International Research Journal of Multidisciplinary Scope (IRJMS), 2026; 7(1): 545-555

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2026.v07i01.09252

Discriminated-SDS: A Novel Hybrid Approach for Optimizing EEG Based Brain-Computer Interface Signals Faced by Metaheuristic Algorithms

Sanoj Chakkithara Subramanian*, Daniel D

Department of Computer Science and Engineering, School of Engineering and Technology, CHRIST (Deemed to be University), Bengaluru, Karnataka, India. *Corresponding Author's Email: sanoj.cs@res.christuniversity.in

Abstract

Brain Computer Interfaces (BCIs) will convert the thoughts of individuals with physical disabilities into commands for devices to enable them autonomous mobility. The Electroencephalogram (EEG) is widely favoured as a control signal due to its ease of acquisition compared to invasive recordings. While the affordability of EEG equipment allows for the use of numerous recording channels, this abundance increases computational complexity, necessitating optimal channel selection strategies to improve efficiency and classification accuracy. Deep Neural Networks (DNNs) often face scalability issues with multidimensional, locally correlated inputs, making them impractical for such applications. Convolutional Neural Networks (CNNs) are efficient for analysing BCI data but require careful hyperparameter tuning to achieve optimal performance. This paper introduces a framework for classifying BCI channel selection using deep learning techniques. The study primarily concentrates on refining the hyper parameters of deep learning algorithms through metaheuristic techniques, specifically employing Discriminated Stochastic Diffusion Search (SDS) to enhance BCI channel selection. The findings indicate that the proposed hyperparameter optimization methods, such as Discriminated-SDS, significantly enhance classification accuracy. The proposed D-SDS balances exploration and exploitation, mitigates the local optima issue, and is especially advantageous for intricate deep learner architectures such as VGGNet, ResNet, and InceptionNet. Hyperparameter optimization in EEG-based BCI systems can substantially improve performance, enhancing their efficiency and reliability.

Keywords: BCI Channel Selection, Deep Learning, Discriminated-SDS, Electroencephalogram (EEG), Hyper-Parameter, Stochastic Diffusion Search (SDS).

Introduction

Brain-Computer Interfaces (BCIs) facilitate the translation of neural impulses into commands for the operation of external equipment. BCIs are categorized based on the recording methodology: non-invasive and invasive. Electroencephalography (EEG), a non-invasive technique that captures signals from the scalp's surface, is the most favoured approach owing to its affordability and easier implementation (1). Invasive BCIs capture signals by many techniques, including Electrocorticography, spiking activity, and Stereo-Electroencephalography. These approaches can yield a superior signal-to-noise ratio compared to non-invasive approaches, so they are of considerable importance in the advancement of high-performance BCI. Recent demonstrations have shown the efficacy of brain-computer interfaces via an invasive stereo electroencephalography paradigm. Ease of acquisition is crucial for EEG-based BCI applications because it ensures non-invasive, portable, and affordable signal collection, making the technology accessible and user-friendly. This practicality enables real-time interaction and scalability for diverse applications, such as assistive devices, gaming, and neuro feedback, ensuring widespread adoption (2). BCIs follow a three-stage process of pre-processing signals and filtering, extraction of features, and signal classification. Pre-processing is primarily dependent upon the method of data capture and typically encompasses broad-band filtering and trend elimination. Feature engineering is conducted to derive valuable information or statistical characteristic from a signal. Frequency representation is a frequently utilized characteristic. Dimensionality reduction is frequently employed to decrease computation time and mitigate over fitting through Principal Component Analysis and Independent Component Analysis. Classification is accomplished by machine learning

(Received 17th September 2025; Accepted 11th December 2025; Published 16th January 2026)

algorithms or deep learning methods (3, 4). The extensive array of channels may result in high-dimensional data, increasing computing complexity and possible noise, which might impair the efficacy of classification algorithms. Processing all channels necessitates considerable computational resources, rendering real-time applications difficult. Not all channels contribute equally to the task, as some carry greater noise or redundant information. Wrapper approaches assess feature subsets according to their impact on the efficacy of a particular classifier (5). The affordability of EEG equipment enables the use of numerous channels for detailed brain activity recording; however, managing multiple channels can increase complexity and computational cost. This drives research into optimal channel selection, focusing on achieving high performance with fewer channels to balance efficiency and practicality in real-world applications.

Deep learning has been widely applied in brain-computer interfaces and has proven effective in overcoming the aforementioned challenges. Deep learning offers two advantages: firstly, it operates directly on raw brain signals to extract important information using back-propagation, thereby circumventing pre-processing stage. Secondly, the intricate architectures of deep neural networks can encapsulate both high-level information and latent dependencies. The learning rate is a crucial hyperparameter in deep learning, influencing the convergence velocity and stability of the training procedure. Determining suitable decay learning rates can improve the efficacy and resilience of deep learning models in EEG categorization (6, 7).

Hyperparameter optimization entails exploring a high-dimensional environment characterized by intricate and non-linear interactions between hyper parameters and model performance. Metaheuristic algorithms are engineered to efficiently explore and exploit intricate search areas. Optimal adjustment of hyper parameters, including batch size, learning rate, dropout rate, layer count, can significantly enhance model accuracy, robustness, and generalization abilities. Appropriately selected hyper parameters can accelerate the training process and enhance resource efficiency, which is essential for computationally intensive deep learning models (8, 9).

Metaheuristic algorithms, such as Particle Swarm Optimization, Genetic Algorithm, and Simulated Annealing, are effective instruments for addressing intricate optimization challenges. A prevalent issue encountered by these algorithms is their tendency to become trapped in local optima, resulting in inferior solutions (10, 11). This study proposes the development of improved variations, including the D-SDS algorithm, to address this constraint and enhance the exploration-exploitation equilibrium. This study's primary contributions include:

This study proposes the use of D-SDS for optimizing channel selection in EEG-based BCIs, reducing computational complexity while maintaining high classification accuracy.

The framework integrates D-SDS with deep learning models such as VGGNet, ResNet, and InceptionNet, effectively optimizing hyper parameters by balancing exploration and exploitation, mitigating local optima issues, and addressing scalability challenges in complex neural networks.

An innovative bagging algorithm that creates interpolation data surrounding misclassified instances utilizing a possibility function, intended for application in BCIs was presented in past study (12). Unlike AdaBoost, which increases the weight of misclassified instances for future training, in this past study they generate virtual data using a membership function focused on these instances and add them right away to subsequent datasets. This expands the training set and allows for finer tuning of the discriminative boundary. They propose a bagging-style ensemble method based on possibility distribution and assess its effectiveness for basic computations with NIRS data.

Research groups comprise the Novel Adaboost Classifier Algorithm and MLP (13). The collection, comprising 64.4 MB, has 30 photos of healthy brains and 30 images of injured brains. Kaggle supplied pictures for the identification of brain strokes. To identify the training dataset, 500 records are necessary, with 80% allocated for training and 20% for testing. A distinctive Adaboost technique and a multilayer perceptron approach were evaluated utilizing ten datasets to enhance research precision. Twenty sets were compared ten times per iteration. The G power test with α=0.05 and β=0.2 results in an approximate power of 80%. A unique Adaboost classifier method (88.84%) outperforms MLP (83.40%). The independent sample test value is significant at 0.01 ($p<0.05$).

The essential challenge of categorizing brain-wave patterns linked to specific brain states, employing various machine learning algorithms (14). The main objectives include constructing and enhancing these models, evaluating the effects of adjusting hyper parameters on measures such as accuracy, consistency, and prediction time. Amongst the various machine learning models studied, Decision Tree exhibits performed the best consistently with 90.03% accuracy. After hyperparameter adjustment, Support Vector Machine and Linear Regression demonstrate significant accuracy improvements of 15.63% and 1.50%, respectively, hence boosting the consistency of all models.

An BCI based on Optimal Deep Learning (ODL) model, termed ODL-BCI, which was refined using hyperparameter tuning methods to overcome the challenges of real-time classification of student confusion (15). Utilizing the "confused student EEG brainwave" dataset, the Bayesian optimization is applied to optimize the hyper parameters of the proposed deep learning model. The model includes input, output layers and multiple hidden layers, where the activation functions, learning rates and number of nodes are established using specified hyper parameters. The researchers compare their model in the past study with leading techniques and traditional classifiers, finding that the ODL-BCI performs best. It enhances accuracy by 4% to 9% compared to existing methods, surpassing all other classifiers in the process.

An innovative method for extracting spatial-temporal features from EEG data using stacked multi-ScaleCNNs was introduced in past research (16). This approach processes raw EEG signals collected from geographically dispersed electrodes, capturing spatial features from a 2D grid of electrode placements via multi-scale 2D convolutions, while temporal features are extracted using varying window sizes. To address the issue of imbalanced class representation, in the past study, the researcher proposed a fine-tuning ensemble strategy with dynamic class weights, ensuring better learning of minority class samples. Testing across datasets with diverse class distributions revealed that integrating this fine-tuning ensemble into deep learning models consistently improves performance across both sessions and subjects. Ensemble Multi-Scale CNN surpasses current methods in cross-subject and cross-session tasks, as shown by thorough experiments.

Methodology

Despite the availability of numerous models in the literature, there is a necessity to improve classification accuracy for BCI applications. As deep learning models become deeper, their parameters increase quickly, causing over fitting. Simultaneously, many hyper parameters substantially influence the efficacy of the CNN model. Specifically, hyper parameters including epoch count, batch size, and learning rate selection are crucial for achieving effective results (17, 18).

Hyper parameters considered in this study are:

a) The learning rate regulates the magnitude of each step in the gradient descent process.
b) Batch size influences the stability and velocity of training.
c) The number of epochs indicates the frequency with which the complete training dataset is processed by the model.

Appropriate weight initialization can avert vanishing or exploding gradients. Dropout is employed to mitigate over fitting by randomly deactivating a proportion of input units during the training process. Due to the laborious and imperfect nature of the trial-and-error approach for hyperparameter tuning, the application of metaheuristic algorithms is useful. This section discusses the hyperparameter tuning of SDS, VGGNET, RESNET, InceptionNet, and discriminated-SDS algorithms.

Hyperparameter Tuning Using SDS

Establish a preliminary cohort of agents. Each agent signifies a hypothesis, which in this instance refers to a particular configuration of hyper parameters for the deep learning model (i.e., learning rate, number of epochs, dropout rate, batch size). Randomly initialize the hyper parameters for each agent within specified ranges. Batch size: [32, 256]; Learning rate: [1e-5, 1e-1]; Dropout rate: [0, 0.5]; Number of epochs: [50, 200]. Train the deep learning model (VGGNet/ResNet) utilizing the hyper parameters given by each agent. Assess the model's performance on a validation set by computing a fitness metric, such as validation accuracy or validation loss. Reiterate the stages of hypothesis development, evaluation, communication, and decision-making for a certain number of iterations (19).

Stochastic Diffusion Search (SDS) presented an innovative probabilistic methodology for addressing optimal pattern recognition and matching challenges. SDS is a distributed computational method employing interaction among simple agents for global search and optimization within a multi-agent population framework (20).

In contrast to numerous search algorithms inspired by nature, SDS provides a well-defined mathematical structure that accounts for its approach to resource distribution, assured global convergence, robustness, minimal prerequisites, and efficient linear time complexity.

The steps in SDS algorithm are.

a) Agents are created with randomly assigned hyper parameters as candidate solutions.
b) The process repeats until the stopping criterion is met.
c) Each agent partially evaluates its current hyper parameters for performance.
d) Agents are sorted as active (promising) or inactive (requiring changes).
e) Active agents share successful hypotheses with others.
f) Agents exchange information and may adopt superior hypotheses.
g) The algorithm ends with selection of the best solution found.

In every SDS search, each agent upholds a hypothesis, h, delineating a potential solution to the problem. Following initialization, two steps ensue, Testing Phase and Diffusion Phase (e.g. gathering and dissemination of knowledge).

SDS tests the agent hypothesis with a partial evaluation during testing, producing a domain-independent Boolean result. Subsequently, depending on the utilized method, effective hypotheses disseminate among the population, hence facilitating the transmission of information regarding possibly optimal solutions throughout the complete agent population. In the Testing phase, every agent conducts a partial evaluation of its own hypothesis, represented as pFE, where pFE is determined by applying a specific function to the agent's hypothesis (pFE = f(h)). In the Diffusion phase, agents interact by selecting peers, allowing them to potentially exchange their current hypotheses.

AdaBoost is an ensemble learning algorithm where set weak classifiers is used to improve performance accuracy by adjusting the weights assigned to each training sample. This algorithm boosts the accuracy of a weak classifier, which may initially make inaccurate predictions, by iteratively refining it into a stronger classifier with higher predictive accuracy (21). AdaBoost's key hyper parameters include the number of estimators (n estimators), which determines number of weak learners, as more estimators may improve accuracy but end up with a over fitted ensemble. The learning rate (learning rate) controls the contribution of each weak learner to the final model; a lower learning rate can enhance generalization but typically requires more estimators. Additionally, the base estimator (base estimator) specifies the type of weak learner, often a decision tree by default, but it can be any classifier capable of fitting the data.

The VGG network is a pre-trained CNN model that gained significant recognition in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC2014). It achieved an impressive 92.7% accuracy on the ImageNet dataset, which contains 14 million images across 1,000 categories. The architecture consists of five convolutional blocks, with each block containing an increasing number of filters—64, 128, 256, 512, and 512, respectively. Each block is followed by a max-pooling layer for spatial pooling. The network concludes with three fully connected layers that employ dropout to prevent over fitting. The final layer uses the SoftMax activation function to output probability values for each of the classes (22).

The ResNet architecture was initially introduced for object detection (23). ResNet facilitated the efficient training of deeper networks by incorporating a shortcut that connects the output of the residual block to its input, so mitigating the vanishing gradient issue. In a study evaluating artificial neural networks for object identification, ResNet-101, a variant of the ResNet family, was identified as one of the most robust models for predicting neural and behavioural data. The ResNet architecture, utilizing fNIRS-based deep learning, demonstrated excellent performance in decoding RPS motions during motor imagery.

Consequently, ResNet101 was chosen for model comparison, utilizing 125-Hz resampled data as input to ensure consistency across various architectures. EEGNet and ResNet had a total of 4,691 and 42,626,435 parameters, respectively.

The Inception architecture has played a pivotal role in driving advancements within deep learning frameworks. This network is available in two forms: the basic inception model and a variant that incorporates dimensionality reduction techniques. The naive inception model comprises several blocks. Each block comprises several convolutional layers operating at the same level. Each layer comprises many filters with designated kernel sizes (1×1, 3×3, and 5×5). The concatenated outputs are diminished by the use of a max-pooling operation and forwarded to the subsequent inception block.

Proposed Discriminated Stochastic Diffusion Search (D-SDS)

The Lévy flight foraging theory posits that because Lévy flights maximize random searches, biological species have consequently evolved to utilize Lévy flights.

A Lévy flight is a process in which, at each time step j, the random walker executes an instantaneous jump lj selected from a probability density function P(l) characterized by a power law tail in the long-distance regime as described in equation [1]:

$$P(l) \sim (1)^{-\mu} \quad [1]$$

with $1 < \mu \leq 3$. For $\mu < 3$, the second moment of P(l) diverges, and for $\mu < 2$, the first moment likewise diverges. Lévy walks and flights result in super diffusion, where the mean squared displacement of the walker's position in equation [2]:

$$(x^2) \sim t^{2H} \quad [2]$$

scales super linearly with time t. Lévy flights permit H > 1/2. This is in contrast to diffusive walks, for which the Hurst exponent H is equal to 1/2.

Lévy flights and walks exhibit scale invariance and fractal characteristics. A magnification of a segment of a genuine Lévy walk trajectory will disclose a substructure with statistically equivalent characteristics, except cut-offs. Lévy walks exhibit uniformity across all scales, with the exception of cut-offs. Their diffusive characteristics are independent of scale. A key notion is to the utilization of μ to compare super diffusive Lévy searches with Brownian searches characterized by normal diffusion. By analysing the search efficiencies as μ is varied, we may ascertain the extent of advantage derived from leveraging diffusivity and stochasticity in this particular method (24).

D-SDS is an advanced iteration of the conventional SDS algorithm, aimed at mitigating particular constraints, including the local optima issue, and enhancing the convergence rate. Levy Flights are a form of random walk distinguished by extensive leaps, which can markedly improve the algorithm's capacity to investigate remote and potentially superior areas of the search space (25, 26).

The D-SDS incorporates two more phases inside the SDS algorithm.

Levy Flight Step: Execute Levy Flight steps to intermittently relocate agents to remote areas of the search space. This facilitates the avoidance of local optima and guarantees enhanced exploration. The Levy Flight step is mathematically expressed as in equation [3]:

$$L(\lambda)=1/〚|\Delta t|〛\text{^}(1+\lambda) \quad [3]$$

Where Δt represents a minor time increment, and λ is the Levy distribution parameter (often $0< \lambda \leq 2$).

Implement an adaptive fitness threshold method to direct the agents in the search process. Modify the fitness threshold adaptively according to the population's performance over iterations. The population's fitness exhibit minimal enhancement over several iterations, up the threshold to promote exploration of novel regions. If substantial improvement is noted, lower the barrier to emphasize exploitation.

D-SDS was chosen over other meta-heuristic techniques due to its unique ability to balance exploration and exploitation, which is critical for optimizing complex deep learning architectures. Unlike conventional methods, D-SDS effectively mitigates the risk of local optima using discrimination strategies, making it particularly suitable for handling the high-dimensional, locally correlated inputs typical of EEG-based BCI systems. The proposed D-SDS methodology can be observed from Figure 1 and the two integral algorithmic components which are represented in subsequent sections (Table 1 and 2).

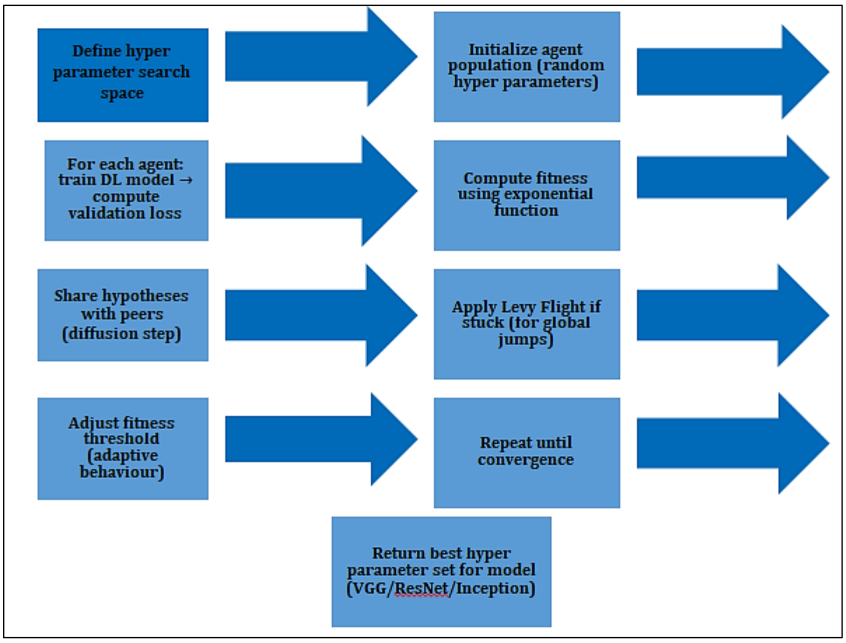

Figure 1: Methodology of Proposed Discriminated SDS

Table 1: Algorithm - Discriminated-SDS for Channel Selection

Algorithm - Discriminated-SDS for Channel Selection

Input:

EEG dataset D with N channels

Initial parameters: α, β, λ, max_iter, δ, threshold bounds

Classifier model (e.g., Adaboost, VGG)

Output:

Optimal subset of EEG channels

1. Initialize population of agents with random channel subsets
2. Set initial fitness threshold T
3. For each agent:
    a. Randomly select a hypothesis (subset of channels)
4. Repeat until max_iter or convergence:
    a. Evaluate fitness using:
        $f(x) = \alpha * \exp(-1/C) + \beta * MSE$
    b. For each agent:
        i. Compare fitness with a randomly selected neighbor
        ii. If worse:
        - Replace hypothesis using neighbor's OR
            - Apply Levy Flight with probability p_levy
        iii. If better:
        - Retain current hypothesis
    c. Update adaptive threshold T:
        If mean fitness improvement < ε for k iterations → T += δ
        Else → T -= δ
5. Select agent(s) with highest fitness scores
6. Return best-performing subset of EEG channels

End

Table 2: Algorithm - Discriminated-SDS for Hyper parameter Tuning

Algorithm - Discriminated-SDS for Hyperparameter Tuning

Input:
EEG dataset D
Deep learning model architecture M (VGGNet / ResNet / InceptionNet)
Hyperparameter search space:
Learning rate ∈ [0.00001, 0.0001, 0.001, 0.01, 0.1]
Batch size ∈ {32, 64, 128, 256}
Dropout rate ∈ [0, 0.1,…,0.5]
Epochs ∈ [50,…., 200]
Number of agents N
Maximum iterations T
Fitness weights α=0.6, β=0.4
Output:
Optimal hyperparameter configuration H* with highest fitness score
Initialize:
Create a population of N agents.
Assign each agent a random hyperparameter configuration Hi.
Evaluate Initial Fitness:
For each agent:
Train model M with Hi
Compute validation loss (MSE) and model complexity penalty
Evaluate fitness
Repeat for each iteration (1 to T):
a. Diffusion Step:
For each agent Ai:
Randomly select peer agent Aj
If f(Hj)>f(Hi):
Replace Hi with Hj
Else:
Retain Hi
b. Stagnation Check and Levy Flight:
If no significant improvement in average fitness for k iterations:
Apply Levy Flight to generate new Hi for exploration.
c. Re-evaluate Fitness:
Train model using updated configurations
Update fitness scores
Select Optimal Agent:
Identify agent A* with highest fitness score
End

Results

In this section, evaluate the SDS and D-SDS algorithm, the classification methods using Adaboost, hyperparameter tunned VGGNET, hyperparameter tunned Resnet and hyperparameter tunned InceptionNet. BCI Competition III – Dataset IVa was used for evaluating the algorithms. Motor Imagery (MI) tasks of 5 subjects were used. Each subject has recordings from multiple sessions. The first two sessions are for training; the third is for testing. The summary of results is presented in Table 3. The accuracy, precision, recall and F-measure as shown in Figure 2 to 5.

Table 3: Summary of Result

	SDS Adaboost	D-SDS Adaboost	SDS Hyperparameter tuned VGGNET	SDS HyperParameter Tuned Resnet	SDS Hyperparameter Tuned InceptionNet	D-SDS Hyperparameter tuned VGGNET	D-SDS HyperParameter Tuned Resnet	D- SDS Hyperparameter Tuned InceptionNet
Classification accuracy	0.9194	0.9324	0.9378	0.9564	0.9617	0.9778	0.9731	0.9903
Precision for right	0.9209	0.9352	0.9378	0.9543	0.9655	0.9804	0.971	0.983
Precision for foot	0.9123	0.9301	0.9272	0.9381	0.9649	0.9894	0.9555	0.9746
Recall for right	0.9363	0.9529	0.9509	0.964	0.9835	0.9819	0.9805	0.9964
Recall for foot	0.8928	0.9104	0.9045	0.9213	0.9408	0.9665	0.9378	0.949
F measure for right	0.9288	0.944	0.9412	0.9511	0.974	0.9936	0.9684	0.9836
F measure for foot	0.9057	0.9231	0.9164	0.9269	0.9518	0.9727	0.9434	0.959

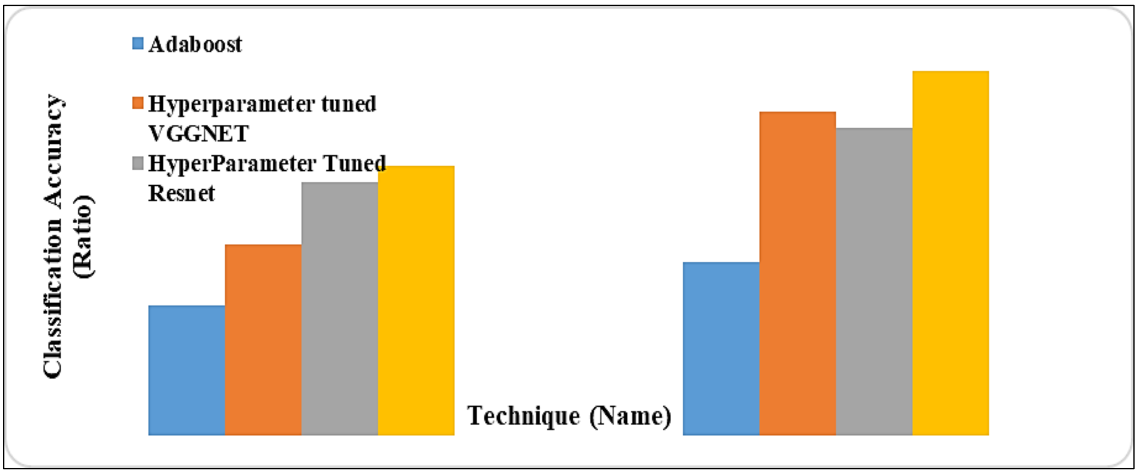

Figure 2: Accuracy for D-SDS

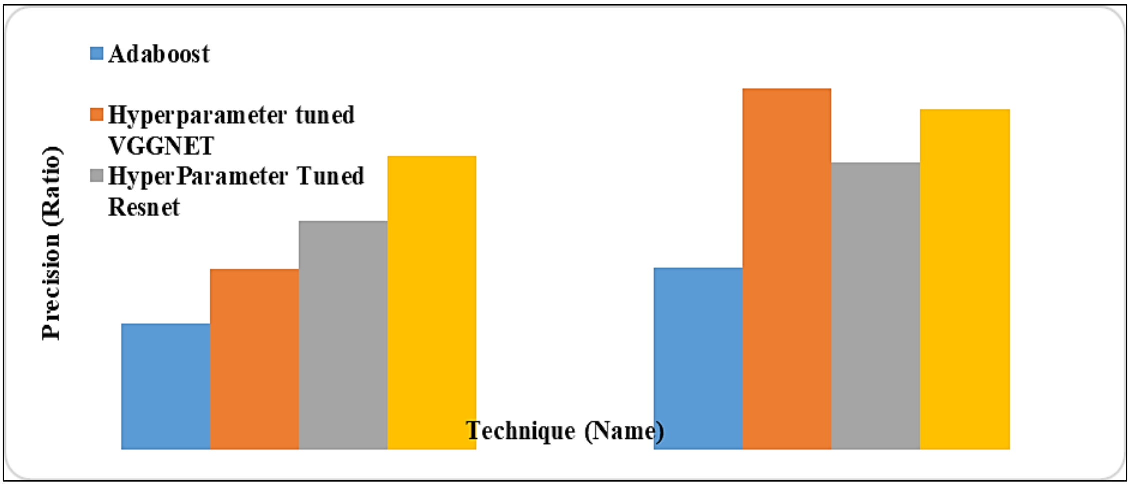

Figure 3: Precision for D-SDS

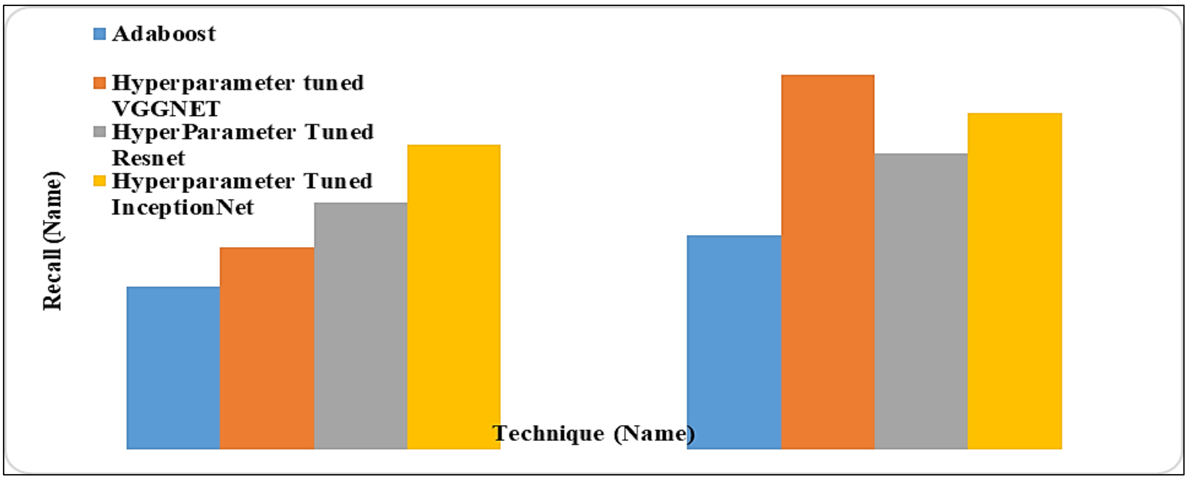

Figure 4: Recall for D-SDS

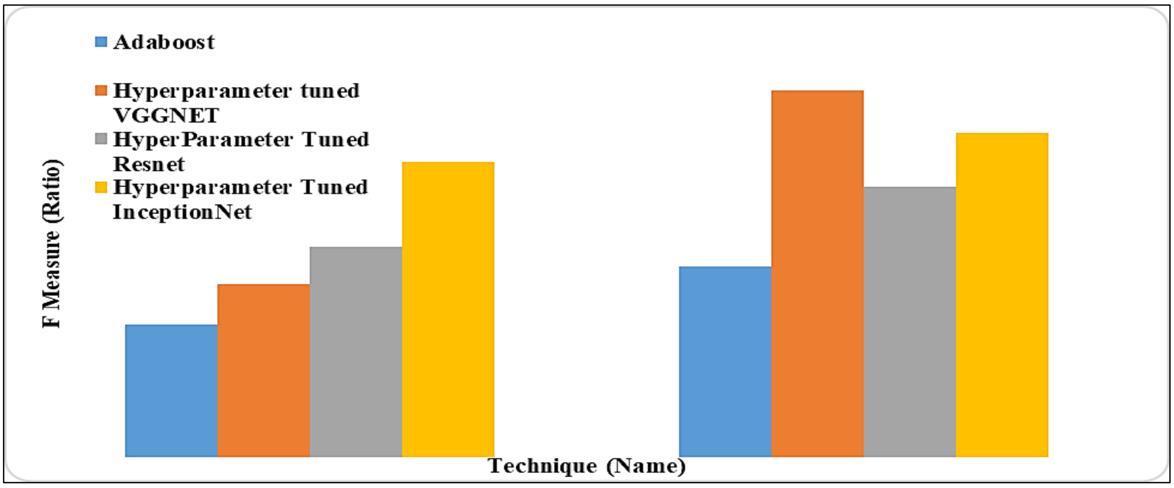

Figure 5: F Measure for D-SDS

Discussion

The baseline Adaboost classifier achieved an accuracy of 0.9324 when optimized using D-SDS as observed in Figure 2. This demonstrates the enhanced capability of Discriminated-SDS to escape local optima and better optimize hyper parameters. D-SDS dramatically improved the accuracy to 0.9778. This substantial enhancement underscores the advantage of D-SDS in fine-tuning the hyper parameters of deep CNNs like VGGNet, leading to near-optimal configurations. The accuracy further increased to 0.9731 with D-SDS Inception. This result indicates that D-SDS effectively addresses the local optima problem and enhances ResNet's performance by better navigating the hyperparameter space.

The baseline Adaboost classifier, when optimised using D-SDS, achieved a precision of 0.932, as shown in Figure 3. Hyperparameter tuned VGGNet classifier achieved precision of 0.984 when optimized using D-SDS. Hyperparameter tuned ResNet classifier achieved the precision of 0.963 when optimized using D-SDS. Hyperparameter tuned InceptionNet classifier achieved the precision of 0.978 when optimized using D-SDS. It can be observed in Figure 4 that the baseline Adaboost classifier achieved the recall of 0.932 when optimized using D-SDS. Hyperparameter tuned VGGNet classifier achieved recall of 0.986 when optimized using D-SDS. Hyperparameter tuned ResNet classifier achieved the recall of 0.959 when optimized using D-SDS. Hyperparameter tuned InceptionNet classifier achieved the recall of 0.973 when optimized using D-SDS.

It can be observed in Figure 5 that the baseline Adaboost classifier achieved the F-measure of 0.9335 when optimized using D-SDS. Hyperparameter tuned VGGNet classifier achieved F-measure of 0.983when optimized using D-SDS. Hyperparameter tuned ResNet classifier achieved the F-measure of 0.956 when optimized using D-

SDS. Hyperparameter tuned InceptionNet classifier achieved the F-measure of 0.971 when optimized using D-SDS.

Conclusion

Hyperparameter optimization plays a crucial role in enhancing machine learning model performance by ensuring better generalization, preventing over fitting, stabilizing training, and improving computational efficiency. This study highlights the efficacy of SDS and D-SDS in navigating the hyperparameter space for EEG-based BCIs. By dynamically refining exploration techniques and fitness evaluation criteria, Discriminated-SDS demonstrated significant improvements across all models. For instance, Adaboost's accuracy improved from 0.9194 with SDS to 0.9324 with D-SDS, VGGNet increased from 0.9378 to 0.9778, ResNet rose from 0.9564 to 0.9731, and InceptionNet achieved the most significant enhancement, from 0.9617 to 0.9903.

The study's findings emphasize the transformative potential of D-SDS for optimizing deep learning architectures and ensemble techniques, setting a benchmark for future research. These advancements provide a scalable framework for addressing high-dimensional data challenges, with implications extending beyond BCIs to other domains such as natural language processing and medical imaging. Future work could focus on applying this framework to real-time BCI systems, enhancing computational efficiency, and integrating it with more advanced architectures to further improve performance and adaptability in diverse applications.

Abbreviations

BCI: Brain Computer Interfaces, CNN: Convolutional Neural Networks, D-SDS: Discriminated Stochastic Diffusion Search, DNN: Deep Neural Networks, EEG: Electroencephalo-gram, fNIRS: Functional Near -Infrared Spectroscopy, InceptionNet: Inception Network, ODL: Optimal Deep Learning, ResNet: Residual Network, SDS: Stochastic Diffusion Search, VGGNet: Visual Geometry Group.

Acknowledgment

None.

Author Contributions

Both authors collaboratively contributed to the conception, methodology, analysis, and manuscript writing of the submitted work. Both authors have reviewed and approved the final draft of the paper.

Conflict of Interest

The authors declare no conflict of interest.

Declaration of Artificial Intelligence (AI) Assistance

I confirm that this work was completed without any AI assistance

Ethics Approval

Not applicable.

Funding

Not applicable.

References

1. Guger C, Grünwald J, Xu R. Noninvasive and invasive BCIs and hardware and software components for BCIs. Handbook of Neuroengineering. Singapore: Springer Nature Singapore, 2023; 1193-1224.
2. Kumar S, Sharma A. Advances in non-invasive EEG-based brain-computer interfaces: Signal acquisition, processing, emerging approaches, and applications. Signal Processing Strategies. 2025; 281–310.
3. Wu X, Jiang S, Li G, Liu S, Metcalfe B, Chen L, Zhang D. Deep learning with convolutional neural networks for motor brain-computer interfaces based on stereo-electroencephalography (SEEG). IEEE J Biomed Health Inform. 2023;27(5):2387-2398.
4. Hossain KM, Islam MA, Hossain S, Nijholt A, Ahad MAR. Status of deep learning for EEG-based brain–computer interface applications. Front Comput Neurosci. 2023; 16:1006763.
5. Alotaiby T, El-Samie FEA, Alshebeili SA, Ahmad IA. review of channel selection algorithms for EEG signal processing. EURASIP Journal on Advances in Signal Processing, 2015;(1),66.
6. Chen L, Deng W, Ji C. Application of convolutional neural network method in brain computer Interface. Journal of Physics Conference Series. 2021; 2078(1):012044.
7. Mansoor A, Usman MW, Jamil N, Naeem MA. Deep learning algorithm for brain–computer interface. Scientific Programming. 2020; 1-12.
8. Abd Elaziz M, Dahou A, Abualigah L, *et al.* Advanced metaheuristic optimization techniques in applications of deep neural networks: a review. Neural Comput Appl. 2021;33(21):14079–14099.
9. Nematzadeh S, Kiani F, Torkamanian-Afshar M, Aydin N. Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases. Comput Biol Chem. 2022; 97:107619.

10. Zhang Y, Wang S, Ji G. A comprehensive survey on particle swarm optimization algorithm and its applications. Mathematical Problems in Engineering. 2015; 1–38.
11. Turgut OE, Turgut MS, Kırtepe E. A systematic review of the emerging metaheuristic algorithms on solving complex optimization problems. Neural Comput Appl. 2023; 35:14275–14378.
12. Hayashi I, Irie H, Tsuruse S. Bagging algorithm based on possibilistic data interpolation for brain-computer interface. J Adv Comput IntellIntell Inform. 2024;28(3):623-633.
13. Manikanta P, Kannan KN, Padmakala S. Detection of Brain Stroke by using the Novel Adaboost Classifier Algorithm Classifier Algorithm Compared with Multi-layer Perceptron. IEEE. 2024; 29:1–6.
14. Lee PF, Chong KY. Machine learning for physical motion identification using EEG signals: a comparative study of classifiers and hyperparameter tuning. J Ambient Intell Humaniz Comput. 2024; 15(4):2455-2466.
15. Miah MO, Habiba U, Kabir MF. ODL-BCI: Optimal deep learning model for brain-computer interface to classify students confusion via hyperparameter tuning. Brain Disorders. 2024;13;100121.
16. Wang X, Dang M, Yang K, Cui X, Zhang D, Chen C. The ensemble multi-scale convolution neural network for visual target detection EEG-based brain-computer interfaces. Biomed Signal Process Control. 2024; 96:106583.
17. Alrowais F, Negm N, Khalid M, Almalki N, Marzouk R, Mohamed A, Alneil A. Modified earthworm optimization with deep learning assisted emotion recognition for human computer interface. IEEE Access. 2023; 11:35089-35096.
18. Liao L, Li H, Shang W, Ma L. An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. ACM Trans Softw Eng Methodol. 2022;31(3):1–40.
19. Kalimeris D, Singer Y, Subbian K, Weinsberg U. Learning Diffusion using Hyperparameters. International Conference on Machine Learning. 2018;2420–8.
20. Al-Rifaie MM, Bishop JM. Stochastic diffusion search review. Paladyn J Behav Robot. 2013;4(3):155-173.
21. Yesilkaya B, Sayilgan E, Yuce YK, Perc M, Isler Y. Principal component analysis and manifold learning techniques for BCIs based on SSVEP. J Comput Sci. 2023; 68:102000.
22. Hossieny RR, Tantawi M, Shedeed H, Tolba MF. Development of electrooculogram-based human computer interface system using deep learning. Bull Electr Eng Inform. 2023;12(4):2410-2420.
23. Bu Y, Harrington DL, Lee RR, Shen Q, Angeles-Quinto A, Ji Z, Huang M. Magnetoencephalogram-based brain–computer interface for hand-gesture decoding using deep learning. Cereb Cortex. 2023;33(14):8942-8955.
24. Viswanathan GM, Raposo EP, Da Luz MGE. Lévy flights and superdiffusion in the context of biological encounters and random searches. Phys Life Rev. 2008;5(3):133-150.
25. Smith PL, Ratcliff R, Sewell DK. Modeling perceptual discrimination in dynamic noise: time-changed diffusion and release from inhibition. J Math Psychol. 2014; 59:95-113.
26. Maroufpoor S, Azadnia R, Bozorg-Haddad O. Stochastic optimization: stochastic diffusion search algorithm. Handbook of Probabilistic Models. Butterworth-Heinemann; 2020;437–448.

How to Cite: Subramanian SC, Daniel D. Discriminated-SDS: A Novel Hybrid Approach for Optimizing EEG Based Brain-Computer Interface Signals Faced by Metaheuristic Algorithms. Int Res J Multidiscip Scope. 2026; 7(1): 545-555. DOI: 10.47857/irjms.2026.v07i01.09252