Brain Decoding over the MEG Signals Using Riemannian Approach and Machine Learning

Zeynep Ozer, Onursal Cetin, Kutlucan Gorur, Feyzullah Temurtas

Abstract—**Brain decoding is an emerging approach for understanding the face perception mechanism in the human brain. Face visual stimuli and perception mechanism are considered as a challenging ongoing research of the neuroscience field. In this study, face/scrambled face visual stimulations were implemented over the sixteen participants to be decoded the face or scrambled face classification using machine learning (ML) algorithms via magnetoencephalography (MEG) signals. This noninvasive and high spatial/temporal resolution signal is a neurophysiological technique which measures the magnetic fields generated by the neuronal activity of the brain. The Riemannian approach was used as a highly promising feature extraction technique. Then Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN) were employed as deep learning algorithms, Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) were implemented as shallow algorithms. The improved classification performances are very encouraging, especially for deep learning algorithms. The LSTM and GRU have achieved 92.99% and 91.66% accuracy and 0.977 and 0.973 of the area under the curve (AUC) scores, respectively. Moreover, CNN has yielded 90.62% accuracy. As our best knowledge, the improved outcomes and the usage of the deep learning on the MEG dataset signals from 16 participants are critical to expand the literature of brain decoding after visual stimuli. And this study is the first attempt with these methods in systematic comparison. Moreover, MEG-based Brain-Computer Interface (BCI) approaches may also be implemented for Internet of Things (IoT) applications, including biometric authentication, thanks to the specific stimuli of individual's brainwaves.**

Index Terms—**Magnetoencephalography, Brain Decoding, Riemannian Approach, Deep Learning.**

ZEYNEP ÖZER, is with Department of Management Information Systems University Bandirma Onyedi Eylul University, Balıkesir, Turkey,(e-mail: zozer@bandirma.edu.tr)

https://orcid.org/0000-0001-8654-0902

ONURSAL ÇETİN, is with Department of Electrical and Electronics Engineering University of Bandirma Onyedi Eylul University, Balıkesir, Turkey,(e-mail: ocetin@bandirma.edu.tr)

https://orcid.org/0000-0001-5220-3959

KUTLUCAN GÖRÜR, is with Department of Electrical and Electronics Engineering University of Bandirma Onyedi Eylul University, Balıkesir, Turkey,(e-mail: kgorur@bandirma.edu.tr)

https://orcid.org/0000-0003-3578-0150

FEYZULLAH TEMURTAŞ, is with Department of Electrical and Electronics Engineering University of Bandirma Onyedi Eylul University, Balıkesir, Turkey,(e-mail: ftemurtas@bandirma.edu.tr)

https://orcid.org/0000-0002-3158-4032

Manuscript received July 17, 2022; accepted June 9, 2023.
DOI: 10.17694/bajece.1144279

I. Introduction

THE brain decoding has obtained great attention from scientific communities in medical applications and is observed as one of the primary goals of the brain analysis literature [1]. Face perception mechanism in individuals is generated by a sequence of cortical activities [2,3]. Neuropsychology and cognitive neuroscience research fields require magnetoencephalography (MEG) and electroencephalography (EEG) signals to decode these brain maps [4]. Different experiments can be implemented to arouse these brain patterns, such as visual stimuli. The brain reacts to different responses for different visual stimuli [4]. If the subject is stimulated by a visual cue, then the related brain activity is recorded from multiple noninvasive sensors. Then each recorded data is named as a trial [5]. In this study, face/scrambled face are used as a visual stimulus; during the same time, MEG signals are collected over the brain activities. MEG signal is a neurophysiological way of measuring magnetic fields generated by neural electrical activities [6]. These signals have some advantages compared to the EEG signals, functional magnetic resonance imaging (fMRI), and positron emission tomography (PET) methods because of the alleviated effects by cerebrospinal fluid, skull, and skin [3]. Moreover, MEG signals present great spatial and temporal resolution [7].

The low prediction performance in multivariate brain decoding is generally caused by the low signal-to-noise ratios (SNRs), high dimensionality recordings of the scalp, and cross-subject variations [8]. Recently machine learning (ML) algorithms offer these problems a very promising approach in signal processing techniques to recognize the activated brain patterns using noninvasive MEG signals [5,9-11]. In this research study, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) were implemented as deep learning algorithms when Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) were used as traditional (shallow) machine learning algorithms. CNN is considered as a state of the art machine learning algorithm especially used for computer vision, natural language processing, and pattern recognition. CNN has convolution layers to extract features of the signal-based images. These features consist of edges, lines, or corners [12]. LSTM is carried out in the prediction of sequential data. The key point of LSTM is that it can remember input for a long time while estimating outputs [13]. The GRU is another applied popular ML algorithm based on the Recurrent

Neural Network (RNN). The reason for the popularity is considered as computation cost and a simple model of topology. This technique is assembled into a single "update gate" with forget and input gates and mounted in the cell state and hidden state [14]. Furthermore, LDA is to project the dataset for classification in a supervised manner. This projected-based method aims to find the best projection direction during the classification task [15]. On the other hand, QDA assumes not equality of the covariance matrices that gain a quadratic decision boundary for two-class problems [16].

The Riemannian approach was implemented in the feature extraction step that enables direct manipulation of multichannel MEG signals to covariance matrices. Then these matrices were employed as features. The Riemannian approach is known as extremely competitive and superior to the other feature extraction techniques, including the Common Spatial Pattern [5,17]. The simultaneous MEG signal recordings were obtained from 16 participants who perform trials during the face and scrambled face visual stimuli to reveal the patterns over the brain dynamics [5]. After that, the Riemannian approach was employed for the feature extraction process [5]. In this study, deep learning algorithms, LSTM and GRU has achieved 92.99% accuracy and 0.977 of AUC (area under the curve) score, 91.66% accuracy, and 0.973 AUC score, respectively. Moreover, CNN has yielded 90.62% accuracy with 0.959 AUC score among the deep learning approaches. Then LDA has determined 78.23% accuracy and 0.861 AUC when QDA has obtained a classification accuracy of 72.24% and 0.796 of AUC score. These results show that LSTM, GRU, and CNN have noticable performances on the MEG signals compared to the previous research study (80.85% accuracy and 0.81 AUC with Deep Neural Network) [5]. Moreover, LDA has also offered improved performance than Support Vector Machine (78.01% accuracy) and the other traditional classifiers in the same study [5]. However, the satisfactory accuracy (79%) of the Generalized Regression Neural Network (GRNN) was not achieved by LDA and QDA results [11]. MEG-based BCI systems have high time resolution and high uniqueness for individuals [4,8-10]. These features can provide effectiveness for IoT applications, such as biometric authentication.

The rest of the article was organized as follows; the method chapter defines dataset description of the dataset details, preprocessing, feature extraction step, evaluation metrics, and machine learning algorithms. Furthermore, the results and discussion chapter explains the performance outcomes of the MLs and statements of the findings. Then the conclusion chapter is to point out the discriminations of the study among the literature of brain decoding in terms of the ML estimations.

II. Material and Method

A. Definition of the dataset

Magnetoencephalography signals were recorded with an Elekta Neuromag VectorView system from 18 participants; thereby, the dataset was created by Henson et al. [18]. The triple sensor group, consisting of a magnetometer and two gradiometers, is located at 102 positions. In total, 306 sensors record the magnetic field caused by brain currents. The z (radial) component of the magnetic field is measured by the magnetometer, while the x and y spatial derivative is measured by the gradiometer. Details of the measuring system are provided by Henson et al. [18].

The dataset, used in this study, was modified within the scope of the [19-20]. The modified dataset encompasses approximately 588 trials for each 16 subjects. Visual stimuli of faces and scrambled faces were randomly presented to the subjects for 1 second, and for each stimulation, subjects were rested for 0.5 seconds. Each trial consists of 1.5 seconds of MEG recording was sampled at 250 Hz from 306 channels. Thus, a total of 9414 trials were presented in a random order [5].

The flowchart showing the overall workflow from MEG recordings to the brain decoding process is given in Fig. 1. A series of operations carried out for this study can be listed as follows: Pre-processing of signals using a bandpass filter, source extraction by spatial filtering, extracting feature vector, classification of the feature vector.

The brain computer interfaces (BCIs) are used to perform various tasks (control, communication, biometric authentication) via processing of brain signals [21]. In the preprocessing phase, the signal was first high-pass filtered at 1Hz. The first 0.5 seconds of the signals were discarded when contemplating the rest time of the subjects. With the start of the feature extraction process, a bandpass filter having 1Hz-20Hz cut-off frequency points is used. Then, by applying a spatial filter to the signal, the dimensionality is reduced, and the signal-to-noise ratio is increased [5].

The Riemannian approach allows the direct manipulation of multichannel MEG signals to covariance matrices and subspaces with proper and special geometry, as defined by Yger et al. [22]. Through the Riemannian geometry, the 2176 features of the dataset were obtained using tangent space representation from the MEG covariance matrices at the end of the feature extraction process [23]. Tangent space is a vector field that allows all Euclidean statistical methods, and Riemannian metrics are more suitable for flat vector spaces rather than conventional Euclidean metrics. Thus, this approach can be applied to the MEG-based brain-computer interface, especially for the implementation of the classifier, and feature representation. Detailed computational issues about the Riemannian geometry can be found in [22].

B. Evaluation metrics for performances

Evaluation metrics are important factors to show the machine learning algorithm performances in the dataset. The suitable evaluation metrics are important keys to discriminate the optimal MLs [24]. In this study, accuracy (ACC), sensitivity (SENS), specificity (SPEC), and the area under the curve (AUC) score were employed as evaluation metrics with k-fold cross-validation technique. In this technique, the dataset is divided into k-subsets. Then the ML is trained with k-1 subsets, and the other one subset is tested. The process is repeated k times so that each subset sample will be trained and tested. After that, the average classification metrics are obtained for each ML algorithm [12]. The formulas for accuracy were presented in Eqs. (1-3) [25]:

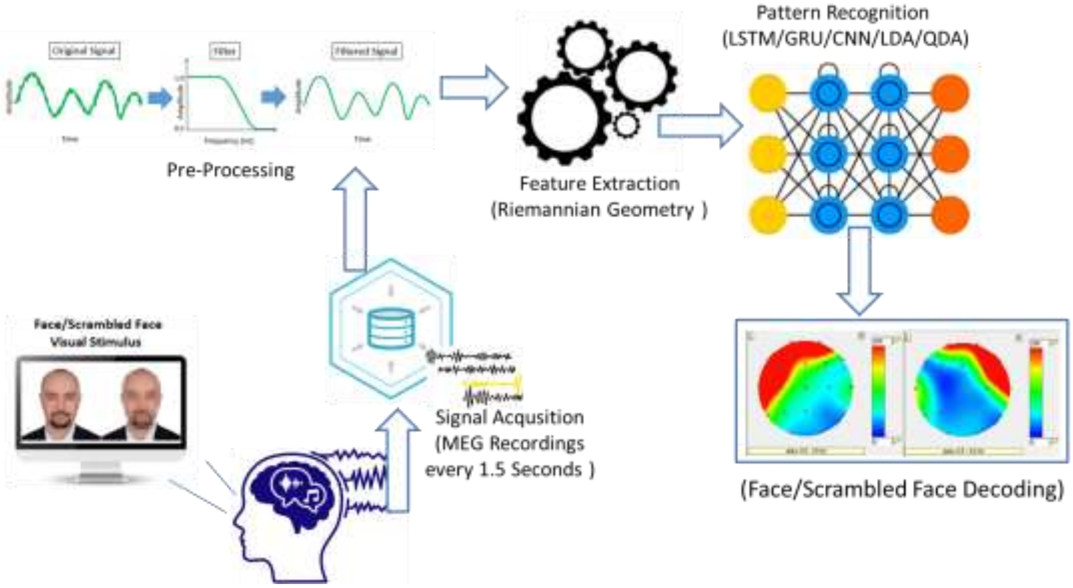

Fig.1. The experiment of MEG recordings and brain decoding process

$$Accuracy(N) = \frac{\sum_{i=1}^{|N|} estimate(n_i)}{|N|}, \quad n_i \in N \tag{1}$$

$$Estimate(n) = \begin{cases} 1, & if\ estimate(n) = cn \\ 0, & otherwise \end{cases} \tag{2}$$

$$Classification\ Accuracy(ML) = \frac{\sum_{i=1}^{|k|} Accuracy(N_i)}{|k|} \tag{3}$$

where N refers the classified (test) dataset, cn defines the class of the value of *n, Estimate(n)* describes the classification result of n, and then the k value is named for the k-fold cross-validation [12].

Different metrics reveal the different characteristics of the ML algorithms induced by the processing [26]. Therefore, it may help easier to make the comparison and analysis of ML algorithms in the robustness observation. In general, the sensitivity and specificity are well-known evaluation metrics for performance analysis of ML algorithms. Thus, they are described as the following equations [27]:

$$Sensitivity = \frac{TP}{TP + FN} \tag{4}$$

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

where: True Positive (TP): The number of face pattern decisions which are targeted as face pattern, True Negative (TN): The number of scrambled face pattern decisions which are targeted as scrambled face pattern, False Positive (FP): The number of scrambled face pattern decisions which are targeted as face pattern, False Negative (FN): The number of face pattern decisions which are targeted as scrambled face pattern.

Thanks to the AUC evaluation metrics, the classification results are presented across the interval of 0-1 scores under the curve of false positive rates and true positive rates. The higher AUC value means better classifier performance. This popular ranking type metric is to prove that the prediction and diagnostic ability of MLs are noteworthy (AUC>0.9) and good discrimination (0.8≥AUC>0.7) if the AUC scores are found in the range of the stated values [28].

C. Linear discriminant analysis and quadratic discriminant analysis Most

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) machine learning methods are both based on the fundamentals of statistical and probabilistic learning. Basically, when LDA is used for linear classification, QDA is employed for quadratic decision problems of the classification process [16]. Kernel Fisher's Discriminant analysis has derived the LDA method that is a type of projection technique. LDA is implemented to classify the dataset in the manner of reducing the dimension. The aim of LDA is to maximize the between-class distance and to minimize within-class distance. If the class samples are defined as C1 and C2, LDA finds the projection direction (w) for maximum separability of the spatial pattern [15-29]. The related Equations are presented below:

$$z = w^T x \tag{6}$$

where x (data samples) are employed to be projected onto *w*. The graphical presentation of LDA is shown in Fig. 2 [15].

where *m1* to ***m1*** describes the means of samples in class *C1* before and after the projection process, respectively. Therefore, $\boldsymbol{m1} \in \Re^d$ defines the multi-dimension, and $m1 \in \Re$ means the projected dimension. Then $m2$ and $\boldsymbol{m2}$ have a similar manner for class *C2*. The samples of scattered dataset around the means are presented as s_1^2 and s_2^2. Then the samples of training dataset are defined as $X\{x^t, r^t\}$:

$$X\{t\} = \begin{cases} r^t = 1, & x^t \in C1 \\ r^t = 0, & x^t \in C2 \end{cases} \tag{7}$$

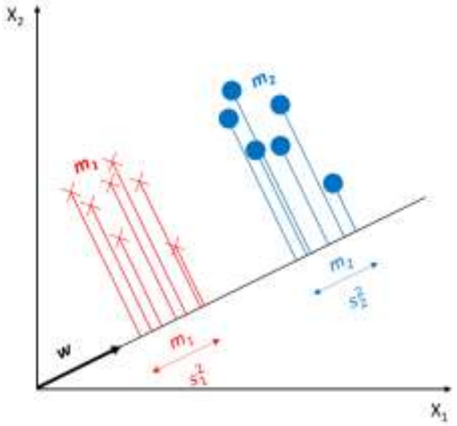

Fig.2. Classification process in LDA via the projection of data samples

$$J(w) = \frac{w^T S_B w}{w^T S_W w} = \frac{|w^T(m1 - m2)|^2}{w^T S_W w} \quad (8)$$

where S_B and S_W are named as the between-class scatter matrix and within-class scatter matrix in $J(w)$, respectively [15].

QDA assumes that the equality of the covariance matrices is not necessary (as shown in Eq. 9) as LDA. This feature yields the advantage to the QDA to be used in the decision boundary of quadratic classification [16].

$$\Sigma_1 \neq \Sigma_2 \quad (9)$$

D. *Recurrent neural network*

The use of artificial neural networks (ANN) in machine learning applications is very common. Over time, many different ANN models have been developed in line with needs. Accordingly, ANNs are specialized to process different types of data. Such as Convolutional Neural Networks are specialized for matrix type data like image. On the other hand, Recurrent Neural Networks (RNN) have been developed to process array data. Traditional feed-forward ANNs take into account existing samples to which they are exposed as input. RNNs, apart from this, apply the samples they perceive over time, as well as the existing ones.

An input sequence is given as $[x_1, x_2, \ldots, x_k]$ with $x_i \in \mathbb{R}^d$. Different examples can have different sequence lengths. Therefore, the k value may vary. In each step of the RNN model, a hidden state is generated as an array $[h_1, h_2, \ldots, h_k]$. Activation of hidden state at time *t* is calculated as a function of the current input x_t and previously hidden state h_t *-1*. This process can be expressed as follows:

$$h_t = f(x_t, h_{t-1}) \quad (10)$$

Unlike traditional feed-forward ANN, RNNs have a repeat layer. By means of this layer, the state information generated by the feed-forward network is stored and re-applied to the network with the input information. That is, RNNs have a memory that holds what has been calculated so far [30]. Fig. 3 shows an exemplary RNN network unit and closed notation of this architecture.

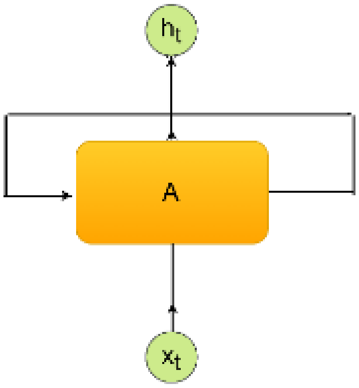

Fig.3. Closed notation of Recurrent Neural Network

1) *Long short-term memory network*

Long Short-Term Memory networks, often called LSTM, are a special type of RNN capable of learning long-term dependencies [21]. This model, which was first proposed in the mid-90s [31], is widely used today. While processing the sequences in RNNs, it is aimed to store and transfer ANN status information. However, it is unlikely to be transferred without disturbing long-term dependencies as a result of transferring the state information by continuously processing it. In other words, while short-term dependencies are transferred very successfully, there are problems in transferring long-term dependencies. LSTMs are designed to address long-term dependency problems.

All RNN networks consist of modules that are repeating like a chain. In standard RNNs, each of these modules usually consists of a *tanh* function or a similar function. The feature that distinguishes LSTMs from standard RNNs is that the internal structure of this module consists of 4 separate structures that interact with each other.

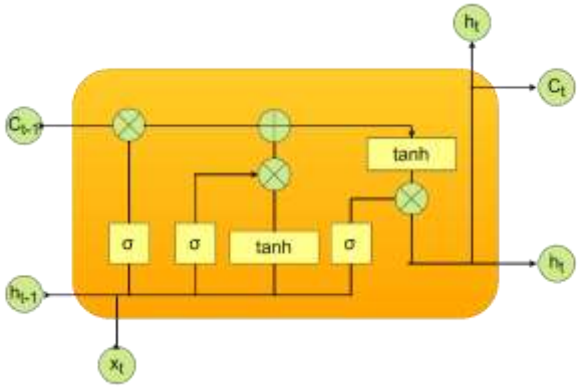

Fig.4. LSTM structure

LSTM module consists of 3 separate gates, as shown in Fig.4. These are the forget gate, the input gate, and the output gate, respectively. Forget gates decides how much of the information will be forgotten and how much of it should be transferred to the next stage. For this process, it uses the sigmoid function, which produces a value between 0 and 1. 0 means that the information will never be transmitted, while 1 means that all must be transmitted.

The next step is to decide what information should be stored. For this, the input layer firstly decides which values should be updated. Then the $tanh$ function forms a vector of the new candidate values of the memory cell defined as $\tilde{C}_t$. Then these two processes are combined. This process is expressed mathematically as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{11}$$

$$\tilde{C} = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{12}$$

In general, W is named as the weight vector, b is called the bias term, σ describes the sigmoid activation function for non-linearity, x_t is used for the input sequence, h_{t-1} is implemented as the output of the neuron at time $t-1$ for feedback into the neuron. Furthermore, i_t, f_t and o_t are defined as the input, forget and output gate, respectively. After this process, the new status information of the memory cell is calculated. The new status information is calculated as follows:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{13}$$

Finally, the output of the system is calculated. This is done at the output gate. The output of the system h_t can be calculated as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{14}$$

$$h_t = o_t * tanh(C_t) \tag{15}$$

The LSTM architecture used in this study consists of two learnable layers. In this study, MEG data is adjusted to 128x17 dimensions and applied to the input layer. The LSTM layer contains 100 units. Dropout value is set to 0.1. At the last stage, there is a fully connected layer containing one neuron. The training process is carried out in 200 epochs. Also, Adam optimizer was used in the training phase.

2) *Gated recurrent unit*

The main difference of the Gated Recurrent Unit (GRU) network from the LSTM network is that each module consists of 2 gates instead of 3, as shown in Fig. 5. A GRU module consists of an update gate and a reset gate. The update gate decides how much of the past information should be transmitted, while the reset gate, on the contrary, decides how much of the past information should be discarded.

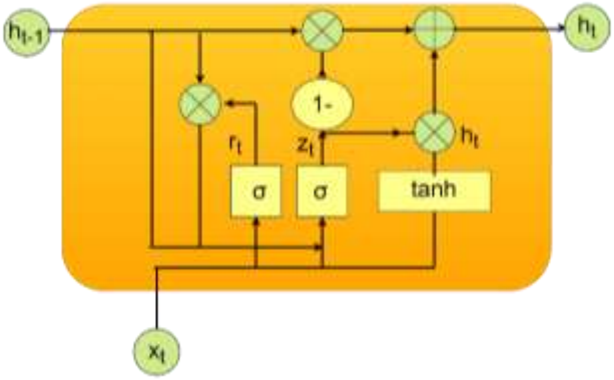

Fig.5. GRU structure

The sigmoid process representing the z_t update gate and the $\tilde{h}_t$ reset operation, GRU can be expressed mathematically as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{16}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{17}$$

$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{18}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{19}$$

GRU architecture also consists of two learnable layers, like LSTM. In the input layer, the dimensions are set to 128x17. The GRU layer contains 32 units. The dropout value is set to 0.1. Finally, there is a fully bonded layer connected to a neuron. The training process is carried out in 105 epochs. Also, Adam optimizer was used in the training phase.

E. *Convolutional neural networks*

CNN is a typical multi-layered neural network [32] that simulates the organization of the animal visual cortex [33]. It is widely used in image-related applications [31,34]. The operation of CNN models takes place in two stages as feature extraction and classification of these features in fully connected layers. CNN architectures are often created by combining convolution, pooling, and fully connected layers.

The convolution layer is the most important structure that makes up CNN. This layer is in principle based on the idea that an image of an object can be in any region above the image. Accordingly, neurons are attached to only a small part of the input and extend across the entire depth of the input. The filter size and number of maps produced are used to define this layer. Filters aim to extract different features related to lines, corners, and edges on the input images [12]. These filters containing pixel values are shifted on the image. During the sliding process, the filter values are multiplied by the values of the image. Then the obtained values are summed and a net result is produced. This process is applied to the whole image and feature maps are obtained. Calculation of feature map values can be expressed as follows:

$$y_l = \sum_{n=0}^{N-1} x_n h_{l-n} \tag{20}$$

where *y* is the feature map, *x* is signal, *h* is the filter, *N* is the number of elements in *x*, and the *nth* vector variable subscripts indicate the subscripts.

Another important structure that forms CNN is the pooling layer. There are different types of this layer commonly used in the literature, such as average pooling and max pooling. In this study, max-pooling was used. In the max-pooling process, the image is divided into blocks that do not overlap, and the biggest value of each block is taken. Therefore, calculation costs and overfitting possibilities are greatly reduced.

Another structure commonly used in CNN architectures is a fully connected layer. This layer is a typical artificial neural network layer. They have connections with all neurons before and after it.

Most of the problems that CNN is trying to solve are not linear. On the other hand, operations such as matrix multiplication and addition are linear. So, the non-saturating activation function is commonly used in CNN to provide non-linearity. This process can be expressed mathematically as follows:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & otherwise \end{cases} \quad (21)$$

The CNN architecture used in this study consists of four learnable layers, as seen in Fig. 6. In this study, 2D CNN architecture was used. MEG data is adjusted to 128x17 dimensions and applied to the input layer. The first convolutional layer takes place after the input layer. The filter number of this layer is 512, and the kernel size is 4. This layer is followed by the max-pooling layer, which is a size of 2. Then there is the second convolution layer. The values of this layer are the same as the first convolutional layer. After the second convolutional layer, there is a max-pooling layer with a size of 2. Then there is the fully connected layer of 50 neurons. Dropout is applied in this layer. The dropout value is 0.2. Finally, there is a fully connected layer with a single neuron.

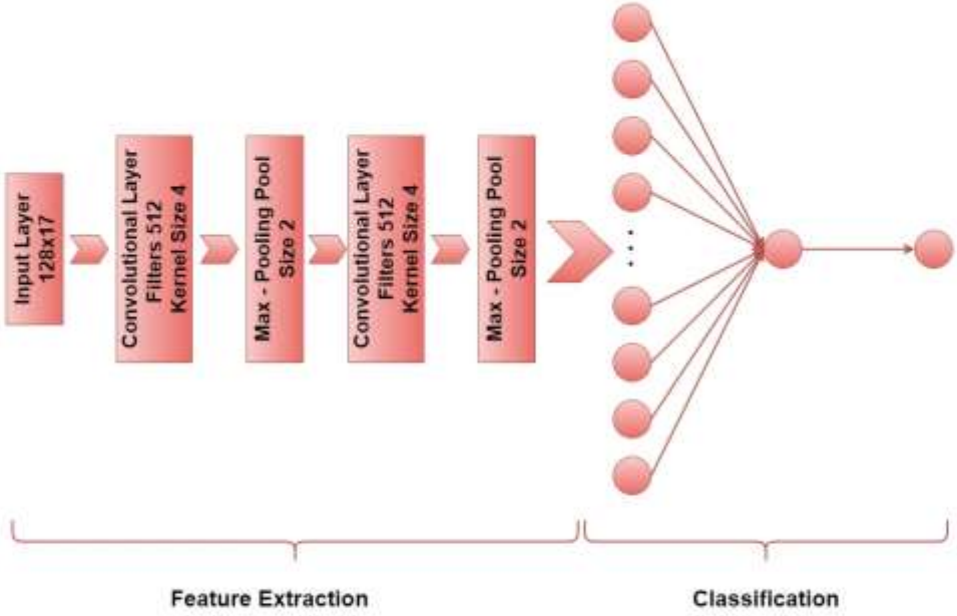

Fig.6. CNN architecture

III. Results and Discussion

In this study, a binary classification process was carried out using MEG signals. Signals are grouped as a face or scrambled face. The dataset consists of 9414 stimuli belonging to 16 individuals. Approximately 588 stimuli were shown to each individual, and recording was performed through 306 channels. The noise-containing MEG signals are filtered as described in the sections above to extract 2176 features for each stimulus. For the classification process, 5 different supervised classifiers, LDA, QDA, LSTM, GRU, and CNN were used. The performance of each machine learning algorithm was compared by taking into account classification performance (ACC), Area under the curve (AUC) score, sensitivity (SENS), and specificity (SPEC) metrics. Results were given with 10k cross-validation for each algorithm. In addition, the results were compared with other studies carried out with the same dataset.

Values obtained as a result of 10k cross-verification are shown in Table 1. With an accuracy value of 92.99%, the best classification performance belongs to LSTM. Similarly, the best scores in specificity and AUC, respectively, with 92.33% and 0.977% belong to LSTM. LSTM and GRU showed quite similar performances in sensitivity value. Sensitivity was 93.64% for LSTM, while 93.73% was obtained in GRU. GRU's performance and specificity values were 91.66% and 89.59%, respectively. CNN showed slightly lower performance in accuracy, sensitivity, and AUC values compared to RNN-based approaches. For all three metrics, 90.62%, 91.16%, and 0.959% values were obtained, respectively. On the other hand, specificity performed quite close to GRU. The specificity for CNN was 90.08%. On the other hand, QDA displayed the worst classification performance with a performance value of 72.24%. At the same time, the worst performance values in sensitivity, specificity and AUC were obtained with QDA. LDA achieved about 6% better classification performance compared to QDA. Fig. 7 shows graphs of the evaluation metrics for each machine learning algorithm.

TABLE I
AVERAGE 10K CROSS-VALIDATION RESULTS

	LSTM	GRU	CNN	LDA	QDA
ACC	**92.99**	91.66	90.62	78.23	72.24
SENS	93.64	93.73	91.16	79.56	78.56
SPEC	92.33	89.59	90.08	76.91	65.92
AUC	0.977	0.973	0.959	0.861	0.796

MEG signals may have adequate (sometimes poor) absolute locality. This means that a noticeable event can be seen at different times and slightly different frequency ranges. CNN emerged from the idea that the image of an object is independent of its location in the picture [32]. Each neuron binds only a small part of the entrance and extends across the entire depth of the entrance. In this case, it provides a distinct advantage compared to traditional machine learning algorithms such as LDA and QDA to detect patterns on poorly localized MEG signals. On the other hand, considering all comparison metrics, RNN-based machine learning algorithms perform significantly better than other methods. Over time, different neural network models have become specialized to process different data types. For example, Convolutional Neural Networks are specialized for processing matrix type information, such as image data, while Recurrent Neural Networks (RNN) have also been developed to process sequence data. The only input that traditional feed-forward neural networks take into account is existing examples to which it is exposed.

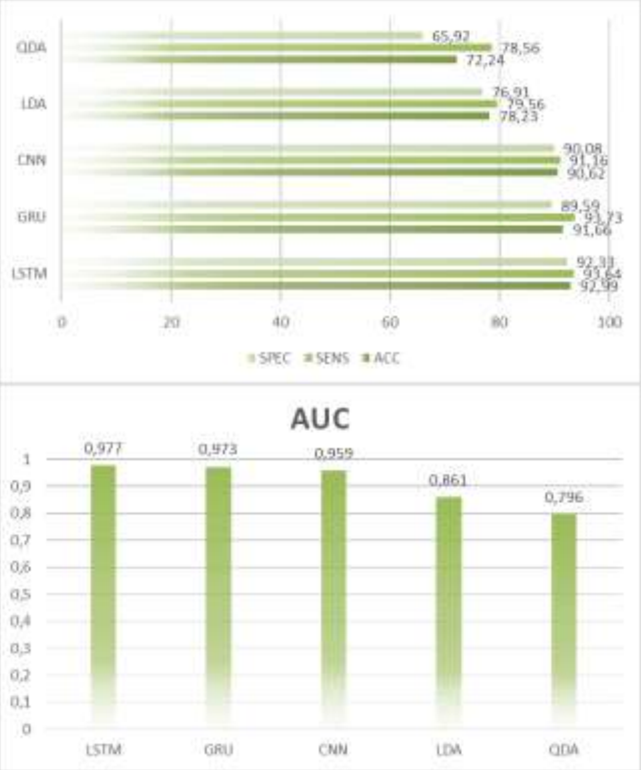

Fig.7. Performances of machine learning algorithms and AUC scores

On the other hand, RNNs also use the information for calculating the overtime, as well as the existing samples. Moreover this memory-based method (RNN) has a repetition layer, unlike traditional feed-forward neural networks. By means of this layer, the state information generated by the feed-forward network is stored and re-applied to the network with the input information. In other words, RNNs have a memory that holds what has been calculated so far. It is evaluated that these capabilities of RNNs cause them to perform better on MEG data compared to other machine learning algorithms.

Table 2 shows detailed results for LSTM 10k cross-validation results. Accordingly, the best classification performance for a fold was 94.37%. The lowest classification success was 91.61%. The highest and lowest scores for sensitivity were 95.12% and 91.30%, respectively. In specificity values, the lowest score was 90.66%, while the highest score was 94.04%. In general, the sensitivity value of the model is higher than the specificity value. In the results in Table 1, it is seen that the sensitivity values are higher than the specificity values. It can be said that all of the 5 different machine learning algorithms used generally differentiate true positive classes (meaningful face) more successfully.

TABLE 2
10K CROSS-VALIDATION RESULTS OF LSTM

	ACC	SENS	SPEC	AUC
1. Fold	91.61	91.30	91.93	0.972
2. Fold	92.46	94.27	90.66	0.978
3. Fold	91.40	90.87	91.93	0.966
4. Fold	94.16	**95.12**	93.21	0.983
5. Fold	93.62	94.06	93.19	0.982
6. Fold	92.88	94.69	91.06	0.975
7. Fold	93.41	92.78	**94.04**	0.978
8. Fold	**94.37**	95.11	93.63	**0.985**
9. Fold	92.56	94.04	91.08	0.977
10. Fold	93.41	94.26	92.57	0.975

In addition, based on Table 2, it is observed that the data distribution between folds occurs randomly in a mutually exclusive structure so that the accuracy between folds is close to each other.

TABLE 3
10K CROSS-VALIDATION RESULTS OF GRU

	ACC	SENS	SPEC	AUC
1. Fold	88.64	94.69	82.59	0.959
2. Fold	91.83	91.51	92.14	0.979
3. Fold	91.30	92.57	90.02	0.973
4. Fold	88.43	90.02	86.83	0.946
5. Fold	91.92	93.63	90.21	0.978
6. Fold	93.09	94.27	91.91	0.978
7. Fold	89.90	94.06	85.74	0.965
8. Fold	**94.58**	**95.96**	**93.21**	**0.985**
9. Fold	93.52	95.96	91.08	0.984
10. Fold	93.41	94.68	92.14	0.984

Table 3 also gives details about the 10k cross-validation results of the GRU algorithm. The highest performance value for GRU was 94.58%. On the other hand, the lowest performance value is 88.64%. The highest score for sensitivity was 95.96%. The lowest sensitivity score is 90.02%. The specificity score is generally lower than the sensitivity values, similar to other machine learning algorithms. The lowest and highest scores in the AUC score were 0.946 and 0.985, respectively.

The results for 10k cross-validation tests of the CNN algorithm are shown in Table 4. The highest accuracy value for CNN was 92.56%. The highest value obtained for a fold is lower compared to LSTM and GRU. The lowest accuracy value was obtained at 87.37%. Here, the lowest score was observed compared to LSTM and GRU. The lowest and highest score for sensitivity is 87.69% and 92.99%, respectively. The lowest score for specificity is 87.04%, and the highest score is 92.36%. The highest value for AUC was 0.973. The lowest score for AUC is 0.932.

Fig. 8 shows the boxplot representation of 10k cross-validation results obtained with LSTM, GRU, and CNN algorithms. Accordingly, the median value of the LSTM algorithm in accuracy is higher compared to GRU and CNN. On the other hand, the lowest median value for accuracy belongs to CNN. Similarly, the lowest median value in sensitivity is the CNN algorithm. Although the median value for Specificity is lower than LSTM, it is quite similar to GRU. The peak values of the CNN algorithm appear to be lower in all comparison metrics compared to RNN-based algorithms.

TABLE 4
10K CROSS-VALIDATION RESULTS OF CNN

	ACC	SENS	SPEC	AUC
1. Fold	91.08	90.45	91.72	0.963
2. Fold	90.76	92.99	88.54	0.962
3. Fold	88.43	87.89	88.95	0.939
4. Fold	87.37	87.69	87.04	0.932
5. Fold	91.07	91.72	90.42	0.962
6. Fold	90.75	90.44	91.06	0.970
7. Fold	91.39	92.14	90.64	0.968
8. Fold	92.56	**94.47**	90.66	0.970
9. Fold	90.44	91.49	89.38	0.957
10. Fold	**92.35**	92.34	**92.36**	**0.973**

In addition, if RNN based algorithms are analyzed, when Table 2, Table 3, and Fig. 13 are evaluated together, it is seen that there are much larger differences in the GRU algorithm between the highest and lowest values for each fold compared to LSTM. Results for GRU performance have been realized in a much wider range. The change interval of the sensitivity value occurred close to each other in both models. However, the peak value of GRU is higher. In addition, one outlier value was realized at the lower point in the GRU. Specificity value stands out as the main factor that reveals the difference between both models. The median of the specificity value of GRU is much lower than the LSTM, and the difference between the highest and lowest values is quite high. It was observed that the false-positive value was significantly higher in the GRU algorithm. Considering both the distribution of the sensitivity value and the specificity values, it can be considered that the positive class trend for the GRU algorithm is higher on the MEG dataset. GRU modules control the flow of information as in LSTM modules. But unlike LSTM, they don't have a memory unit. For this reason, LSTMs can remember longer sequences compared to GRU [35]. Therefore, LSTMs are more successful in this task, as the evaluation of MEG signals also requires modeling long-term relationships.

In Fig. 9, the signal analysis of Subject-1 over the MEG signal recordings for face and scrambled face were shown to explore the cortical decoding at the first trial and the third trial, respectively. Notably, the face visual stimulation related power increment has been investigated between 8-18 Hz after the visual stimulation, as reported in the previous research study [18,36]. Moreover, this initial power increment of the evoked component has arisen around 170ms and/or 220ms. Then again, as expected, the negative deflection of N170 (the Event Related Potential-ERP) has occurred greater for face visual stimuli than scrambled visual stimuli around 170ms [37].

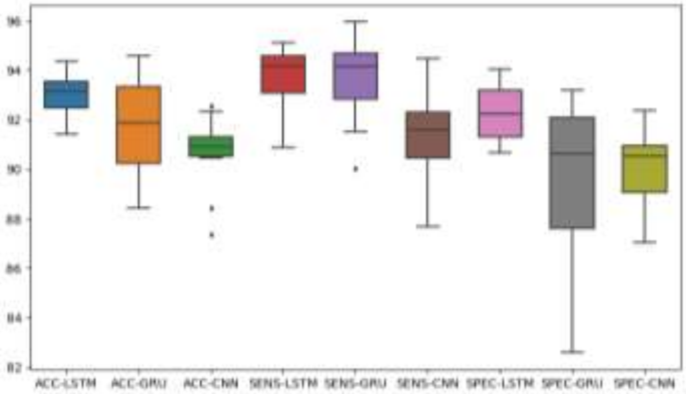

Fig.8. Boxplot graphics of LSTM, GRU and CNN algorithms

All these results are coherent with the previous outcomes connected to the N/M170 cortical activation [18,36-37]. Furthermore, the latency for scrambled face visual stimulation was observed that this might be due to the brain perception mechanism. The functioning of face perception has been observed as an automatic, rapid, and subconscious process which already has been seen in human newborns. Preferably, the simple schematic (such as scrambled face) may be seen for face-like patterns. Therefore, the face-like stimuli pattern can be perceived as faces for participants in the experiment. The tendency of the brain to see the face is called the phenomenon of pareidolia. Hence, the face-specificity of the N170 is a challenge for neurological studies [36]. Nonetheless, the development of more generative, complex and realistic comments from the neuroimaging data requires more multi-subject and multi-modal analysis.

Furthermore, according to the reported study, the stages of the stimulation performances in terms of the machine learning algorithms show that the perception stage can provide much higher accuracy than the pre-stimuli stage. In our study, deep learning algorithms may have yielded distinguished performances due to the better perception stage representation of the spatial and temporal features of the MEG signals [38]. t-SNE visualization was shown in Fig.10.

The confusion matrices were also determined for GRU and LSTM deep methods, as shown in Fig.11. Finally, some other studies performed with MEG dataset in Table 5 and the results of RNN-based models and CNN model that provide the best results in this study are presented together. Considering the results in the table, it can be seen that RNN-based approaches perform significantly better. With the LSTM model, an improvement of 12.14% was achieved, which corresponds to the 63.39% relative error reduction rate compared to the DAE model, which provided the best results before. Besides, the CNN model appears to give significantly better results compared to other studies in the literature.

IV. Conclusion

The main purpose of this study is to investigate the classification performances of MLs over the MEG signals which were recorded during the human brain's response to visual stimuli to be decoded the brain functioning of face perception mechanism. There are two classes: face and scramble face in the classification process. MEG signals are very difficult to classify as they contain high amounts of noise. In this study, the classification performances were compared by using LSTM, GRU, CNN, LDA, and QDA algorithms. The CNN algorithm appears to provide a distinct advantage in capturing weakly localized MEG signals compared to LDA, QDA, and other studies with the same dataset. With the CNN algorithm, 90.62% and 0.959% values were obtained for accuracy and AUC, respectively.

On the other hand, the best results were obtained with RNN based algorithms. RNN algorithms cannot use only the existing information they are exposed to as input. In addition to this information, they use the information they calculate overtime. Therefore, they differ from traditional neural networks. In this study, the best results in all comparison metrics except sensitivity were obtained with the LSTM algorithm. Quite similar values were obtained with the GRU for the sensitivity metric. The LSTM model, 92.99%, 93.64%, 92.33%, and 0.977 values were obtained for accuracy, sensitivity, specificity, and AUC, respectively.

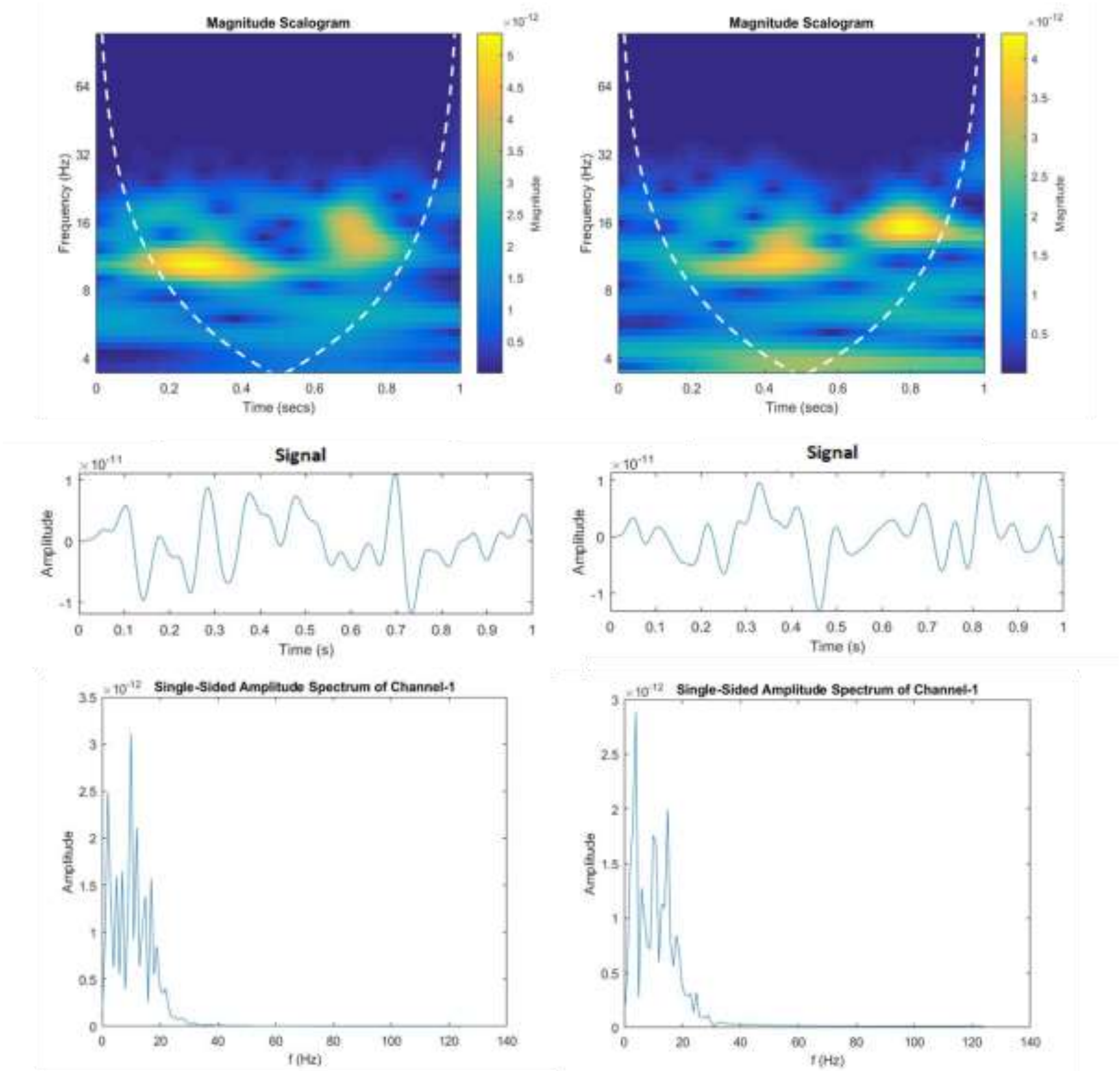

Fig.9. Signal, Scalogram and Single-Sided amplitude spectrum presentation for face visual stimuli (*left*) Signal, Scalogram and Single-Sided amplitude spectrum presentation for scrambled face visual stimuli (*right*) (Channel-1 signals are represented for all figures)

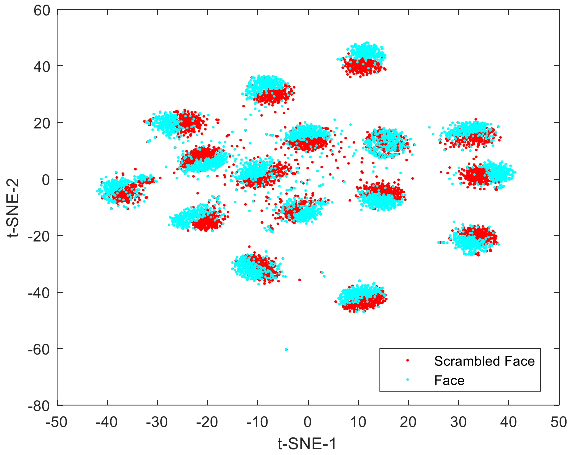

Fig.10. t-SNE visualization for feature extracted dataset via Riemannian approach

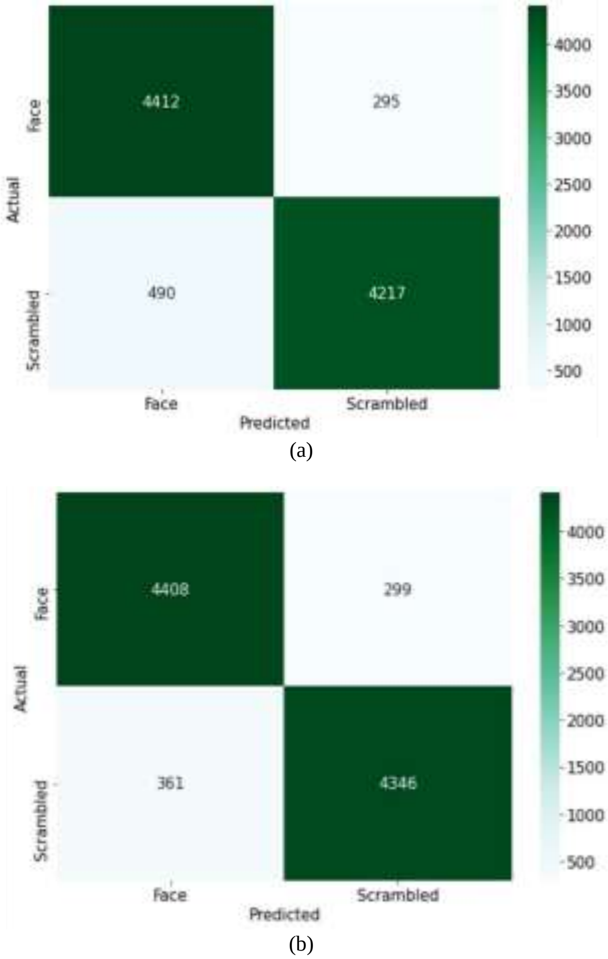

Fig. 11. The confusion matrices for GRU (*a*) and LSTM (*b*) related to the average results of 10-fold cross validation

TABLE 5
EXISTING METHODS USING THE MEG DATA AND CLASSIFICATION ACCURACIES FOR FACE/SCRAMBLED FACE RECOGNITION IN THE LITERATURE

STUDY	METHOD	NEURAL DECODING	ACC (%)
[5]	DNN SVM KNN NB DT	Face/Scramble Face Decoding	**80.85** 78.01 72.84 71.92 68.36
[10]	LVQ	Face/Scramble Face Decoding	**69.39**
[39]	SVM	Face/Scramble Face Decoding	**74.85**
[11]	GRNN	Face/Scramble Face Decoding	**79**
[40]	Hybrid GRU	Face/Scramble Face Decoding	**71.20**
[41]	SVM	Face, Tool, Animal, Scene Decoding	**84**
[9]	MLNN PNN	Face/Scramble Face Decoding	77.78 **82.36**
This Study	LSTM GRU CNN LDA QDA	Face/Scramble Face Decoding	**92.99** 91.66 90.62 78.23 72.24

The ability of LSTMs to learn long and short-term dependencies has provided a distinct advantage over other algorithms used in the MEG dataset. In future studies, the LSTM algorithm can be used to study and compare cortical activities of various regions of the brain. Moreover, applications of intelligent Internet of Things (IoT) need universal and trustworthy biometric authentication system [42-43]. To address these issues, paradigm of a visual presentation (face/scrambled face) can be proposed to use the MEG signals of subjects due to the great spatial and temporal resolution with specific stimulation of individual’s brainwave pattern.

Human and animal rights

This article does not contain any studies with human participants or animals performed by any of the authors. The used data in this paper is taken from the Kaggle data science repository. It is a public dataset.

ACKNOWLEDGMENT

This study was supported by Scientific Research Project Unit of the Bandırma Onyedi Eylül University under Project No: BAP-18-MF-1003-005.

REFERENCES

[1] Zarief, C. N., & Hussein, W. (2019). Decoding the Human Brain Activity and Predicting the Visual Stimuli from Magnetoencephalography (MEG) Recordings. In Proceedings of the 2019 International Conference on Intelligent Medicine and Image Processing - IMIP '19 (pp. 35–42). New York, New York, USA: ACM Press. https://doi.org/10.1145/3332340.3332352

[2] Lin, J.-F. L., Silva-Pereyra, J., Chou, C.-C., & Lin, F.-H. (2018). The sequence of cortical activity inferred by response latency variability in the human ventral pathway of face processing. Scientific Reports, 8(1), 5836. https://doi.org/10.1038/s41598-018-23942-x

[3] Watanabe, S., Miki, K., & Kakigi, R. (2005). Mechanisms of face perception in humans: A magneto- and electro-encephalographic study. Neuropathology, 25(1), 8–20. https://doi.org/10.1111/j.1440-1789.2004.00603.x

[4] Tadel, F., Bock, E., Niso, G., Mosher, J. C., Cousineau, M., Pantazis, D., … Baillet, S. (2019). MEG/EEG Group Analysis With Brainstorm. Frontiers in Neuroscience, 13(FEB), 1–21. https://doi.org/10.3389/fnins.2019.00076

[5] Caliskan, A., Yuksel, M. E., Badem, H., & Basturk, A. (2017). A deep neural network classifier for decoding human brain activity based on magnetoencephalography. Elektronika Ir Elektrotechnika, 23(2), 63–67. https://doi.org/10.5755/j01.eie.23.2.18002

[6] Özkaya, Ş. N., & Yıldırım, T. (2018). Assessment of Components and Methods Used to Identify Responses and Regions of Brain Related with Face Recognition and Perception. Procedia Computer Science, 131, 38–44. https://doi.org/10.1016/j.procs.2018.04.183

[7] http://web.mit.edu/kitmitmeg/whatis.html. (n.d.). Retrieved from http://web.mit.edu/kitmitmeg/whatis.html

[8] Kia, S. M., Vega Pons, S., Weisz, N., & Passerini, A. (2017). Interpretability of Multivariate Brain Maps in Linear Brain Decoding: Definition, and Heuristic Quantification in Multivariate Analysis of MEG Time-Locked Effects. Frontiers in Neuroscience, 10. https://doi.org/10.3389/fnins.2016.00619.

[9] Cetin, O., & Temurtas, F. (2020). A comparative study on classification of magnetoencephalography signals using probabilistic neural network and multilayer neural network. Soft Computing. https://doi.org/10.1007/s00500-020-05296-7

[10] Çetin, O., & Temurtaş, F. (2018). A Study on Brain Computer Interface using Learning Vector Quantization. Sakarya University Journal of Computer and Information Sciences, 1(2), 1–7.

[11] Çetin, O., & Temurtaş, F. (2019). Classification of Magnetoencephalography Signals Regarding Visual Stimuli by Generalized Regression Neural Network. Dicle Tıp Dergisi, 45(3), 19–25. https://doi.org/10.5798/dicletip.534819.

[12] Gorur, K., Bozkurt, M., Bascil, M., & Temurtas, F. (2019). GKP Signal Processing Using Deep CNN and SVM for Tongue-Machine Interface. Traitement Du Signal, 36(4), 319–329. https://doi.org/10.18280/ts.360404

[13] Yeom, H. G., Kim, J. S., & Chung, C. K. (2020). LSTM Improves Accuracy of Reaching Trajectory Prediction From Magnetoencephalography Signals. IEEE Access, 8, 20146–20150. https://doi.org/10.1109/ACCESS.2020.2969720

[14] Alom, M. Z., Moody, A. T., Maruyama, N., Van Essen, B. C., & Taha, T. M. (2018). Effective Quantization Approaches for Recurrent Neural Networks. In Proceedings of the International Joint Conference on Neural Networks (Vol. 2018-July). https://doi.org/10.1109/IJCNN.2018.8489341

[15] Alpaydın, E. (2010). Introduction to Machine Learning. MIT Press, Cambridge, Massachusetts.

[16] Ghojogh, B., & Crowley, M. (2019). Linear and Quadratic Discriminant Analysis: Tutorial, (4), 1–16. Retrieved from http://arxiv.org/abs/1906.02590

[17] Barachant, A. (2014). MEG decoding using Riemannian Geometry and Unsupervised classification. Notes on the winner of the Kaggle "DecMeg2014 - Decoding the Human Brain" competition., 1–8.

[18] Henson, R. N., Wakeman, D. G., Litvak, V., & Friston, K. J. (2011). A Parametric Empirical Bayesian Framework for the EEG/MEG Inverse Problem: Generative Models for Multi-Subject and Multi-Modal Integration. Frontiers in Human Neuroscience, 5(August), 1–16. https://doi.org/10.3389/fnhum.2011.00076

[19] "DecMeg2014-Decoding the Human Brain". [Online]. Available: https://www.kaggle.com/c/decoding- the-human-brain. (n.d.), 2014.

[20] Olivetti, E., Kia, S. M., & Avesani, P. (2014). MEG decoding across subjects. Proceedings - 2014 International Workshop on Pattern Recognition in Neuroimaging, PRNI 2014. https://doi.org/10.1109/PRNI.2014.6858538

[21] Roy, P. P., Kumar, P., & Chang, V. (2020). A hybrid classifier combination for home automation using EEG signals. Neural Computing and Applications, 32(14), 1–19. https://doi.org/10.1007/s00521-020-04804-y

[22] Yger, F., Berar, M., & Lotte, F. (2017). Riemannian Approaches in Brain-Computer Interfaces: A Review. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25(10), 1753–1762. https://doi.org/10.1109/TNSRE.2016.2627016

[23] Barachant, A., Bonnet, S., Congedo, M., & Jutten, C. (2012). Multiclass brain-computer interface classification by Riemannian geometry. IEEE Transactions on Biomedical Engineering, 59(4), 920–928. https://doi.org/10.1109/TBME.2011.2172210

[24] Uçar, M. K. (2020). Classification Performance-Based Feature Selection Algorithm for Machine Learning: P-Score. IRBM, 41(4), 229–239. https://doi.org/10.1016/j.irbm.2020.01.006

[25] Bascil, M. S., Tesneli, A. Y., & Temurtas, F. (2015). Multi-channel EEG signal feature extraction and pattern recognition on horizontal mental imagination task of 1-D cursor movement for brain computer interface. Australasian Physical and Engineering Sciences in Medicine, 38(2), 229–239. https://doi.org/10.1007/s13246-015-0345-6

[26] Hossin, M., & Sulaiman, M. . (2015). A Review on Evaluation Metrics for Data Classification Evaluations. International Journal of Data Mining & Knowledge Management Process, 5(2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201

[27] Toğaçar, M., Ergen, B., Cömert, Z., & Özyurt, F. (2020). A Deep Feature Learning Model for Pneumonia Detection Applying a Combination of mRMR Feature Selection and Machine Learning Models. IRBM, 41(4), 212–222. https://doi.org/10.1016/j.irbm.2019.10.006

[28] Yang, S., & Berdine, G. (2017). The receiver operating characteristic (ROC) curve. The Southwest Respiratory and Critical Care Chronicles, 5(19), 34. https://doi.org/10.12746/swrccc.v5i19.391

[29] Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., & Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks, 12(2), 181–201. https://doi.org/10.1109/72.914517

[30] Koudjonou, K. M., & Rout, M. (2020). A stateless deep learning framework to predict net asset value. Neural Computing and

Applications, 32(14), 1–19. https://doi.org/10.1007/s00521-019-04525-x

[31] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791

[32] Ozer, I., Ozer, Z., & Findik, O. (2018). Noise robust sound event classification with convolutional neural network. Neurocomputing, 272, 505–512. https://doi.org/10.1016/j.neucom.2017.07.021

[33] Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology, 195(1), 215–243. https://doi.org/10.1113/jphysiol.1968.sp008455

[34] Ma, J., Wu, F., Zhu, J., Xu, D., & Kong, D. (2017). A pre-trained convolutional neural network based method for thyroid nodule diagnosis. Ultrasonics, 73, 221–230. https://doi.org/10.1016/j.ultras.2016.09.011

[35] Golmohammadi, M., Ziyabari, S., Shah, V., Von Weltin, E., Campbell, C., Obeid, I., & Picone, J. (2017). Gated recurrent networks for seizure detection. In 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB) (Vol. 2018-Janua, pp. 1–5). IEEE. https://doi.org/10.1109/SPMB.2017.8257020

[36] Hadjikhani, N., Kveraga, K., Naik, P., & Ahlfors, S. P. (2009). Early (M170) activation of face-specific cortex by face-like objects. NeuroReport, 20(4), 403–407. https://doi.org/10.1097/WNR.0b013e328325a8e1

[37] Wakeman, D. G., & Henson, R. N. (2015). A multi-subject, multi-modal human neuroimaging dataset. Scientific Data, 2(1), 150001. https://doi.org/10.1038/sdata.2015.1

[38] Dash, D., Ferrari, P., & Wang, J. (2020). Decoding Imagined and Spoken Phrases From Non-invasive Neural (MEG) Signals. Frontiers in Neuroscience, 14(September 2004), 8–20. https://doi.org/10.3389/fnins.2020.00290

[39] Gupta, S., & Gandhi, T. (2020). Identification of Neural Correlates of Face Recognition Using Machine Learning Approach. In Computer Vision and Machine Intelligence in Medical Image Analysis (pp. 13–20). https://doi.org/10.1007/978-981-13-8798-2_2

[40] Li, J., Pan, J., Wang, F., & Yu, Z. (2021). Inter-Subject MEG Decoding for Visual Information with Hybrid Gated Recurrent Network. Applied Sciences, 11(3), 1215. https://doi.org/10.3390/app11031215

[41] Liu, C., Kang, Y., Zhang, L., & Zhang, J. (2021). Rapidly Decoding Image Categories From MEG Data Using a Multivariate Short-Time FC Pattern Analysis Approach. IEEE Journal of Biomedical and Health Informatics, 25(4), 1139–1150. https://doi.org/10.1109/JBHI.2020.3008731

[42] Huang, H., Hu, L., Xiao, F., Du, A., Ye, N., & He, F. (2019). An EEG-Based Identity Authentication System with Audiovisual Paradigm in IoT. Sensors, 19(7), 1664. https://doi.org/10.3390/s19071664

[43] Sooriyaarachchi, J., Seneviratne, S., Thilakarathna, K., & Zomaya, A. Y. (2021). MusicID: A Brainwave-Based User Authentication System for Internet of Things. IEEE Internet of Things Journal, 8(10), 8304–8313. https://doi.org/10.1109/JIOT.2020.3044726

BIOGRAPHIES

ZEYNEP ÖZER was born in Ankara, Turkey, in 1986.

In 2010, she graduated from Dumlupınar University with a bachelor's degree in mathematics. She received a M.Sc. degree from the Department of Computer Engineering at the University of Sakarya, Turkey, in 2012. Also, she received a Ph. D. degree from the Computer Engineering at the University of Karabük, Turkey, in 2019.

Since 2019, she has been working as an assistant professor in the Department of Management Information Systems at Bandırma Onyedi Eylül University.

Her current research interests include natural processing language, artificial intelligence, applied mathematics and advanced machine learning techniques.

ONURSAL ÇETİN was born in Kayseri, Turkey, in 1982. He received the B.S. and M.S. degrees from Erciyes University, Kayseri, Turkey, in 2006 and 2009, respectively, and the Ph.D. degree from Sakarya University, Sakarya, Turkey, in 2014, all in electrical and electronics engineering.

From 2007 to 2014, he was a Research Assistant with Bozok University, Yozgat, Turkey. From 2014 to 2018, he worked as an Assistant Professor with Bozok University. Since 2018, he has been an Assistant Professor with the Electrical and Electronics Engineering Department, Bandırma Onyedi Eylul University, Bandırma, Turkey. His research interests include artificial intelligence, signal processing, and embedded systems.

KUTLUCAN GÖRÜR was born on April 13, 1985 in Adana. He completed his primary, high school and university education in this city. He graduated from Adana Anatolian High School in 2003. After graduating from Çukurova University Electrical and Electronics Engineering in 2009, he worked as a Research Assistant at Bozok University Electrical and Electronics Engineering Department in 2010.

started his life. After graduating from Bozok University Institute of Science and Mechatronics Engineering master's program in 2014, he enrolled in a doctoral program at Sakarya University Institute of Science, Department of Electrical and Electronics Engineering in 2015. He is currently working as an assistant professor at Bandırma Onyedi Eylül University, Department of Electrical and Electronics Engineering.

FEYZULLAH TEMURTAŞ received the B.Sc. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 1991, the M.Sc. degree in electrical and electronics engineering from Kütahya Dumlupınar University, Kütahya, Turkey, in 1995, and the Ph.D. degree in electrical and electronics engineering from Sakarya University, Serdivan, Turkey, in 2000.

He is currently a Professor with Bandırma Onyedi Eylül University, Bandırma, Turkey. He has substantial experience of data processing, machine learning, electronic nose, biomedical, control and computer architecture.