Biomedical Signal Processing and Control 88 (2024) 105018

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

CNN-FEBAC: A framework for attention measurement of autistic individuals

Manan Patel [a], Harsh Bhatt [a], Manushi Munshi [a], Shivani Pandya [a], Swati Jain [a,*], Priyank Thakkar [a], SangWon Yoon [b]

[a] CSE, Nirma University, Ahmedabad, Gujarat, India
[b] State University of New York at Binghamton, NY, USA

ARTICLE INFO

Keywords:
EEG signals
Autism Spectrum Disorder
EEGNet
Feature Extractor
Shallow classifier
CNN-FEBAC

ABSTRACT

Electroencephalogram (EEG) signals are a cost-effective and efficient method to measure and analyse neurological data and brain-related ailments. Autism Spectrum Disorder (ASD) is a globally prevalent neurological disorder that is of significant concern to the medical research community regarding its diagnosis and treatment. Artificial Intelligence (AI) algorithms utilized to study EEG signals of autistic patients have shown promising results to make progress in this domain. In this study, the authors have used the BCIAUT-P300 dataset for attention measurement and analysis of EEG signals of autistic patients. The dataset comprises the EEG signal data of ASD patients when they are exposed to external stimuli in a controlled environment. The authors propose a Convolutional Neural Network based Feature Extractor for BCI Attention Classification (CNN-FEBAC) framework to achieve the research objective of predicting the response of ASD patients by studying their EEG signal recordings. The CNN-FEBAC framework consists of a feature extractor architecture followed by a shallow classifier to predict the patient's response to the stimuli. The proposed model was evaluated using performance metrics such as — confusion matrix, accuracy and F1 scores. The best accuracy achieved by the proposed model was 91%. The authors have explored and described the limitations of previously established methods and highlighted the performance improvements achieved with the proposed CNN-FEBAC framework. The authors further highlight the challenges encountered in the study and suggest the scope for improvement.

1. Introduction

Electroencephalogram (EEG) is the electrical recording of the brain's activity, represented by voltage fluctuations due to ionic current flows within the brain neurons. It is a non-invasive recording technique of brain signals and is performed by attaching electrodes to the brain. The amplitude range of EEG signals is 10 to 200 V, and the frequency range is 0.5 to 40 Hertz (Hz). EEG signals are non-stationary and non-linear in nature [1,2].

EEG signals find significant use in the research community in studying brain functions, attention and alertness. They are also used to diagnose various neurological ailments like epilepsy, autism, seizure and other brain traumas [3,4]. The strength and intensity of an individual's brain waves vary throughout the day. The EEG signals can be divided into five frequency bands, depending upon the level of awakening of the brain (Table 1) [5].

Autism Spectrum Disorder (ASD), or autism, is a developmental disorder affecting the individual's social interaction and behaviour. It can be defined by the presence of difficulty or impairments while dealing with social interactions and communication [6]. Here, the term "spectrum" signifies that the disorder can occur in different forms with varying severity levels in different individuals. Each autistic individual faces different kinds and levels of impairment, symptoms, strengths and challenges [7,8].

Table 1
EEG Frequency Bands.

EEG band	Frequency range	Brain state
Delta	0.5 to 4 Hz	Deep, restful, unconscious sleep
Theta	4 to 8 Hz	Light sleep
Alpha	8 to 13 Hz	Awake, rest state
Beta	13 to 30 Hz	Awake and fully alerted state
Gamma	Greater than 30 Hz	Visual Simulation response

The behavioural patterns of ASD individuals are analysed via neurological data. EEG signal recordings of such individuals can help analyse their neurological functioning. An autistic patient's EEG signals have significantly more spiking as compared to normal individuals [9]. The

* Corresponding author.
E-mail addresses: 19bce112@nirmauni.ac.in (M. Patel), harsshbhatt0201@gmail.com (H. Bhatt), 19bce119@nirmauni.ac.in (M. Munshi), 21ftphde57@nirmauni.ac.in (S. Pandya), swati.jain@nirmauni.ac.in (S. Jain), priyank.thakkar@nirmauni.ac.in (P. Thakkar), yoons@binghamton.edu (S. Yoon).

https://doi.org/10.1016/j.bspc.2023.105018
Received 9 March 2023; Received in revised form 27 April 2023; Accepted 9 May 2023
Available online 2 June 2023
1746-8094/

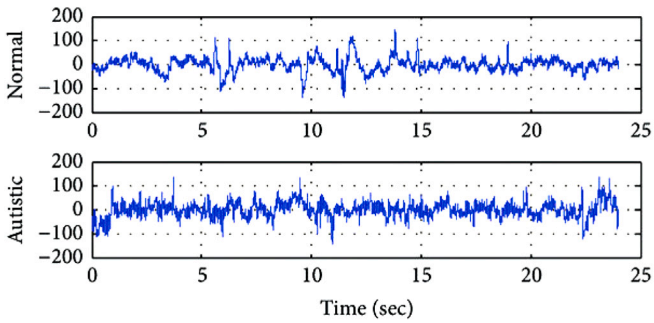

Fig. 1. EEG signal recordings of normal v/s autistic individuals [10].

magnitude of spikes in their EEG signals is also greater in comparison. Different timestamps in the signal can help analyse responses to different stimuli. Increased activity of delta, theta, beta, and gamma bands, with reduced activity of alpha frequencies, is a frequent pattern observed in patients with ASD [7]. Fig. 1 shows the difference in EEG signals of normal individuals vs autistic patients. An erratic and volatile pattern of signals is observed in autistic individuals [10].

The analysis of autism using EEG signals deals with certain concepts such as Event-Related Potential (ERP), P-300 wave and Brain Computer Interface (BCI). An ERP can be explained as an electro-physiological response to a stimulus by a human brain. The P-300 wave is the ERP component elicited by the brain during any decision-making process. It is related to the person's neurological response to any stimulus and not the physical attributes of the stimulus. A BCI is a collaboration between a brain and a device that enables the brain signals to direct certain external activities, such as control of a cursor or a prosthetic limb.

Artificial Intelligence (AI) has been widely used to analyse and classify EEG signals. Machine learning (ML) has consistently enhanced results in such sectors. It is a sub-field of AI that involves the development of algorithms and statistical models that enable computers to learn from and make decisions without explicit programming. ML consists of several types, such as supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [11,12]. Deep Learning (DL) is an AI domain that uses deep neural networks to model and analyse complex patterns in data. It has been particularly successful in areas such as computer vision, natural language processing and speech recognition. While DL requires large amounts of data and computational power, their ability to automatically extract and learn features often helps them perform better than traditional ML algorithms. DL models can handle non-linearity in data, which is crucial as many real-world problems have non-linear relationships between inputs and outputs [13,14]. Subsequently, DL techniques have found their applications in various everyday domains such as healthcare [15], satellite image analysis [16], robotics [17] and automation [18], among others.

This study aims to propose a DL framework to study and analyse the changes in EEG signals of ASD individuals when they react to a stimulus. The BCIAUT-P300 dataset from the International Federation of Medical and Biological Engineering (IFMBE) Scientific Challenge, organized during MEDICON 2019 [19], has been used for this study. The dataset consists of EEG signal recordings of ASD patients as a response to certain controlled stimuli. The task is to predict whether an ASD patient successfully identified the target object or not using the EEG data. The authors have explored the approaches from previous studies and aim to overcome their shortcomings such as – inconsistent performance on subjects, lack of evaluation metrics, and session-specific analysis – by proposing their framework — **C**onvolutional **N**eural **N**etwork based **F**eature **E**xtractor for **B**CI **A**ttention **C**lassification (CNN-FEBAC).

1.1. Motivations

1 in 160 children worldwide has ASD. Analysing ASD patients and their behaviour can help better understand the disorder and develop more effective treatment strategies. EEG signals are a practical and affordable method of measuring brain activity and can be used to analyse autism. The potential for EEG to be employed as a functional brain imaging modality is growing. It can be analysed and explored with the aid of cutting-edge DL and ML algorithms [7]. The authors hope to contribute to the medical research community with the results of this study.

1.2. Contributions

The major contributions of this paper are as follows.

- The authors propose a novel framework comprising data augmentation techniques, a CNN-based Feature Extractor model to capture the general EEG signal information and a shallow classifier model on top for patient-specific information.
- The authors compare and analyse the results of the proposed framework with previously established methods on the same dataset. The authors also highlight the shortcomings of the previous approaches and detail the reasons for improved performance in the proposed framework.
- The results of the study can aid medical experts in analysing the response and attention patterns of ASD individuals, which can help aid and develop treatment programs. The authors also highlight further research avenues to expand the scope of the study.

1.3. Organization

The rest of the paper is organized as follows. Section 2 describes the other works related to the domain and the literature study. Section 3 presents the details and description of the dataset. Section 4 describes the framework system. The methodology and implementation are described in Section 5. The results of the study are presented in Section 6. Finally, the paper is concluded in Section 7, which also includes the future scope of the study.

2. Related work

EEG signals have been employed in different domains of neurological analysis, such as emotion recognition, epilepsy analysis, autism analysis etc. Past research has shown that ML methods have been applied for emotion recognition and classification using EEG signals [30]. The authors in [30] have used the DEAP dataset formed after aggregating data from different tests and sampling rates. The 2D EEG channel data was converted to a single dimension feature vector to feed into ML algorithms like Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbours (KNN), Decision Trees and Linear Discriminant Analysis (LDA). They concluded the best performance of SVM with an accuracy of 63% (see Table 2).

ML algorithms have also been applied in studies for detecting autism using EEG signal data. Liao et al. in [20] describe a study which employs behavioural data such as eye fixation, facial expressions and physiological data like EEG signals to detect autism in individuals. The authors have used ML techniques like Random Forest, SVM and KNN for autism detection. Random Forest outperformed other algorithms with a classification accuracy of 83.75%.

DL approaches have also been employed for autism detection using EEG signals. In study [21], the 2D channel EEG data was passed to the CNN model for classification and the model's performance was evaluated using a confusion matrix and accuracies. The authors achieved an accuracy of 80%. Radhakrishnan et al. [22] present a comparative

Table 2
Comparison of literature.

Study	Data	Approach	Best accuracy	Other evaluation metrics	Limitations
Detect children with ASD using physiological and behavioural data by applying ML techniques [20]	Custom dataset	Random Forest, SVM, KNN	83.75%	Confusion Matrix	Data recorded in varying environments, limited samples
Detect autism by identifying patterns in EEG signals between autistic and normal individuals using DL [21]	Dataset from University King Abdul Aziz	CNN architecture	80%	Confusion Matrix	Limited size of dataset resulting in low accuracies and inconsistent outputs
Analyse performance of various deep CNN architectures to detect autism [22]	Custom dataset	AlexNet, ResNet50, Resnet101	81%	Precision, Recall, F1 score	Heavy overfitting of models due to deep architectures, limited number of samples in the dataset
Detect autistic individuals by applying deep convolution neural network based architectures on EEG signal data [23]	Dataset from University King Abdul Aziz	ResNet101, ResNet50, ResNet18	98.32%	Precision, F1 score, Recall	Limited dataset of 29 individuals, computationally intensive processing
ASD detection in individuals using EEG signal data and different ML based classification algorithms [24]	Custom Dataset	Probabilistic Neural Network, SVM-RBF, KNN	98.7%	Sensitivity, Specificity, Positive Predictive Value	Proposed model only works with smaller data size, feature selection is done manually
Attention Measurement in 13-year old ASD individual using EEG signals [25]	Custom Dataset	SVM, MLP-NN, Random forest	92.99%	Confusion Matrix, Precision, Recall, F1 score, Area Under Curve, Cohen's Kappa Coefficient, Hamming Loss, Matthews Correlation Coefficient	Absence of BCI and individual EEG information which can enhance performance and the quality of the data
Attention Measurement of ASD patients using EEG signals based on response to external stimuli [26]	BCIAUT P-300 Dataset	EEGNet architecture	92.47%	Not Mentioned	Hyperparameters of the base model have not been tuned, performance on other evaluation metrics not provided
Attention Measurement of ASD patients using EEG signals based on response to external stimuli [27]	BCIAUT P-300 Dataset	BLSTM-CNN architecture	84%	Not Mentioned	Hyperparameters have not been tuned to check if there is a performance increase, limited number of samples in the dataset
Attention Measurement of ASD patients using EEG signals based on response to external stimuli [28]	BCIAUT P-300 Dataset	VB-ARD model	81.4%	Computation time	Accuracy variance is very high, performance on other evaluation metrics of classification is not provided
Attention Measurement of ASD patients using EEG signals based on response to external stimuli [29]	BCIAUT P-300 Dataset	LDA, LDA+Boosting, LSVM, RSVM	80%	F1 score	Limited number of samples in the dataset, Inconsistent and insufficient accuracies across different individuals

study of different DL-based convolutional architectures applied to EEG signals to perform feature extraction and classification for autism detection. The authors applied architectures such as AlexNet, Inception v1, ResNet50, Resnet101, SqueezeNet etc. The architecture of ResNet50 outperformed other algorithms with an average accuracy of 81%.

Another domain wherein artificial intelligence is applied to EEG signals is attention measurement and analysis of ASD patients [25]. In this study, the authors have explored the use of different algorithms like Naïve Bayes, KNN, Decision Trees, SVM, Multi-layer Perceptron — Neural Networks (MLP-NN), Random forest etc., for attention measurement and analysis of autistic individuals. The authors concluded the best performances of MLP-NN, Random Forest and SVM-RBF algorithms with accuracies of 92.99%, 92.94% and 89.33%, respectively.

Cecotti et al. [31] were the first to establish the implementation of neural networks while dealing with P300 waves, which have also been used in our study. The study detected P300 waves using a CNN model with an accuracy of 95.5%. This study concludes the application of DL and CNN-based approaches while dealing with P300-based signal data.

The winners of the IFMBE challenge 2019 employed a CNN approach for the research objective of attention span and analysis of autistic patients when they are asked to identify target objects from a series of flashed objects [26]. The authors implemented the EEGNet Model with a few modifications. The model was trained within the set in which each patient session was treated separately and all sessions of one patient were trained together for the cross-set. The within-set accuracy achieved 84.43% and the cross-set accuracy was 92.47%.

Other approaches in the IFMBE 2019 competition included the Bidirectional Long Short Term Memory (BLSTM)-CNN approach [27] and Linear v/s Non-linear classification approach [29]. In another study [27], the authors developed a model with one CNN layer followed by 2 BLSTM layers and achieved an accuracy of 84%. Arancibia et al. [29] utilized linear approaches such as LDA and SVM and non-linear approaches like Reduced Support Vector Machines (RSVM) for session-specific classification. The study showed that linear algorithms did not work for complex cases, and they were more generalized. It also proved that the non-linear approaches did not perform significantly better than the linear approaches. The overall data prediction accuracy achieved was 82%. The authors in the study concluded that subject-specific classification outperformed inter-subject classification.

Based on the comprehensive literature study, the authors have identified and summarized certain research gaps. Several studies in autism analysis using EEG signals incorporate machine learning approaches and require prior knowledge of the domain for manual feature selection. This process varies for different datasets and requires expert knowledge and human intervention. This study addresses the limitation above by utilizing DL techniques that facilitate automatic feature selection.

Furthermore, the EEG signal recordings in this application are collected from multiple nodes simultaneously, forming multi-channel data. These channels must be processed together to retain maximum signal information. This study uses CNN architectures that capture spatial information using convolutional layers that learn filters to extract features from different signal parts. These filters are applied across all channels simultaneously to recognize local patterns and structures effectively. Hence CNNs are highly efficient at retaining spatial information [31,32] as well as extracting the underlying features [15].

Attention analysis of an ASD patient is generally conducted over several sessions with similar stimuli. The patient's response differs slightly in each session, resulting in minor variances in the signal recordings. Current studies perform a session-specific evaluation that does not incorporate such variations. The authors aim to address this limitation by combining the data from each subject's sessions to train the proposed architecture.

3. Dataset

The BCIAUT-P300 dataset was used in the International Federation of Medical and Biological Engineering (IFMBE) Scientific Challenge, organized during MEDICON 2019. It comprises the complete EEG signal recordings of a conducted clinical trial to test a P300-based Brain–Computer Interface to train ASD patients in joint social attention and following social cues.

It contains the EEG data of 15 ASD patients who underwent seven training sessions over four months. Hence a total of 105 sessions were conducted. Data acquisition was made using the g. The Nautilus system is a device measuring EEG data and needs to be worn on the head. The device recorded data from 8 electrodes at C3, CZ, C4, CPZ, P3, PZ, P4, and POZ locations. The participant's outcomes were assessed at baseline, i.e., session 0; post the training, i.e., session seven and the follow-up (6 months after training).

The P300-based virtual reality BCI paradigm was presented to participants in a virtual environment using a Virtual Reality headset and EEG cap. The 8 electrodes mentioned above recorded the EEG data. The virtual environment was in the form of a normal bedroom having certain objects of interest. 8 objects in the environment were flashed in front of the participant in random order. These 8 objects were — a wooden plane hanging from the ceiling, a printer on a shelf, a cork board on the wall, a laptop on a table, a ball on the ground, a radio on top of a dresser, a picture on the wall and books on a shelf.

The task was divided into 3 phases. The first two phases were a part of the BCI calibration and the last phase was the online phase. In Phase 1, Participants were directly and explicitly instructed to attend to the target object to remove potential errors in identifying the same. Phase 2 involved asking the participants which object was flashed to guarantee they learned to read the attention cue accurately and correctly use this information. In Phase 3, the participants were asked to respond to the head cue of an avatar in the centre of the scene, looking at the object of interest.

Each trial had 10 sequential runs; each run consisted of the flashing of all 8 objects in random order. Flash of each object was done with an inter-stimulus interval of 200 ms and each flash had a duration of 100 ms. This gave a total of 80 flashes per trial. Participants performed a total of 70 trials (10 in the first phase, 10 in the second, and 50 in the online phase). Fig. 2 shows the pictorial representation of blocks, runs and events in the experimental setup. Fig. 2 (A) shows a block used to identify a target object. Each block consisted of K runs. Fig. 2 (B) shows a run. Each run consisted of 8 events, each event corresponding to the flashing of one of the 8 objects in the setup. Fig. 2 (C) shows an event which comprised of flashing of the corresponding object for 100 ms, with an interval of 200 ms between 2 flashes.

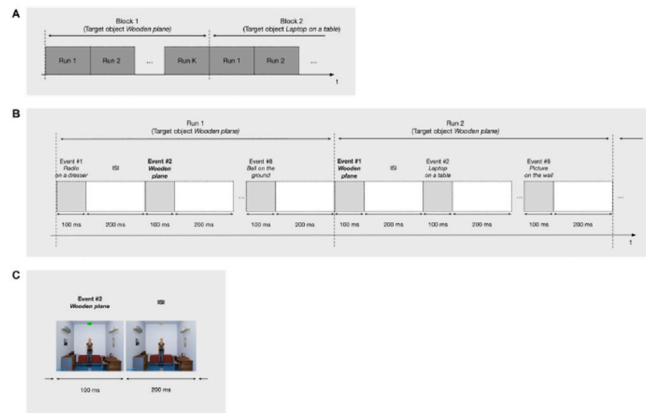

Fig. 2. Description of experiment environment with blocks runs and events [19].

The first 20 calibration trials stored the P300 responses that occurred when the object of interest flashed. This was part of the training set. Statistical classifiers were used to identify this response. These classifiers were then used in the online phase to determine whether participants were counting the flashes of the avatar's object of interest. If it was correctly done by the participant, the BCI gave positive feedback, i.e., the object of interest turned green at the end of the trial. If not, the object turned red.

The train folder consisted of the following files-

- train data.mat — Calibration phase data, structured as [channels x epoch x event], epoch corresponding to data samples from 200–1000 ms, relative to the event stimulus onset (epoch length of 1200 ms; 350 data samples). Final shape = (8, 1600, 350)
- trainEvents.txt — One label per line (from 1 to 8) corresponding to the order of the flashed objects.
- trainTargets.txt — 1 or 0 per line, indicating if the flashed object was the target object or not, respectively.
- trainLabels.txt — Label of the target object per line (from 1 to 8), one for each block.

The test folder consisted of the following files-

- testData.mat — Data from the online phase in the same structure as the train data.
- testEvents.txt — One label per line (from 1 to 8) corresponding to the order of the flashed objects.
- testTargets.txt — 1 or 0 per line, indicating whether the flashed object was the target, respectively.
- testLabels.txt — Label of the target object per line (from 1 to 8), one for each block.
- runs_per_block.txt — File containing only one number, corresponding to the number of runs per block used in the online phase (from 3 to 10).

The epochs were calculated as (events per run*runs per block*blocks). For training data, there were 8 events per run, 10 runs per block and 20 blocks which summed up to 1600 epochs. In the testing data, the number of runs varied between sessions; there were 50 blocks and 8 events in each run. The epochs summed up to 400 K, where K was the variable number of runs between sessions.

4. Framework system model

Fig. 3 visually represents the proposed CNN-FEBAC system. It showcases a step-by-step process of using the BCIAUT-P300 dataset to predict an ASD individual's response to external stimuli.

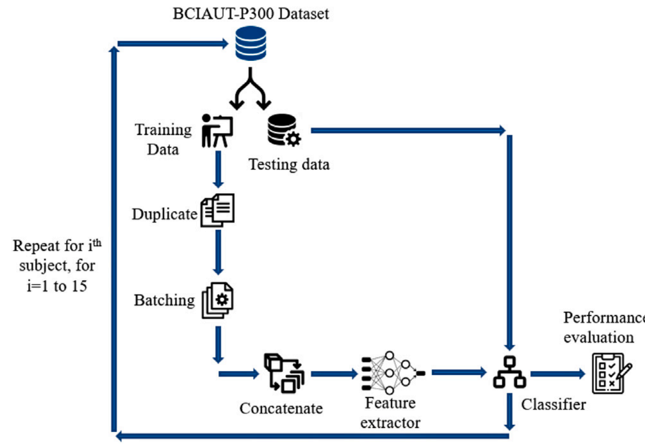

Fig. 3. System model and description of the CNN-FEBAC Framework.

The study is conducted on the data of 15 subjects ($n \in \{1, 2, \ldots 15\}$). The training dataset (D) is severely imbalanced with samples of class 0 (D^0) and samples of class 1 (D^1) in the ratio of 7:1. Hence, the data is pre-processed and transformed to correct the target class imbalance. To overcome this, data augmentation techniques are used to obtain samples of both classes in the ratio of 1:1. Subsequently, augmentation, batching and shuffling techniques are carried out and the processed data (I) is returned. The pre-processing and augmentation process is described in detail in Section 5.1. Next, the Feature Extractor (FE) is trained on all subjects iteratively to capture the intrinsic features of the EEG signal data from all subjects. Steps 2 to 5 of Algorithm 1 showcase the pre-processing and feature extraction training. After the FE training is completed, the trainable attribute of the feature extractor is changed to *False* to stop further training. Following this, a Shallow Classifier (SC) is joined on top of the feature extractor. The final model (M) consists of both combined models. The final model is then trained on each subject individually to gain subject-specific information. Steps 6 to 9 of Algorithm 1 describe the final model compilation process.

Algorithm 1 Training process.

Inputs: Subjects (n), Training Data (D), Actual Label (L), Feature Extractor (FE), Shallow Classifier (SC), Final Model (M)
Output: Obtain trained model M
Preparation: Number of epochs = z

```
1:  procedure MODEL(n, D, L, FE, SC, M)
2:      for each n ∈ {1, 2, … 15} do
3:          I_n = pre-processing(D_n, L_n)
4:          train FE (I_n, z)
5:      end for
6:      FE.trainable = False
7:      M ← join(FE, SC)
8:      train M(I_n, z)
9:      return M
10: end procedure
```

After the complete training of model (M) is completed, it is used to make predictions on the test data (T). The model predicts the probabilities of either class (P). The argmax function obtains the final predictions (H). This process is depicted in steps 3 and 4 of Algorithm 2. The final predictions are then used with the actual labels (L) to evaluate the accuracy and F1 score. The final results (R) are combined for analysis. The evaluation process is depicted in steps 5 to 8 of Algorithm 2.

The results table R includes subject-wise evaluation metrics (accuracy and F1 score of both classes) for the final model M and is summarized in Table 6 in Section 6.

Algorithm 2 Prediction and Evaluation.

Inputs: Subjects (n), Testing Data (T), Predicted Class Probabilities (P), Predicted Labels (H), Actual Labels (L), Results Table (R)
Output: Obtain predicted labels H and performance metrics table R for all subjects

```
1:  procedure EVALUATE(n, T, P, H, L, R)
2:      for each n ∈ {1, 2, … 15} do
3:          P_n = predict M (T_n)
4:          H_n → argmax(P_n)
5:          R_A = evaluate_accuracy(H_n, L_n)
6:          plot_classification_matrix(H_n, L_n)
7:          R_F = evaluate_F1score(H_n, L_n)
8:          R ← join (R_A, R_F)
9:      end for
10:     return R
11: end procedure
```

5. Methodology

This section presents the details of the proposed CNN-FEBAC framework, including the dataset pre-processing, the EEGNet base model architecture, the framework parameters and the training processes.

5.1. Dataset pre-processing

The data of each patient has been recorded over multiple sessions. The session-specific data has a limited sample size. Hence, the data of all the sessions of one subject have been merged and used as one training set. In addition, this dataset is highly imbalanced where the class targets are distributed in the ratio 7:1 for 0 and 1, respectively (Eq. (1)). This causes significant overfitting and can result in negative outcomes and negatively affect architecture performance. Data augmentation techniques can address this problem by increasing the size of the undersampled class.

$$D^0 : D^1 = 7 : 1 \tag{1}$$

Oversampling or undersampling techniques can either increase the undersampled class or decrease the oversampled class, respectively. However, due to such a significant imbalance, undersampling the larger class would result in a loss of information. Hence, the smaller class was oversampled. The samples were duplicated without any changes to increase the samples and bring the sample distribution ratio of the classes to 1:1 (Eq. (2), (3)).

$$D^1_{aug} \xleftarrow{\text{Duplication}} D^1 \tag{2}$$

$$D^0 : D^1_{aug} = 1 : 1 \tag{3}$$

Since the duplicates were essentially the same samples, the model was prone to overfitting. Hence it was trained in batches of 64 consisting of equal samples of both classes — 32 of each. Samples of both classes were separated, and shuffled individually, and 32 random samples were drawn from each class to form a batch of 64. The following equations depict the aforementioned batching process, where D_B represents the dataset obtained after batching:

$$D^0_B \xleftarrow{\text{random sampling}} D^0, \text{ Size}(D^0_B) = 32 \tag{4}$$

$$D^1_B \xleftarrow{\text{random sampling}} D^1_{aug}, \text{ Size}(D^1_B) = 32 \tag{5}$$

$$D_B = D^0_B + D^1_B, \text{ Size}(D_B) = 64 \tag{6}$$

Algorithm 3 represents the entire flow of the pre-processing and augmentation process.

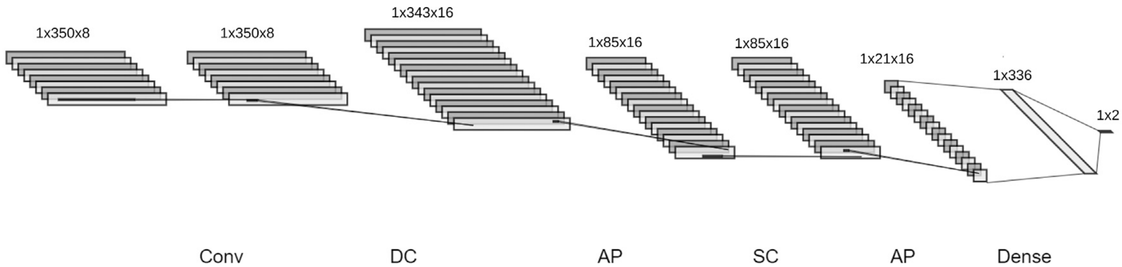

Fig. 4. EEGNet architecture.

Algorithm 3 Data Pre-processing and Augmentation.

Inputs: Training Data (D), Actual Label (L), Batch (B)
Output: Obtain final training inputs I

```
procedure MODEL(D, L, S, B, I)
    D → (D^0, L^0), (D^1, L^1)
    while Size(D^1) ≤ Size(D^0) do
        D^1_dup = duplicate(D^1)
        append(D^1_aug, D^1_dup)
    end while
    while Size(D^0) > 0 ς Size(D^1_aug) > 0 do
        D^0_B ← random sampling(D^0), Size(D^0_B) = 32
        D^1_B ← random sampling(D^1_aug), Size(D^1_B) = 32
        D_B ← concatenate(D^0_B, D^1_B)
        shuffle(D_B)
        append(I, D_B)
    end while
    return(I)
end procedure
```

5.2. EEGNet

After data augmentation and batching, the EEGNet model – a pre-established compact convolutional neural network for EEG-based brain–computer interfaces – was reconstructed and trained on the batched dataset. The model was compiled using Adam optimizer [33] and Sparse Categorical Crossentropy loss function. The model was trained for 100 epochs. The EEGNet architecture is depicted in Fig. 4.

At the cost of slight accuracy reductions, the f1 score was successfully increased through the EEGNet architecture. However, the EEGNet model is designed as a general-purpose architecture for EEG signals. Certain modifications were required in the model that would capture the specific features of the dataset on hand. These modifications are described in the section below. EEGNet utilized kernels of size (1,64). The researchers of EEGNet provided an explanation to keep the kernel size equal to the batch size as per the dataset. However, this results in a loss of information on the dataset used in this study. Autistic patients tend to have rapid frequency changes and it is essential to capture this information. Hence, the kernel size is changed to (1,8).

EEGNet used filters equal to the channel depth of the dataset. However, the maximum available information cannot be captured by this structure. Hence, the number of filters is expanded to 32 to capture more information from all channels. The 'depth_multiplier' parameter in the depthwise convolutional layer replicates the existing channels by the factor provided. However, since the data is directly duplicated and no new information is generated, this leads to overfitting. Hence, the multiplier is changed to 1 to reduce overfitting.

It was observed from visual plotting of the signal data that there were significantly higher spikes in the signals when the patient observed the target object. This information was lost in the average-pooling layers as the neighbouring values would reduce the peak value. Hence max-pooling layers are used in place of average-pooling. A 64-unit dense layer is added after the flattening layer to minimize the rapid collapse of neurons from 1344 units to 2 units.

Table 3
Custom modifications of CNN-FEBAC.

Layer parameter	EEGNet	CNN-FEBAC
Conv (number of filters)	8	32
Conv (kernel size)	(1,64)	(1,8)
DC (depth multiplier)	2	1
SC (number of filters)	16	64
Pooling	Average pooling	Max pooling

5.3. CNN-FEBAC

To address the overfitting issue, the CNN architecture of the customized EEGNet model is developed to be used as a Feature Extractor. The model is trained iteratively on the data of all the subjects to generalize the learning patterns and signal features. After training, the Feature Extractor weight updates are frozen. The dense layers are discarded and a shallow classifier is added on top of it. The classifier is subject-specific and the whole model is then trained on single subjects only. The CNN architecture captures generalized signal data and drastically reduces overfitting while maintaining and even improving the accuracy. Three dense layers in the classifier help capture subject-specific information which differs across individuals. A softmax activation provides the final probabilistic output of both classes. The class with the maximum probability is chosen as the predicted label. The architecture of the Feature Extractor and classifier model is shown in Fig. 5.

Adam optimizer and Sparse Categorical Cross entropy loss functions are used to compile both the Feature Extractor and the shallow classifier model. Both models are trained for 100 epochs. 'Early Stopping' [34] and 'Reduction of Learning Rate on Plateau' [35] techniques are utilized. 'Early Stopping' utilizes the validation loss metric and stops training the model when there is no significant change in the validation loss over a specified number of training epochs. This technique is beneficial to reduce the overfitting of the model. 'Reduction of Learning Rate on Plateau' reduces the learning rate by a specified factor when there is no reduction in the training loss over the course of a specified number of training epochs. This technique reduces redundant training rounds and helps the model reach an optimized state faster. The data augmentation techniques, the Feature Extractor model and the shallow classifier, are combined to form the entire CNN-FEBAC framework.

Table 3 depicts the layer-wise parameter changes between EEGNet and CNN-FEBAC architectures.

6. Results and analysis

This section details the results obtained by the proposed architecture, its performance on the classification task and the performance comparison with the results of other studies.

6.1. Experimental setup

The experimental setup utilized several computer libraries. The NumPy library, version 1.22.4, is used for mathematical calculations

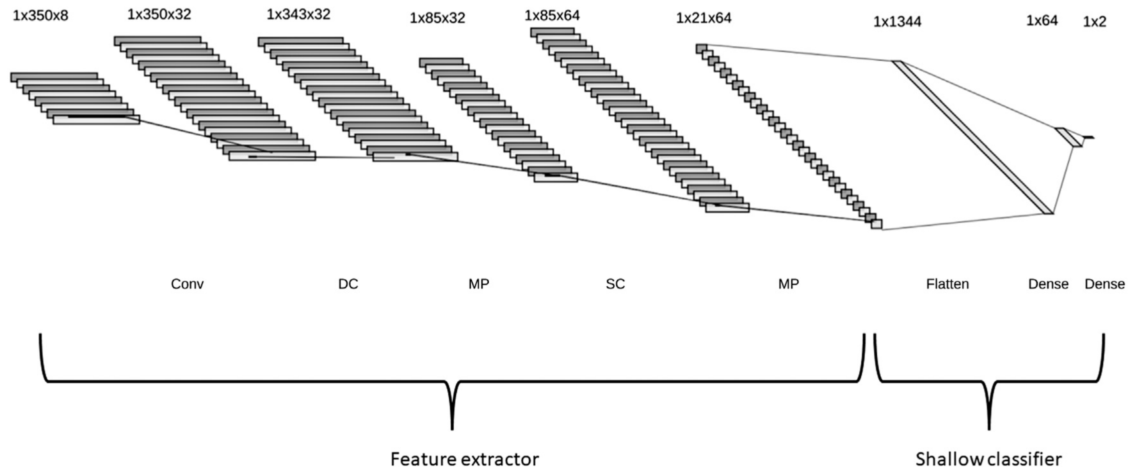

Fig. 5. Proposed feature extractor and shallow classifier architecture in CNN-FEBAC framework.

Table 4
Simulation parameters of the proposed architecture.

Parameter	Value
Environment Parameters	
NumPy environment seed	0
Tensorflow environment seed	0
Feature Extraction & Classifier Model Parameters	
Number of Training Epochs	100
Validation Data split for each Training Epoch	20%
Number of Workers	8
Optimizer	Adam
Early Stopping Metric	Validation Loss
Early Stopping Patience	5 epochs

and matrix operations. The Scikit-learn library, version 1.2.2, is used for several pre-processing and metrics evaluation processes. The pre-processing and model_selection packages from Scikit-learn are used for scaling, encoding, and data splitting processes. The metrics package from Scikit-learn is used for evaluating the proposed architecture, which includes accuracy, precision, recall, and F1 score. The SciPy library 1.10.1 visualizes the signal data from .mat files. The Tensorflow library, version 2.11.0, and Keras library, version 2.11.0, are used to develop the deep learning architectures, including the compilation, training, and testing processes. The Keras 'Model' application programming interface (API) is used to build the entire neural network models, including all layers, individual activation functions, and regularizers, compile the models, and simulate their training and testing.

The Matplotlib library, version 3.5.3, Seaborn library 0.11.2, and Plotly library, version 5.5.0, are used to construct and visualize all the plots and graphs for the simulation results. The proposed architecture requires several hyperparameters to enhance its performance. All hyperparameters involved in the simulation have been listed in Table 4 along with their respective values.

The simulation environments and processes are created and run on a system with specifications, such as a 2.50 GHz Intel Core i5 processor, 8 GB installed RAM, and NVIDIA GeForce GTX 1650 Ti graphics processor with 4 GB RAM.

6.2. Evaluation metrics

The models were evaluated using the following performance metrics [36]:

- Confusion Matrix — It depicts the prediction results of a classification problem. It showcases the true positive, false positive, false negative and true negative values (Table 5).

Table 5
Confusion matrix structure.

Confusion	Actual true value	Actual false value
Predicted True Value	True Positive (TP)	False Positive (FP)
Predicted False Value	False Negative (FN)	True Negative (TN)

- Accuracy — It is the ratio of total correct predictions to total predictions (Eq. (7)).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

- F1 Score — It is the harmonic mean of precision and recall (Eq. (8), (9), (10)).

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

6.3. EEGNet vs CNN-FEBAC

The EEGNet architecture provides an average accuracy of 69.32%, and average F1 scores for class 0 and 1 of 66.45% and 73.16%, respectively. The EEGNet provides a good baseline for the classification task. However, it fails to capture dataset-specific information as it is a generalized model. Hence, a custom and dataset-specific model must be developed to increase the performance.

The CNN-FEBAC model provides an average accuracy of 87.93%, and average F1 scores for class 0 and 1 of 86.73% and 89.13%, respectively. The feature-extraction architecture addresses the dataset-specific features. Training the architecture on the data of all subjects helps generalize the model and a subject-specific shallow classifier fine-tunes the weights.

Fig. 6 depicts the confusion matrices for best-case results of the EEGNet model and the CNN-FEBAC model.

It can be observed that while the EEGNet model is able to achieve high true positives, it lacks in minimizing the false negatives and false positives efficiently. The proposed model successfully minimizes the false negatives to a greater extent. Furthermore, it also reduces false positives, indicating better performance in resolving the overfitting issue.

Table 6 compares and contrasts other different performance metrics for both models.

Table 6
Performance metrics for custom EEGNet and CNN-FEBAC model.

Approach	Case	Accuracy (%)	F1 Score Class 0 (%)	F1 score class 1 (%)	Subject No.
Custom EEGNet model	Best	71	65	75	10
	Second Best	70	66	74	9
	Average	69.32	66.45	73.16	NA
	Second Worst	68	67	69	4
	Worst	68	67	79	13
CNN-FEBAC model	Best	91	91	92	9
	Second Best	90	90	91	10
	Average	87.83	86.73	89.13	NA
	Second Worst	84	83	85	13
	Worst	82	80	83	4

Table 7
CNN-FEBAC model performance metrics for 15 subjects.

Subject No.	Accuracy (%)	F1 Score Class 0 (%)	F1 Score Class 1 (%)
1	88	87	90
2	89	87	90
3	88	87	89
4	82	80	83
5	88	87	90
6	89	87	90
7	89	88	90
8	85	83	87
9	91	91	92
10	90	90	91
11	89	88	90
12	89	88	89
13	83	83	85
14	89	87	90
15	88	88	89

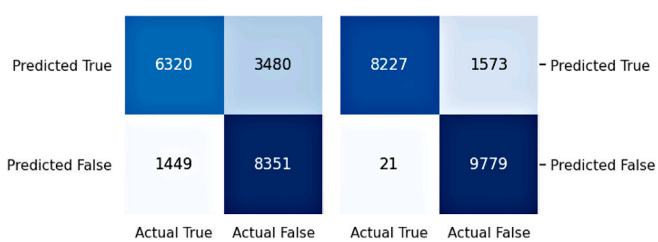

Fig. 6. Confusion matrices for best case result of EEGNet model and CNN-FEBAC model.

Although the generalized parametric values of the EEGNet model can be applicable to a large spectrum of EEG signals data, the specific case in hand does not adhere to this generalization. The custom modifications made to the CNN-FEBAC address the specific changes required to analyse the data in this study and is hence, able to outperform EEGNet.

6.4. CNN-FEBAC performance analysis

Instead of training a single model individually on different subjects, a feature-extraction model trained on all the subjects helps capture data and salient features of all subjects while focusing on the dataset. Training a shallow classifier to be subject-specific helps capture minute individual variations and build on the generalized information. As a result, the CNN-FEBAC framework performs consistently on all the subjects by successfully capturing generic EEG patterns and individual variations. This also results in a significant performance boost of 15%–20% across all metrics compared to the former approach. Table 7 depicts the performance metrics of subject-specific classification for 15 subjects for the proposed CNN-FEBAC model.

Fig. 7 depicts the training accuracy of the proposed model for successive epochs and Fig. 8 illustrates the training loss of the proposed model for successive epochs until training is stopped by the Early

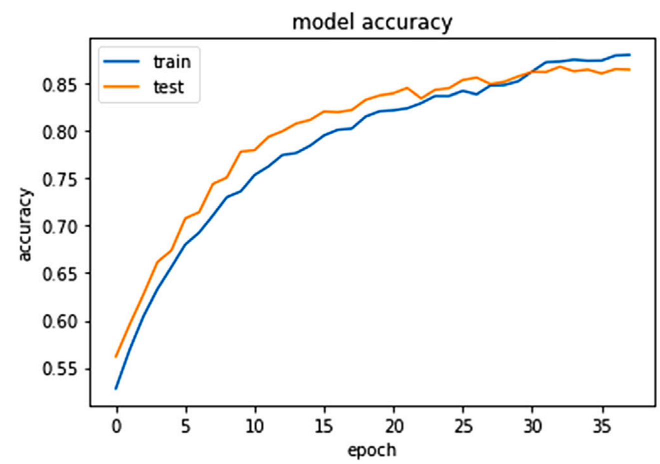

Fig. 7. Change in accuracy over each training epoch of CNN-FEBAC model.

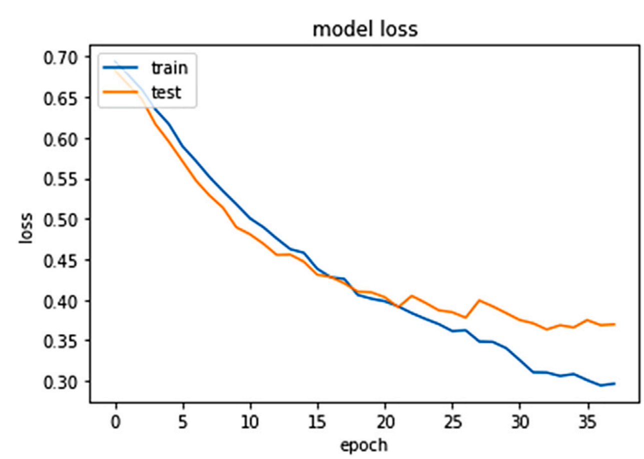

Fig. 8. Change in loss over each training epoch of CNN-FEBAC model.

Stopping method. It can be observed that a continuous and sharp decrease in training loss accompanies a continuous and sharp increase in the training accuracy. This signifies that the model is successful in continuously improving its learning process.

As shown in Table 8, other architectures and algorithms deliver average accuracies close to the baseline of 80%. However, they do not provide consistent performances and there is a sharp drop in the accuracy for a few specific subjects. E. Santamaria-Vazquez et al. [27] implemented a CNN-BLSTM approach and obtained an accuracy of 56% for subject 1. H. Zhao et al. [38] implemented SVM, LDA and CNN approaches and obtained an accuracy of 51% for subject 1. However, the author's proposed feature extraction model and shallow classifier deliver an accuracy of 88% for the same subject. Comparing the metrics

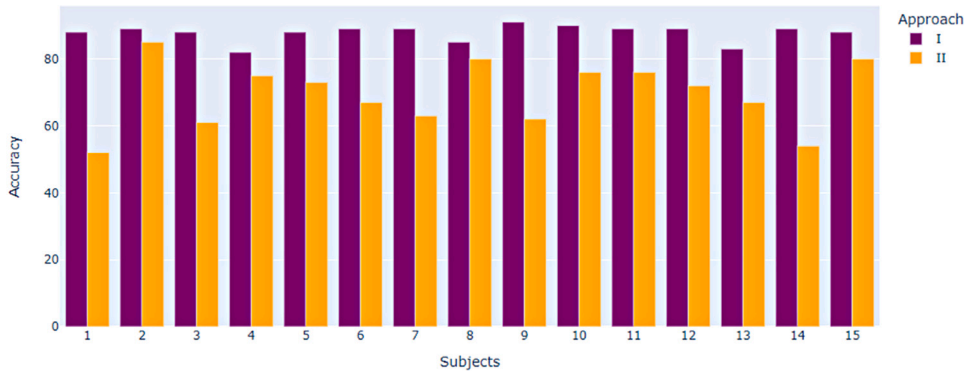

Fig. 9. Accuracy comparison : I. CNN-FEBAC (proposed) and II. LDA+SVM [37].

Table 8
Comparative performance metrics of the proposed model with other established methods on the same dataset.

Model	Pre-processing	Average accuracy (%)	F1 score(%)
CNN-LSTM [27]	NA	82	Not Mentioned
LDA+Boosting [29]	Signal averaging and downsampling	79	73
CNN-BLSTM [27]	NA	84	Not Mentioned
RSVM [29]	Signal averaging and downsampling	79	76
VB-ARD [28]	Signal time stamp modification	81.4	Not Mentioned
LDA [29]	Signal averaging and downsampling	80	76
Proposed CNN-FEBAC Framework	**Duplication and batching**	**87.93**	**88**

of other subjects, the standard deviations of accuracies are high at an average of 8.6%. The Feature Extractor model has a standard deviation of 2.2% hence, providing improved and more consistent performances.

Fig. 9 depicts a subject-wise accuracy comparison of CNN-FEBAC with the approach used by Zhao et al. [37]. They utilized a combination of a 20 Hz Butterworth low-pass filter, a linear support vector regression pre-selector and Linear Discriminant Analysis (LDA). CNN-FEBAC outperforms the above-stated methods in the accuracy of all the subjects.

7. Conclusion and future scope

Attention measurement of autistic patients using EEG signals is a significant research domain in healthcare. ML and DL models outperform conventional methods due to their capacity to work on and capture more information from large and multi-channel datasets. The authors have used the BCIAUT-P300 dataset for the objective of attention measurement and analysis of autistic individuals. The dataset contains EEG signals of autistic patients when they react to an external stimulus of identifying a target object from a series of flashed objects. The dataset is highly imbalanced and data augmentation using conventional methods was challenging due to the time-series nature of EEG signal data. Hence, data duplication was used to balance the classes in a 1:1 ratio. An EEGNet architecture was implemented to form a baseline model, which gave the best accuracy of 71%. However, EEGNet is a generalized model which does not capture dataset-specific information. The authors propose the CNN-FEBAC model comprising a CNN Feature Extractor to address the dataset-specific information and a shallow classifier added on top of it to fine-tune and capture individual variations in the signal. This model achieved the best accuracy of 91%.

The proposed framework has certain limitations. The dataset has a limited sample size, due to which the class imbalance issue hinders the performance of the model. Publicly available datasets need more data to train a generic model that can effectively capture the variations associated with EEG signals. Such models can help expand the scope of the study, which is currently limited to subject-specific classification. Thus, larger datasets can be used to overcome this problem. Additionally, deeper neural network architectures can help increase the accuracy of the outcome on such datasets. Certain research gaps need to be addressed to make further progress in this domain. Introducing well-balanced datasets can help prediction models train effectively on each class and improve model performance. Further studies can also be conducted to enhance signal data augmentation techniques to expand small datasets in order to deal with class imbalance. Furthermore, the analysis of EEG signals can be combined with the analysis of other techniques, such as eye fixation, to provide more insights and improve end results.

CRediT authorship contribution statement

Manan Patel: Software, Writing – original draft, Visualization. **Harsh Bhatt:** Software, Writing – original draft, Visualization. **Manushi Munshi:** Software, Writing – original draft, Visualization. **Shivani Pandya:** Investigation, Data curation. **Swati Jain:** Conceptualisation, Methodology, Writing – review & editing. **Priyank Thakkar:** Methodology, Validation. **SangWon Yoon:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Publicly accessible data.

References

[1] S.D. Puthankattil, P. Joseph, U.R. Acharya, C. Lim, EEG signal analysis: a survey, J. Med. Syst. 34 (2010) 195–212, http://dx.doi.org/10.1007/s10916-008-9231-z.

[2] S. Vaid, P. Singh, C. Kaur, EEG signal analysis for BCI interface: A review, in: 2015 Fifth International Conference on Advanced Computing & Communication Technologies, 2015, pp. 143–147, http://dx.doi.org/10.1109/ACCT.2015.72.

[3] A. Khosla, P. Khandnor, T. Chand, A comparative analysis of signal processing and classification methods for different applications based on EEG signals, Biocybern. Biomed. Eng. 40 (2) (2020) 649–690, http://dx.doi.org/10.1016/j.bbe.2020.02.002, URL https://www.sciencedirect.com/science/article/pii/S0208521620300231.

[4] D.M. Praveena, D.A. Sarah, S.T. George, Deep learning techniques for EEG signal applications – A review, IETE J. Res. 68 (4) (2022) 3030–3037, http://dx.doi.org/10.1080/03772063.2020.1749143.

[5] S. Nacy, S. Kbah, H. Jafer, I. Al-Shaalan, Controlling a servo motor using EEG signals from the primary motor cortex, Am. J. Biomed. Eng. 2016 (2016) 139–146, http://dx.doi.org/10.5923/j.ajbe.20160605.02.

[6] C. Lord, M. Elsabbagh, G. Baird, J. Veenstra-Vanderweele, Autism spectrum disorder, Lancet 392 (10146) (2018) 508–520, http://dx.doi.org/10.1016/S0140-6736(18)31129-2, URL https://www.sciencedirect.com/science/article/pii/S0140673618311292.

[7] W. Bosl, H. Tager-Flusberg, C. Nelson, EEG analytics for early detection of autism spectrum disorder: A data-driven approach, Sci. Rep. 8 (2018) http://dx.doi.org/10.1038/s41598-018-24318-x.

[8] S. Raj, S. Masood, Analysis and detection of autism spectrum disorder using machine learning techniques, Procedia Comput. Sci. 167 (2020) 994–1004, http://dx.doi.org/10.1016/j.procs.2020.03.399.

[9] M. Magán-Maganto, A. Bejarano, C. Fernandez Alvarez, A. Narzisi, P. García-Primo, R. Kawa, M. De la Paz, R. Canal, Early detection and intervention of ASD: A European overview, Brain Sci. 7 (2017) 159, http://dx.doi.org/10.3390/brainsci7120159.

[10] R. Djemal, K. Alsharabi, S. Ibrahim, A. Alsuwailem, EEG-based computer aided diagnosis of autism spectrum disorder using wavelet, entropy, and ANN, BioMed Res. Int. 2017 (2017) 1–9, http://dx.doi.org/10.1155/2017/9816591.

[11] H.-D. Wehle, Machine learning, deep learning, and AI: What's the difference?, 2017.

[12] M.I. Jordan, T.M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260, http://dx.doi.org/10.1126/science.aaa8415, URL https://www.science.org/doi/abs/10.1126/science.aaa8415.

[13] H. Alaskar, T. Saba, Machine Learning and Deep Learning: A Comparative Review, 2021, pp. 143–150, http://dx.doi.org/10.1007/978-981-33-6307-6_15.

[14] N. Hordri, S. Yuhaniz, S.M. Shamsuddin, Deep learning and its applications: A review, 2016.

[15] W. Wang, X. Zhang, S. Wang, Y. Zhang, Covid-19 diagnosis by WE-SAJ, Syst. Sci. Control Eng. 10 (2022) http://dx.doi.org/10.1080/21642583.2022.2045645.

[16] T.T. Nguyen, T.D. Hoang, M.T. Pham, T.T. Vu, T.H. Nguyen, Q.-T. Huynh, J. Jo, Monitoring agriculture areas with satellite images and deep learning, Appl. Soft Comput. 95 (2020) 106565, http://dx.doi.org/10.1016/j.asoc.2020.106565, URL https://www.sciencedirect.com/science/article/pii/S1568494620305032.

[17] K. Aggarwal, S. Singh, M. Chopra, S. Kumar, F. Colace, Deep learning in robotics for strengthening industry 4.0.: Opportunities, challenges and future directions, 2022, pp. 1–19, http://dx.doi.org/10.1007/978-3-030-96737-6_1,

[18] V.M. Deshmukh, R. B, G.B. Krishna, G. Rudrawar, An overview of deep learning techniques for autonomous driving vehicles, in: 2022 4th International Conference on Smart Systems and Inventive Technology, ICSSIT, 2022, pp. 979–983, http://dx.doi.org/10.1109/ICSSIT53264.2022.9716433.

[19] M. Simões, D. Borra, G.-U. Santamaría-Vázquez, M. Bittencourt-Villalpando, D. Krzemiński, A. Miladinović, N, T. Schmid, H. Zhao, C. Amaral, B. Direito, J. Henriques, P. Carvalho, M. Castelo-Branco, BCIAUT-p300: A multi-session and multi-subject benchmark dataset on autism for P300-based brain-computer-interfaces, Front. Neurosci. 14 (2020) http://dx.doi.org/10.3389/fnins.2020.568104, URL https://www.frontiersin.org/articles/10.3389/fnins.2020.568104.

[20] M. Liao, H. Duan, G. Wang, Application of machine learning techniques to detect the children with autism spectrum disorder, J. Healthcare Eng. 2022 (2022) 1–10, http://dx.doi.org/10.1155/2022/9340027.

[21] N. Ali, Autism spectrum disorder classification on electroencephalogram signal using deep learning algorithm, IAES Int. J. Artif. Intell. (IJ-AI) 9 (2020) 91, http://dx.doi.org/10.11591/ijai.v9.i1.pp91-99.

[22] M. Radhakrishnan, K. Ramamurthy, K.K. Choudhury, D. Won, T.A. Manoharan, Performance analysis of deep learning models for detection of autism spectrum disorder from EEG signals, Traitement Du Signal 38 (3) (2021) 853–863, URL https://search.ebscohost.com/login.aspx?direct=true&db=bsu&AN=151674010&site=eds-live.

[23] B. Ari, N. Sobahi, Ö.F. Alçin, A. Sengur, U.R. Acharya, Accurate detection of autism using Douglas-Peucker algorithm, sparse coding based feature mapping and convolutional neural network techniques with EEG signals, Comput. Biol. Med. 143 (2022) 105311, http://dx.doi.org/10.1016/j.compbiomed.2022.105311, URL https://www.sciencedirect.com/science/article/pii/S0010482522001032.

[24] T.-H. Pham, J. Vicnesh, J.K.E. Wei, S.L. Oh, N. Arunkumar, E.W. Abdulhay, E.J. Ciaccio, U.R. Acharya, Autism spectrum disorder diagnostic system using HOS bispectrum with EEG signals, Int. J. Environ. Res. Public Health 17 (3) (2020) http://dx.doi.org/10.3390/ijerph17030971, URL https://www.mdpi.com/1660-4601/17/3/971.

[25] J.J. Esqueda-Elizondo, R. Juárez-Ramírez, O.R. López-Bonilla, E.E. García-Guerrero, G.M. Galindo-Aldana, L. Jiménez-Beristáin, A. Serrano-Trujillo, E. Tlelo-Cuautle, E. Inzunza-González, Attention measurement of an autism spectrum disorder user using EEG signals: A case study, Math. Comput. Appl. 27 (2) (2022) http://dx.doi.org/10.3390/mca27020021, URL https://www.mdpi.com/2297-8747/27/2/21.

[26] D. Borra, S. Fantozzi, E. Magosso, Convolutional neural network for a P300 brain-computer interface to improve social attention in autistic spectrum disorder, 2019, pp. 1837–1843, http://dx.doi.org/10.1007/978-3-030-31635-8_223.

[27] E. Santamaría-Vázquez, V. Martínez-Cagigal, J. Gomez-Pilar, R. Hornero, Deep Learning Architecture Based on the Combination of Convolutional and Recurrent Layers for ERP-Based Brain-Computer Interfaces, 2020, pp. 1844–1852, http://dx.doi.org/10.1007/978-3-030-31635-8_224.

[28] A. Miladinović, M. Ajčević, P.P. Battaglini, G. Silveri, G. Ciacchi, G. Morra, J. Jarmolowska, A. Accardo, Slow cortical potential BCI classification using sparse variational Bayesian logistic regression with automatic relevance determination, in: J. Henriques, N. Neves, P. de Carvalho (Eds.), XV Mediterranean Conference on Medical and Biological Engineering and Computing – MEDICON 2019, Springer International Publishing, Cham, 2020, pp. 1853–1860.

[29] L. Arancibia, P. Sánchez-González, E. Gómez Aguilera, M.E. Hernando, I. Oropesa, Linear vs nonlinear classification of social joint attention in autism using VR P300-based brain computer interfaces, 2020, pp. 1869–1874, http://dx.doi.org/10.1007/978-3-030-31635-8_227.

[30] S. Vijayakumar, R. Flynn, N. Murray, A comparative study of machine learning techniques for emotion recognition from peripheral physiological signals, in: 2020 31st Irish Signals and Systems Conference, ISSC, 2020, pp. 1–6, http://dx.doi.org/10.1109/ISSC49989.2020.9180193.

[31] H. Cecotti, A. Graser, Convolutional neural networks for P300 detection with application to brain-computer interfaces, IEEE Trans. Pattern Anal. Mach. Intell. 33 (3) (2011) 433–445, http://dx.doi.org/10.1109/TPAMI.2010.125.

[32] W. Wang, Y. Pei, S. Wang, J. Gorriz, Y. Zhang, PSTCNN: Explainable COVID-19 diagnosis using PSO-guided self-tuning CNN, BIOCELL 47 (2023) 373–384, http://dx.doi.org/10.32604/biocell.2023.025905.

[33] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, http://dx.doi.org/10.48550/ARXIV.1412.6980, URL https://arxiv.org/abs/1412.6980.

[34] L. Prechelt, Automatic early stopping using cross validation: quantifying the criteria, Neural Netw. 11 (4) (1998) 761–767, http://dx.doi.org/10.1016/S0893-6080(98)00010-0, URL https://www.sciencedirect.com/science/article/pii/S0893608098000100.

[35] L.N. Smith, A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018, http://dx.doi.org/10.48550/ARXIV.1803.09820, URL https://arxiv.org/abs/1803.09820.

[36] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recognit. Lett. 30 (1) (2009) 27–38, http://dx.doi.org/10.1016/j.patrec.2008.08.010, URL https://www.sciencedirect.com/science/article/pii/S0167865508002687.

[37] H. Zhao, S. Yu, J. Prinable, A. Mcewan, P. Karlsson, A feasible classification algorithm for event-related potential (ERP) based brain-computer-interface (BCI) from IFMBE scientific challenge dataset, 2019, pp. 1861–1868, http://dx.doi.org/10.1007/978-3-030-31635-8_226.

[38] H. Zhao, S. Yu, J. Prinable, A. Mcewan, P. Karlsson, A feasible classification algorithm for event-related potential (ERP) based brain-computer-interface (BCI) from IFMBE scientific challenge dataset, 2019, pp. 1861–1868, http://dx.doi.org/10.1007/978-3-030-31635-8_226.