INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS

Received: 28-10-2022

Accepted: 19-01-2023

Published: 11-02-2023

Citation: Vishwesh J, Raviraj P (2023) Improved Differential Evolution with Stacked Auto Encoder for EEG Motor Imagery Classification. Indian Journal of Science and Technology 16(6): 391-400. https://doi.org/10.17485/IJST/v16i6.2076

* **Corresponding author**.

vishweshj@gsss.edu.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Improved Differential Evolution with Stacked Auto Encoder for EEG Motor Imagery Classification

J Vishwesh[1]*, P Raviraj[2]

1 Research Scholar, Department of Computer Science & Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, Affiliated to VTU, Belagavi, Karnataka, India
2 Professor, Department of Computer Science & Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, Affiliated to VTU, Belagavi, Karnataka, India

Abstract

Objectives: To develop an improved version of Differential Evolution (DE) algorithm to overcome the complexity in extracting the features from the Electroencephalogram (EEG) based Brain-Computer Interfaces (BCI) systems; To develop a Stacked Auto Encoder (SAE) for classifying motor imagery signals into left, right, feet and tongue movements, respectively. **Methods:** Improved Differential Evolution Optimization Algorithm (IDEOA) is proposed for the selection of features which is extracted by the hybrid CSP-CNN feature extraction model. Extracted features will undergo the classification process by using SAE. **Findings:** The proposed IDEOA has an accuracy of 97.34% compared to the existing Sinc-based convolutional neural networks that obtained 75.39% and TSGL-EEG-Net of 81.34%. **Novelty:** The proposed IDEOA improves the mutation strategy results in improved convergence effect.

Keywords: BrainComputer Interfaces; Convolutional Neural Networks; Electroencephalogram; Improved Differential Evolution Optimization Algorithm; Stacked Auto Encoder

1 Introduction

The BCI system bridges a gap between computers and humans which translates the thoughts for controlling the signals. The developed models are used for controlling the external devices that assist severely disabled people in living a quality life that showed minimum or non-dependence[1]. The BCI system can convert the EEG signals to peripheral signals to control the decoding function[2]. The developed model has been used widely in the field of rehabilitation during training. As per the existing classification models, the motor imagery EEG signals are utilized for the process of feature extraction which has performed recognition of relatively and independent parts are important[3,4]. The multi-band feature space is having a dimension that is reduced by using the feature selection model. Each trial is having the multi-EEG signal that represents the MI tasks that are composed to have the finite set of narrow band signals[5]. The neighborhood components are analyzed based on the feature selection algorithm that is implemented for selecting the features which are relevant to perform the classification accurately[6]. The EEG data is converted to multi-dimensional tensor images which are used with novel hybrid kernel function where it combines both the local and global kernel functions, but due to the complexity in the algorithm, computation load is increased[7]. The Common Sparse Spectral Spatial Pattern (CSSSP) simultaneously includes the spatial filter and impulse response filter that received the spatial pattern[8]. The solutions obtained by the filter coefficients are dependent on the initial parameters[9,10]. Transformed based approaches have been used to classify the EEG signals with the spatial-temporal characteristics of EEG, with good percentage of accuracy but there is change of improving the accuracy even more[11].

In this research, the optimization method-based feature selection is proposed to improve the performance of signal classification. The BCI dataset showed improvement in the developed model performances. The feature selection method selects the relevant features which are extracted and applied for classification. The classifier model based on selected features performs signal classification and performance metrics are measured.

Musallam et al.[12] developed Temporal Convolutional Network Fusion (TCN-Fusion) which consisted of fixed hyperparameters. Here, the CNN model uses various techniques like TCN, Separable Convolution, depth-wise convolution, and the fusion layers. The developed model used the filters which were unable to perform well with the resources and leads to the 83.73%, 84.13%, 83.78%, 83.62% of Accuracy, Precision, Recall and F1-score respectively.

Altuwaijri and Muhammad[13] developed a Multi branch CNN model for EEG based motor imagery classification. The developed model uses a multi-branch-based CNN model for addressing the issue that effectively extracted the temporal and spatial features from raw EEG data that branches corresponding to distinct kernel size filters. This model has limited to precision classification with 82.01%. However, the BCI-MI has shown limitations in terms of accuracy of 85.59% and needed to develop the model that would be used for the BCI system.

Ghumman et al.[14] developed a Filter Bank Common Spatial Pattern (FBCSP) with a Support Vector Machine (SVM) for performing the process of feature selection and extraction. The process of feature classification worked based on SVM enhanced the performances using the optimization approach with the polynomial kernel parameters. However, the developed model was required for investigating different optimization techniques with different classifiers for enhancing the results and showed the accuracy of only 67% Accuracy.

Deng et al.[15] developed Temporary Constrained Sparse Group Lasso (TCSGL) for finding the motor imagery-based BCI systems using a deep learning model. Through the visualization process, the developed TCSGL model proved that the developed model required meaningful features to reflect on ERD and ERS bands. The deep learning EEGNet model used with TCSGL failed to achieve an end to end model for hyperparameters learning and failed to show higher classification accuracies.

Bria et al.[16] developed a Sinc-based CNN for the classification of motor imagery which has learned effective features and the raw data from the EEG signals were read by the classifier. The Sinc-EEG-Net has used the lightweight CNN model that has combined the bandpass and learnable features having depth-wise convolutional filters which showed an improvement to evaluate the accuracy. Compared to the other methods, achieved accuracy is only of 75.39% which is better than FBCSP-SVM method.

2 Methodology

2.1 Proposed Method

Figure 1 shows the block diagram of the proposed IDEOA with the SAE model for EEG motor signal classification. Initially, the BCI competition IV dataset 2a is used the evaluate the results of the proposed method. At the first, the pre-processing is performed using a sliding window and butter worth bandpass filter for the evaluation of EEG signals. The noises are removed by using the butter worth bandpass filter. Once the pre-processed signals are obtained, the feature extraction is undergone by using the hybrid CSP-CNN feature extraction model. The obtained features were now undergone the process of feature extraction based on the proposed Improved Differential Evolution Optimization Algorithm for the selection of features. The features selected were undergone the classification process by using SAE which is a DNN based model for the classification.

2.2 BCI Dataset

The proposed method has evaluated the experiment on the dataset like BCI Competition IV Dataset 2a. The proposed method has recorded the signals from 4 class of motor imagery that has performed the tasks related to motor imagery. Each of the subjects chosen to perform classification for the left hand, right hand, tongue, or foot movements. The dataset consists of a total of 22 EEG channels that are operating between 0.5 to 100Hz, 3 EOG channels, 250Hz sampling rate, 4 classes, and 9 subjects. A visual signal will be displayed on the computer monitor screen as the task begins, from which the subject begins to perform a motor imagery task of 4s. In order to maintain a balance between the two classes, 200 motor imagery tasks are completed.

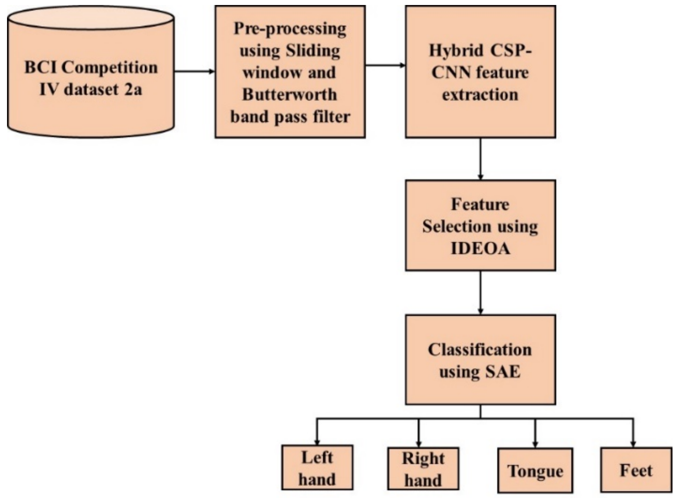

Fig 1. The block diagram of the proposed hybrid feature extraction

Voice commands are followed by predetermined instructions to perform the task, ranging in duration from 1.5 seconds to 8 seconds. The “a,” “b,” “f,” and “g” are the four human subject classes that are used in the work.

2.3 Pre-processing

The present research work uses the sixth order Butter worth filters with a sliding window used to remove the unwanted noises using the filter. The main aim of using the pre-processing technique is to eliminate the disturbances which are in the form of noises that should be detected and removed. The noise components consist of high-frequency components which include EMG and power line interference which has low frequency based Electrooculography (EOG) signals. Thus, the pre-processing technique is used to identify and eliminate such disturbances[17].

2.4 Hybrid features extraction using CSP-CNN

2.4.1 CSP

CSP is included to extract features from EEG multi-channels when hand movements are presented. The main purpose of the CSP model is to consider the weight of each channel in the current set. A covariance matrix’s two classes—hand and foot—get the most out of the transformation that was carried out.

2.4.2 Convolution Neural Network

For further processing, the prominent features of the signals will be extracted by the CNN. Features from the high dimensional data will be extracted by the CNN model. Feature extraction is similar to the input space, except using standard techniques for manually extracting features. The classifier receives these features so that it can carry out the classification. The large amount of training data used leads to overfitting of the data. The CNN will deal with the issues connected with nearby network, pooling or sub-sampling, and neighborhood availability[17].

The diagonalization of both classes of the covariance matrix serves as the foundation for the CSP model. Resnet18 architecture is used in CNN for feature extraction. 88 features from CSP 4 features from resnet18 92 features extracted. The selected feature length is 78.

2.5 Feature Selection

The Differential evolution algorithm evaluates the differences among individuals for guiding the Differential evolution which is used for searching the space under the solution. The DE has included various steps of initialization, mutation operation, selection, and crossover operations. The main aim of the proposed IDEOA model is to scale and differentiate various individual vectors which have the same population that is added as a third individual vector of the population. The developed model obtains a mutation for an individual having a vector that has crossed with the individual vector is having the probability of users to attempt. This gets an individual vector and the process of evolution is done by using DE which is as follows:

2.5.1 Initialization

The DE algorithm has utilized D dimensional vectors (M) which are provided with the initial solution. The population number has Nset for each individual that is expressed as $x_i(G) = (x_{i1}(G), x_{i2}(G), \ldots, x_{iD}(G)$. At the first, the population is generated in the range of $[x_{min}, x_{max}]$. M is known as the number of D-dimensional vectors, the population number is N, $x_i(G)$ is the i^{th} an individual which is represented as shown in (Equation 1).

$$x_{iD} = x_{min} + rand(0,1) \times (x_{max} + x_{min}) \tag{1}$$

From the above Equation (1), G is for G^{th} generation, x_{max} is known as the maximum search space value, x_{min} is known as the minimum search space value, and $rand(0,1)$ is known as the random value number that has to reach within normal distribution (0, 1)

2.5.2 Mutation Operation

The mutation operation is performed for generating the mutation vector $V_iG$for each of the individuals in the current population x_iG which is the target vector. The generated target vector corresponds to the mutation vector which is generated for a particular mutation strategy. The different generation methods for mutual individuals are generated with different mutation strategies for forming DE.

2.5.3 Cross-over Operation

The target vectors $x_{i,G}$ of each of them has the corresponding mutation vectors which were crossed and are represented as $V_{i,G}$ generates the test vector. The test vector is represented as $U_{i,G} = (u_{1,G}, u_{2,G} .., u_{i,G})$. The binomial crossover for the DE algorithm is represented which is defined in (Equation 2)

$$u_{i,G} = \begin{Bmatrix} v_{i,G} & if & \left(rand_j(0,1) \leq CR\right) or \left(j = j_{rand}, j = 1,2,3,\ldots,D\right) \\ x_{i,G} & otherwise & \end{Bmatrix} \tag{2}$$

From the above Equation (2), crossover rate (CR) is represented with a constant value that is ranging between [0,1] which is used to control the duplicate proportion. This is used for the mutation vector and j_{rand} is called the random number selected that is ranging between $[1, D]$.

2.5.4 Selection Operation

The parameter values are exceeding the lower and upper bounds that are corresponding to reinitializing uniformly and randomly in the range. The test vectors are evaluated and operated for the objective function that performs the selection operations. The objective function is represented as $f(U_i, G)$ where each of the test values is compared to the objective value. The current population is resembling with the target vector. The main objective function is to test the vector that is consisting of the target vector or the equal value. Therefore, the target vector has been replaced by the test vector that is used in the upcoming generation. The target vector has remained the same with the upcoming generation and has performed the same operation in the selection process which is expressed in (Equation 3)

$$X_{i,G+1} = \begin{Bmatrix} U_{i,G}\, if\, (f(U_{i,G}) \leq f(X_{i,G}) \\ X_i, G \quad Otherwise \end{Bmatrix} \tag{3}$$

The neighborhood mutation strategy population has been performed on evolutionary algorithms like PSO and DE that faces technical problem in the evolution process. The global search ability has shown improvement with exploration ability and convergence speed. Some distinct researchers are successively used by the neighborhood is based the search which is overcome by the problem of poor local search. The shortcoming and DE are easy for falling up to the local optimum which has improved the speed and accuracy. The conventional neighborhood model has the vector which has set the vectors that are connected to other vectors. The performances of the neighbor are directly used for the evolution process that is affected by individual metrics. The neighbor's performance is used directly for the evolution process that affects the individual performances. The geographic location of proximity evaluates based on the weight factors. The increase in the weight factors would lead to diversity for the neighbor vector also.

2.5.5 Improved Differential Evolution Algorithm

There is an increase in the diversity of the neighborhood vector that sorts out the current generation on the basis of the fitness values to perform the mutation. The developed model has selected randomly a part of individual which is ranked in the range of 1-30. The global mutation reduces the operations for generating the higher quality of individuals that improves convergence rate. The model effectively avoids the premature convergence and the mutation operators for the DE have affected directly to search for the algorithm's ability. Therefore, it is important to select the strategy based on the neighborhood. The proposed mutation strategy used for performing the mutation operation that has good global search ability and showed improvement with respect to accuracy convergence and speed. The $DE/best/k$ has introduced the individual information which is the best and can fall in the local optimum. The strategies like $DE/randtobest/k$, $DE/current-to-best/k$, and $DE/current-to-rand/k$ has combined the search capabilities of $DE/rand/k$ showed the convergence speed of $DE/best/k$. The differential evolution algorithm enhances the stability of an algorithm that has been compared with performance for mutation strategies. The JADE has combined $DE/rand/kandDE/current-to-best/k$mutation strategies that have obtained DE/$current-to-pbest$ mutation strategy. The mutation strategy has shown improvement with a breakthrough for improving the convergence effect. The differential evolution algorithm showed higher time complexity but showed limitations because of high dimensional optimization problems. Thus, the shortcomings are solved by the DE/ current-to-p_{best} a strategy that has rules in the selection of individuals to consider the local neighborhood and $DE/current-to-best/1$ mutation strategy that is having the strong ability for improving the convergence effect better. Therefore, the new strategy for neighborhood mutation is named after DE/neighbor-to-neighbor/1, and is designed as shown in (Equation 4)

$$V_i^g = X_{r_3}^g + F_1 \times \left(X_{\text{best}}^g - X_{r_3}^g\right) + F_2 \times \left(X_{r_1}^g - X_{r_2}^g\right)$$

The neighborhood mutation strategy $(DE/neighbor-toneighbor/1)$ has replaced the current evolutionary individual X_i^g with the $DE/current-to-best/1$ having the mutation strategy that is generating random individuality $X_{r_3}^g$ present among the neighborhood. The high-quality information from each of the individual's end in the neighborhood is needed to be executed to perform the operations. An excellent individual among the population has performed better competitiveness during the evolution. Therefore, it is important to show an excellent individual to evolve the descendants that consist of the best fitness value functions. The difference among the excellent individuals is present in the later stages of the evolution which has shown a smaller generation vector. It has shown improvement in the convergence effect provided a higher quality of individuals with respect to the evolutionary generation. The differential evolution algorithm speeded up the convergences and improved the accuracy values.

2.6 Classification using SAE-based DNN

The datasets are having a relationship complexity with features. The single Autoencoder is not sufficient and is unable to reduce the input features' dimensionality. Thus, in such cases, the stacked autoencoder is used that consists of multiple encoders stacked on top of one another. The SAE with 2 encoders is stacked on each other as shown in the following description. The architecture is shown in (Figure 2) and input data is provided to the autoencoder 1. The output obtained by encoder 1 and the input of autoencoder 1 is provided as input to autoencoder 2. Therefore, the input length of autoencoder 2 will be double the auto encoder 1's input. Thus, the SAE based DNN model solves the insufficient data problem to some extent. Figure 2 shows the structure of the Stacked Auto Encoder based on a Deep Neural Network.

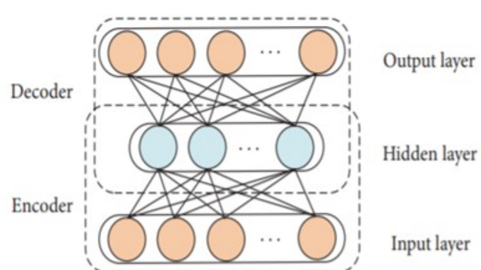

Fig 2. Stacked Auto Encoder based on Deep Neural Network

2.6.1 Implementing Stacked Autoencoders using Matlab

The autoencoder is designed over dense layers of both sides such as encoder and decoder. The number of neurons in each of the encoder and decoder are the same. The mirror reflection of the decoder is the encoder. The autoencoder reduces the number of features that build and compile. The SAE model is built and compiled for fitting the training data. The autoencoder targets the output which is the same as that of the input. The Stacked Auto Encoder is trained at first by the and all the output generated are concatenated. The auto-encoder 2 is ready for considering the input. The SAE is built compiled and trained to autoencoder 2 on a new dataset. The autoencoder 2 is trained that moves toward training the 3rd autoencoder. For the second autoencoder, the input to the 3rd autoencoder helps to build the last two encoders for compiling and training the new data.

Pseudo Code:

Begin

Parameters: The iteration is executed maximal times and the population is having a size of N Dimension of data(D), scale is having a factor of 'F', and crossover probability.

Input: The dataset is initialized for the process of classification.

Output: The Pareto-optimal solutions generated are on the basis of correspondent feature subset.

Step 1: Initialize the individuals

Step 2: Let t = 0. //: set the iteration steps

Step 3: Iteration

The fitness functions of individuals are evaluated.

The individuals are sorted out according to fitness values

Step 3.1: The selection of three vectors is randomly used for the population

Step 3.2: The best feature has been selected by the three vectors which are called the base vector

Step 3.3: The new mutation vector is generated for an individual according to the Neighborhood mutation strategy.

Step 3.4: The trial vector is generated according to Cross over the operation.

Step 3.5: The fitness function is evaluated in the trial vector

Step 3.6: The individuals are Compared

Step 3.7: If the size is larger individuals are removed with higher ranks that are crowded with shorted distances by using a selection algorithm.

The features considered are selected by an individual which is having best fitness value;

Step 4: Else, return to Step 3; otherwise, algorithm is terminated for not obtaining optimal solutions

End

3 Results and Discussion

The proposed hybrid CSP-CNN feature with Multi SVM model is implemented using the MATLAB R2018 software tool and a system with 16 GB of RAM and an i7 processor. The accuracy, precision, sensitivity, and specificity of the performance measures used to analyze the classified arrhythmia signals.

3.1 Performance Analysis

- **Accuracy**

The ratio of the correctly predicted observation to the total number of observations is the accuracy measure. The accuracy is determined by Eq. (5)

$$Accuracy(\%) = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100 \tag{5}$$

- **Specificity**

The ratio of correctly identified negatives to the total number is referred to as specificity as shown in Eq. (6)

$$Specificty(\%) = \frac{TN}{TN+FP} \times 100 \tag{6}$$

- **Recall**

The correctly identified positives that are expressed as shown in Eq. (7) constitute recall.

$$Recall(\%) = \frac{TP}{TP+FN} \times 100 \tag{7}$$

- **F-measure**

The evaluation metric for determining the EEG signals' abnormality and normality effectiveness is the F-measure. The expression contains the F-measure (Equation 8)

$$F - measure(\%) = \frac{2TP}{(2TP+FP+FN)} \times 100 \tag{8}$$

- **Precision**

Precision is known as the ratio of the overall number of positives that are truly classified with the predicted positives. The (Equation 9) is the expression for Precision.

$$Precision = \frac{TP}{TP+FP} \times 100 \tag{9}$$

Whereas, TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative respectively.

3.2 Quantitative Analysis

Table 1 shows the results obtained using various classifiers without the IDEOA evaluated in terms of accuracy, sensitivity, specificity, precision, and f-score.

Table 1. Results obtained by using various classifiers without IDEOA

Without feature selection	Accuracy (%)	Specificity (%)	Precision (%)	Sensitivity (%)	F1-score (%)
SVM	56.43	56.24	57.56	54.24	55.85
KNN	94.57	88.48	90.02	88.59	89.30
RF	91.99	92.99	91.66	92.15	91.90
DT	94.45	94.59	92.27	93.49	92.87
NN	95.02	94.02	93.99	94.78	94.39
DNN	95.73	93.46	94.54	94.32	94.43

The value of accuracy for the Neural Network was obtained as 95.02%, Sensitivity of 94.78%, Specificity of 94.02 %, F1-sore of 94.39 %, and Precision of 93.99 %. The network reduced the certain error value for the sample when the process of training was completed. Similarly, in the case of the SVM model, when the number of features is exceeded for each data point then the training data samples are resulting underperformances. The SVM model obtained 56.43% of accuracy, Sensitivity of 54.24 %, specificity of 56.24 %, F1-sore of 55.85 %, and Precision of 57.56 %. In the case of the Random Forest classifier, the smaller data changes would lead to larger change in the structure. Thus, an optimal decision tree obtained an accuracy of 94.45 %, a sensitivity of 93.49%, specificity of 94.59 %, F1-sore of 92.87 %, and Precision of 92.27 %. Also, the KNN classifier does not work well with high dimensionality as it is difficult to calculate the distance from each of the dimensions thus obtaining 94.57 % of accuracy, the sensitivity of 88.59 %, specificity of 88.48%, F1-sore of 89.30%, and precision of 90.02 %. The present research utilized an ensemble learning model that consisted of multiple classifiers such as NN, SVM, DT, and KNN that are strategically generated by combining to solve the computational intelligence problem. Whereas, the proposed research used SAE acts as a DNN model and obtained accuracy of 95.73%, a sensitivity of 94.32 %, specificity of 93.46 %, F1-sore of 94.43%, and precision of 94.54%. All the aforementioned models have obtained the classification results without the feature selection algorithms. Without the feature selection algorithm, there is a moderate achievement of results due to the complexity created among the features. Whereas, Table 2 shows the results obtained using various classifiers with IDEOA evaluated in terms of accuracy, sensitivity, specificity, precision, and f-score.

The main purpose of feature selection algorithm usage is to increase classification accuracy. If the IDEOA removes the unwanted features which showed complexity during the classification, then the enhancement was performed during

Table 2. Results obtained by using various classifiers with IDEOA

With feature selection	Accuracy (%)	Specificity (%)	Precision (%)	Sensitivity (%)	F1-score (%)
SVM	93.79	94.94	97.31	93.24	95.23
KNN	94.82	94.96	95.57	95.76	95.66
RF	96.69	96.57	92.99	95.18	94.07
DT	96.96	94.09	95.97	95.80	95.88
NN	97.24	97.35	94.98	95.02	95.00
DNN	97.34	98.01	98.54	98.89	98.72

classification. The proposed IDEOA showed improvement in the classification that trains the model faster. The feature selection algorithm has a better representation for searching large spaces that are computationally expensive and thus selected the optimal subset without any of the prior information of features. (Figure 3 a) shows the results obtained in terms of accuracy with and without feature selection.

The proposed technique is compared with existing feature selection techniques of Slap Swarm Algorithm (SSA), Firefly Optimization Algorithm (FOA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO). The PSO algorithm which leads to the low convergence rate in its iterative process with an accuracy of 87.5% where as the ACO algorithm achieves a better accuracy of 92.2% with its limited exploitation and convergence rate. The FOA showed limitations of being trapped in local optima because of it is a local search algorithm that obtained an accuracy of 94.78%. The SSA failed to perform better for real-time applications and thus obtained 96% of accuracy. Whereas the proposed IDEOA uses local neighborhood mutation which showed improvement for high search efficiency which is replaced with large-scale global mutation and opposition based learning model for optimizing the initial population. The direct convergence accelerates the convergence showed improvement in efficiency, enhances stability and avoids falling to the local optimum and thus obtains 97.34 % of accuracy, a sensitivity of 98.89 %, specificity of 98.01%, F1-sore of 98.72%, the precision of 98.54%. (Figure 3 b) shows comparative results for the proposed IDEOA and the existing optimization approaches.

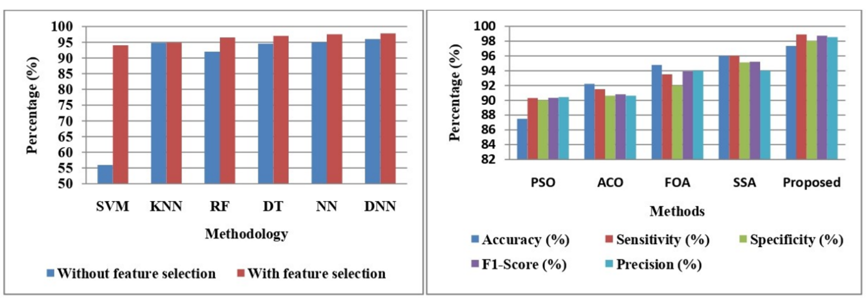

Fig 3. Quantitative Analysis. (a) Results obtained in terms of accuracy for with and without feature selection (b) Results obtained by the proposed IDEOA with other optimization algorithms

3.3 Comparative Analysis

Table 3 represents proposed and the existing methods comparative analysis which are evaluated in terms of accuracy, precision, recall, F1-sore terms.

The existing model showed an increase in the number of filters which were unable to perform better and showed lower accuracies and limited the precision of classification that required improvement in terms of performance for the BCI system. The existing deep learning EEGNet with TSGL model needed hyperparameters for learning but failed to obtain better classification accuracy compared to the MBEEGNet and the existing model used the Difference of Gaussian (DoG) filter which is subjected to discrete approximation issues and has shown variants for exploring the filtering techniques for the combinational approaches. The proposed IDEOA shows an improvement in the convergence effect and obtained better accuracy of 97.34% compared with the existing Sinc-based convolutional neural networks that obtained 75.39 % and TSGL-EEGNet of 81.34 %.

Table 3. Comparative Analysis

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-sore (%)	Specificity (%)
Temporal Convolutional Network Fusion (11)	83.73	84.13	83.78	83.62	-
MBEEGNet (12)	85.59	82.01	82.44	82.22	-
Filter bank common spatial pattern-SVM (13)	67.0	-	-	-	-
TSGL-EEGNet (14)	81.34	-	-	-	-
Sinc-based convolutional neural networks (15)	75.39	-	-	-	-
Proposed method	97.34	98.54	98.89	98.72	98.01

4 Conclusion

In this research, the Improved Differential Evolution Optimization Algorithm (IDEOA) model considers the local neighbor that improves the mutation strategy and showed improvement in the convergence effect. The results obtained by the proposed IDEOA gained 97.34 % of accuracy over the existing Sinc-based convolutional neural networks (75.39 %) and TSGL-EEGNet (81.34 %) respectively. The datasets are having a relationship complexity with features and the existing modules which used single Autoencoder which is not sufficient and is unable to reduce the input features' dimensionality. The proposed module uses stacked Autoencoder that consists of multiple encoders stacked on top of one another. Thus, the SAE based DNN model solves the insufficient data problem to some extent.

References

1) Altuwaijri GA, Muhammad G. Electroencephalogram-Based Motor Imagery Signals Classification Using a Multi-Branch Convolutional Neural Network Model with Attention Blocks. *Bioengineering*. 2022;9(7):323. Available from: https://doi.org/10.3390/bioengineering9070323.
2) Rahman MKM, Joadder MAM. A space-frequency localized approach of spatial filtering for motor imagery classification. *Health Information Science and Systems*. 2020;8(1):15–15. Available from: https://doi.org/10.1007/s13755-020-00106-8.
3) Molla MKI, Shiam AA, Islam MR, Tanaka T. Discriminative Feature Selection-Based Motor Imagery Classification Using EEG Signal. *IEEE Access*. 2020;8:98255–98265. Available from: https://doi.org/10.1109/ACCESS.2020.2996685.
4) Shovon TH, Nazi ZA, Dash S, Hossain MF. Classification of Motor Imagery EEG Signals with multi-input Convolutional Neural Network by augmenting STFT. *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*. 2019;p. 398–403. Available from: https://doi.org/10.1109/ICAEE48663.2019.8975578.
5) Tang R, Li Z, Xie X. Motor imagery EEG signal classification using upper triangle filter bank auto-encode method. *Biomedical Signal Processing and Control*. 2021;68:102608. Available from: https://doi.org/10.1016/j.bspc.2021.102608.
6) Attallah O, Abougharbia J, Tamazin M, Nasser AA. A BCI System Based on Motor Imagery for Assisting People with Motor Deficiencies in the Limbs. *Brain Sciences*. 2020;10(11):864. Available from: https://doi.org/10.3390/brainsci10110864.
7) Dong E, Zhou K, Tong J, Du S. A novel hybrid kernel function relevance vector machine for multi-task motor imagery EEG classification. *Biomedical Signal Processing and Control*. 2020;60:101991. Available from: https://doi.org/10.1016/j.bspc.2020.101991.
8) Xu L, Xu M, Jung TP, Ming D. Review of brain encoding and decoding mechanisms for EEG-based brain–computer interface. *Cognitive Neurodynamics*. 2021;15(4):569–584. Available from: https://doi.org/10.1007/s11571-021-09676-z.
9) Xu S, Zhu L, Kong W, Peng Y, Hu H, Cao J. A novel classification method for EEG-based motor imagery with narrow band spatial filters and deep convolutional neural network. *Cognitive Neurodynamics*. 2022;16(2):379–389. Available from: https://doi.org/10.1007/s11571-021-09721-x.
10) Huang Z, Qiu Y, Sun W. Recognition of motor imagery EEG patterns based on common feature analysis. *Brain-Computer Interfaces*. 2021;8(4):128–136. Available from: https://doi.org/10.1080/2326263X.2020.1783170.
11) Xie J, Zhang J, Sun J, Ma Z, Qin L, Li G, et al. A Transformer-Based Approach Combining Deep Learning Network and Spatial-Temporal Information for Raw EEG Classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2022;30:2126–2136. Available from: https://doi.org/10.1109/TNSRE.2022.3194600.
12) Musallam YK, Alfassam NI, Muhammad G, Amin SU, Alsulaiman M, Abdul W, et al. Electroencephalography-based motor imagery classification using temporal convolutional network fusion. *Biomedical Signal Processing and Control*. 2021;69:102826. Available from: https://doi.org/10.1016/J.BSPC.2021.102826.
13) Altuwaijri GA, Muhammad G. A Multibranch of Convolutional Neural Network Models for Electroencephalogram-Based Motor Imagery Classification. *Biosensors*. 2022;12(1):22. Available from: https://doi.org/10.3390/bios12010022.
14) Ghumman MK, Singh S, Singh N, Jindal B. Optimization of parameters for improving the performance of EEG-based BCI system. *Journal of Reliable Intelligent Environments*. 2021;7(2):145–156. Available from: https://doi.org/10.1007/s40860-020-00117-y.
15) Deng X, Zhang B, Yu N, Liu K, Sun K. Advanced TSGL-EEGNet for Motor Imagery EEG-Based Brain-Computer Interfaces. *IEEE Access*. 2021;9:25118–25130. Available from: https://doi.org/10.1109/ACCESS.2021.3056088.
16) Bria A, Marrocco C, Tortorella F. Sinc-Based Convolutional Neural Networks for EEG-BCI-Based Motor Imagery Classification. In: Pattern Recognition. ICPR International Workshops and Challenges;vol. 1. Springer International Publishing. 2021;p. 526–535. Available from: https://doi.org/10.1007/978-3-030-68763-2_40.

17) Jayashekar V, Pandian R. Hybrid Feature Extraction for EEG Motor Imagery Classification Using MultiClass SVM. *International Journal of Intelligent Engineering and Systems*. 2022;15(4):20–30. Available from: https://doi.org/10.22266/ijies2022.0831.03.