre-trained image models – Inception V3 [32] and MobileNet V2 [33]. However, their performances were significantly worse than other models and so, they were excluded from further analyses.

1) NEURAL NETWORKS

a) Bidirectional GRU - A 2-layer bidirectional GRU network was used to learn sequence relationships in both the forward and backward direction. A tanh activation was used with a dropout of 0.4, to avoid overfitting. Sequence models are known to be good at learning time relations in data and so, we applied the network for this purpose.

b) Deep Net - The network was inspired by the Deep Convnet [24]. A 5-layer modification of the original 4-layer network, introduced in Lawhern *et al.*'s work [34] was used. We used AveragePooling layers rather than MaxPooling, since the latter performed better. The default exponential linear unit (ELU) activation was also replaced with the scaled exponential linear unit (SELU) activation, as that seemed to perform better. A dropout of 0.4 was used to curb overfitting.

c) Multibranch - The network comprises 4 branches of the same CNN architecture - the Shallow network - introduced by Schirrmeister *et al.* [24]. We used a multibranch structure since that generally performed better than an unbranched one and made slight modifications, considering the data specifications. Increasing the number of branches beyond 4, generally yielded no gains. The output from the branches were concatenated and fed into a convolution layer before a final fully connected layer.

The choice of the 3 networks was for the following reasons:

a) To have a representative of sequence and convolutional models, which have been mostly used.

b) To explore CNN branching with an intent to discover how performance varies in a deep CNN as compared with a multibranched shallow one. A branched architecture tended to give more stable results across runs.

Across both SOTA and NNs, an 80:20 ratio was used to split the data into train and test sets. The training set was further split using an 80:20 ratio for train and validation sets, in the case of the NNs. For the networks, the optimal batch size was found to be 32. The networks were each trained for 50 epochs, with model checkpointing to save only the best weights, giving the least loss. Structures of the networks are presented in Tables B1, B2 and B3 of Appendix B. Data from subject A was used for all optimizations – SOTA and NNs, since subject A had plausible performance and with the expectation that average optimal values might not vary greatly compared with subject-specific ones. Optimizations could also be done on a per-subject and per-session basis. However, that could get cumbersome very quickly.

2) TRAINING AND TESTING APPROACHES

a) Trial-wise (TW)approach - This approach entailed the use of the whole 1-second trial block for training and testing. The 200 * 19 matrix for each trial in the training set was used for training and in the same way, test trials were used for evaluation.

b) Cropped approach - For this approach, crops of the 1-second block of data were used for training and testing. The main idea behind the use of crops is data augmentation for possible improvement of model performances. Crops were separately generated on train and test sets. The steps in generating crops were as follows:

i) The window length, wlen, was defined, ranging from 0.1 (100ms) to 1 (1000ms). A window length of 1000ms meant no cropping.

ii) ii. The amount of overlap was defined, *n_overlap*, with values ranging from 0 to 95, meaning no overlap to 95% overlap. In practice, any value less than 100 could be used for the overlap, since 100% overlap would be infeasible.

iii) Next, a sliding window was applied to extract crops of length, *wlen*, with the chosen overlap, from the start to the end of the trial. The number of crops generated is represented by the formula:

$$\left\lfloor \frac{((trial\_length * sampling\_rate) - crop\_size)}{(1 - n\_overlap) * crop\_size} \right\rfloor + 1 \qquad (2)$$

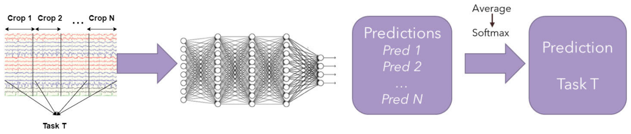

FIGURE 3. Illustration of the cropped approach.

where

$$crop\_size = wlen * sampling\_rate \qquad (3)$$

For instance, to generate crops of $wlen = 0.3$, with n_overlap = 50% and the 200Hz sampling rate, on a trial length of 1 second, the number of crops generated is 5.

$crop\_size = 0.3 * 200 = 60$

$Number\ of\ crops = floor\ [((1 * 200) - 60)/((1\text{-}0.5) * 60)] + 1 = floor(4.667) + 1 = 4 + 1 = 5$

Crops originating from the same trial were assigned the same label.

iv) For testing, the prediction scores generated on the crops originating from a test trial were averaged and the trial assigned the class with the highest mean score.

For the cropped approach, the optimal window length was found to be between 500ms and 600ms and the optimal overlap was 90%. We, therefore, used a window length of 600ms with 90% overlap. An illustration of the cropped approach is seen in Figure 3.

IV. RESULTS AND DISCUSSION

Results for both SOTA and deep learning techniques are presented in following subsections A and B. Repeated measures one-way analysis of variance (ANOVA) was performed on model results across all SOTA and deep learning methods. p-values for comparisons are reported in Table 1 and the threshold, α was set to 0.05. We do not expect a strict conformance of the results to normality and urge the reader to be aware of that.

A. STATE-OF-THE-ART TECHNIQUES

A summary of results for SOTA methods is seen in Table 2. Figure 4 also shows summarized mean accuracies in descending order. Details of our results and those reported by the dataset authors are given in appendix tables A1-A3. The authors reported their results for different train, validation and test partitions. However, for comparison with theirs, we chose the best most consistent result achieved for each subject, irrespective of the partition ratio used.

From the results, we make the following observations and deductions:

I) **Optimal parameters were more generalizable for CSP as compared with spectrograms**

In the feature extraction parameter searches for CSP and spectrograms, optimal parameters for CSP seemed to have less variability across subjects, compared to spectrograms. Before reaching a decision to use subject A's optimal values for feature extraction, across all subjects, we performed parameter searches on different subject's data. In initial subject-specific parameter searches, we noticed wider variability in optimal values for spectrograms. For CSP, on the other hand, most optimal values were in a smaller range, making it more generalizable across subjects. We infer based on these that optimal parameters and, in turn, features from CSP were more stable and generalizable across subjects and classifiers than spectrograms, which seemed to be more subject-specific.

TABLE 1. Summary of results for SOTA classifiers.

Method	Grand mean ± std. (%)
Kaya et al.	64.52 ± 16.44
CSP + LDA	67.72 ± 18.41
CSP + SVM	**68.60 ± 15.92**
FREQ.+ LDA	54.76 ± 13.95
FREQ. + SVM	53.64 ± 13.10

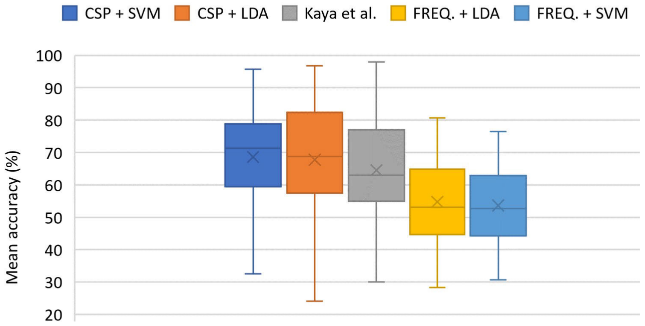

FIGURE 4. Summary box plot of SOTA methods stated in the study. Bars are shown in descending order of mean accuracy. Crosses depict the mean value.

II) **CSP features yield better results than spectrograms**

CSP yielded better results, than spectrograms, with p-values showing a statistically significant difference (3.42E-07, 7.62E-08, 8.67E-09, 1.48E-09 < 5E-02). The reduced performance of spectrograms may be partly attributed to the relative non-generalizability and instability of the optimal parameter values. Our approach in the use of CSP SOTA techniques gave slightly better results than the authors' approach, albeit not significantly (1.85E-01, 6.89E-02 > 5E-02), but both the authors' and ours compared better than with the use of spectrograms.

B. DEEP LEARNING-BASED APPROACHES

We present the results from using 3 neural network architectures, with the data in trial-wise and cropped form. Also, CSP features and spectrograms were used with the networks to determine if the neural networks needed these feature extraction techniques to give optimal performance or were

TABLE 2. **p-values for comparisons across techniques ($\alpha = 5E\text{-}02$).**

	Kaya et al.	CSP + LDA	CSP + SVM	FREQ. + LDA	FREQ. + SVM	TW + Bi-GRU	TW + Deep Net	TW + Multi-branch	Crops + BiGRU	Crops + Deep Net	Crops + Multi-branch	CSP + Bi-GRU	CSP + Deep Net	CSP + Multi-branch	FREQ + BiGRU	FREQ + Deep Net	FREQ + Multi-branch
Kaya et al.	-	1.85E-01	6.89E-02	**1.61E-04**	**2.77E-05**	**5.64E-07**	**7.10E-08**	**3.51E-06**	**4.54E-06**	**4.78E-04**	**4.02E-07**	**2.26E-08**	**3.20E-08**	**4.08E-07**	**1.39E-03**	8.94E-02	1.49E-01
CSP + LDA	-	-	2.63E-01	**3.42E-07**	**7.62E-08**	**7.04E-06**	**9.04E-09**	**4.94E-04**	**1.44E-03**	**1.48E-02**	**3.52E-05**	**6.68E-08**	**3.48E-08**	**4.30E-05**	**4.41E-02**	7.73E-01	8.49E-01
CSP + SVM	-	-	-	**8.67E-09**	**1.48E-09**	**1.39E-06**	**1.12E-09**	**9.83E-05**	**4.30E-04**	**1.30E-02**	**7.57E-06**	**1.82E-09**	**4.64E-10**	**9.20E-06**	**4.14E-02**	9.10E-01	8.68E-01
FREQ. + LDA	-	-	-	-	1.43E-01	**1.62E-13**	**5.74E-17**	**8.49E-13**	**2.95E-11**	**2.36E-09**	**3.17E-12**	**1.75E-15**	**4.40E-17**	**1.64E-13**	**1.76E-12**	**3.25E-10**	**9.92E-10**
FREQ. + SVM	-	-	-	-	-	**3.39E-15**	**3.91E-18**	**1.61E-14**	**3.85E-12**	**3.23E-10**	**2.66E-13**	**3.63E-17**	**2.79E-18**	**2.08E-15**	**1.16E-14**	**6.63E-12**	5.99E-11
TW + Bi-GRU	-	-	-	-	-	-	**5.05E-03**	6.92E-01	9.15E-01	**6.53E-03**	**1.64E-02**	**3.28E-03**	**3.41E-03**	9.64E-02	**1.24E-03**	**1.79E-06**	**8.17E-06**
TW + Deep Net	-	-	-	-	-	-	-	7.15E-02	7.15E-02	**4.37E-05**	8.19E-01	9.33E-01	6.35E-01	5.15E-01	**9.06E-06**	**1.79E-11**	**1.63E-08**
TW + Multi-branch	-	-	-	-	-	-	-	-	7.14E-01	**7.58E-03**	**1.32E-02**	**3.45E-02**	6.17E-02	**2.42E-02**	**4.28E-05**	**1.91E-07**	**6.62E-09**
Crops + BiGRU	-	-	-	-	-	-	-	-	-	**3.06E-03**	**2.24E-04**	5.90E-02	5.32E-02	6.27E-02	**1.32E-03**	**3.05E-05**	**1.41E-06**
Crops + Deep Net	-	-	-	-	-	-	-	-	-	-	**6.48E-08**	**3.84E-05**	**8.55E-06**	**9.80E-05**	6.07E-01	**1.20E-02**	**6.82E-03**
Crops + Multi-branch	-	-	-	-	-	-	-	-	-	-	-	7.76E-01	5.93E-01	2.58E-01	**2.26E-06**	**2.27E-07**	**1.96E-08**
CSP + Bi-GRU	-	-	-	-	-	-	-	-	-	-	-	-	7.34E-01	5.30E-01	**2.03E-06**	**8.58E-11**	4.60E-09
CSP + Deep Net	-	-	-	-	-	-	-	-	-	-	-	-	-	6.42E-01	**5.07E-07**	**1.62E-10**	**4.67E-09**
CSP + Multi-branch	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**5.55E-07**	**3.21E-09**	**2.83E-10**
FREQ + BiGRU	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**7.42E-03**	**2.89E-04**
FREQ + Deep Net	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9.14E-01

TABLE 3. **Summary of raw data with deep learning-based classifiers.**

Method	Grand mean ± std. (%)
Trial-wise + Bi-GRU	77.70 ± 11.06
Trial-wise + Deep Net	**81.49 ± 11.43**
Trial-wise + Multibranch	78.32 ± 7.53
Crops + Bi-GRU	77.90 ± 7.35
Crops + Deep Net	73.52 ± 10.55
Crops + Multibranch	**81.92 ± 6.48**

TABLE 4. **Summary of SOTA feature extraction with deep learning-based classifiers.**

Method	Grand mean ± std. (%)
CSP + Bi-GRU	**81.40 ± 11.16**
CSP + Deep Net	81.05 ± 10.79
CSP + Multibranch	80.42 ± 7.68
FREQ + Bi-GRU	72.63 ± 10.18
FREQ + Deep Net	68.38 ± 11.48
FREQ + Multibranch	68.23 ± 9.35

capable of capturing relevant relationships from the raw data. Summaries of results for deep learning methods are seen in Tables 3 and 4. Table 3 shows results for the use of the raw data in trial-wise and cropped forms, while Table 4 shows the use of SOTA features with the networks. Figure 5 also shows summarized mean accuracies of all the methods in descending order.

The following observations and deductions were made:

I) **All subjects gave plausibly discriminatory signals**
While some subjects performed significantly better than others, all subjects achieved greater than chance level performance (approximately 20%) [35]. This shows that subjects were able to give signals discriminatory enough for the decoding.

II) **Crops yielded better performance in shallow networks compared with the deep net**
The results from the use of crops across the neural networks yielded varying performances, which we attribute to their respective architectures. For the multi-branch network, results show that using crops gave significantly better results than with the trial-wise form ($p = 1.32E\text{-}02 < 5E\text{-}02$). The reverse was observed

TABLE 5. Results of SOTA classifiers.

Subject	Session (Date)	Kaya et al. (%)	CSP + LDA (%)	CSP + SVM (%)	FREQ. + LDA (%)	FREQ. + SVM (%)
A	160223	68	71.52	72.12	52.73	55.15
	160308	69	85.26	76.84	73.68	72.63
	160310	77	76.28	75.64	63.46	62.82
B	160218	66	61.17	61.70	44.15	38.30
	160225	44	44.38	51.69	37.64	35.96
	160229	40	73.91	73.91	47.83	46.74
C	160224	88	88.07	82.95	75	71.59
	160302	77	91.98	89.3	80.75	76.47
E	160219	57	59.88	59.28	43.11	45.51
	160226	61	51.33	59.33	50	52.67
	160304	54	68.75	71.35	56.25	51.56
F	160202	58	67.72	66.67	65.08	62.96
	160203	63	56.45	62.37	64.52	55.38
	160204	78.95	60	55.79	63	77.37
G	160301	79	80.14	76.71	49.32	47.26
	160322	58	66.12	71.04	38.25	38.80
	160412	77	84.67	78.67	53.33	56.67
H	160720	40	26.67	32.67	31.33	30.67
	160722	40	44.38	41.88	32.50	35.62
I	160609	62	46.89	51.98	45.2	42.94
	160628	30	24.10	32.53	28.31	30.72
J	161121	98	96.77	95.70	77.42	67.20
K	161027	65	59.32	61.02	53.11	50.28
	161108	55	65.43	64.36	60.11	61.17
L	161116	83	78.88	81.37	52.17	58.39
	161205	83	89.89	92.13	69.10	75.28
M	161108	88	58.56	59.67	52.49	52.49
	161117	73	90.76	89.67	73.91	73.91
	161124	55	77.37	77.89	57.37	50.53

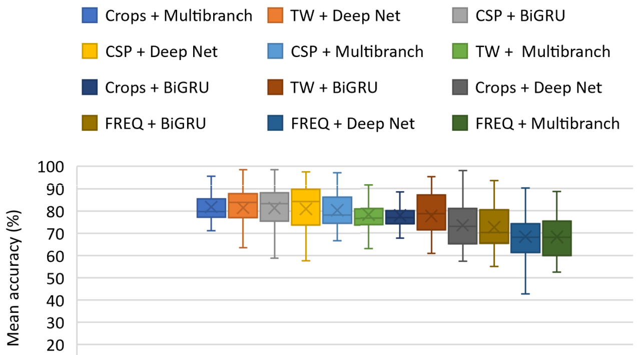

FIGURE 5. Summary box plot of deep learning methods stated in the study. Bars are shown in descending order of mean accuracy. Crosses depict the mean value.

in the case of the deep net, for which the trial-wise approach gave significantly better results than the use of crops ($p = 4.37E\text{-}05 < 5E\text{-}02$). Contrary to the observed results of these two, the BiGRU showed no difference in performance using either approach ($p = 9.15E\text{-}01 > 5E\text{-}02$). From these, we infer based on the network architectures that performance with crops tends to degrade with a deep network but with shallow networks, performance obtained is similar to that of the trial-wise form, as in the case of the BiGRU, or better as with the multibranch network. We infer that the multibranch network particularly had better performance since the branches were shallow networks and the branched structure provided more stability compared to others with unbranched structures. Branches may provide stability by combining knowledge learnt across branches to form a more robust classification decision. We, therefore, recommend the use of multi-branch shallow networks with crops, for improved and more stable results. This is somewhat comparable with a related work [24], where improvements were noticed with crops, but only in high frequencies. This then means that crops-based improvements might be dependent on different factors, such as the frequencies

TABLE 6. Results of raw data with deep learning-based classifiers.

Subject	Session (Date)	Trial-wise + Bi-GRU (%)	Trial-wise + Deep Net (%)	Trial-wise + Multibranch (%)	Crops + Bi-GRU (%)	Crops + Deep Net (%)	Crops + Multibranch (%)
A	160223	86.98	84.90	78.89	72.62	72.99	79.90
	160308	84.85	91.90	77.93	78.92	75.93	86.04
	160310	80.21	83.87	85.35	82.83	71.11	77.08
B	160218	73.33	77.03	67.11	72.81	64.65	72.52
	160225	65.15	59.09	68.57	76.77	57.84	75.98
	160229	78.79	84.34	75.49	70.20	74.51	83.33
C	160224	83.85	90.32	80.21	88.43	90.20	94.44
	160302	95.05	92.93	91.67	76.67	77.96	83.33
E	160219	78.65	82.81	80.30	78.57	69.19	79.03
	160226	68.28	77.22	81.67	77.62	66.19	76.32
	160304	72.55	79.52	77.14	79.44	77.08	79.17
F	160202	77.14	87.96	74.77	69.61	70	79.80
	160203	75.98	82.88	72.86	76.39	69.05	79.29
	160204	71.93	85.96	79.63	76.47	70	78.10
G	160301	90.91	86.27	76.67	77.78	83.89	88.24
	160322	70.72	80.30	76.47	80.63	75.71	83.33
	160412	87.37	86.87	74.29	76.04	84.34	85.14
H	160720	43.89	60.22	72.04	73.81	57.53	77.60
	160722	64.44	73.66	63.24	58.59	61.90	71.26
I	160609	69.66	63.64	75.68	75.52	60.75	77.78
	160628	60.95	48.92	73.12	67.71	58.59	78.65
J	161121	95.37	98.48	96.46	93.94	92.86	93.43
K	161027	73.23	82.83	71.72	76.96	62.50	81.82
	161108	79.73	87.50	76.39	74.54	66.19	80.56
L	161116	88.17	83.87	86.76	83.33	84.34	93.14
	161205	90.95	94.12	88.89	94.12	86.36	91.92
M	161108	74.77	70.95	76.19	79.52	78.12	77.08
	161117	87.14	97.22	93.63	78.43	74.12	75.93
	161124	83.33	87.62	78.28	90.74	98.10	95.45

present, length of signals or architecture of networks. It would be worth investigating these factors in detail.

III) **Neural networks generally perform optimally but can be exploited to yield improved results in suitable scenarios**
Using the trial-wise approach, the Deep net outperformed BiGRU ($p = 5.05E\text{-}03 < 5E\text{-}02$) but had no significant difference compared to the multibranch network ($p = 7.15E\text{-}02 > 5E\text{-}02$). Using crops, on the other hand, resulted in the multibranch network performing significantly better than Deep net while attaining similar performance as the BiGRU. Comparisons amongst the networks do not generally show one as better compared to others, as each network performed best of all in different scenarios, while performing optimally in general. We, therefore, infer that all networks perform optimally but can be more suited to specific scenarios. For instance, it might be preferable to use multibranch shallow networks when cropping is applied and deep networks when not applied.

IV) **The use of SOTA features with NNS could yield improved results**
The use of CSP features yielded better or similar performance with most networks. With the BiGRU, for instance, using CSP significantly outperformed TW ($p = 3.28E\text{-}03 < 5E\text{-}02$) but yielded similar performances with TW, using the deep net ($p = 6.35E\text{-}01 > 5E\text{-}02$) and slightly better performance with the multibranch network ($p = 2.42E\text{-}02 < 5E\text{-}02$). As noticed in other cases, the use of spectrograms gave worse results. This leads us to conclude, in this case, that the use of CSP does help improve neural network performance. However, not always by a significant margin. The benefits of using CSP for decoding with neural networks would have to be measured against the corresponding computational cost associated with

TABLE 7. Results of SOTA feature extraction with deep learning-based classifiers.

Subject	Session (Date)	CSP + Bi-GRU (%)	CSP + Deep Net (%)	CSP + Multibranch (%)	FREQ. + Bi-GRU (%)	FREQ. + Deep Net (%)	FREQ. + Multibranch (%)
A	160223	88.54	81.67	73.66	69.89	70.83	65.2
	160308	91.67	91.20	86.19	86.27	82.83	80.88
	160310	85.56	84.80	85	74.44	72.92	77.59
B	160218	82.86	73.87	75.24	59.46	64.14	56.37
	160225	58.82	69.27	74.32	70.27	45.10	62.25
	160229	80.88	74.07	76.26	64.29	63.24	63.81
C	160224	87.88	85.71	86.27	82.83	70.10	75.98
	160302	95.96	95.37	90.91	87.88	88.24	79.41
E	160219	82.78	81.25	80.73	65.59	60.75	56.77
	160226	82.76	76.11	82.26	65.56	76.11	71.08
	160304	85.14	84.29	73.61	77.93	66.20	72.92
F	160202	85.71	85.04	83.81	76.77	68.14	72.97
	160203	72.40	83.82	75.23	69.05	73.44	70.71
	160204	83.33	85.19	78.28	62.63	62.04	65.28
G	160301	87.93	90.32	78.49	68.14	72.92	60.48
	160322	83.33	79.90	74.76	68.69	56.25	59.60
	160412	74.29	88.17	87.14	71.72	65.69	57.84
H	160720	55	60.61	77.42	55.21	59.77	52.53
	160722	65.15	60.94	72.73	59.31	60.95	63.73
I	160609	65.71	63.89	69.12	71.93	52.25	66.15
	160628	61.76	57.78	72.58	59.80	42.71	52.60
J	161121	98.48	97.47	97.14	93.63	90.20	83.85
K	161027	77.6	73.53	66.67	66.67	65.66	56.86
	161108	82.35	85.78	76.75	72.86	70.10	68.14
L	161116	94.44	89.35	92.19	84.31	75	73.12
	161205	93.55	94.44	90.62	90.28	84.31	75
M	161108	76.67	72.40	76.67	67.14	65.74	73.23
	161117	95.71	93.94	94.76	90.69	88.38	88.57
	161124	84.34	90.20	83.33	73.04	69.05	75.76

TABLE 8. Structure of the Bi-GRU in sequential order.

Layer Type	Units	Dropout rate	Activation	Output
Bidirectional GRU	64	-	tanh	(None, 200, 128)
Dropout	-	0.4	-	(None, 200, 128)
Bidirectional GRU	32	-	tanh	(None, 200, 64)
Dropout	-	0.4	-	(None, 200, 64)
Flatten	-	-	-	(None, 12800)
Dense	6	-	linear	(None, 6)
Activation	-	-	softmax	(None, 6)

computing CSP features before use in the neural networks.

V) **Neural network approaches significantly outperformed authors' approaches**

For most approaches using the neural nets, results were significantly better than the authors' reported results. For instance, p-value comparisons of the TW + NN against the authors' results, give 5.64E-07, 7.10E-08 and 3.51E-06 (all < 5E-02) for the BiGRU, Deep Net and multibranch networks, respectively.

This can be expected given a number of factors to consider such as pre-processing, which was not done by the authors; optimal feature and hyperparameter searches, use of more sophisticated neural networks with a careful consideration of network architectures for optimality. Pre-processing helped to remove

TABLE 9. Structure of the Deep Net in sequential order.

Layer Type	Filters	Kernel size	Pool size	Strides	Dropout rate	Output
Input	-	-	-	-	-	(None, 1, 19, 200)
Conv	25	(1, 6)	-	-	-	(None, 25, 19, 195)
Conv	25	(1, 6)	-	-	-	(None, 25, 1, 195)
Batch Norm	-	-	-	-	-	(None, 25, 1, 195)
Activation (SELU)	-	-	-	-	-	(None, 25, 1, 195)
Average pooling	-	-	(1, 3)	(1, 2)	-	(None, 25, 1, 97)
Dropout	-	-	-	-	0.4	(None, 25, 1, 97)
Conv	50	(1, 6)	-	-	-	(None, 50, 1, 92)
Batch Norm	-	-	-	-	-	(None, 50, 1, 92)
Activation (SELU)	-	-	-	-	-	(None, 50, 1, 92)
Average pooling	-	-	(1, 3)	(1, 2)	-	(None, 50, 1, 45)
Dropout	-	-	-	-	0.4	(None, 50, 1, 45)
Conv	100	(1, 6)	-	-	-	(None, 100, 1, 40)
Batch Norm	-	-	-	-	-	(None, 100, 1, 40)
Activation (SELU)	-	-	-	-	-	(None, 100, 1, 40)
Average pooling	-	-	(1, 3)	(1, 2)	-	(None, 100, 1, 19)
Dropout	-	-	-	-	0.4	(None, 100, 1, 19)
Conv	200	(1, 6)	-	-	-	(None, 200, 1, 14)
Batch Norm	-	-	-	-	-	(None, 200, 1, 14)
Activation (SELU)	-	-	-	-	-	(None, 200, 1, 14)
Max pooling	-	-	(1, 3)	(1, 2)	-	(None, 200, 1, 6)
Dropout	-	-	-	-	0.4	(None, 200, 1, 6)
Flatten	-	-	-	-	-	(None, 1200)
Dense	6	-	-	-	-	(None, 6)
Activation (Softmax)	-	-	-	-	-	(None, 6)

different forms of artifacts in the signals, strengthening the signal quality and therefore, yielding better results from the learning. Also, optimal feature and hyperparameter tuning helped to get a more general set of values yielding optimal features and learning for the networks. The architectural choices made also contributed to this. Our choices considered the best-performing networks with as few parameters as possible. All these resulted in improved results over authors' reported results.

VI) **Neural networks significantly outperformed SOTA** For other SOTA techniques, as applied in this study, all feature parameter tuning and data enhancement techniques were applied. With that, the results would reveal how well SOTA performs, as compared with NNs, as these enhancements were applied to both categories. Across neural network approaches, results show that for all except where spectrograms were used with the networks, performances were significantly better (p-values from 4.94E-04 - 9.04E-09; all < 5E-02) than for all SOTA approaches. This shows that neural networks are more sophisticated and capable of giving better results over the SOTA. They should therefore be explored more for motor imagery decoding, as they tend to capture task-relevant relationships in the data much better than the prevalent techniques. The challenge of having small number of data points might be surmounted by carefully choosing architectures suitable for such sizes. Also, transfer learning and data augmentation might be helpful in such situations.

TABLE 10. Structure of the Multibranch network in sequential order.

Layer Type	Filters	Kernel size	Pool size	Strides	Dropout rate	Output	Connected to
Input	-	-	-	-	-	[(None, 1, 19, 200)]	-
Conv$_1$	40	(1, 11)	-	-	-	(None, 40, 19, 190)	Input
Conv$_2$	40	(1, 11)	-	-	-	(None, 40, 19, 190)	Input
Conv$_3$	40	(1, 11)	-	-	-	(None, 40, 19, 190)	Input
Conv$_4$	40	(1, 11)	-	-	-	(None, 40, 19, 190)	Input
Conv$_5$	40	(1, 11)	-	-	-	(None, 40, 1, 190)	Conv$_1$
Conv$_6$	40	(1, 11)	-	-	-	(None, 40, 1, 190)	Conv$_2$
Conv$_7$	40	(1, 11)	-	-	-	(None, 40, 1, 190)	Conv$_3$
Conv$_8$	40	(1, 11)	-	-	-	(None, 40, 1, 190)	Conv$_4$
Batch Norm$_1$	-	-	-	-	-	(None, 40, 1, 190)	Conv$_5$
Batch Norm$_2$	-	-	-	-	-	(None, 40, 1, 190)	Conv$_6$
Batch Norm$_3$	-	-	-	-	-	(None, 40, 1, 190)	Conv$_7$
Batch Norm$_4$	-	-	-	-	-	(None, 40, 1, 190)	Conv$_8$
Activation$_1$ (Square)	-	-	-	-	-	(None, 40, 1, 190)	Batch Norm$_1$
Activation$_2$ (Square)	-	-	-	-	-	(None, 40, 1, 190)	Batch Norm$_2$
Activation$_3$ (Square)	-	-	-	-	-	(None, 40, 1, 190)	Batch Norm$_3$
Activation$_4$ (Square)	-	-	-	-	-	(None, 40, 1, 190)	Batch Norm$_4$
Average pooling$_1$	-	-	(1, 33)	(1, 7)	-	(None, 40, 1, 23)	Activation$_1$
Average pooling$_2$	-	-	(1, 33)	(1, 7)	-	(None, 40, 1, 23)	Activation$_2$
Average pooling$_3$	-	-	(1, 33)	(1, 7)	-	(None, 40, 1, 23)	Activation$_3$
Average pooling$_4$	-	-	(1, 33)	(1, 7)	-	(None, 40, 1, 23)	Activation$_4$
Activation$_5$ (Log)	-	-	-	-	-	(None, 40, 1, 23)	Average pooling$_1$
Activation$_6$ (Log)	-	-	-	-	-	(None, 40, 1, 23)	Average pooling$_2$
Activation$_7$ (Log)	-	-	-	-	-	(None, 40, 1, 23)	Average pooling$_3$
Activation$_8$ (Log)	-	-	-	-	-	(None, 40, 1, 23)	Average pooling$_4$
Dropout$_1$	-	-	-	-	0.5	(None, 40, 1, 23)	Activation$_5$
Dropout$_2$	-	-	-	-	0.5	(None, 40, 1, 23)	Activation$_6$
Dropout$_3$	-	-	-	-	0.5	(None, 40, 1, 23)	Activation$_7$
Dropout$_4$	-	-	-	-	0.5	(None, 40, 1, 23)	Activation$_8$
Concatenate	-	-	-	-	-	(None, 40, 1, 92)	Dropout$_1$, Dropout$_2$, Dropout$_3$, Dropout$_4$
Conv$_8$	64	(10, 1)	-	-	-	(None, 64, 1, 92)	Concatenate
Activation$_9$ (RELU)	-	-	-	-	-	(None, 64, 1, 92)	Conv$_8$
Flatten	-	-	-	-	-	(None, 5888)	Activation$_9$
Dense	6	-	-	-	-	(None, 6)	Flatten
Activation$_{10}$ (Softmax)	-	-	-	-	-	(None, 6)	Dense

Given the approach detailed here, we recommend the use of neural networks over the SOTA.

VII) **Comparisons with other works**

Not many works have explored the dataset used in this study. Of the few works that have [36]–[38], only two performed classifications. In Shahbakhti *et al.* [36], the authors investigated the detection and elimination of eye blinks from EEG trials but performed no classification. Other works by Phang and Ko [37] and Mwata-Velu *et al.* [38] performed classification, though with limited number of classes. Phang and Ko [37] focused on left- and right- foot distinction using CSP, band power and Pearson's correlation-based connectivity features with traditional SOTA algorithms - SVM, LDA and KNN. While they reported plausible results for the best-performing method (86.26 ± 9.95%), it should be noted that their result is based on a binary decoding task. Also, the authors did not explore deep learning methods, as done in this study. Mwata-Velu *et al.* [38], on the other hand, explored a hybrid CNN-LSTM architecture for the classification of signals. The accuracy of their three-class classifier of left-, right- and no- hand imagined movements was reported to be 79.2%. As compared with ours, they used a smaller number of classes and reported less performance as compared with some of our deep learning methods. Also, they did not report exploring multiple architectures and making comparisons based on those. Finally, many of these works did not perform pre-processing in the manner reported in this study.

V. CONCLUSION

In this comparative study, we investigated the different approaches to motor imagery decoding. SOTA techniques, which have mostly been the use of CSP and frequency transforms with SVM and LDA classifiers, were compared with neural networks. We have presented our results categorized by the different approaches and provided summaries of the results and p-values for comparisons.

From the results, we conclude that neural networks are suitable for motor imagery decoding and offer some improvement over the SOTA. They are more sophisticated and capable of modelling underlying task relevant relationships in the data and do not need specific feature extraction to perform well or always boost their performance. As seen from the results, using the raw data is suitable for motor imagery decoding and while the use of specific feature extraction might give some gains in performance, it's not required for optimal performance. Also, we conclude that the use of cropping for data augmentation enhances performance, depending on a factor such as the network architecture. As seen from the results, cropping improved results in shallow networks but worsened performance in the deep network. This leads us to infer that the performance of crops in neural networks might be affected by these factors - range of frequencies present (as seen in Schirrmeister et al's work [24]), network architecture and, possibly, the length of the trial. Further investigation needs to be done on other datasets of varying lengths to also determine how much effect the length of trials has when applying cropping for enhancements.

Considering the performances of the CSP-based SOTA approaches, our conclusion is that when following a SOTA approach, CSP is preferable for feature extraction. This is because it significantly outperforms other SOTA approaches and offers more in terms of stability of features and results. While we cannot state clearly which SOTA classifier performed best, the results tend to show that SVM performed better than LDA. From the results, we conclude in this case, that deep learning-based techniques are better than SOTA, as they show that deep learning-based techniques outperformed SOTA approaches. Taking it further, deep learning-based techniques might be improved using data augmentation and model enhancement techniques, such as transfer learning.

A limitation of this work is that we have not provided time comparisons for each of the different approaches. All experiments were run on Google's Colab GPU - Tesla T4. We, however, could not provide direct time comparisons since the neural networks were optimized for running on a GPU, while the core library used for the SOTA techniques was not.

In future, we will apply transfer learning specifically with the neural networks. This will involve intra-subject and inter-subject transfer learning, to provide a solution to the non-stationarity problem in the EEG experiments. Intra-subject transfer learning would involve making use of knowledge learnt from a previous session in another session, for the same subject. Inter-subject transfer learning, on the other hand involves making use of knowledge learnt from other subjects across different sessions, for a target subject. With this, we opine that there would be improvements in performance of the neural networks, since previously learnt knowledge can be useful in future sessions. Also, with model adaptation, improvements are anticipated since the model is adapted periodically. There, however, is the challenge of finding the optimal adaptation frequency, with this approach.

APPENDIX A
TABLES OF RESULTS FROM ALL APPROACHES

See Tables 5–7.

APPENDIX B
TABULAR STRUCTURE OF THE NEURAL NETWORK MODELS

See Tables 8–10.

REFERENCES

[1] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," *Proc. IEEE*, vol. 89, no. 7, pp. 1123–1134, Jul. 2001, doi: 10.1109/5.939829.

[2] J. Zhuang and G. Yin, "Motion control of a four-wheel-independent-drive electric vehicle by motor imagery EEG based BCI system," in *Proc. 36th Chin. Control Conf.*, Jul. 2017, pp. 5449–5454, doi: 10.23919/ChiCC.2017.8028220.

[3] B. A. S. Hasan and J. Q. Gan, "Hangman BCI: An unsupervised adaptive self-paced brain–computer interface for playing games," *Comput. Biol. Med.*, vol. 42, no. 5, pp. 598–606, May 2012, doi: 10.1016/j.compbiomed.2012.02.004.

[4] G. Pfurtscheller and C. Neuper, "Motor imagery activates primary sensorimotor area in humans," *Neurosci. Lett.*, vol. 239, nos. 2–3, pp. 65–68, 1997, doi: 10.1016/S0304-3940(97)00889-6.

[5] J. Kevric and A. Subasi, "Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system," *Biomed. Signal Process. Control*, vol. 31, pp. 398–406, Jan. 2017, doi: 10.1016/j.bspc.2016.09.007.

[6] N. Padfield, J. Zabalza, H. Zhao, V. Masero, and J. Ren, ''EEG-based brain-computer interfaces using motor-imagery: Techniques and challenges,'' *Sensors*, vol. 19, no. 6, p. 1423, Mar. 2019, doi: 10.3390/s19061423.

[7] Y. Kim, J. Ryu, K. K. Kim, C. C. Took, D. P. Mandic, and C. Park, ''Motor imagery classification using mu and beta rhythms of EEG with strong uncorrelating transform based complex common spatial patterns,'' *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–13, Jan. 2016, doi: 10.1155/2016/1489692.

[8] P. Wierzgała, D. Zapała, G. M. Wojcik, and J. Masiak, ''Most popular signal processing methods in motor-imagery BCI: A review and meta-analysis,'' *Frontiers Neuroinform.*, vol. 12, p. 78, Nov. 2018, doi: 10.3389/fninf.2018.00078.

[9] O. George, R. Smith, P. Madiraju, N. Yahyasoltani, and S. I. Ahamed, ''Motor Imagery: A review of existing techniques, challenges and potentials,'' in *Proc. 45th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jul. 2021, pp. 1893–1899.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[11] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, ''Deep learning earth observation classification using ImageNet pretrained networks,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 105–109, Jan. 2016, doi: 10.1109/LGRS.2015.2499239.

[12] N. Srivastava, E. Mansimov, and R. Salakhutdinov, ''Unsupervised learning of video representations using LSTMs,'' in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 1, 2015, pp. 843–852.

[13] D. Steyrl, R. Scherer, F. Oswin, and R. M. Gernot, ''Motor imagery brain-computer interfaces: Random forests vs regularized LDA-non-linear beats linear,'' in *Proc. 6th Int. Brain-Comput. Interface Conf.*, vol. 1, 2014, pp. 241–244, doi: 10.3217/978-3-85125-378-8-61.

[14] D. Steyrl, R. Scherer, and G. R. Müller-Putz, ''Random forests for feature selection in non-invasive brain-computer interfacing,'' in *Proc. Int. Workshop Hum.-Comput. Interact. Knowl. Discovery Complex, Unstruct., Big Data*. Berlin, Germany: Springer, Jul. 2013, pp. 207–216.

[15] S. L. Wu, C. W. Wu, N. R. Pal, C. Y. Chen, S. A. Chen, and C. T. Lin, ''Common spatial pattern and linear discriminant analysis for motor imagery classification,'' in *Proc. Symp. Comput. Intell., Cogn. Algorithms, Mind, Brain*, 2013, pp. 146–151, doi: 10.1109/CCMB.2013.6609178.

[16] J. Jin, Y. Miao, I. Daly, C. Zuo, D. Hu, and A. Cichocki, ''Correlation-based channel selection and regularized feature optimization for MI-based BCI,'' *Neural Netw.*, vol. 118, pp. 262–270, Oct. 2019, doi: 10.1016/j.neunet.2019.07.008.

[17] J. Feng, E. Yin, J. Jin, R. Saab, I. Daly, X. Wang, D. Hu, and A. Cichocki, ''Towards correlation-based time window selection method for motor imagery BCIs,'' *Neural Netw.*, vol. 102, pp. 87–95, Jun. 2018, doi: 10.1016/j.neunet.2018.02.011.

[18] B. Blankertz. (2008). *BCI Competition IV*. [Online]. Available: http://www.bbci.de/competition/iv/

[19] S. Sakhavi, C. Guan, and S. Yan, ''Learning temporal information for brain-computer interface using convolutional neural networks,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5619–5629, Nov. 2018, doi: 10.1109/TNNLS.2018.2789927.

[20] S. Sakhavi, C. Guan, and S. Yan, ''Parallel convolutional-linear neural network for motor imagery classification,'' in *Proc. 23rd Eur. Signal Process. Conf.*, 2015, pp. 2736–2740, doi: 10.1109/EUSIPCO.2015.7362882.

[21] H. Dose, J. S. Møller, H. K. Iversen, and S. Puthusserypady, ''An end-to-end deep learning approach to MI-EEG signal classification for BCIs,'' *Expert Syst. Appl.*, vol. 114, pp. 532–542, Dec. 2018, doi: 10.1016/j.eswa.2018.08.031.

[22] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, ''PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,'' *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000, doi: 10.1161/01.cir.101.23.e215.

[23] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, ''BCI2000: A general-purpose brain-computer interface (BCI) system,'' *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1034–1043, Jun. 2004, doi: 10.1109/TBME.2004.827072.

[24] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, ''Deep learning with convolutional neural networks for EEG decoding and visualization,'' *Hum. Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, Nov. 2017, doi: 10.1002/hbm.23730.

[25] Z. Wang, L. Cao, Z. Zhang, X. Gong, Y. Sun, and H. Wang, ''Short time Fourier transformation and deep neural networks for motor imagery brain computer interface recognition,'' *Concurrency Comput., Pract. Exp.*, vol. 30, no. 23, p. e4413, Dec. 2018, doi: 10.1002/cpe.4413.

[26] Z. Zhang, F. Duan, J. Sole-Casals, J. Dinares-Ferran, A. Cichocki, Z. Yang, and Z. Sun, ''A novel deep learning approach with data augmentation to classify motor imagery signals,'' *IEEE Access*, vol. 7, pp. 15945–15954, 2019, doi: 10.1109/ACCESS.2019.2895133.

[27] M. Kaya, M. K. Binli, E. Ozbay, H. Yanar, and Y. Mishchenko, ''A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces,'' *Sci. Data*, vol. 5, no. 1, pp. 1–6, Dec. 2018, doi: 10.1038/sdata.2018.211.

[28] A. Gramfort, ''MEG and EEG data analysis with MNE-Python,'' *Frontiers Neurosci.*, vol. 7, p. 267, Oct. 2013, doi: 10.3389/fnins.2013.00267.

[29] M. Jas, D. A. Engemann, Y. Bekhti, F. Raimondo, and A. Gramfort, ''Autoreject: Automated artifact rejection for MEG and EEG data,'' *NeuroImage*, vol. 159, pp. 417–429, Oct. 2017, doi: 10.1016/j.neuroimage.2017.06.030.

[30] O. Ledoit and M. Wolf, ''A well-conditioned estimator for large-dimensional covariance matrices,'' *J. Multivar. Anal.*, vol. 88, no. 2, pp. 365–411, 2004, doi: 10.1016/S0047-259X(03)00096-4.

[31] F. Pedregosa *et al.*, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, 2011.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ''Rethinking the inception architecture for computer vision,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, ''MobileNetV2: Inverted residuals and linear bottlenecks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[34] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, ''EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces,'' *J. Neural Eng.*, vol. 15, no. 5, Oct. 2018, Art. no. 056013, doi: 10.1088/1741-2552/aace8c.

[35] E. Combrisson and K. Jerbi, ''Exceeding chance level by chance: The caveat of theoretical chance levels in brain signal classification and statistical assessment of decoding accuracy,'' *J. Neurosci. Methods*, vol. 250, pp. 126–136, Jul. 2015, doi: 10.1016/j.jneumeth.2015.01.010.

[36] M. Shahbakhti, M. Beiramvand, M. Nazari, A. Broniec-Wojcik, P. Augustyniak, A. S. Rodrigues, M. Wierzchon, and V. Marozas, ''VME-DWT: An efficient algorithm for detection and elimination of eye blink from short segments of single EEG channel,'' *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 408–417, 2021, doi: 10.1109/TNSRE.2021.3054733.

[37] C.-R. Phang and L.-W. Ko, ''Global cortical network distinguishes motor imagination of the left and right foot,'' *IEEE Access*, vol. 8, pp. 103734–103745, 2020, doi: 10.1109/ACCESS.2020.2999133.

[38] T. Mwata-Velu, J. Ruiz-Pinales, H. Rostro-Gonzalez, M. A. Ibarra-Manzano, J. M. Cruz-Duarte, and J. G. Avina-Cervantes, ''Motor imagery classification based on a recurrent-convolutional architecture to control a hexapod robot,'' *Mathematics*, vol. 9, no. 6, p. 606, Mar. 2021, doi: 10.3390/math9060606.

OLAWUNMI GEORGE received the B.Sc. and M.Sc. degrees, in 2011 and 2020, respectively. She is currently pursuing the Ph.D. degree in computer science with Marquette University. She is a member of the Ubicomp Laboratory, where she is directly supervised by Dr. Sheikh Iqbal Ahamed. Prior to starting her Ph.D., she acquired professional experience working as a Software Engineer in different software and blockchain companies. Her research interests include big data, neuroscience, machine learning, and deep learning.

SARTHAK DABAS is currently pursuing the Ph.D. degree in computer science focusing in computational neuroscience with Marquette University. He is also a Professional in people analytics and also working in HR analytics and insights COE at Johnson Controls Inc. Traditionally having a background in electrical and electronics engineering, his main focus is always to create value and help drive business decisions through developing optimized architecture that leverages data science and data analytics.

ABDUR SIKDER (Member, IEEE) received the Bachelor of Science degree from The University of Waikato, Hamilton, New Zealand, the master's degree in computer science and the Ph.D. degree in computer science from The University of Sydney, Australia, and the M.B.A. degree from Lincoln University, Oakland, CA, USA. He is currently a Visiting Assistant Professor with the Department of Computer Science, Marquette University.

ROGER SMITH received the B.A. degree in social sciences (psychology and communications) from the Goshen College, Indiana, USA, the master's degree in health sciences (occupational therapy) from the University of Washington, and the Ph.D. degree in industrial engineering from the University of Wisconsin–Madison. He is currently the Director of the R2D2 Center, University of Wisconsin–Milwaukee, which provides an interdisciplinary home for basic research, applied research and development, as well as innovative instruction related to technology and disability. He has served as a primary author and the Director for more than 30 grant and contract awards of over $8 million of extramural-sponsored research and training programs. His research interests include measurement related to disability and the application of assistive technology and universal design.

PRAVEEN MADIRAJU received the Ph.D. degree. He is currently an Associate Professor of computer science and is also the Graduate Chair with the Computer Science Program, Marquette University. He is the Director of the Data Science and Text Analytics Research (DATA) Laboratory, which focuses on solving real-world problems by applying techniques from the broad area of data science and data analytics on both structured and unstructured data. The laboratory also conducts research on applying machine learning techniques to analyze textual and social media data.

NASIM YAHYASOLTANI received the B.Sc. degree in electrical and computer engineering from the University of Tehran, Tehran, Iran, in 2003, the M.Sc. degree in electrical engineering from the Iran University of Science and Technology, Tehran, in 2006, and the Ph.D. degree in electrical engineering from the University of Minnesota, Twin Cities, in June 2014. She was a Research Associate with the Digital Technology Center, University of Minnesota, from 2014 to 2017. From 2018 to 2019, she worked as a Senior Data Scientist at Harley-Davidson Motor Company. Since August 2019, she has been a Northwestern Mutual Assistant Professor with the Department of Computer Science, Marquette University. Her research interests include statistical signal processing, machine learning, optimization theory and network science with applications to wireless communications and networking, big data analytics, and smart grid.

SHEIKH IQBAL AHAMED (Senior Member, IEEE) received the Ph.D. degree in computer science from Arizona State University, USA, in 2003. He is currently a Professor and the Chair of computer science and the Director of the Ubicomp Laboratory, Marquette University, USA. He is active in system and application development of mHealth projects for Native American, Hispanic community, and other underserved populations like Nepal, Bangladesh. Most of his mHealth projects are supported by the NIH, Industry and Philanthropic organizations. He has published more than 200 peer-reviewed journals, conference, and workshop papers. His research interests include mHealth, affective computing, and non-intrusive technologies. He is a Senior Member of the ACM and the IEEE Computer Society. He has received 12 best paper/posters awards in last five years. He serves regularly on international conference program committees in software engineering and pervasive computing, such as COMPSAC, PERCOM, and SAC. He has been serving as the Standing Committee Vice Chair for the IEEE COMPSAC (compsac.org), which is a signature conference of IEEE, since 2015. He is the Guest Editor of *Computer Communications* journal (Elsevier).