International Journal of Applied Mathematics

Volume 38 No. 10s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ENHANCING CLASSIFICATION PERFORMANCE OF SVM USING HYBRID PSO ALGORITHM: A CASE STUDY ON LEARNING DISABILITY DIAGNOSIS

Anu P J[1][0009-0002-6245-297X], Dr. K. Ranjith Singh[2][0000-0003-2651-2509]

[1]Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India. anujohnson123@gmail.com.

[2]Assistant Professor & Research Guide, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore.

ranjithsingh.koppaiyan@kahedu.edu.in

Abstract

Learning disabilities are conditions that impact an individual's capacity to comprehend or employ spoken or written languages. This research paper, it is aimed to classify students with and without learning disabilities using machine learning techniques. A real dataset of 348 students was used for this purpose. It was found that traditional classification methods such as decision tree, k-nearest neighbors, etc. achieved an accuracy of over 90%, while the support vector machine (SVM) performed poorly with only 80% accuracy.

In this paper, the main objective is to improve SVM by using the advantage of optimization techniques. Firstly, a grid search algorithm was applied, resulting in an accuracy of 92.7%. Furthermore, a global best PSO algorithm was employed, which improved the accuracy to 94.5%, and a local best PSO algorithm, which improved the accuracy to 95.4%. Finally, a new hybrid PSO algorithm that combined the advantages of both local and global PSO was applied, resulting in an accuracy of 97.27%.

The outcomes indicate that the hybrid PSO-SVM algorithm is a highly efficient approach for distinguishing between students with learning disabilities and those without. This could be of great help to educators in identifying students who may require additional support in their studies. Additionally, the study highlights the limitations of traditional classification methods and the potential of machine learning techniques, specifically the use of optimization algorithms with SVM, in improving classification accuracy.

Keywords— Learning Disabilities, Machine Learning, SVM, Grid search, particle swarm optimization

I. INTRODUCTION

Learning disabilities [1] represent a prevalent neurodevelopmental condition that can exert a substantial influence on a child's academic and social well-being. Early identification and timely intervention are pivotal factors in enhancing outcomes for children grappling with learning disabilities. In recent years, machine learning algorithms have shown great promise in identifying and predicting learning disabilities in children. In this study, we aimed to classify students with and without learning disabilities using a real dataset of 348 students and various cognitive and academic features. The results unveiled that conventional classification techniques like decision trees, k-nearest neighbors, and random forests achieved impressive accuracy rates exceeding 90%. Conversely, the support vector machine (SVM) algorithm

Received: August 05,2025

Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

exhibited subpar performance, achieving only 80% accuracy. Nonetheless, through the application of grid search and particle swarm optimization (PSO) algorithms, significant

improvements were achieved, and the SVM's performance improved. In addition, a new hybrid PSO algorithm that combines the strengths of both local and global PSO algorithms was proposed to optimize the SVM. The hybrid PSO algorithm achieved a maximum accuracy rate of 97.27%, outperforming all other classification methods. The findings from the study imply that the hybrid PSO algorithm serves as a valuable tool for identifying students with learning disabilities. Additionally, they underscore the promise of machine learning techniques in enhancing classification accuracy within educational contexts.

II. Literature Review

Support Vector Machines (SVM) have long been a cornerstone in the field of machine learning, providing robust classification capabilities across various domains. Traditional SVMs, however, often struggle with large-scale data and non-linear decision boundaries. Recent developments have introduced Quantum SVMs, which leverage quantum computing to handle complex, high-dimensional datasets more efficiently than classical SVMs (Innan, N., Khan, M., Panda, B. et al. (2023)) [2]. Moreover, adaptive kernel SVMs have been proposed to dynamically adjust to data characteristics, enhancing the flexibility and accuracy of the model in non-linear classification scenarios (Wainer & Fonseca, 2021) [3].

Particle Swarm Optimization (PSO), known for its efficacy in navigating complex search spaces, has seen innovative adaptations that improve convergence rates and optimization accuracy. Adaptive PSO variants now feature dynamic adjustment of parameters such as inertia and acceleration coefficients, tailored to the specific topology of the search space (Musa et al., 2023) [4]. Furthermore, hybrid PSO models integrating mechanisms from genetic algorithms and ant colony optimizations have demonstrated superior performance in maintaining diversity in the search swarm and avoiding local optima (Aggarwal et al., 2020) [5].

The integration of PSO and SVM into a hybrid framework has been particularly promising for enhancing SVM’s performance in classification tasks. These hybrid models combine the global search capability of PSO with the margin maximization of SVM, effectively addressing the dual challenge of feature selection and hyperparameter optimization. In fields ranging from bioinformatics to cybersecurity, hybrid PSO-SVM models have outperformed traditional methods, achieving higher accuracy and stability (Moukhafi, Mehdi & El Yassini, Khalid & Bri, Seddik,2018) [6].

Application-specific enhancements have also been reported, with hybrid PSO-SVM models deployed for neurological disease prediction and automated speech recognition systems. These applications underscore the model's adaptability and potential in processing complex signals and images, where conventional SVMs often falter due to their linear limitations and sensitivity to hyperparameter settings (Zemmal et al., 2020) [7].

Jothi Prabha, A. et al [8] proposed a model which proposed Particle Swarm Optimization model for the prediction of dyslexia in individuals and proposed statistical methods for the differentiation of dyslexia and non-dyslexia individuals and used Principal Component analysis (PCA) to identify eye movements by the eye tracker raw data.

Received: August 05,2025

Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Looking forward, the application of hybrid PSO-SVM models could be extended beyond supervised classification tasks to include unsupervised and reinforcement learning scenarios. These models' capability to refine feature selection and hyperparameter tuning autonomously makes them suitable candidates for complex learning tasks where manual parameter adjustments are impractical.

III. Materials and Methods

A. Dataset and Implementation

The study used a real-time dataset consisting of various cognitive and academic features of children, including age, gender, class, academic performance, and skills related to reading, writing, spelling, copying, language, decoding, fine motor skills, maths, attention, hyperactivity, impulsivity, processing speed, auditory discrimination, auditory memory, visual memory, visual discrimination, and visuomotor skills. Table 1. shows the details of the attributes. The dataset included 348 rows, and the target variable was binary, indicating whether a child has a learning disability (LD) or not. The model was trained to predict LD status as yes or no.

Table 1. Dataset attributes

Attributes	Explanation
Age	Child's chronological age.
Academic performance	Child's overall academic achievement.
Reading skills	Ability to read and understand text.
Writing skills	Proficiency in written expression.
Spelling	Accuracy in spelling words.
Copying skills	Accuracy in replicating written or visual information.
Language skills	Competence in language use.
Decoding skills	Ability to decipher written words.
Fine Motor Skills	Coordination and dexterity of small muscles.
Mathematical Skills	Proficiency in math concepts and operations.

Received: August 05,2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Attention	Ability to focus and sustain attention.
Hyperactivity	Excessive motor activity or restlessness.
Impulsivity	Tendency to act spontaneously.
Processing Speed	Speed of perceiving and responding to information.
Auditory Discrimination	Ability to distinguish between sounds.
Auditory Memory	Capacity to retain and recall auditory information.
Visual Memory	Ability to remember visual information.
Visual Discrimination	Ability to perceive differences in visual stimuli.
Visuomotor Skills	Coordination between visual perception and motor skills. Top of Form

This work is carried out in different stages. First, the dataset is analyzed and performs data preprocessing tasks such as dropping irrelevant columns, integer encoding categorical variables, and oversampling to address the class imbalance. The dataset is then split into training and testing sets, and it is standardized using a StandardScaler. The mean values obtained are very small and are close to zero, which indicates that the data has been centered around zero. Also, the standard deviation of all 19 features is equal to 1 after the standardization process is done. That means the data has been scaled to a similar range across all the features. Subsequently, the algorithm undergoes training using the designated training dataset and is assessed through cross-validation. Following this, the algorithm is applied to make predictions on the testing dataset, and its performance is evaluated by measuring accuracy and Cohen's kappa score. Additionally, the algorithm's effectiveness is further assessed through the calculation of the ROC curve and the computation of the Area Under the Curve (AUC) score. Together, these metrics offer a comprehensive assessment of the algorithm's performance.

Summarizing overall work as follows:

- Data collection
- Data preprocessing

Received: August 05,2025

Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Splitting the dataset
- Training and testing ML models
- Tuning and optimization techniques
- Evaluate the performance metrics of each model
- Data visualization

B. Classification Techniques Used for Prediction [9,10]

1)Decision Tree (DT) [11]

Training:

- Select the best feature to split the data based on some criterion.
- Partition of the data based on the selected feature into subsets that maximize the separation of the classes.
- Iterate through the process recursively for each subset until a predetermined stopping criterion is satisfied.
- Assigning class labels to leaf nodes means finding the most common class among the instances in that node

Testing:

- To classify a new instance with feature vector X, follow the tree's branches from the root node to a leaf node using X's values for each split feature
- Assign the class label of the leaf node to the new instance

2)Random Forest (RF) [12]

Training:

- Generate a "bootstrapped" dataset by randomly selecting a subset of the training data with replacement.
- For each decision tree within the forest, randomly pick a subset of features to consider for splitting at each node during tree construction.
- Train each decision tree using the bootstrapped dataset and the chosen subset of features. The training process follows the same methodology as a standard decision tree but with these specific variations.
- For each leaf node in every decision tree, determine its class label by identifying the majority class among the instances contained within that particular node. This ensures that each leaf node provides a class prediction based on the dominant class within its associated data subset.

Testing:

- To classify a new instance represented by a feature vector X, you would navigate through each decision tree within the forest, starting at the root node and proceeding to a leaf node by evaluating the feature values in X at each split node along the way.
- Assign the class label of the new instance based on the majority vote of the class labels assigned by all the decision trees in the forest.

3)Extra Trees (ET)

Training:

Received: August 05,2025

Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Create a "bootstrapped" dataset by randomly sampling a subset of the training data with replacement.
- For every decision tree within the forest, randomly pick a subset of features to consider when making splits at each node.
- Randomly choose a threshold for splitting for each feature and split the data based on the threshold that maximizes the separation between classes.
- Continue splitting iteratively until a stopping condition is met, like reaching a max tree depth or a minimum instance per leaf.
- Assign a class label to each leaf node by determining the majority class among the instances contained within that node

Testing:

- For a new instance with feature vector X, traverse each decision tree in the forest from the root node to a leaf node based on the values of X for each split feature.
- Assign the class label of the new instance based on the majority vote of the class labels assigned by all the decision trees in the forest.

4)K-Nearest Neighbors (KNN) [13]

Training:

- The training data's feature vectors and corresponding class labels are stored.
- Prior probability P(Y) for each class is calculated, which is the frequency of each class label in the training data.
- Conditional probability P(Xi|Y) is calculated for each feature Xi and each class Y, which is the frequency of the feature Xi occurring in the training data for class Y.
- The learned hypothesis is formed by the set of prior and conditional probabilities calculated in the training phase.

Testing:

- For a new instance with feature vector X, the distance between X and all training feature vectors is calculated.
- Based on the calculated distances, the k nearest neighbors to X are selected.
- The classification of the new instance is determined by identifying the majority class label among its k nearest neighbors.

5)Support Vector Machines (SVM) [14]

Training:

- In SVM, to find a hyperplane that maximizes the margin between two classes in the training data, effectively separating them.
- The learned hypothesis consists of the support vectors that define the hyperplane.

Testing:

- For a new instance with feature vector X, calculate the distance to the hyperplane.
- Classify the new instance based on which side of the hyperplane it falls

To enhance SVM's performance, it is planned to use a hyperparameter tuning technique grid search to identify the optimal hyperparameter values for the SVM model.

Received: August 05,2025

International Journal of Applied Mathematics

Volume 38 No. 10s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

C. Hyperparameter tuning using a grid search

The most frequently employed technique for hyperparameter optimization is grid search. This method entails establishing a grid of values for each hyperparameter and assessing each combination on a validation set using a performance metric. The set of hyperparameters that achieve the best performance metric is then selected as the optimal hyperparameters for the model [15]. From some studies, it is noted that random search was faster than grid search, but grid search was more stable in selecting hyperparameters across different runs [16]. In this study for enhancing the SVM model grid search tuning method is selected. The generalized performance of a grid search for hyperparameter tuning of a support vector machine (SVM) model can be summarized as follows:

Input:

- Hyperparameter grid: Dictionary defining hyperparameters and their values or ranges.
- Training data: Features and labels for model training.
- Validation data: Features and labels for performance evaluation.
- Performance metric: Metric used to evaluate model performance.
- Cross-validation folds: Number of folds for cross-validation.

Output:

- Optimal hyperparameters: Best-performing hyperparameter values.
- Best model: Model trained with optimal hyperparameters.
- Performance score: Score of the best model on validation data.

D. Hyperparameter tuning using PSO Algorithm.

Particle Swarm Optimization (PSO) is a metaheuristic optimization algorithm that leverages a group of candidate solutions, known as particles, to explore and find the optimal solution [17]. PSO is inspired by bird flock behavior, with each particle representing a solution navigating the search space using its position and velocity [18]. PSO is a more powerful and efficient method for hyperparameter tuning compared to grid search, especially for complex problems with high-dimensional search spaces and interdependent hyperparameters.

1)Global-best Particle Swarm Optimization (Global-best PSO)

Global-best PSO, commonly known as gbest PSO, represents a variant of PSO employing a star topology. In this configuration, every particle is attracted toward the swarm's best-performing particle, termed the global best position. In each iteration, the position and velocity of each particle undergo updates according to the following equations [19]:

Position update: $x_i(t+1) = x_i(t) + v_i(t+1)$ (1)

Here, xi(t) represents the position of each particle at time t, and vi(t) represents the velocity of the particle at time t. The new position of each particle at time t+1 is calculated by adding the computed velocity at that time.

Velocity update: $v_{ij}(t+1) = w*v_{ij}(t) + c_1r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2r_{2j}(t)*[\hat{y}_j(t) - x_{ij}(t)]$ (2)

where vij(t+1) is the velocity of particle j along dimension i at time t+1,'w' controls momentum (high explores, low exploits).'c1' and 'c2' influence personal and global effects.'r1j(t)' and 'r2j(t)' guide direction randomly. Adjust parameters to customize PSO for specific problems.

The idea behind using SVM with Global PSO is to enhance the SVM model's performance by fine-tuning its hyperparameters through the Global PSO algorithm. Hyperparameters such as C

Received: August 05,2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

(the penalty parameter) and gamma (the kernel coefficient) need to be tuned for optimal performance when using SVM. In this approach, the objective function is the SVM model's cost function, and the hyperparameters to be optimized are C and gamma. The bounds for these hyperparameters can be set, and 8 the Global PSO algorithm can be used to search for the best values of C and gamma that minimize the SVM model's cost function.

2)Local-best Particle Swarm Optimization (Local-best PSO)

lbest PSO is a PSO variation that employs a ring topology rather than a star topology. In lbest PSO, each particle is drawn towards its specific neighborhood instead of the global best position. Similar to global-best PSO, this algorithm updates each particle's position using calculated velocity, and velocity adjustments are made using cognitive and social parameters [18,19].

Position update: $x_i(t+1) = x_i(t) + v_i(t+1)$ (3)

Velocity update: $v_{ij}(t+1) = m*v_{ij}(t) + c_1r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2r_{2j}(t)*[\hat{y}_j(t) - x_{ij}(t)]$ (4)

The concept of using Local PSO with SVM is similar to that of using Global PSO with SVM, except that the algorithm is optimized for local search rather than global search. The objective function remains the SVM model's cost function, with the hyperparameters for optimization still being the penalty parameter 'C' and the kernel coefficient 'gamma'. However, instead of searching the entire parameter space, Local PSO only searches within a neighborhood of each particle.

Compared to global-best PSO, local-best PSO converges more slowly but explores more effectively due to its ring topology and neighbor comparisons.

3)Hybrid Particle Swarm Optimization (Hybrid PSO)

Hybrid PSO combines elements of both Global and Local PSO algorithms to strike a balance between exploring and exploiting the search space effectively. In Hybrid PSO, the algorithm starts with Global PSO to explore the search space and find good regions. It switches to Local PSO to exploit the good regions found in the previous step and refine the search toward the best solution. By combining the strengths of both algorithms, Hybrid PSO can provide better convergence to the global optimum while avoiding premature convergence to local optima.

When using Hybrid PSO with SVM, the objective is to optimize the hyperparameters of the SVM model using a combination of global and local search.

The following algorithm is used for getting the result.

Input:

- Training dataset (x_train, y_train) and testing dataset (x_test, y_test)
- Bounds for hyperparameters: lower_bound, upper_bound
- Number of particles N
- Number of dimensions (hyperparameters) D
- PSO parameters for global and local search: inertia weight w, cognitive coefficient c1, social coefficient c2

Received: August 05,2025

Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Number of iterations for global and local search:
  max_iter_global, max_iter_local

Output:

- Best hyperparameters found by the HPSO-SVM algorithm

Algorithm:

1. Define the objective function:
   -Input: List of hyperparameters h
   -Output: Cost of the SVM classifier trained with hyperparameters h on the training dataset
2. Define the bounds for the hyperparameters: lower_bound, upper_bound
3. Initialize global and local PSO optimizer options:
   - Number of particles: N
   - Number of dimensions: D
   - PSO parameters: inertia weight w, cognitive coefficient
   c1, social coefficient c2
4. Initialize global and local PSO optimizers with the options and bounds:
   - Global optimizer with parameters: N, D, w, c1, c2,
   max_iter_global, bounds (lower_bound, upper_bound)
   - Local optimizer with parameters: N, D, w, c1, c2,
   max_iter_local, bounds (lower_bound, upper_bound)
5. Loop through 10 iterations:
   a. If the current iteration is less than 5:
      - Use global search mode with the global optimizer for
      max_iter_global iterations
   b. If the current iteration is greater than or equal to 5:
      - Use local search mode with the local optimizer for
      max_iter_local iterations
   c. Print the current iteration number, best cost found so far,
   and corresponding best hyperparameters
6. Return the best hyperparameters found by the algorithm

Combining both HPSO and SVM can potentially find the optimal hyperparameters of the SVM model more quickly and accurately than using either approach alone.

IV. RESULTS AND DISCUSSION

Various tests and their results have been carefully studied in this section. The main objective of the study was to improve the performance of the SVM algorithm compared to other machine

Received: August 05,2025

Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

learning models such as Decision Trees, The Random Forest, Extra Trees, and K-nearest neighbor. initially, the SVM shows lower performance than other models, and hence grid search is used.

Grid Search was effective but can be quite computationally intensive, particularly when it deals with numerous hyperparameters. Hence, the study examined the use of optimization algorithms like Global and Local PSO to find optimal SVM hyperparameters more efficiently.

Global and local PSO can further improve the SVM algorithm performance when it is compared with the combination of Grid search and SVM, according to the study. However, the study suggested a new approach: the hybrid PSO algorithm that combines the benefits of global and local PSO algorithms to further, improve efficiency

The hybrid PSO algorithm is a high-performance algorithm, that can efficiently balance the exploration and exploitation of the search space, leading to the discovery of optimal hyperparameters for the SVM model. Using hybrid PSO algorithms, the performance of the SVM algorithm was much improved compared with grid search and other optimization algorithms.

A. Evaluation matrices

To measure the system's performance, various parameters are used, such as precision, recall, f1-score, accuracy,[20] AP (Average Precision) score, kappa measurement, ROC-AUC (Receiver Operating Characteristic- Area Under the Curve) score, TPR (True Positive Rate), and FPR(False Positive Rate). [21,22]

1)Precision:

Precision=true positive/ (true positive + false positive) (5)

2)Recall:

Recall=true positive/ (true positive + false negative) (6)

3)F1-score:

F1-score=2*(precision*recall)/ (precision + recall) (7)

4)Accuracy:

Accuracy= (true positive + true negative)/ (true positive + true negative + false positive + false negative) (8)

5)TPR

TPR=true positive/ (true positive + false negative) (9)

6)FPR

FPR=false negative/ (false positive + true negative) (10)

Received: August 05,2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

7)Average Precision (AP)

This score quantifies the precision-recall trade-off by averaging precision values at different recall levels in precision-recall curves.

8)Kappa measurement

It assesses agreement between observed and expected classifications, accounting for the possibility of chance agreements, with higher coefficients indicating stronger agreement or better performance.

9)ROC-AUC score

It evaluates binary classifier performance by plotting True Positive Rate against False Positive Rate at various thresholds, with higher values representing superior performance.

B. Result analysis

1)Comparison of classification models

In this case, various parameters are used for calculating the performance of models. Table 2. shows the results of various parameter calculations done in the models for the evaluation.

Algorithm	Precision	Recall	F1-Score	Kappa value	AP	ROC-AUC	TPR	FPR	Accuracy
Random forest	97	96	96	92.71	99	98.30	93.33	0	96.36
Decision tree	96	95	95	90.79	95	95.10	90.19	0	95.45
KNN	97	96	96	92.64	96	96.10	92.15	0	96.36
Extra tree	96	95	95	90.78	100	99.70	93.65	0.021	95.45
SVM	80	80	80	60.01	86	85.60	78	0.185	80

Table 2. Performance Evaluation Metrics of the classification models

When comparing the performance metrics of the algorithms in the given Table 2., the following observations can be made.

Random Forest achieved a precision score of 97%, indicating a high proportion of correctly predicted positive instances. It also had a recall score of 96% and an F1-score of 96%, suggesting effective capturing of actual positive instances. The Kappa value of 92.71 reflects a substantial level of agreement between predicted and actual classifications. Additionally, it achieved high scores in Average Precision (AP) and ROC-AUC, indicating excellent precision and discrimination performance.

Received: August 05,2025

Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

In a similar vein, Decision Tree and KNN both performed well overall. They identified positive cases accurately, as evidenced by their strong recall, precision, and F1 scores. There is good agreement between the Kappa values of 90.79 and 92.64 and the actual classifications. In addition, these algorithms performed competitively when it came to AP, ROC-AUC, and total accuracy.

The Extra Tree algorithm also demonstrated notable performance with precision, recall, and F1-scores of 96% and 95%, along with a Kappa value of 90.78. It achieved a perfect AP score and a high ROC-AUC score, indicating exceptional precision and discrimination capabilities. The TPR of 93.65% suggests accurate identification of positive instances, while the low FPR of 0.021 signifies minimal misclassification of negative instances.

In contrast, the SVM algorithm had lower performance metrics across precision, recall, F1-score, and Kappa value. It exhibited relatively lower precision and recall scores of 80% and a moderate Kappa value of 60.01. However, it achieved reasonable scores in AP and ROC-AUC, indicating acceptable precision and discrimination performance. The TPR of 78% suggests a lower ability to correctly identify positive instances, while the higher FPR of 0.185 indicates a higher misclassification rate of negative instances. The overall accuracy score of 80% is lower compared to the other algorithms.

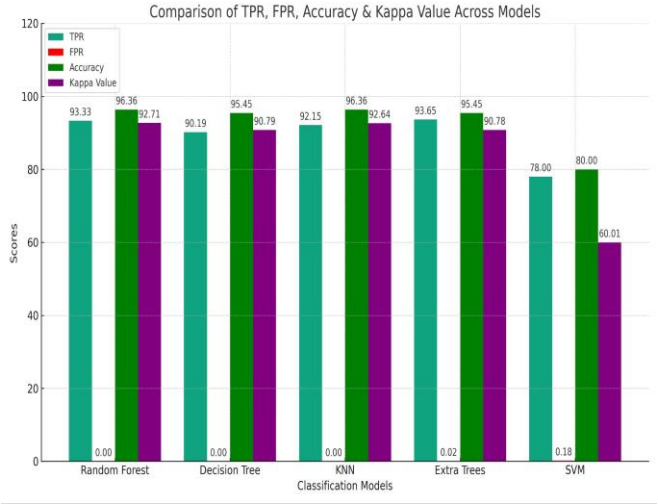


Fig. 1. Comparison of TRP, FPR, Accuracy, and kappa value across models.

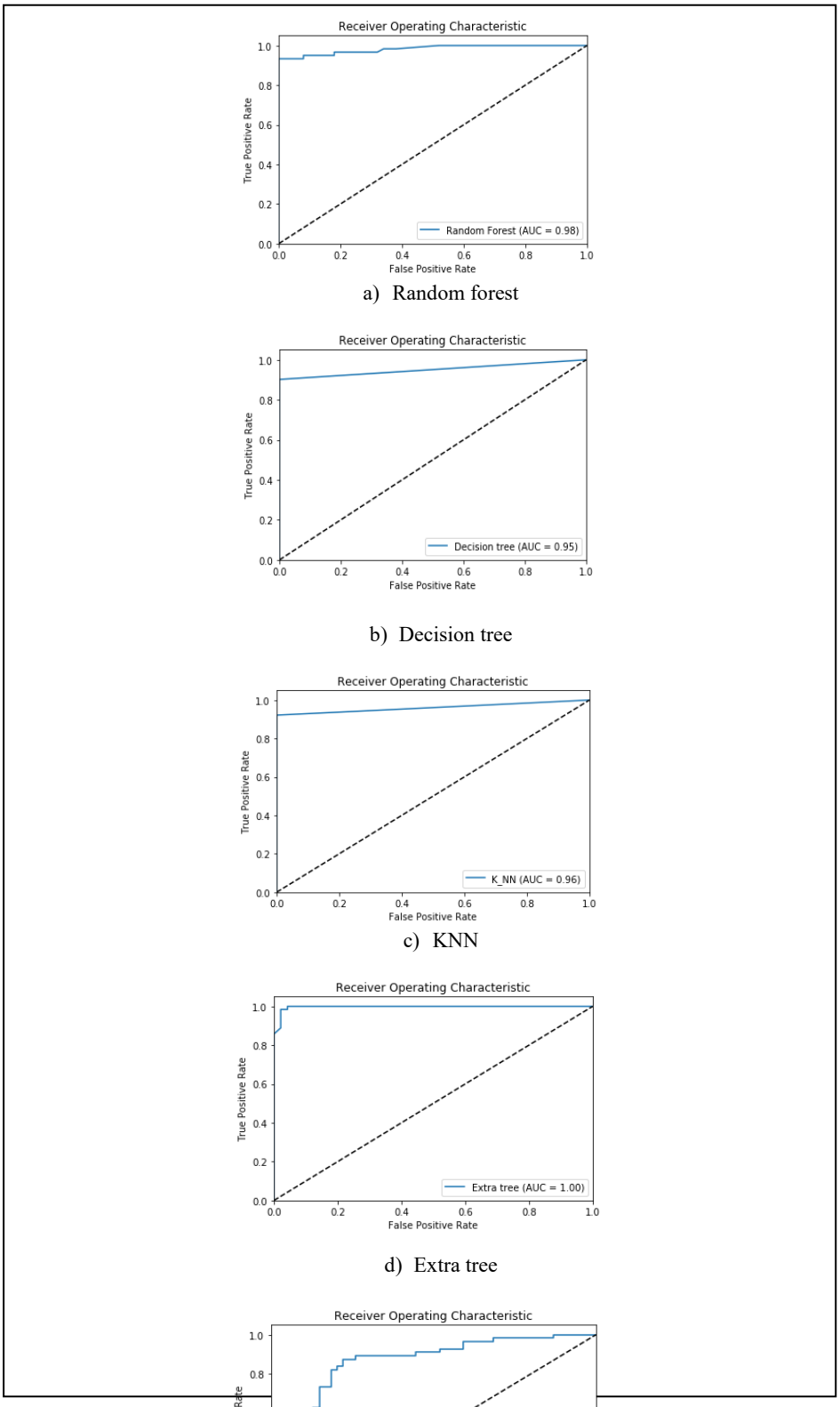
The bar chart provides in Fig. 1. Shows a comparative visualization of performance metrics across five classification models: Random Forest, Decision Tree, K-Nearest Neighbors (KNN), Extra Trees, and Support Vector Machines (SVM). It distinctly illustrates each model's True Positive Rate (TPR), False Positive Rate (FPR), Accuracy, and Kappa Value.

Received: August 05,2025

International Journal of Applied Mathematics

Volume 38 No. 10s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)



a) Random forest

b) Decision tree

c) KNN

d) Extra tree

The Fig.2. shows the ROC-AUC [illegible] s of Random Forest, Decision Tree [illegible] M.

Received: August 05,2025

International Journal of Applied Mathematics

Volume 38 No. 10s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2)Performing Grid search on Support Vector Machines (SVM)

SVM is a popular algorithm for classification and regression analysis. The main hyperparameters to tune in SVM are [23]:

● C: Regularization parameter. Determines the trade-off between finding a decision boundary that correctly classifies training data points and a smooth decision boundary.

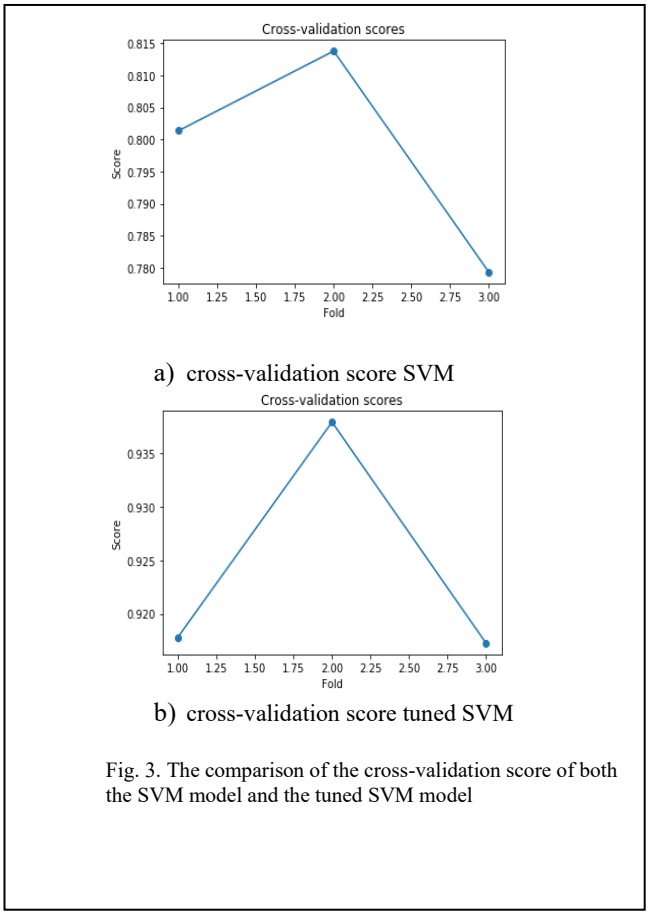
● Kernel: Specifies the kernel function used for data transformation into a higher dimensional space.

● Gamma: Parameter for kernel function. Controls the shape of the decision boundary.

To perform a grid search on SVM, define a parameter grid with different values for 'C', 'gamma', and 'kernel'. Use GridSearchCV with the SVM estimator, and the parameter grid, and specify cross-validation with accuracy scoring to find the best hyperparameters. Create a new SVC model with the best hyperparameters and evaluate its performance using 3-fold cross-validation.

In this case, the mean cross-validation score is 0.924, which indicates that the model has high accuracy. The standard deviation of 0.009 indicates that the model is consistent in its performance across the folds. The cross-validation score of the simple SVM model is 0.798 with a standard deviation of 0.014. Overall, the model's performance is moderate

Fig. 3. shows the comparison of the cross-validation score of both the SVM model and the tuned SVM model.



a) cross-validation score SVM

b) cross-validation score tuned SVM

Fig. 3. The comparison of the cross-validation score of both the SVM model and the tuned SVM model

In comparison to the SVM model, the grid-searched SVM model has a significantly higher

Received: August 05,2025

Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

mean cross-validation score and a lower standard deviation. This suggests that the grid-searched SVM model is more accurate and consistent in its predictions. Therefore, the grid-searched SVM model is the better model in this case.

3)Performing Global best PSO on Support Vector Machines (SVM)

The Global Best PSO algorithm is used with SVM to optimize hyperparameters C and gamma for maximum SVM performance. The objective function calculates the cost based on accuracy, and the PSO algorithm searches for the best hyperparameter values within specified bounds. The resulting best hyperparameters are used to create and evaluate a new SVM model using cross-validation.

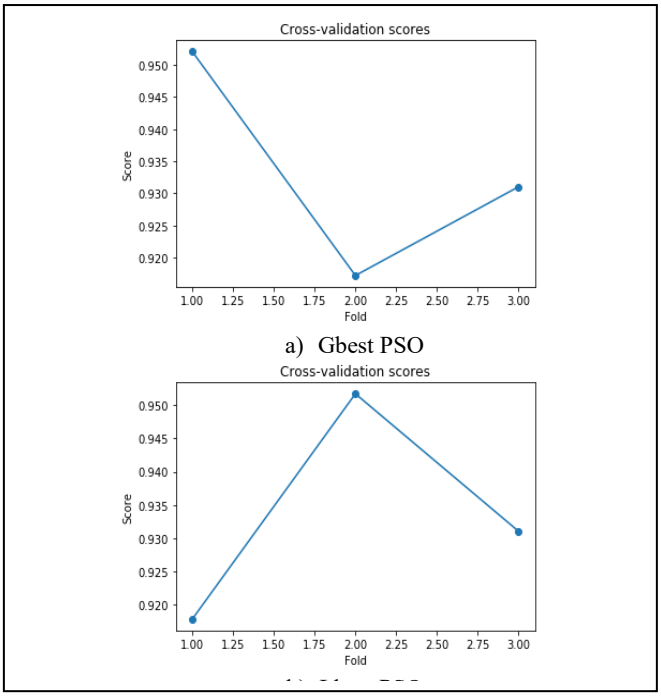
In this case, the mean cross-validation score is 0.933, which indicates that the model has high accuracy. The standard deviation of 0.014 indicates that the model is consistent in its performance across the folds.

4)Performing Local best PSO on Support Vector Machines (SVM)

The performance of the Local best PSO is similar to the Global best PSO but there is a slight difference in the options and parameters provided to the two algorithms. In the Local best PSO algorithm additional options k and p are provided. These parameters are specific to the Local Best PSO algorithm's implementation and represent the number of neighbors to consider (k) and the power parameter (p) for determining the neighborhood's best position. These parameters allow for more localized search behavior and can affect the convergence speed and exploration capabilities of the algorithm.

In this case, the mean cross-validation score is 0.933, which indicates that the model has high accuracy. The standard deviation of 0.013 indicates that the model is consistent in its performance across the folds.

5)Performing Hybrid PSO on Support Vector Machines (SVM)



a) Gbest PSO

Received: August 05,2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The hybrid Particle Swarm Optimization (HPSO) algorithm is used to find the best combination of hyperparameters for a given objective function. The algorithm consists of two search modes: global and local. In the global search mode, the algorithm explores a wide range of the search space using a global PSO optimizer. In the local search mode, it refines the search around promising regions using a local PSO optimizer. The algorithm iterates for a set number of steps, with the first half for global search and the second half for local search. It identifies the best cost and corresponding hyperparameters in each iteration. Then the iteration number, best cost, and best parameters for each iteration were found to allow for tracking the optimization progress. By combining both strategies, the algorithm aims to find the optimal hyperparameters that minimize the cost of the objective function and improve the model performance.

Using the best hyperparameters, a new SVM model is created and evaluated using cross-validation. The model achieves a mean cross-validation score of 0.958, indicating high accuracy and consistent performance across the folds. The model's stability and reliability are confirmed by the low standard deviation of 0.009 in predicting the target variable. Refer to Fig. 5 for the cross-validation score of the Hybrid PSO-SVM model.

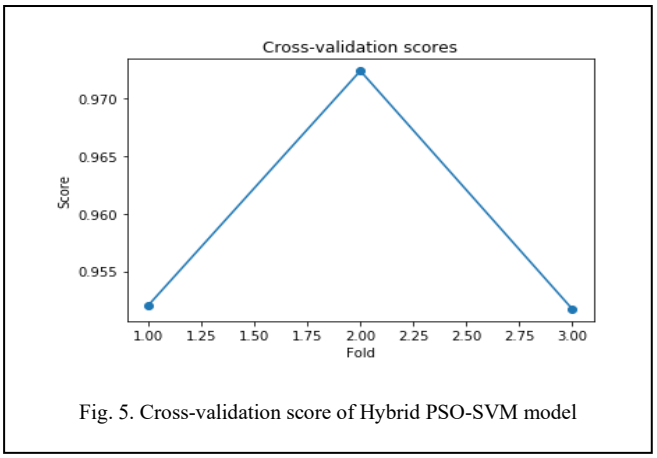


Fig. 5. Cross-validation score of Hybrid PSO-SVM model

In all these scenarios, cross-validation is utilized. It entails partitioning the available dataset into multiple subsets, often referred to as "folds," for both training and evaluation. Cross-validation estimates the model's generalization to new data by simulating its performance on different test sets, providing a robust evaluation and mitigating overfitting concerns.[24]

6)Results of hyperparameter tuning done on SVM using Grid Search and PSO

Table 3. displays the performance metrics of different algorithms applied to SVM for a specific problem. The algorithms include Grid Search-SVM, Global Best PSO-SVM, Local Best PSO-SVM, and Hybrid PSO-SVM. The metrics evaluated are precision, recall, F1-score, Kappa value, average precision (AP), ROC-AUC score, true positive rate (TPR), false positive rate (FPR), and accuracy.

By comparing the algorithms, Grid Search-SVM achieves a precision, recall, and F1-score of 93, with a Kappa value of 85.18. Global Best PSO-SVM improves the precision, recall, and F1-score to 95, with a Kappa value of 88. Local Best PSO-SVM further enhances the precision, recall, and F1-score to 96, with a Kappa value of 90.63. Finally, the Hybrid PSO-SVM algorithm demonstrates the best performance with precision, recall, and F1-score of 97, and a

Received: August 05,2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Kappa value of 94.33.

All algorithms achieve high average precision (AP) and ROC-AUC scores, indicating their ability to rank positive samples higher than negative samples. The true positive rate (TPR) remains consistently high across all algorithms, indicating their effectiveness in identifying positive samples. The false positive rate (FPR) decreases as the algorithms improve, A decreasing FPR is desirable because it indicates a higher level of specificity in the classification model. A lower FPR means that the algorithm is making fewer false positive errors and is better able to identify true negative samples. This is particularly important in applications where the cost or impact of false positives is high, such as in medical diagnoses or fraud detection.

Furthermore, as the algorithms improve and the FPR decreases, the accuracy of the algorithms also improves. Accuracy measures a classification model's correctness. In the case of the Hybrid PSO-SVM algorithm, an accuracy of 97.27% indicates a high level of overall correctness in the classification results.

These results highlight the effectiveness of the different optimization algorithms (Grid Search, PSO) in improving the performance of SVM, with the Hybrid PSO-SVM algorithm achieving the highest overall performance.

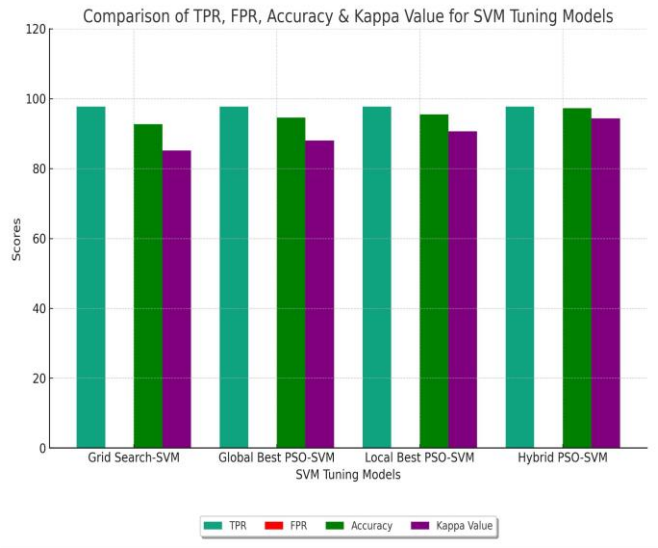


Fig. 6. Comparison of TRP, FPR, Accuracy, and kappa value for SVM Tuning models.

The bar chart shown in Fig. 6. shows how effectively delineates the performance metrics of four SVM tuning models: Grid Search-SVM, Global Best PSO-SVM, Local Best PSO-SVM, and Hybrid PSO-SVM, focusing on True Positive Rate (TPR), False Positive Rate (FPR), Accuracy, and Kappa Value. All models exhibit an identical TPR of 97.72%, illustrating their effectiveness in identifying true positives. However, distinctions emerge in their FPR, Accuracy, and Kappa metrics, highlighting the varying efficiencies and reliabilities. The Hybrid PSO-SVM model demonstrates superior performance with the lowest FPR (0.03%) and the highest Accuracy (97.27%) and Kappa Value (94.33), indicating its robustness and consistency in classification tasks.

Received: August 05,2025

International Journal of Applied Mathematics

Volume 38 No. 10s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

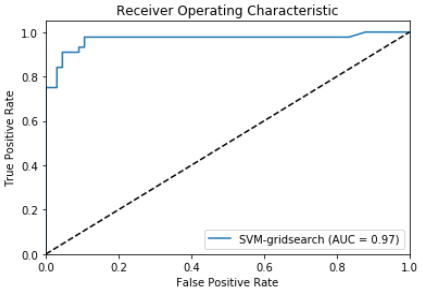
Table 3. Performance evaluation metrics of SVM tuning models

Algorithm	Precision	Recall	F1-Score	Kappa value	AP	ROC-AUC	TPR	FPR	Accuracy
Grid Search-SVM	93	93	93	85.18	97	96.8	97.72	0.10	92.72
Global Best PSO-SVM	95	95	95	88	96	97.4	97.72	0.075	94.54
Local Best PSO-SVM	96	95	95	90.63	97	97.9	97.72	0.060	95.45
Hybrid PSO-SVM	97	97	97	94.33	98	97.3	97.72	0.030	97.27

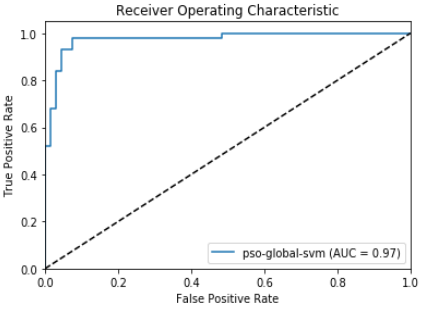
The Fig.7. shows the ROC-AUC score plotting for the models of Grid Search-SVM, Global Best PSO-SVM, Local Best PSO-SVM, and Hybrid PSO-SVM.

Received: August 05,2025

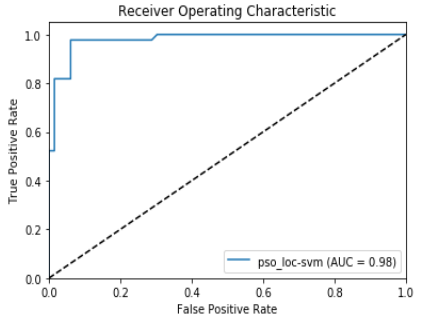
International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)



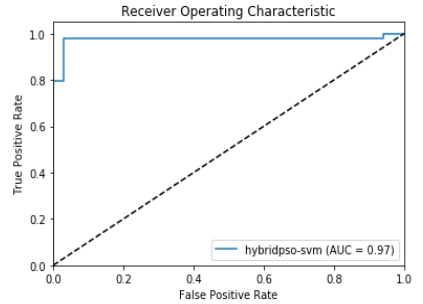
a) Grid Search SVM



b) GlobalBest PSO-SVM



c) LocalBest PSO-SVM



d) Hybrid PSO-SVM

Fig. 7. ROC-AUC curve plotting for optimization models

V. CONCLUSION AND FUTURE WORK

In conclusion, the study conducted a comparative analysis of various optimization techniques applied to the SVM algorithm for classifying children with and without learning disabilities. The optimization methods evaluated comprised Grid Search, Global Best PSO, Local Best PSO, and Hybrid PSO. While Grid Search yielded moderate results, Global Best PSO and Local Best PSO exhibited enhancements in precision, recall, F1-score, and reduced false

Received: August 05,2025

Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

positive rates (FPR). Notably, the Hybrid PSO technique emerged as the most effective, achieving the highest precision, recall, F1 score, and accuracy across all methods. Furthermore, Hybrid PSO demonstrated superior discrimination capabilities, with a significantly lower false positive rate (FPR) compared to other optimization approaches. These findings highlight Hybrid PSO's effectiveness in improving SVM model performance for classifying children with learning disabilities. The outcomes suggest that the Hybrid PSO-SVM algorithm is a highly efficient method for distinguishing between students with learning disabilities and those without. This capability could greatly benefit educators in identifying students who may require additional support in their studies. Future research avenues may explore fine-tuning the parameters of the Hybrid PSO-SVM algorithm, extending its applicability to other classification problems, and assessing its performance on large-scale datasets.

REFERENCES

[1] Karande, S., & Kulkarni, M. (2020). Learning Disabilities in Children: An Overview. Journal of Pediatric Neurosciences, 15(2), 54-60. doi: 10.4103/jpn.JPN_107_19.

[2] Innan, N., Khan, M., Panda, B. et al. Enhancing quantum support vector machines through variational kernel training. *Quantum Inf Process* 22, 374 (2023). https://doi.org/10.1007/s11128-023-04138-3

[3] Wainer, J., Fonseca, P. How to tune the RBF SVM hyperparameters? An empirical evaluation of 18 search algorithms. *Artif Intell Rev* 54, 4771–4797 (2021). https://doi.org/10.1007/s10462-021-10011-5

[4] Zulkifli Musa, Zuwairie Ibrahim, Mohd Ibrahim Shapiai, and Yusei Tsuboi. "Cubature Kalman Optimizer: A Novel Metaheuristic Algorithm for Solving Numerical Optimization Problems". *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 33, no. 1, Oct. 2023, pp. 333-55, doi:10.37934/araset.33.1.333355.

[5] Aggarwal, G., Monga, R. & Gochhayat, S.P. A Novel Hybrid PSO Assisted Optimization for Classification of Intellectual Disability Using Speech Signal. *Wireless Pers Commun* 113, 1955–1971 (2020). https://doi.org/10.1007/s11277-020-07301-6

[6] Moukhafi, Mehdi & El Yassini, Khalid & Bri, Seddik. (2018). A novel hybrid GA and SVM with PSO feature selection for intrusion detection system. *International Journal of Advances in Scientific Research and Engineering*. 4. 129-134. 10.31695/IJASRE.2018.32724.

[7] Zemmal, N., Azizi, N., Sellami, M. et al. Particle Swarm Optimization Based Swarm Intelligence for Active Learning Improvement: Application on Medical Data Classification. *Cogn Comput* 12, 991–1010 (2020). https://doi.org/10.1007/s12559-020-09739-z

[8] Jothi Prabha, A. and Bhargavi, R., 2019. Prediction of dyslexia from eye movements using machine learning. *IETE Journal of Research*, pp.1-10.

[9] Alpaydin, E. (2010). *Introduction to machine learning* (2nd ed.). Cambridge, MA: MIT Press. (Chapter 6)

[10]Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York: Springer. (Chapter 9)

Received: August 05,2025

Volume 38 No. 10s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[11]Bahzad Taha Jijo, Adnan Mohsin Abdulazeez, Classification Based on Decision Tree Algorithm for Machine Learning. January 2021, *Journal of Applied Science and Technology Trends* Vol. 02, No. 01, pp. 20 – 28 (2021), doi: 10.38094/jastt20165.

[12]Sarica, Alessia et al. "Random Forest Algorithm for the Classification of Neuroimaging Data in Alzheimer's Disease: A Systematic Review." *Frontiers in aging neuroscience* vol. 9 329. 6 Oct. 2017, doi:10.3389/fnagi.2017.00329.

[13]Syriopoulos, P.K., Kalampalikis, N.G., Kotsiantis, S.B. et al. kNN Classification: a review. *Ann Math Artif Intell* (2023). https://doi.org/10.1007/s10472-023-09882-x.

[14]Omid Naghash Almasi, Modjtaba Rouhani. Fast and de-noise support vector machine training method based on fuzzy clustering method for large realworld datasets[J]. *Turkish Journal of Electrical Engineering and Computer Sciences,* 2016, 24(1):219-233.

[15]Feurer, M., Lienhart, R. and Eggensperger, K., 2019. Hyperparameter optimization. In Automated Machine Learning (pp. 3-38). Springer, Cham.

[16]Empirical Evaluation of Random Search and Grid Search for Hyperparameter Tuning in Support Vector Machines" by Zhang, Y., Qin, X., Chen, J., & Li, J. (2021). IEEE Access, 9, 32272-32283

[17]Musa, Zulkifli, et al (2023)"Cubature Kalman Optimizer: A Novel Metaheuristic Algorithm for Solving Numerical Optimization Problems." *Journal of Advanced Research in Applied Sciences and Engineering* Technology, vol. 33, no. 1, pp. 333-355, https://doi.org/10.37934/araset.33.1.333355.

[18]Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43.

[19]J. Kennedy and R.C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[20]Pandiyarajan, Mariappan, and R. S. Valarmathi. (2024) "Deep Learning Techniques in Classification of Stages in Dementia: An Ensemble Approach." Journal of Advanced Research in Applied Sciences and Engineering Technology, vol. 36, no. 1,pp. 131-146, https://doi.org/10.37934/araset.36.1.131146.

[21]J. Cohen (1960). "A coefficient of agreement for nominal scales." Educational and Psychological Measurement, 20(1), 37-46.

[22]Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. *In Proceedings of the 23rd International Conference on Machine Learning* (ICML), 233-240.

[23]Wainer, Jacques, and Pablo Fonseca. "How to tune the RBF SVM hyperparameters? An empirical evaluation of 18 search algorithms." Artificial Intelligence Review 54.6 (2021): 4771-4797.https://doi.org/10.1007/s10462-021-10011-5

[24]Refaeilzadeh, P., Tang, L., Liu, H. (2009). Cross-Validation. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_565.

Received: August 05,2025