Industrial Electronics Society

Physics-Informed Attention temporal convolutional network for EEG-based motor imagery classification

Journal:	*IEEE Transactions on Industrial Informatics*
Manuscript ID	TII-22-2210.R1
Manuscript Type:	SS on Scientific and Physics-Informed Machine Learning for Industrial Applications
Keywords:	

http://www.ieee-ies.org/

Physics-informed attention temporal convolutional network for EEG-based motor imagery classification

Hamdi Altaheri, *Member*, *IEEE*, Ghulam Muhammad, *Senior Member*, *IEEE*, and Mansour Alsulaiman, *Senior Member*, *IEEE*

Abstract— **The brain-computer interface (BCI) is a cutting-edge technology that has the potential to change the world. Electroencephalogram (EEG) motor imagery (MI) signal has been used extensively in many BCI applications to assist disabled people, control devices or environments, and even augment human capabilities. However, the limited performance of brain signal decoding is restricting the broad growth of the BCI industry. In this paper, we propose an attention-based temporal convolutional network (ATCNet) for EEG-based motor imagery classification. The ATCNet model utilizes multiple techniques to boost the performance of MI classification with a relatively small number of parameters. ATCNet employs scientific machine learning to design a domain-specific DL model with interpretable and explainable features, multi-head self-attention to highlight the most valuable features in MI-EEG data, temporal convolutional network (TCN) to extract high-level temporal features, and convolutional-based sliding window to augment the MI-EEG data efficiently. The proposed model outperforms the current state-of-the-art techniques in the BCI Competition IV-2a dataset with an accuracy of 85.38% and 70.97% for the subject-dependent and subject-independent modes, respectively.**

***Index Terms*— Deep learning, convolution neural network (CNN), temporal convolution networks (TCN), attention, scientific machine learning, EEG, motor imagery, classification**

I. Introduction

THE brain-computer interface (BCI) is a system that interprets brain activity and converts it into commands to control an external device, such as a wheelchair or a drone. BCI is a cutting-edge technology that has the potential to transform the world and further enhances the quality of life, with a wide range of industrial applications spanning from medical applications to human augmentation [1], [2].

The brain signal can be recorded using various techniques, such as electroencephalography (EEG). EEG is a non-invasive method that records the electrical activities of the brain. The EEG signal is captured on the scalp as a two-dimensional matrix of real values (time and channel). EEG is widely used and preferred over other techniques due to its ease of use, low cost, low risk, portability, and high temporal resolution, making it suitable for industrial applications.

Motor imagery (MI) is the activity of thinking about moving a part of the human body without physically moving it. EEG-based MI (MI-EEG) activities have been employed in a variety of medical applications, including stroke rehabilitation, wheelchair control, prostheses control, exoskeleton control, cursor control, speller, and thought-to-text conversion. MI-EEG signals have also been used in non-medical applications such as vehicle control, drone control, environment control, smart home, security, gaming, and virtual reality [2]. Therefore, MI-EEG signals have great applicability in a variety of medical and non-medical industry applications. However, the real-world applications are still limited by the decoding performance and generalization ability of the MI-EEG signal.

One of the main challenges for the real-world application of MI-EEG BCI is accurately recognizing human intention from low signal-to-noise ratio and nonstationary brain signals with various sources of artifacts, including biological artifacts (e.g., muscle movements, eye blinking), electronic equipment (e.g. computers and wireless devices), and environmental noise (e.g., light and sound). These artifacts, along with channel correlation, subject dependency, and high dimensionality of EEG signals make the analysis and classification of brain signal a challenging task.

Several conventional machine learning (ML) and deep learning (DL) approaches have been proposed to address the difficulties involved in classifying MI-EEG tasks. Among the conventional ML approaches that rely on manual feature extraction, filter bank common spatial patterns (FBCSP) [3] and its variants achieved the best performance in MI classification. In contrast to conventional methods, DL can learn distinct and latent features from raw EEG data without the requirement for pre-processing or manual feature extraction. DL has been used

Manuscript received May 9, 2022. Acknowledgment should be put here. Paper titles should be written in uppercase and lowercase letters, not all uppercase. The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (DRI-KSU-1354).

All authors are affiliated with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia. E-mail: haltahei@ksu.edu.sa (H.A.); ghulam@ksu.edu.sa (G.M.); msuliman@ksu.edu.sa (M.A.).

Corresponding author: Ghulam Muhammad (e-mail: ghulam@ksu.edu.sa)

effectively in a variety of applications, including image, video, audio, and text analysis [4]–[7]. Recently, motivated by the significant success of DL techniques in other applications, many researchers have employed DL algorithms to classify MI tasks.

In the past five years, the number of studies using DL methods to classify MI tasks has increased rapidly [2]. Different DL architectures was proposed for MI classification including convolutional neural network (CNN) [8]–[13], recurrent neural network (RNN) [14], [15], deep belief network (DBN) [16], Auto-encoder (AE) [17], and hybrid DL models [8], [18]. CNN was the most widely used architecture for MI classification [2]. Standard CNN models with light [12] and deep architectures [19] have been proposed, as well as many other CNN varieties, including inception-based CNN [10], [11], residual-based CNN [20], 3D-CNN [20], multi-scale CNN [13], multi-layer CNN [18], multi-branch CNN [9], [20], and attention-based CNN [8]–[11], [13]. Several other DL models have also been suggested by some studies for classifying MI tasks. Xu et al. [16] proposed a DBN model based on restricted Boltzmann machines (RBMs) for feature extraction and a support vector machine (SVM) for classification. Hassanpour et al. [17] proposed a stacked auto-encoder (SAE) to classify MI tasks using frequency features. In other studies, researchers have attempted to extract temporal information from the MI-EEG signal using recurrent neural networks. For example, researchers in [14] proposed a long short-term memory (LSTM) model combined with FBCSP features and an SVM classifier. In another study [15], the authors also adopted FBCSP features and used them as inputs to a gated recurrent unit (GRU) model. The study showed that the GRU model performed better than the LSTM. In general, CNN models have shown better performance in MI task classification than other DL models [2], e.g., RNN, SAE, and DBN. Therefore, many researchers have suggested integrating CNN with other DL models, such as LSTM [8] and SAE [18], and encouraging results have been obtained.

Recently, a new CNN variant called temporal convolutional network (TCN) was specifically designed for time series modeling and classification [21]. TCN outperformed other recurrent networks such as LSTM and GRU in many sequence-related tasks [21]. In contrast to typical CNNs, TCN can exponentially expand the size of the receptive field with a linear increase in the number of parameters, and unlike RNNs it does not suffer from vanishing or exploding gradients. Some recent studies have used TCN architectures to classify MI tasks [22], [23]. Ingolfsson et al. [22] proposed a TCN model named EEG-TCN that combines TCN with the well-known EEGNet architecture [12]. A recent study in [23] attempted to improve the EEG-TCN model using the feature fusion technique. Our research is an ongoing contribution to these works, which utilizes scientific machine learning (SciML) and attention mechanism with TCN architecture.

Scientific machine learning is a new field that combines machine learning and scientific computing to produce domain-aware ML models that are reliable, robust, scalable, and interpretable. SciML aims to derive insights from scientific data to reduce ML model parameters, prevent overfitting, enhance extrapolation, and overcome domain-specific data challenges, including noisy data, high dimensionality, and low signal-to-noise. SciML can produce the next wave of data-driven scientific discovery in the engineering, physical, and medical sciences [24].

The attention mechanism is an effort to emulate human brain behavior of selectively focusing on a few significant elements while ignoring others. Integrating the attention mechanism with DL models helps to focus automatically (by learning) on the most important parts of the input data. The first attention-based model (RNN model) was proposed in 2015 by Bahdanau et al. [25], known as additive attention. In the same year, Luong et al. [26] proposed an attention layer with a multiplication scoring function, known as multiplicative attention. In 2017, Google researchers proposed a pure attention model with multi-head attention, which consists of several self-attention layers [27]. These attention-based models were originally proposed for natural language processing (NLP) and have subsequently been used in other fields. For computer vision, several attention blocks have been proposed, such as squeeze-and-excitation (SE) [28] and convolutional block attention module (CBAM) [29].

Recently, researchers have used attention-based DL models to classify MI-EEG signals [8]–[11], [13]. For instance, the authors in [8] employed self-attention with LSTM and graph neural representation to decode MI tasks. The researchers in [10], [11] combined attention layers with inception-CNN and LSTM. In a more recent study, Altuwaijri et al. [9] proposed an attention-based multi-branch CNN model for classifying MI tasks using raw data. The authors used three SE attention blocks as intermediate layers in three CNN branches.

Although the current studies showed promising results in decoding MI-EEG signals, the classification performance is still limited and requires further improvement.

In this paper, we propose an attention-based temporal convolutional network, ATCNet, to decode MI-EEG brain signals. This research utilizes SciML to address domain-specific MI-EEG data challenges, which results in a robust, interpretable, and explainable DL model specifically designed for decoding MI-EEG brain signals. The proposed DL model processes the MI-EEG data in three steps: first, encode the MI-EEG signal into a sequence of high-level temporal representations using conventional layers, then, highlight the most valuable information in the temporal sequence using an attention layer, and finally, extract high-level temporal features from the highlighted information using a temporal convolutional layer. The proposed model utilizes a multi-head self-attention and convolutional-based sliding window to boost the performance of MI classification. This research highlights the following contributions:

1. We propose a high-performance ATCNet model, which utilizes the powerful of TCN, SciML, attention mechanism, and convolutional-based sliding window.
2. Performing sliding window using convolution helps augment MI data and efficiently enhance accuracy, by parallelizing the process and reducing computations.

3. Self-attention helps the DL model to attend to the most effective MI information in the EEG data, and the multiple heads help to focus on multiple positions, resulting in multiple attention representations.
4. The proposed model achieves outstanding results in the BCI Competition IV-2a (BCI-2a) dataset [30].

For reproducibility, the code for this research and the trained models will be released on GitHub. The remainder of the article is organized as follows. Section II describes the proposed ATCNet model. In Section III, we present and discuss the results. Then we finally conclude in Section IV.

II. Proposed ATCNet Model

The proposed ATCNet model consists of three main blocks: convolutional (CV) block, attention (AT) block, and temporal convolutional (TC) block, as shown in Figure 1. CV block encodes low-level spatio-temporal information within the MI-EEG signal through three convolutional layers: temporal, channel depth-wise, and spatial convolutions. The output of the CV block is a temporal sequence with a higher-level representation. The AT block then highlights the most important information in the temporal sequence using a multi-head self-attention (MSA). Finally, the TC block extracts high-level temporal features within the temporal sequence using TCN and feeds them into a fully connected (FC) layer with a SoftMax classifier.

The temporal sequence, output from CV block, can be split into multiple windows and each is fed to AT/TC blacks separately. The output of all windows is then concatenated and fed to a SoftMax classifier. This helps efficiently augment the data and enhances accuracy. The details of ATCNet blocks are described in the following subsections.

A. Preprocessing and Input Representation

In this research, we feed raw MI-EEG signals into the proposed model without preprocessing, that is, the full frequency band, all channels, and without artifact removal.

ATCNet model takes as input a motor imagery trial $X_i \in \mathbb{R}^{C \times T}$ consisting of C channels (EEG electrodes) and T time points. The objective of the ATCNet model is to map the input MI trial X_i to its corresponding class y_i, given a set of m labeled MI trials $S = \{X_i, y_i\}_{i=1}^{m}$, where $y_i \in \{1, \dots, n\}$ is the corresponding class label for trial X_i and n is the total number of defined classes for set S. For the BCI-2a [30] dataset, $T = 1125$ time points, $C = 22$ EEG channels, $n = 4$ MI classes, and $m = 5184$ MI trials.

B. Convolutional (CV) block

The CV block is similar to the EEGNet architecture proposed in [12]. CV block differs from EEGNet by using 2D convolution instead of separable convolution, which showed better performance. CV block also uses different parameter values than those used in [12].

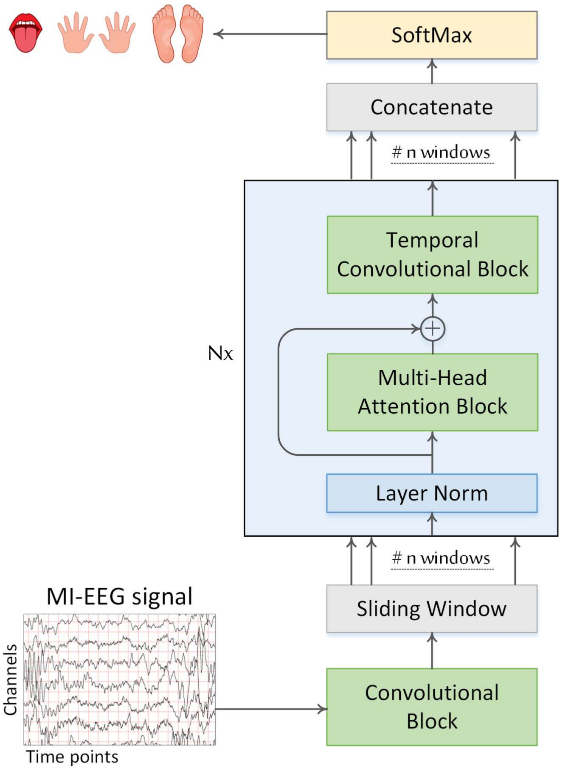

Figure 1. The components of the proposed ATCNet model.

CV block consists of three convolutional (conv) layers, as shown in Figure 2. The first layer performs a temporal convolution using F_1 filters of size $(1, K_C)$, where K_C is the filter length in the time axis. K_C was set to be one-fourth of the sampling rate (64 for BCI-2a). This allows the filters to extract temporal information associated with frequencies above 4 Hz. The output of this layer is F_1 temporal feature maps. The second layer is a depth-wise convolution with F_2 filters of size $(C, 1)$, where C is the number of EEG channels. Using depth-wise convolution, each filter extracts spatial features (i.e., related to EEG channels) from a single temporal feature map. Therefore, the output of this layer is $F_1 \times D$ feature maps, where D is the number of filters linked to each temporal feature map in the previous layer. D is set empirically to 2. $F_1 \times D$ determine the output dimension of the CV block. The depth-wise convolution is followed by an average pooling layer of size (1, 8) to abstract the temporal data by a factor of 8. This reduces the sampling rate of the signal to ~32Hz. The third convolutional layer consists of F_2 filters of size $(1, K_{C2})$. K_{C2} was set to 16 to decode MI activities within 500 ms (for 32 Hz sampled data). Finally, a second average pooling layer with a size of (1, P_2) is used to reduce the sampling rate to ~32/P_2 Hz. P_2 is used to control the length of the temporal sequence produced by CV block. The second and third conv layers are followed by batch normalization [31] to speed up network training and then by exponential linear unit (ELU) activation for nonlinearity.

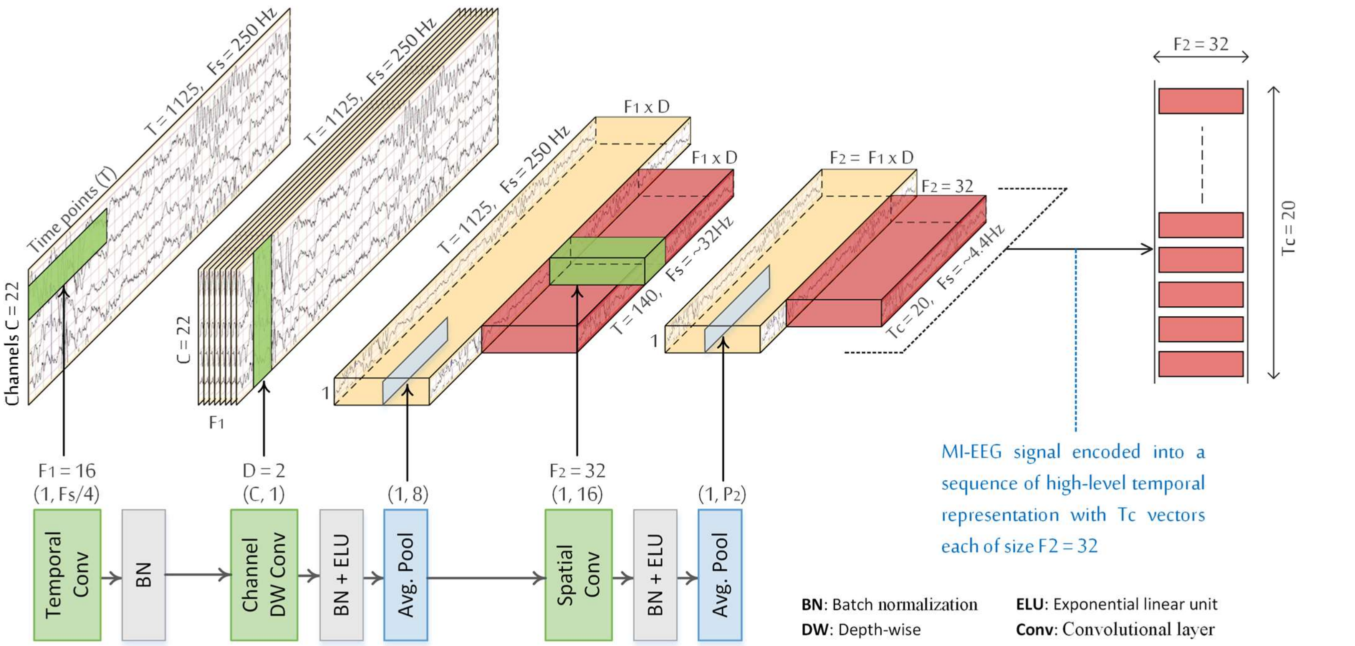

Figure 2. CV block performs spatio-temporal encoding through three convolutional layers. The CV block receives a raw MI-EEG signal and outputs a temporal sequence with T_C elements. Each element is a vector of size F_2.

CV block output a sequence $z_i \in \mathbb{R}^{T_c \times d}$ of temporal representation consisting of T_c temporal vectors each with dimension $d = F_2 = F_1 \times D$. We empirically set d *to* 32. The length of the temporal sequence z_i is determined by

$$T_c = T/8P_2 \tag{1}$$

where T refers to the time points of the original EEG signal.

C. Convolutional-based sliding window (SW)

Instead of entering the whole T_c samples of z_i to the later layers, a sliding window has been used to divide the temporal sequences into multiple windows. This helps to augment the data and enhance the decoding accuracy. However, the sliding window raises the computations, because it requires the input data to be passed through the DL model n times (instead of once), where n stands for the number of windows. As a result, the computations are incremented n times. But in our approach, we used a sliding window as integration with convolutional layers (in the convolutional block). In this approach, convolution computations are performed once for all windows, which reduces training and inference time by parallelizing the process. This technique was originally used in sliding-window-based object detection. The convolutional-based sliding window has been described in detail by Schirrmeister et al. [32].

We used a sliding window of length T_w with one step stride to divide the temporal sequence z_i into multiple windows $z_i^w \in \mathbb{R}^{T_w \times d}$ with $w = 1, \dots, n$ denoting the window index, and n is the total number of windows. Each window z_i^w is then entered separately to the later AT block and then to the TC block. The window length T_w is determined by:

$$T_w = T_c - n + 1, \qquad T_c > n \geq 1 \tag{2}$$

$$T_w = {}^{T}\!/_{8P_2} - n + 1 \tag{3}$$

If the CV block performs two temporal pooling of size $P_1 = 8$ and $P_2 = 7$, CV will produce a temporal sequence z_i consisting of $T_c = 20$ vectors (Eq. 1, where $T = 1125$). Each vector will represent 56 (8×7) time-points in the original MI-EEG signal x_i. Therefore, performing one step sliding in the z_i is equivalent to 56 time-steps sliding in the original signal x_i.

D. Attention (AT) block

In psychology, the cognitive process of selectively focusing on one or a few things while disregarding others is known as attention. In deep neural networks, the attention mechanism is an effort to emulate the human brain behavior of selectively focusing on a few significant elements while ignoring others. In the visual world, subjects use both volitional and nonvolitional cues to selectively focus attention. The former is task-dependent, and the latter is based on the conspicuity and saliency of things in the surroundings. Inspired by the voluntary and involuntary attention cues, the attention mechanism can be emulated using three components: values (sensory inputs), keys (nonvolitional cues), and queries (volitional cues). The interaction of queries and keys creates attention pooling that biases the selection of values, as demonstrated in Figure 3.

The attention mechanism can be implemented based on attention scores or by different machine learning algorithms such as reinforcement learning. This research adopts an attention scores-based approach, i.e., MSA, due to its large success in various fields such as NLP and computer vision.

The attention block consists of an MSA layer as described in [27]. MSA consists of several self-attention layers (i.e., scaled dot-product attention) called heads, as shown in Figure 4. Each self-attention layer consists of three main components: query Q, keys K, and values V. Interactions between query and keys produce attention scores that guide selection bias over values. The detailed implementation of this interaction is as follows. Given a window representation z_i^w, encoded by CV block, the query/key/value vectors are calculated for each batch using linear transformation as:

$$q_t^h = W_Q^h\, LN\left(z_{i,t}^w\right) \quad \in \mathbb{R}^{d_H}, \quad W_Q^h \in \mathbb{R}^{d\times d_H} \tag{4}$$

$$k_t^h = W_K^h\, LN\left(z_{i,t}^w\right) \quad \in \mathbb{R}^{d_H}, \quad W_K^h \in \mathbb{R}^{d\times d_H} \tag{5}$$

$$v_t^h = W_V^h\, LN\left(z_{i,t}^w\right) \quad \in \mathbb{R}^{d_H}, \quad W_V^h \in \mathbb{R}^{d\times d_H} \tag{6}$$

where LN stands for Layer Normalization [33], $t = 1, \dots, T_w$ is an index over the temporal vectors in window w and T_w is the length of the window (the total number of temporal vectors), $h = 1, \dots, H$ is an index over multiple attention heads and H is the total number of heads. The diminution of the attention head is set empirically to $d_H = d/2H$.

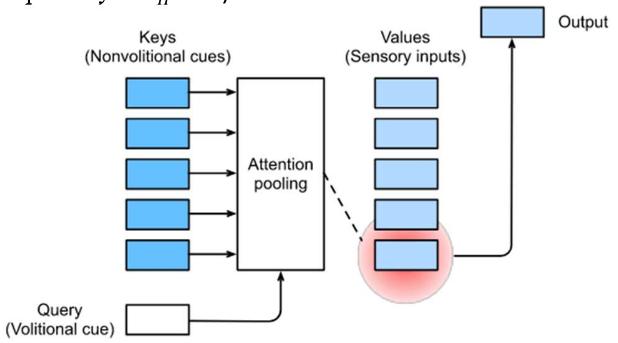

Figure 3. The interaction of queries and keys creates attention pooling that biases the selection of values.

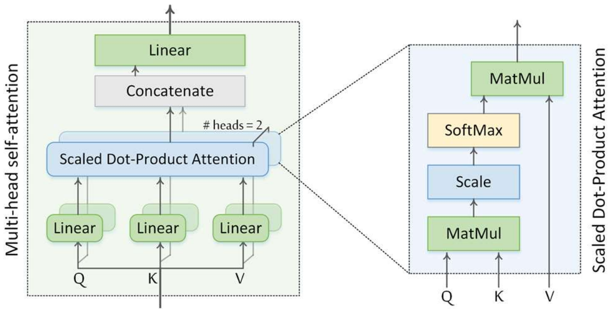
Multi-head self-attention
Linear
Concatenate
# heads = 2
Scaled Dot-Product Attention
Linear
Linear
Linear
Q
K
V
MatMul
SoftMax
Scale
MatMul
Q
K
V
Scaled Dot-Product Attention

Figure 4. Multi-head self-attention.

Given a query $q_t^h \in \mathbb{R}^{q=d_H}$ and T_w key-value pairs $(k_1^h, v_1^h), \dots., (k_{T_w}^h, v_{T_w}^h)$, where $k_{t'}^h \in \mathbb{R}^{k=d_H}$ and $v_{t'}^h \in \mathbb{R}^{v=d_H}$. The attention pooling f that generates the context vector c_t^h is defined as a weighted sum of the values $v_{t'}^h$:

$$c_t^h = f\left(q_t^h, k_{t'}^h, v_{t'}^h\right) = \sum_{t'=1}^{T_w} \alpha_{tt'}^h v_{t'}^h \in \mathbb{R}^{d_H}, \sum_{t'=1}^{T_w} \alpha_{tt'}^h = 1 \tag{7}$$

The attention weight (scalar) $\alpha_{tt'}^h$ of the query q_t^h and key $k_{t'}^h$ is calculated by applying the SoftMax function on the corresponding alignment scores $e_{tt'}^h$ as follows

$$\alpha_{tt'}^h = \text{softmax}\left(e_{tt'}^h\right) = \frac{\exp\left(e_{tt'}^h\right)}{\sum_{k=1}^{T_w} exp\left(e_{tk}^h\right)} \in \mathbb{R} \tag{8}$$

The alignment scores $e_{tt'}^h$ are calculated using the attention scoring function a as in Eq. 9. Distinct selections for the attention scoring function a result in different attention pooling behaviors. Two common scoring functions have been proposed: additive attention (Bahdanau attention [25]) and multiplicative attention (Luong attention [26]). In this paper, we use multiplicative attention, specifically scaled dot-product attention defined by Vaswani et al. [27], which is more computationally efficient. The dot product operation, however, necessitates that both the query and the key have the same vector length. The scoring function of scaled dot-product attention is defined in Eq. 10. The dot product is divided by $\sqrt{d_H}$ to ensure that the variance of the dot product remains constant regardless of vector length.

$$e_{tt'}^h = a\left(q_t^h, k_{t'}^h\right) \quad \in \mathbb{R} \tag{9}$$

$$a = \frac{(q_t^h)^T k_{t'}^h}{\sqrt{d_H}} \quad \in \mathbb{R} \tag{10}$$

For each head, the context vectors of the scaled dot-product attention for a minibatch with $n = T_w$ queries and $m = T_w$ key-values pairs (global attention) are determined by Eq. 11, where keys and queries of length d_H and values of length v (in this study $v = d_H = 8$). Attention context vectors manage and quantify the interdependence either between the input and output components (general attention) or within the input components (self-attention). In this research, we adopt the self-attention mechanism as it helps parallel computing attention to all inputs at the same time.

$$C^h = \text{softmax}\left(\frac{Q^h (K^h)^T}{\sqrt{d_H}}\right) V^h \in \mathbb{R}^{n=T_w \times v=d_H} \tag{11}$$

Where $Q \in \mathbb{R}^{n\times d_H}, K \in \mathbb{R}^{m\times d_H}$, and $V \in \mathbb{R}^{m\times v}$

Then, the MSA is computed by projecting the concatenation of the context vectors from all heads and adding the result to the input window z_i^w using a residual connection, as in Eq. 12.

$$z_i^w = W_O\left[C^1, \dots, C^H\right] + z_i^w \in \mathbb{R}^{T_w \times d}, \quad W_O \in \mathbb{R}^{d_H \times d} \tag{12}$$

E. Temporal Convolutional (TC) block

The TC block has the same architecture as the TCN described in [22]. TCN consists of a stack of residual blocks. The residual block composes of two dilated causal convolutional layers, each one followed by batch normalization [31] and ELU activation, as shown in Figure 5.

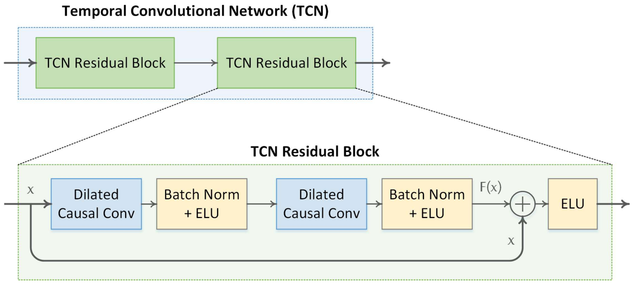

Figure 5. The architecture of the temporal convolutional network (TCN) consisting of two residual blocks.

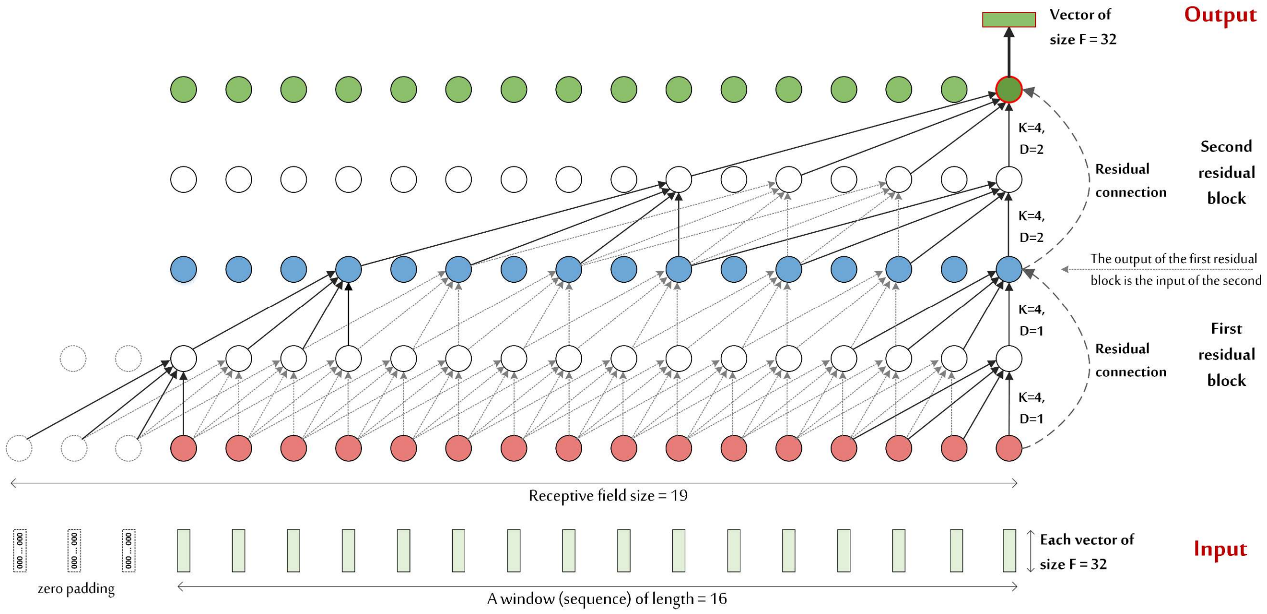

Figure 6. Visualize a TCN block with a depth of 2, i.e., consisting of two residual blocks, kernel size = 4, and number of filters = 32. The output of the first residual block is the input of the second. The receptive field size for this TCN will be 19, so the input length T_w should be ≤ 19. This figure shows a sequence of 16 temporal elements (T_w= 16) entering the TCN.

Causal convolutions are used to prevent any information from traveling from the future to the past, i.e., the output at time t depends only on the inputs from time t and earlier. Dilated convolutions allow the receptive field to be expanded exponentially while increasing the network depth. Therefore, dilated causal convolutions can learn relationships in long sequences. The residual connection performs an element-wise addition of the input and output feature map, $F(x) + x$, which is effective in deep networks due to its ability to learn the identity function. In the residual block, we use identity mapping because the input and output dimensions are identical (32), otherwise, a linear transformation, i.e., 1×1 convolution, is used.

The receptive field size (RFS) of the TCN increases exponentially with the number of stacked residual blocks, L, due to the exponential increase in dilation D with each succeeding block. The RFS is controlled by two parameters: the number of residual blocks L and the kennel size K_T, as defined in Eq. 13.

$$\mathrm{RFS} = 1 + 2(K_T - 1)(2^L - 1) \qquad (13)$$

In the proposed ATCNet, the TC block consists of a TCN with $L = 2$ residual blocks and 32 filters of size $K_T = 4$ for all convolutional layers. With this TCN, the RFS is 19, that is, the TCN can process up to 19 elements in a sequence, as shown in Figure 6. Therefore, the temporal sequence entered in the TC block should be less than or equal to 19 to allow TCN to process all temporal information without loss. For sequences that are longer than RFS, they can be split into multiple windows each with a length less than RFS. Each window is then entered separately into the TC block. In this research, we fixed RFS to 19 and changed the length of windows entering the TC block (T_w), as defined in Eq. (3).

Figure 6 shows a sequence of 16 temporal elements (T_w= 16) entering the TCN. Each element is a vector of size F_2 (#filters in CV block). The output of the TCN is the last element in the sequence, which is a vector of size F_T (# filters in TCN). In this study, $F_2 = F_T = 32$. The outputs of the TC block from all windows are concatenated and then fed to an FC layer with 4 neurons, as the number of MI classes, followed by a SoftMax classifier, as shown in Figure 1.

Unless otherwise noted, hyperparameters used for all experiments in this article are shown in Table 1. These parameters were set empirically based on several experiments and were fixed for all subjects.

Table 1. The hyperparameter setting that used for all subjects.

Attention (AT) block		Convolutional (CV) block	
# of attention heads (H)	2	# Temporal filters (F_1)	16
Head size (d_H)	8	Kernel size (K_c)	64
Dropout rate (p_a)	0.5	Depth multiplier (D)	2
		2nd pooling size (P_2)	7
		Dropout rate (p_c)	0.3
Temporal Convolutional (TC) block		# of windows (n)	5
# of residual blocks (L)	2		
Kernel size (K_T)	4		
# Filters (F_T)	32		
Dropout rate (p_t)	0.3		

III. Experimental Results and Discussion

A. Selected Dataset and Evaluation Approaches

BCI Competition IV-2a (BCI-2a) dataset [30] is used to train and evaluate the proposed model. BCI-2a is a well-known public MI-EEG dataset created by Graz University of Technology in 2008. BCI-2a has been widely used in the research community and is thus considered a benchmark dataset in MI-EEG decoding. It contains a limited number of samples captured under uncontrolled conditions with a considerable amount of artifacts, which makes decoding MI tasks using this dataset a challenging process.

BCI-2a dataset consists of 5184 trials (samples) of MI-EEG data recorded using 22 EEG electrodes from 9 subjects (576 trials per subject). MI trials last 4 seconds and were sampled at 250 Hz and filtered between 0.5 and 100 Hz. Each trial belongs to one of four MI tasks: imagining of movement of the left hand (class 1), right hand (class 2), both feet (class 3), and tongue (class 4). For each subject, two sessions were recorded on different days. Each session consists of 288 trials per subject. One of these sessions is used to train the model and the other for evaluation.

The proposed model is evaluated using subject-dependent (subject-specific) and subject-independent approaches. For subject-dependent, we used the same training and testing data as the original competition, i.e., 288 x 9 trials in session 1 for training, and 288 x 9 trials in session 2 for testing. For subject-independent, we used cross-subject evaluation, known as "Leaving One Subject Out" (LOSO). In LOSO, the model is trained and evaluated by several folds, equal to the number of subjects, and for each fold, one subject is used for evaluation and the others for training. The LOSO evaluation technique ensures that separate subjects (not visible in the training data) are used to evaluate the model.

B. Performance Metrics

The proposed models in this research are evaluated using accuracy, Eq. 14, and Kappa score, Eq. 15.

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} TP_i / I_i}{n} \tag{14}$$

where TP_i is the true positive, i.e., the number of correctly predicted samples in class i, l_i is the number of samples in class i, and n indicates the number of classes.

$$\kappa\_score = \frac{1}{n}\sum_{a=1}^{n} \frac{P_a - P_e}{1 - P_e}, \tag{15}$$

where n is the number of classes, P_a is the actual percentage of agreement, and P_e is the expected percentage chance of agreement.

C. Training Procedure

The models were trained and tested by a single GPU, Nvidia GTX 2070 8GB, using the TensorFlow framework. For all experiments, we used the following training configurations. Glorot uniform initializer is used to initialize the weights. The models are trained using the Adam optimizer with a learning rate of 0.0009, batch size of 64, and a categorical cross-entropy loss over 1000 epochs with a patience of 300. These hyperparameters were determined through several experiments to help the models generalize well.

The proposed ATCNet model achieves an overall accuracy of 85.38% and a κ-score of 0.81, which is better than the state-of-the-art results.

D. The contributions of ATCNet blocks

In this subsection, we perform an ablation analysis to measure the effectiveness of each block in the ATCNet model. Table 2 presents the impact of removing one or more blocks in the ATCNet model on the performance of MI classification using the BCI-2a dataset. Blocks were removed before training and validation operations. The results showed that the AT block increased the overall accuracy by 1.54% and SW by 2.28%. The addition of the TC block also increased accuracy by 1.04% compared to using the CV block only. The results showed that each block adds its contribution regardless of the other blocks except for the attention block. Attention block improves accuracy if followed by TC block. If the TC block is removed, the accuracy drops to 79.44%, which is lower than the accuracy after removing both AT and TC blocks, 82.60%, and even removing all blocks, 81.71%. This means that placing the attention layer at the end of the model harms the performance unless followed by an additional classification layer.

Table 2. Contribution of each block in the ATCNet model to the performance of MI classification using the BCI-2a dataset. AT: attention, SW: sliding window, TC: temporal convolution.

Removed block	Accuracy %	κ-score
None (ATCNet)	85.38	0.805
AT	83.84	0.784
SW	83.10	0.775
SW + AT	82.75	0.770
TC	79.44	0.726
SW + TC	80.48	0.740
AT + TC	82.60	0.768
SW + AT + TC	81.71	0.756

E. Varying the temporal sequence length

In the following experiments, we investigate the effect of changing the length of the temporal sequence (T_c) produced by the CV block as well as the number of windows n. The sequence length is controlled by the size of the second pooling layer (P_2) in the CV block, as defined in Eq. 1. Figure 7 shows the accuracy of MI classification using three temporal sequences of lengths 17, 20, and 28, which encode the original MI-EEG signal with a resolution of 64 samples (256 ms), 56 samples (225 ms), and 40 samples (160 ms), respectively. Each sequence is studied while increasing the number of windows from 1 to 16. ATCNet works best when the window length is less than RFS (19). Using a longer window than RFS significantly reduces accuracy due to information loss, as shown in the 28-sample sequence (while T_W = 20 to 28). In general, increasing the number of windows improved classification accuracy as this helps to augment the data and helps the model learn the changing MI information from different time positions. However, this increase in accuracy reaches a point where the window contains a narrow signal that may not contain enough MI information to train the model. For example, by dividing the 20-sample sequence into 12 windows with a stride of one, each window will have 6 samples corresponding to 384 samples (~1.5 s) in the original signal with a stride of 64. This justifies the decrease in accuracy for sequences of lengths 17 and 20 starting in 6 windows. The 20-sample sequence performed better than the other sequences in many windows and the best performance was achieved using 5 windows (each window of length 16). This indicates that encoding the original MI-EEG signal at a resolution of 56 samples (225 ms) provides a good representation compared to a lower (e.g., 160 ms) or higher (e.g., 256 ms) resolution.

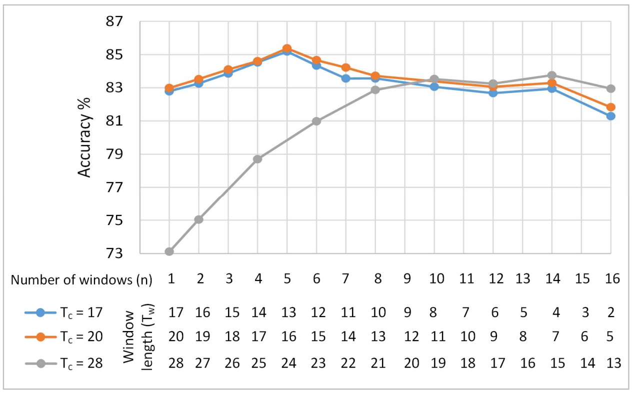

Figure 7. Accuracy on BCI-2a as a function of the number of windows using three temporal sequences of length 17, 20, and 28. These sequences were studied while varying the number of windows from one window, that is, the whole sequence, to 16 windows. Each window has a different length (T_w) depending on the length of the original sequence (T_c). The 20-sample sequence showed better performance than the others and the best performance was using 5 windows.

F. Comparing different attention schemes

Figure 8 compares the performance of the MSA block with a different number of heads using dimension sizes of 8 and 16, i.e., the size of each attention head for query/key/value vectors. The results showed that using 2 heads each of size 8 gave the best results. This is because the MI-EEG dataset has a limited number of samples, which requires a light MSA layer to converge well. In addition, the temporal data entered into the MSA layer has a light representation, i.e., sequence length = 16 and embedding size = 32, which requires few parameters to train.

In Table 3, we compare the performance of the proposed model using three attention mechanisms: MSA [27], SE [28], and CBAM [29]. The number of MSA heads was set to 2 and the head size was set to 8 and 16. The reduction ratio for both SE and CBAM was experimentally set to 8. The results showed that all attention mechanisms improved the performance of the ATCNet model while the best performance was achieved by MSA, indicating that MSA is more suitable for a two-dimensional EEG representation.

G. Comparison to recent studies

Table 4 summarizes the accuracy and κ-score of the proposed ATCNet model using the BCI-2a dataset and its comparison with the reproduced EEGNet [12], EEG-TCNet [22], and TCNet_Fusion [23], as these models have some similarities with the proposed model. The results of the reproduced models are based on the hyperparameters identified in the original articles, while pre-processing, training, and evaluation followed the same procedure defined in this research. Table 4 shows that ATCNet performed better than EEGNet, EEG-TCNet, and TCNet_Fusion for all subjects with an average accuracy of 85.38% and a κ-score of 0.81. This represents a 4.71% increase in accuracy over these models. In addition, the proposed model achieved the best standard deviation among subjects with a value of 9.08%, indicating that the accuracy is more robust over all subjects. The average confusion matrices of ATCNet and the reproduced models are shown in Figure 9. ATCNet demonstrated an improvement in MI decoding for all MI classes compared to the other models.

Table 5 presents the reported overall accuracy and κ-score of recent studies in the subject-specific MI-EEG classification using the BCI-2a dataset. The proposed ATCNet model performs better than the recent studies using raw EEG data and without pre-processing. In addition to the subject-specific (subject-dependent) results, we evaluated the performance of the proposed model in subject-independent classification, which is a measure of the model's generalization ability. The proposed model achieved the best subject-independent performance on the BCI-2a dataset as shown in Table 6.

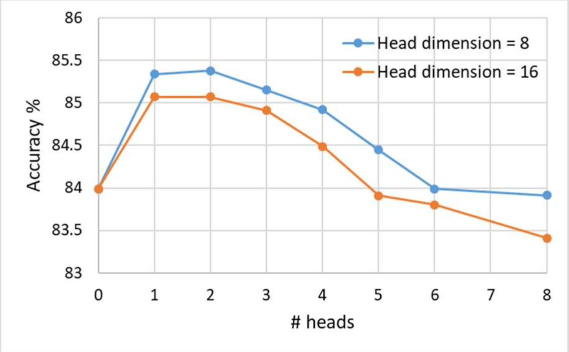

Figure 8. Accuracy on BCI-a2 as a function of the number of attention heads using head sizes of 8 and 16. Reducing head size as well as the number of heads showed better performance due to the small size of the dataset and its light representation. The best performance was using two 8-size heads.

Table 3. ATCNet model performance using different attention schemes: multi-head self-attention (MSA) with 8 and 16 head size, squeeze-and-excitation (SE), and convolutional block attention module (CBAM).

Attention mechanism	Accuracy %	κ-score
No Attention	83.84	0.784
MSA-8	**85.38**	**0.805**
MSA-16	85.07	0.801
SE	84.07	0.788
CBAM	84.30	0.791

Table 4. Performance (accuracy (%) and κ-score) comparison of subject-specific classification using BCI-2a dataset for the proposed model with other reproduced models.

	Proposed (ATCNet)		EEGNet [12]		EEG-TCNet [22]		TCNet Fusion [23]	
Sub.	%	κ	%	κ	%	κ	%	κ
1	88.5	0.85	88.5	0.85	84.0	0.79	86.1	0.81
2	70.5	0.61	66.0	0.55	66.3	0.55	66.0	0.55
3	97.6	0.97	95.1	0.94	94.1	0.92	93.4	0.91
4	81.0	0.75	73.6	0.65	72.6	0.63	72.6	0.63
5	83.0	0.77	75.4	0.67	76.0	0.68	79.9	0.73
6	73.6	0.65	64.2	0.52	62.9	0.50	66.7	0.56
7	93.1	0.91	90.3	0.87	89.9	0.87	90.3	0.87
8	90.3	0.87	85.8	0.81	84.7	0.80	85.8	0.81
9	91.0	0.88	86.5	0.82	85.4	0.81	85.4	0.81
Mean	**85.4**	**0.81**	80.6	0.74	79.6	0.73	80.7	0.74
St.D.	**9.1**	**0.12**	11.1	0.15	10.7	0.14	10.1	0.13

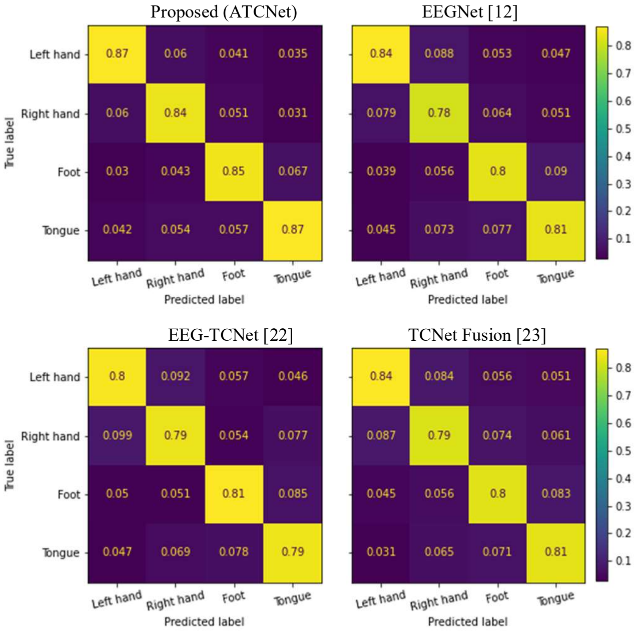

Figure 9. Average confusion matrices of the proposed ATCNet and the reproduced EEGNet, EEG-TCNet, and TCNet_Fusion models. The results showed that ATCNet improved MI decoding for all MI tasks compared to equivalent models.

Table 5. Subject-specific performance on the BCI-2a dataset using the same original competition division (hold-out approach: 50% training trials and 50% test trials). Accuracy (%) and κ-score are the averages for all subjects.

Method	Accuracy	κ-score
Shallow CNN [32]	74.31	0.66
EEGNet: CNN [12]*	80.59	0.74
DBN-AE [17]	71.0	_
Multi-layer-CNN and MLP [18]	75.0	_
EEG-TCNet: CNN and TCN [22]*	79.55	0.73
Attention multi-scale CNN [13]	79.9	_
TCNet_Fusion: multi-layer CNN + TCN [23]*	80.67	0.74
Attention-inception CNN & LSTM [10]	82.84	_
Attention multi-branch CNN [9]	82.87	0.772
ATCNet: Attention-CNN and TCN (Proposed)	**85.38**	**0.805**

** Reproduced.*

Table 6 Subject-independent performance on the BCI-2a dataset using leave-one-subject-out (LOSO) cross-validation. Accuracy (%) and κ-score are the averages for all subjects.

Method	Accuracy	κ-score
Attention graph convolutional network [8]	60.1	-
Multi-layer-CNN and AE [18]	55.3	-
EEGNet: CNN [12]*	68.79	0.584
Attention multi-branch CNN [9]	69.10	-
EEG-TCNet: CNN and TCN [22]*	69.52	0.594
TCNet_Fusion: multi-layer CNN + TCN [23]*	70.58	0.608
ATCNet: Attention-CNN and TCN (Proposed)	**70.97**	**0.613**

** Reproduced.*

IV. Conclusion

This study proposed a novel attention-based temporal convolutional network (ATCNet) for EEG-based motor imagery classification. ATCNet consists of three main blocks: the convolutional (CV) block, to encode the raw MI-EEG signal into a compact temporal sequence, the multi-head self-attention (AT) block, to highlight the most effective information in the temporal sequence, and the temporal convolutional (TC) block, to extract high-level temporal features from the temporal sequence. This study also implemented a convolutional-based sliding window (SW) combined with CV block to improve the performance of MI classification efficiently by parallelizing the process. The ablation analysis showed that each block in the ATCNet model made a significant contribution to the performance of the ATCNet model. The AT block increased overall accuracy by 1.54%, the SW by 2.28%, and the TC by 1.04% compared to using the CV block only. The proposed ATCNet model outperformed recent techniques in MI-EEG classification using the BCI-2a dataset with an accuracy of 85.38% and 70.97% for the subject-dependent and subject-independent modes, respectively. These high results came with a relatively small number of parameters (115.2K), which makes ATCNet applicable to industrial devices with limited resources. The proposed model demonstrated a powerful ability to extract MI features from a raw EEG signal without artifact removal and with minimal pre-processing using a limited-size and challenging dataset. ATCNet showed an overall improvement in EEG decoding for all MI classes and all subjects in the BCI-2a dataset proving that ATCNet can learn to find generic EEG representations across classes and subjects.

For future work, the proposed model can be further improved by using attention mechanisms in several domains. Effective MI information occurs in the EEG data at specific time intervals, channel locations, and frequency bands; Thus, developing a DL model that automatically attends to the most important information in these domains is a promising direction for improving the performance of MI-EEG classification. The proposed model can also be refined using preprocessing methods to remove artifacts and deep generative models to increase the size of the dataset.

References

[1] I. Ahmed, G. Jeon, and F. Piccialli, "From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where," *IEEE Trans. Ind. Informatics*, vol. 18, no. 8, pp. 5031–5042, 2022.

[2] H. Altaheri *et al.*, "Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: a review," *Neural Comput. Appl.*, pp. 1–42, 2021.

[3] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b," *Front. Neurosci.*, vol. 6, p. 39, 2012.

[4] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic fruit classification using deep learning for industrial applications," *IEEE Trans. Ind. Informatics*, vol. 15, no. 2, pp. 1027–1034, 2018.

[5] H. Altaheri, M. Alsulaiman, and G. Muhammad, "Date Fruit Classification for Robotic Harvesting in a Natural Environment Using Deep Learning," *IEEE Access*, vol. 7, no. 1, pp. 117115–117133, Aug. 2019.

[6] I. Ahmed, S. Din, G. Jeon, and F. Piccialli, "Exploring deep learning models for overhead view multiple object detection," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5737–5744, 2019.

[7] M. Qamhan, H. Altaheri, A. H. Meftah, G. Muhammad, and Y. A. Alotaibi, "Digital Audio Forensics: Microphone and Environment Classification Using Deep Learning," *IEEE Access*, vol. 9, pp. 62719–62733, 2021.

[8] D. Zhang, K. Chen, D. Jian, and L. Yao, "Motor imagery classification via temporal attention cues of graph embedded EEG signals," *IEEE J. Biomed. Heal. informatics*, vol. 24, no. 9, pp. 2570–2579, 2020.

[9] G. A. Altuwaijri, G. Muhammad, H. Altaheri, and M. Alsulaiman, "A Multi-Branch Convolutional Neural Network with Squeeze-and-Excitation Attention Blocks for EEG-Based Motor Imagery Signals Classification," *Diagnostics*, vol. 12, no. 4, p. 995, 2022.

[10] S. U. Amin, H. Altaheri, G. Muhammad, M. Alsulaiman, and A. Wadood, "Attention-Inception and Long Short-Term Memory-based Electroencephalography Classification for Motor Imagery Tasks in Rehabilitation," *IEEE Trans. Ind. Informatics*, 2022.

[11] S. U. Amin, H. Altaheri, G. Muhammad, M. Alsulaiman, and W. Abdul, "Attention based Inception model for robust EEG motor imagery classification," in *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2021, pp. 1–6.

[12] 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, p. 56013, 2018.

[13] D. Li, J. Xu, J. Wang, X. Fang, and J. Ying, "A Multi-Scale Fusion Convolutional Neural Network based on Attention Mechanism for the Visualization Analysis of EEG Signals Decoding," *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2020.

[14] S. Kumar, R. Sharma, and A. Sharma, "OPTICAL+: a frequency-based deep learning scheme for recognizing brain wave signals," *PeerJ Comput. Sci.*, vol. 7, p. e375, 2021.

[15] T. Luo and F. Chao, "Exploring spatial-frequency-sequential relationships for motor imagery classification with recurrent neural network," *BMC Bioinformatics*, vol. 19, no. 1, p. 344, 2018.

[16] J. Xu, H. Zheng, J. Wang, D. Li, and X. Fang, "Recognition of EEG signal motor imagery intention based on deep multi-view feature learning," *Sensors*, vol. 20, no. 12, p. 3496, 2020.

[17] A. Hassanpour, M. Moradikia, H. Adeli, S. R. Khayami, and P. Shamsinejadbabaki, "A novel end-to-end deep learning scheme for classifying multi-class motor imagery electroencephalography signals," *Expert Syst.*, vol. 36, no. 6, p. e12494, 2019.

[18] S. U. Amin, M. Alsulaiman, G. Muhammad, M. A. Mekhtiche, and M. S. Hossain, "Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion," *Futur. Gener. Comput. Syst.*, vol. 101, pp. 542–554, 2019.

[19] M.-A. Li, J.-F. Han, and L.-J. Duan, "A Novel MI-EEG Imaging With the Location Information of Electrodes," *IEEE Access*, vol. 8, pp. 3197–3211, 2019.

[20] T. Liu and D. Yang, "A Densely Connected Multi-Branch 3D Convolutional Neural Network for Motor Imagery EEG Decoding," *Brain Sci.*, vol. 11, no. 2, p. 197, 2021.

[21] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv Prepr. arXiv1803.01271*, 2018.

[22] T. M. Ingolfsson, M. Hersche, X. Wang, N. Kobayashi, L. Cavigelli, and L. Benini, "EEG-TCNet: An Accurate Temporal Convolutional Network for Embedded Motor-Imagery Brain-Machine Interfaces," *arXiv Prepr. arXiv2006.00622*, 2020.

[23] Y. K. Musallam *et al.*, "Electroencephalography-based motor imagery classification using temporal convolutional network fusion," *Biomed. Signal Process. Control*, vol. 69, p. 102826, 2021.

[24] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next," *arXiv Prepr. arXiv2201.05624*, 2022.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv Prepr. arXiv1409.0473*, 2014.

[26] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv Prepr. arXiv1508.04025*, 2015.

[27] A. Vaswani *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[29] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[30] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "BCI Competition 2008–Graz data set A," *Inst. Knowl. Discov. Graz Univ. Technol.*, vol. 16, pp. 1–6, 2008.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[32] R. T. Schirrmeister *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapp.*, vol. 38, no. 11, pp. 5391–5420, 2017.

[33] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv Prepr. arXiv1607.06450*, 2016.

Hamdi Altaheri received the master's degree in computer engineering from King Saud University, in 2019, where he is currently pursuing the Ph.D. degree with the Department of Computer Engineering, College of Computer and Information Sciences. His research interests include computer vision, bioengineering, machine learning, and deep learning.

Ghulam Muhammad is a Professor in the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. Prof. Ghulam received his Ph.D. degree in Electrical and Computer Engineering from Toyohashi University and Technology, Japan in 2006, M.S. degree from the same university in 2003. He received his B.S. degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology in 1997. He was a recipient of the Japan Society for Promotion and Science (JSPS) fellowship from the Ministry of Education, Culture, Sports, Science and Technology, Japan. His research interests include AI, machine learning, image and speech processing, and smart healthcare. He holds two U.S. patents. Prof. Ghulam has authored and co-authored more than 300 publications including IEEE/ACM/Springer/Elsevier journals, and flagship conference papers.

Mansour Alsulaiman is a Professor in the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. Prof. Mansour the Ph.D. degree from Iowa State University, USA, in 1987. His research areas include automatic speech/speaker recognition, automatic voice pathology assessment systems, computer-aided pronunciation training system, and robotics. He was the Editor-in-Chief of the King Saud University Journal Computer and Information Systems. He is the director of Center of Smart Robotics Research at King Saud University.