ISSN 1840-4855
e-ISSN 2233-0046

Original scientific article
http://dx.doi.org/10.70102/afts.2025.1834.1307

ENHANCING MACHINE LEARNING CLASSIFIERS WITH GLOBALBESTPSO FOR CLASSIFYING BANK CUSTOMERS

Sufaira Shamsudeen[1*], Dr.K. Ranjith Singh[2]

[1]Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India. e-mail: sufaira@mesmarampally.org, orcid: https://orcid.org/0009-0003-3783-9117*
[2]Assistant Professor, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India.
e-mail: ranjithsingh.koppaiyan@kahedu.edu.in, orcid: https://orcid.org/0000-0003-2651-2509

Received: October 15, 2025; Revised: November 20, 2025; Accepted: December 17, 2025; Published: December 30, 2025

SUMMARY

The innovation of Machine Learning (ML) techniques is evolving from basic techniques to optimized techniques, considerably improving the performance of prediction models. In the proposed work, the study primarily explores fundamental ML classification methods to classify banking customers based on their credit information. The classification of customers targets five categories: Outstanding, Excellent, Good, Satisfactory, and Bad. The aim is to assess the profitable customer categories and gain successful business by offering resources. The basic classification algorithms used in the proposed work are K-Nearest Neighbour (K-NN), Support Vector Machines (SVM), Decision Tree (DT), and Random Forest (RF) Classifiers. Using standard evaluation metrics, the performance of the classifiers are evaluated. Based on the metrics the comparative analysis is conducted, and comprehended the performance metrics need to be elevated. To manipulate this, Hyperparameter GridSearchCV (HGSCV) optimization is adopted, which is putative for its exhaustive search capabilities. However, the present accuracy scores of algorithms could be slightly improved while applying the HGSCV. Subsequently, the analysis moves on to an advanced optimization meta-heuristic optimized technique known as Particle Swarm Optimization (PSO). In this approach, the GlobalBestPSO method is implemented to tune the classifiers. The performance of the optimized classifiers such as GlobalBestPSO-SVM (gbestPSO-SVM), GlobalBestPSO-KNN (gbestPSO-KNN), GlobalBestPSO-DT (gbestPSO-DT), and GlobalBestPSO-RF (gbestPSO-RF) classifier are evaluated by analyzing the chosen set of parameters. The comparison of test results demonstrates the outstanding performance metrics in the optimized method, with accuracy outperformed with exceeding 0.95 score. The proposed hybrid model, integrating GlobalBestPSO with basic classifiers, superiors both traditional classifiers and tuned model HGSCV. The analysis is concluded to figure out the performance metrics of boosted classifiers, which optimized with the GlobalBestPSO, offers superior performance than others beyond all metrics.

Key words: *random forest, decision tree, support vector classifier, k-nearest neighbor, particle swarm optimization, globalbestpso.*

INTRODUCTION

The banking and financial industry is one of the most data intensive diverse environments, where customer profiling is an important part of credit risk management, profitability models, and strategic allocation of resources. Contemporary financial institutions handle large amounts of heterogeneous data on customers such as demographic, transactional and credit related data. The proper analysis of this data is necessary to differentiate between profitable and high-risk customers as well as to reduce the financial losses related to defaults.

Emerging technologies such as Machine Learning (ML) has significantly expanded the exploration and analysis of big data, allowing more knowledgeable decision-making in multi-class classification applications. ML techniques are gradually being used by many companies to improve decision-making processes and thereby increase business revenues. Recently, in the banking sector, these technologies are driving solid revenue growth and improving operational efficiency. Due to the fast evolvement of technologies like AI and ML, the banking and finance industry widely used this technique for various operations and gain a reliable and cost-efficient banking services [1]. Evolution of novel techniques, has significantly understand the customer behavior. In the proposed analysis, mainly focus is on the analysis of the customer's demographic and credit information in a refined way, dividing the customer into multi-class: Class0 (outstanding customers), Class1 (excellent customers), Class2 (good customers), Class3 (satisfactory customers) and Class4 (bad customers) by using a real-world dataset containing over one lakh records and 43 features. For the classification, the simple machine learning algorithms are used and stated the performance metrics. Based on the evaluation and to further improve the performance of the model, the hyperparameter tuning techniques like GridSearchCV method is adopted and evaluated, resulting a slight progress in the performance measure. In the recent studies, the advanced hyperparameter tuning techniques such as an optimized swarm intelligence are espoused. In the proposed study, the swarm intelligence technique Particle Swarm Optimization is executed. Based on the dataset landscape, GlobalBestPSO is upright approach. The performance of four optimized classifiers such as gbestPSO-KNN, gbestPSO-SVC, gbestPSO-DT, and gbestPSO-RF are gained more than others because of their capacity to learn the models by tuning the parameters very accurately. Though, nowadays, optimized models combined with machine learning and unconventional optimization techniques have proved to be more reliable models for multi-class classification.

The key findings of this study can be summarized as, Multiclass Banking Customer Classification: The experiment concerns a five-class customer classification issue based on the real-life banking data. Hybrid Optimization Framework: A new combination of GlobalBestPSO and several ML classifiers is suggested to help optimize hyperparameters effectively and precisely. Comprehensive Comparative Analysis: The simpler classifiers, GridSearchCV-tuned models and GlobalBestPSO-optimized models have been systematically compared with both parametric and non-parametric analysis measures. Performance Enhancement Validation. It has been shown by experiment that GlobalBestPSO-optimized classifiers are always more accurate than traditional and grid-tuned models, with accuracy and ROC-AUC scores of more than 95 percent, and in some cases, 99 percent.

The rest of this paper will be structured in the following way. Section 2 provides a review of related literature on customer classification and optimization algorithms based on machine learning. Part 3 is a description of the dataset used, preprocessing steps, and the proposed methodology. Section 4 discusses the evaluation metrices used. In section 5, the results, discussions and comparison of the suggested models are included along with practical implications. Lastly, Section 6 wraps up the paper and provides future research directions.

LITERATURE REVIEW

In our time, wide varieties of sectors are working with very gigantic data sets. Manually processing the massive amount of data, the time scale it would take much longer may not even be worthwhile in the end. Because of the significant expansion of new technologies such as artificial intelligence, machine learning and deep learning models generate outcomes that are more precise than those produced by earlier models. Because of their widespread use across various domains and their capacity to classify new observations using training data, K-Nearest Neighbors, Support Vector Classifier, Decision Tree, and Random Forest are well-known classifiers for classifying beneficial customers. By creating the excellent models, the sectors can reduce their stress to fulfil the decision. Later, the tuning techniques are successfully employed with hike performance metrics for the research work due to their capacity to modify the hyperparameters of the four classifiers using the present data. However, the research article proved that various ML classification algorithms and advanced swarm optimization techniques are used to create a consistent hybrid model shows immaculately reliable performance measures.

Assorted works have been recommended so far based on the models stated. Wael Etaiwi et al. [2] developed a model based on the Naïve Bayes (NB) and the Support Vector Machine (SVM). Comparison of two classifiers are evaluated and revealed that NB outperforms SVM in terms of performance metrics. Iqbal H. Sarker [3] reviewed the machine learning algorithms which used to develop an intelligent system and executed in various aspects. The main aim of this article is to analysis the machine learning techniques and provide the significance of these to the research domains. Based on this paper we can comprehended the classification algorithms deeply and make sense of the sphere of the applications where they are pertinent. Ion Smeureanu1 et al., [4] analyzed that in the private banking industry, categorizing customers' is a decisive phase for the success of the organization due to achieve the target outcome. In the new era, the customer is the asset for the smooth-running of business. Most of the business is shifted to customer-centric. The article concentrates on support vector machines and neural network for the classification. Both are the well-suited machine learning algorithms which sort out well in this area. Elzhan Zeinulla et al., [5] assessed different classification models for predicting the outcomes of bank telemarketing. Based on the study, Random Forest and Artificial Neural Network are more efficient because of their high accuracy.

Emad Abd Elaziz Dawood et al., [6] aims to study in detail the customer profiling artificial intelligence techniques used for the better improvement of banks' revenue by finding the valuable customer which may lead to trademark in the business profit. Bahzad Taha Jijo et al., [7] recommended decision tree classifier implementation with prominent parameters in different datasets and the study re-emphasize the researchers to apprehend the classifier acutely for the future work. Anuradha et al., [8] paper reviewed and analysed decision tree classification with the tuning parameters and measures used for the evaluation which would be worthwhile for the researchers. Here, in the work studied all the approaches used for building decision tree classifier. Harsh H. Patel et al., [9] presented a comprehensive study on the decision tree classifier and the various parameters involved in the classifier. Taha Chicho et al., [10] proposed to organize and predict a group of objects. The K-Nearest Neighbours, Decision Tree, and Random Forest methods are used in the study, and the findings proved that the K-nearest neighbours outperformed more than the other classifiers, nevertheless, it is excellent for certain grouping tasks like economic forecasting [11][12]. N. S. Ahmed et al., [13] reviewed and build something up using classification algorithm Random Forest for making it up to a successful decision system. Here, the proposed decision system is built up for forecasting student's performance. Priyanka et al., [14] summarized that essential features of decision tree can be tuned to build up better decision-making system. Briefly the paper reviewed on different algorithms implemented in literature for finding relevant attribute and methods. Knowledge of tuning DT, ensemble methods, fuzzy technique, etc. are grasped. Ashish Kumar Pandey et al., [15] comprehended the machine learning methods for cancer image recognition and classification.

Shangzhou Wang et al., [16] proposed hyperparameter tuning feature selection and evaluation in ML models. The study also revealed that the researchers can save their time by searching the tuning method of ML by analysing the paper. Here, the GridSearchCV are expounded and executed. Veeralagan. J et al., [1] recommended a hyper parameter tuning technique GridSearchCV in machine learning models for detecting the Alzheimer's disease very precisely and streamlined the metrics. The study proved that the tuning model is elegant because of its exhaustive search and strength to find out the best parameter value for tuning the model. The metrics drawn up for the computation is precision, recall, f1-score, and accuracy and figure out which model fits with the given data. Daniel Mesafint Belete et al., [17] presented a hyperparameters optimization using gridsearch in eight machine learning models by optimizing the parameters of models for forecasting the output of HIV tests. In the study of the paper the main encounter is to find the optimal parameters of the best model. The technique GridSearchCV used here is very strong technique for the finest prediction. The valuation is done in two stages. First with default parameters in machine learning models and then, the GridSearch HyperParameter Optimization (GSHPO) approach is adopted in the models. The evaluation metrics are estimated in each stage and employing GSHPO approach for the prediction might be acceptable.

Yoga Religia et al., [18] recommended that the innovative technology developed to tune the machine learning model, namely, Naive Bayes classification algorithm using credit data. Here, swarm intelligence tuning technique, Particle Swarm Optimization is employed. The combination of Naïve Bayes with PSO is developed and can improve the recall, and accuracy value. B. Chopard et al., [19] summarized that K. K. Vardhini et al., [20] comprehensive analysis is done on nature-based swarm intelligence algorithms and figure out the ways of implementing the optimization techniques. Jun Li et al., [21] described the swarm optimization technique with the classifiers in feature selection method and find out the optimal solution. A. Lamba et al., [22] evaluated and implemented the Particle Optimised Scored K-Nearest Neighbour hybrid strategy in this study as a new way to enhance the performance of the KNN classifier to address the issue of multidimensional data in KNN. I. Ibrahim et al., [23] comprehended the role of machine learning classification in the different domains.

Machine learning classifiers, including SVM, KNN, Decision Tree, and Random Forest have been successfully used in classifying banking customers, although their effectiveness greatly relies on the ability to select the optimal hyperparameters. Conventional tuning strategies provide a low performance improvement and high computational overheads in cases where operators are used to high-dimensional and large datasets. Recent research has shown that optimization methods based on swarm intelligences, especially Particle Swarm Optimization, have better global search performance and convergence speed than exhaustive search methods. Nevertheless, systematic optimization of hyperparameters using GlobalBestPSO on multiple classifiers in fine-grained multiclass banking customer segmentation has not been fully exploited, which justifies the suggested solution.

Based on the summaries stated above, classification-based optimized model using GlobalBest PSO (gbestPSO), a nature-based swarm optimization techniques are employed. The combination of gbestPSO with ML methods can accurately search a large parameter space, and figure out the best parameter values than the grid and random search hyper tuning techniques. Without much iteration, this boosted model can quickly reach the optimal or almost optimal solution and the overall performance may be high compared to the traditional tuning methods. Additionally, the hybrid approach for feature selection or dimensionality reduction using the PSO may cause an omission of relevant attributes or bias may occur. The combination of the use of gbestPSO-KNN, gbestPSO-SVC, gbestPSO-DT, and gbestPSO-RF algorithm, a novel approach could mark it more beneficial.

MATERIALS AND METHODS

In this section discussed in detail about the dataset, and methodology used for multiclass classification of bank customer.

Dataset

The dimensions and features of the dataset estimate the accuracy of classification model precisely. However, finding the real time data from the bank was very tedious task. The main challenge in collecting the data from the bank is they are not willing to explore the customer details due to confidentiality. In this work, the dataset with more than 1 lakh of records, with 43 features, collected from Indian Bank are used. Due to privacy concerns, the name of the bank will not be disclosed. The real dataset is probably messy and need data pre-processing such as analysing the data, denoising, imputation, data reduction, integer encoding categorical variables, oversampling, normalization, splitting dataset into two sets and feature scaling. The algorithm is then trained using the training set, then evaluated using cross-validation techniques and other non-parametric measures.

The bank customers are classified into five categories – Class0: Outstanding Customers, Class1: Excellent Customers, Class2: Good Customers, Class3: Satisfactory Customers, and Class4: Bad Customers. Based on the attribute loan_status, which describes the credit information of the customers and consist of the values - 'Fully Paid', 'Charged Off', 'Current', 'Default', 'Late (31-120 days)', 'In Grace Period', 'Late (16-30 days)', 'Does not meet the credit policy. Status: Fully Paid', 'Does not meet the credit policy. Status: Charged Off '. Fully Paid means the debt has been fully reimbursed, either by prepayment or at the end of the loan period. The Charged Off value indicates that the account has been closed due to additional charges and written off as a loss by a lender or creditor. The value Current means the debtor is paying back the loan on schedule. Default specifies that the borrower does not return the money in accordance with the original arrangement. Grace period is the period of time after the due date that the borrower has to make payment without incurring penalties. In the Class0, the values Fully Paid and Current are considered. The Class1 consist of Does not meet the credit policy. Status: Fully Paid, and In Grace Period. In Class2 specifies Late (16-30) days. The Class3 includes Late (31-120) days, and Default. In Class4 – Charged Off, Does not meet the credit policy: Charged Off. The Class0 and Class1 are taken as the profitable Customer, i.e., the Outstanding Customers and Excellent Customers, based on their credit history and thereby these classes of customers are cost-effective customer for the bank.

Methodology

The banking and financial sector is chosen as the subject of the study because it highly relies on proper customer classification to handle credit risks, profitability, and strategic decision-making. The customer data processed by banks is non-linear, classes are imbalanced, and there is uncertainty because it is handled in large scale, high-dimensional, and heterogeneous. Conventional statistical and rule-based methods tend to be ineffective in the modeling of these complexities to produce sub-optimal risk assessment and customer segmentation. Besides, multiclassification of customers is necessary at the fine-grained level to determine different degrees of creditworthiness and profitability instead of using binary results. The fact that a large real-life banking data comprising over one lakh records and heterogeneous attributes are available, further supports the relevance and validity of the study. Figure 1 illustrates the general framework of a proposed methodology. The study includes various steps for customer classification, i.e., data acquisition (Bank Customer data), data preprocessing (Imputation, Encoding, and Normalizing dataset), training ML classifiers (KNN, SVM, DT, RF), build models using tuning (GridSearchCV) and optimization techniques (GlobalBestPSO) for better enhancement, evaluation of performance indicators (Precision, Recall, F1-score, Accuracy, TPR, FPR, AUC-ROC, Kappa value), and data visualization (plots and graphs) based on the metrics. Model was carefully selected based on visualization and metrics.

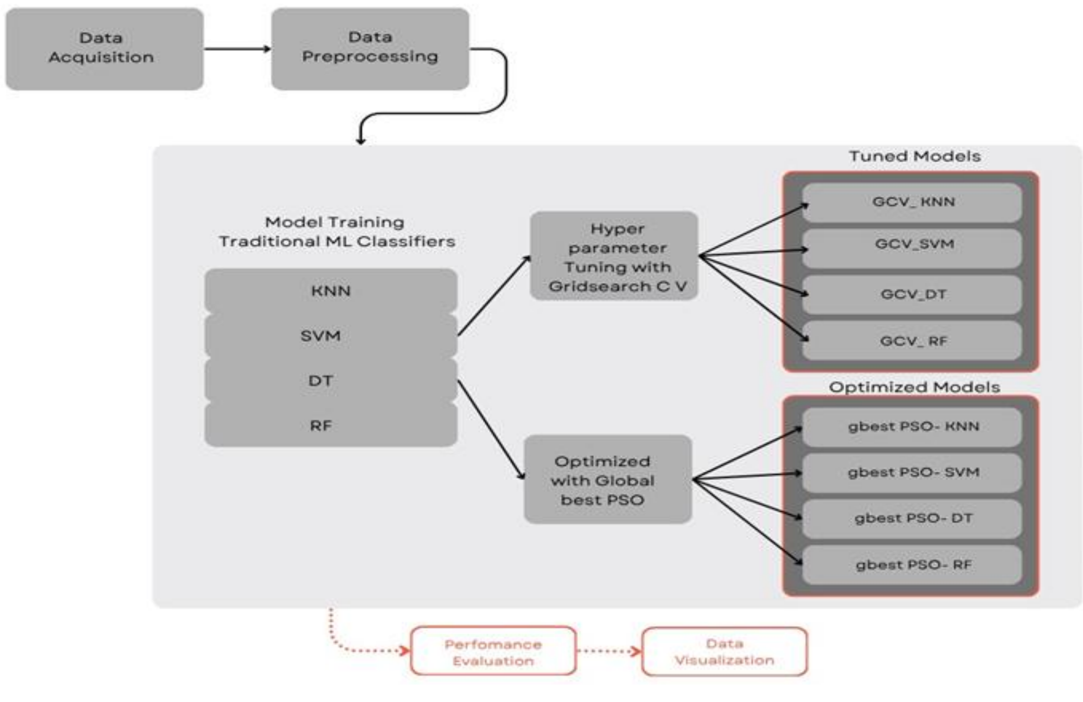


Figure 1. The framework of the proposed methodology

Machine Learning Classifiers

In this research article, classification ML models are categorized into Simple, Hyper Tuned GridSearchCV, and Swarm Optimized models, based on methods employed for the classification. The various data classification techniques are used in the study like support vector classifier, k- nearest neighbour classifier, decision tree classifier, and random forest classifier. The exploration done in four classifiers that are detailed below.

Support Vector Machine (SVM)

SVM [24][25] is a powerful classification algorithm in which the objects may or may not be linearly separable using kernels, a complex mathematical function. It can manage structured and semi-structured data, and it can manage complex functions if the appropriate kernel function is defined. The plus of the classifier is to manage high dimensional data, which does not become trapped in local optima. The linear SVC performance will be low when the size of the dataset is high. But in the non-linear SVC due to kernel function, the higher dimension can handle very effectively and algorithm classifies the data. In the present study, the SVC without default parameters is estimated and by tuning the parameters using function for multi-class classification are measured.

Decision Tree (DT)

The decision tree capture information in a tree-like structure, which may also be simplified by writing it as a set of distinct rules. Their strength lies in the ability to several feature subsets and decision rules at multiple stages of the classification [7]. The main advantage of DT is it is very efficient in multi-dimensional data and is more valuable for exploratory knowledge discovery. Decision tree classifier is an inductive learning with achieving high accuracy by pruning these trees to make it small subtree and provide information more ease and efficiently within a limited period of time, and further improve the computation time for large dataset. [24] Decision trees' key drawbacks include their potential for instability, difficulty in managing tree size, prone to sampling error, and only provide locally optimal solutions rather than globally ideal ones. Entropy as shown in Equation 1 and information gain as in Equation 2 determine how a node in a decision tree splits [26]. Gini index as shown in Equation 3 and Entropy are used to determine a dataset's impurity or uncertainty. In the current dataset the uncertainty of data is very high based on the calculation of the Entropy value of given dataset is 17.455 and Gini index to 1.0. The higher the entropy value, the randomness of data is more. So, the depth of the tree may increase to 10 to reach the target-entropy value equal to 0.551. The good value of entropy is within the range of 0 to 1. A Gini value of 1 indicates that the data are distributed randomly among the various classes. The index value of Gini ranges from 0 to 1, 0 indicates purity of classification, i.e., there is no uncertainty and 0.5 represents equal distribution of classes [7][26].

$$\mathrm{Entropy} = -\sum (\mathrm{P(x = k)} * \mathrm{log2}\ (\mathrm{P(x = k)}) \quad (1)$$

$$\mathrm{InformationGain(feature)} = \mathrm{Entropy(Dataset)} - \mathrm{Entropy(feature)} \quad (2)$$

$$\mathrm{GiniIndex} = 1 - \sum \left(\mathrm{P(x = k)}\right)^2 \quad (3)$$

The most informative feature is one that does the best job of separating knowledge about the target feature from ambiguity [27]. We continue looking for the most informative feature until we find only pure leaf nodes.

K-Nearest Neighbour (KNN)

One of the earliest and most straightforward classification techniques is the KNN [27] algorithm. The training of the data is done apace. The KNN algorithm relies on the objects that are similar, exist in proximity. That is why this algorithm is named as K Nearest Neighbour. Here k is equal to the number of nearest neighbours. The concept of KNN algorithm is similarity, based on this classification done. It is also known as lazy learners because it does not learn from the training set.

Random Forest (RF)

Many scientific fields have successfully used the Random Forest (RF) technique to reduce large dimensional and multi-source data. RF are thought to be high stable in the presence of outliers and in very high dimensional parameter spaces than other ML models [28]. The Gini impurity criterion index is used to evaluate the implicit feature selection of variable importance carried out by RF using a random subspace methodology [28]. An approach known as an ensemble learners employs a number of weak learners or independent models, some of which may be similar or dissimilar, to produce an output or make predictions. An ensemble of multiple decision trees formed a Random Forest.

Hyper-Parameter Tuning Using GridSearchCV

Machine learning deals with two types of parameters – model parameters and hyper-parameter. The first parameters that is model parameter are variables that can be learned from training data. The hyper-parameter are the adjustable parameters which are used to get the best outcome. These parameters are shown in Table 1 and it regulate how the model learns. There is a popular tuning approach – GridSearchCV. GridSearchCV tests all possible combinations of the values accepted in the dictionary and evaluates the model for each combination using the cross-validation method as shown in Algorithm 1. As a result, after employing this function, we can determine the accuracy and loss for each set of hyper parameters and select the best combination of parameters which grades best performance.

Table 1. Hyper-parameters of the classifier model that are tuned

ML Classifiers	Parameter Name	Default Parameter Value	Tuned Parameter Value
SVM [24][25]	C, Gamma, Kernel	C = 1.0, Gamma = scale, Kernel = rbf	C: [0.1, 1, 10, 100, 1000], gamma: [1, 0.1, 0.01, 0.001, .0001], kernel: ['rbf']
KNN [27]	n_neighbors, weights	n_neighbors: 5, 'Weights: uniform	n_neighbors: [5, 7, 9, 11], weights: ['uniform', 'distance']}
DT	criterion, max_depth	criterion: gini, max_depth: None	criterion: ['Gini', 'Entropy'], max_depth: [4,5,6,7,8,9,10,11,12,15]}
RF [28]	n_estimators, max_depth, random_state	n_estimators :10, max_depth :2, random_state= 0	n_estimators: [2,4,6,8], max_depth: [3,5,7,9], random_state: [10,20]

Given that the settings for the hyperparameters and their effectiveness heavily depend on the ML algorithm and the kind of hyperparameter, discrete or continuous values, the Hyperparameters Optimisation (HPO) problem requires an in-depth knowledge of the ML model [29]. The aim behind decision-theoretic techniques is to search through combinations of hyperparameters in the hyperparameter space, calculate their accuracy, and then select the combination that performed the best. The grid search is available if we test over a fixed domain of hyperparameter values [29]. Here's a generalized algorithm as pseudo code for performing a grid search hyperparameter tuning in four classifiers- SVM, KNN, DT, RF.

Algorithm 1: Hyper-Parameter Tuning Using GridSearchCV

Begin

1. Initialize the classifier-Set classifier, Set default values in classifier. Hyperparameters

a) Define possible values for hyperparameters-Set param_grid {hyperparameter_1: [v1, v2, v3], hyperparameter_2: [v4, v5], hyperparameter_3: [v6, v7]}

b) Apply Grid Search with cross-validation- Set grid_search as GridSearchCV (classifier, param_grid) *c) Train grid search on training data- CALL grid_search.fit (training_data, training_labels)*

*d) Find best hyperparameters- Set best_params as grid_search. best_params_*

2. Create a new classifier with best hyperparameters- Set optimized_classifier ← New Classifier with best_params

3. Train the optimized classifier- CALL optimized_classifier.fit (training_data, training_labels)

4. Predict labels for test data- Set predicted_labels from optimized_classifier. predict(test_data)

End

Hyperparameter Tuning Using PSO

In the recent research studies, the role of optimization technique in any field is decisive. The purpose of optimization is to determine the most optimal solution to the problem. Particle Swarm Optimization is an intelligence system, scale the search space and attempt to model improved ML versions. In PSO, individuals referred to as particle, are then flown across the hyper dimensional space [30]. Each particle upholds track of the hyperspace coordinates that correspond to the best fitness it has so far found. That fitness's value known as pbest is also saved. A second-best value is also kept track of. The gbest, global version of the particle swarm optimizer, keeps track of the overall best value and its position thus far achieved by each particle in the population [30][31]. Each particle has velocity and position. In the multidimensional space each particle is flying to search for an optimal solution. Using PSO we can easily convert the problems into functional optimization problems. The main advantage of this algorithm is its fast convergence compared to others.

Global_Best Particle Swarm Optimization

In the study, the Global_Best Particle Swarm Optimization (gbestPSO) algorithm a variant of PSO, is used to compare a set of potential solutions in an effort to select the best one and implemented as a star-topology in which each particle is drawn to the particle that is performing the best. That is the global best position. The two factors are considered in each iteration, i.e., position and velocity of each particle and are altered. The inertia weight W, determines how quickly the swarm moves. While a low value of W can increase the exploitation of the present solution, a high value of W can improve the search space's exploration. The cognitive and social parameters c1 and c2, respectively, determine the particle's optimal position and the swarm's global best position as shown in Algorithm 2.

The concept of combining the traditional classifiers with the advanced optimization technique gbestPSO, explore the performance of each classifier with the tuned parameters. Based on the classifier the hyper parameters may be changed. For example, in the SVM, the parameters are C–the penalty parameter, gamma, and the kernel coefficient to be tuned for the best performance. Other classifier parameters and values are mentioned in the above Table1. If the boundaries for these hyper parameters can be defined, the gbestPSO algorithm is used to look for the optimal values of the parameters that minimize the model's cost function. The benefit of gbestPSO is that it quickly converges to the best solution and expands the viable region widely to add diversity.

Algorithm 2: Hyper-Parameter Tuning Using gbestPSO

Begin

1. Define the objective function

Function ObjectiveFunction (hyperparameter_list)

SET cost_list as empty list

For each set in hyperparameter_list Do

- *Train the model using set and evaluate model performance*
- *Compute cost using error or loss value and Add cost to cost_list*

End For

Return cost_list

End Function

2. Define hyperparameter boundaries

Set lower_bounds as [lb1, lb2, lb3, ..., lbn], upper_bounds as [ub1, ub2, ub3, ..., ubn]

3. Initialize Global PSO optimizer

- *SET number_of_particles as P, SET dimensions as number of hyperparameters*
- *SET options ← {c1, c2, w}*

Initialize PSO with number_of_particles, dimensions, options, bounds

4. Set iteration count as SET max_iterations from I

5. Optimize using Global PSO

For iterations 1: max_iterations Do

- *Compute costs from ObjectiveFunction(current_particle_positions)*
- *Update particle velocities, positions, global best position & best cost*

End For

6. Display results

Print best_hyperparameters & best_cost

End

For each classifier, the above general algorithm is followed. The parameter list may be varied depending upon the classifier. The best hyperparameters for the four classifiers' may be discovered more quickly and precisely by combining gbestPSO with classifiers. The efficiency of the proposed hybrid global best PSO with four classifiers is evaluated by comparing its performance with benchmark models. The effectiveness of the Optimized GlobalBestPSO-SVM, Optimized GlobalBestPSO-K-NN, Optimized GlobalBestPSO-DT, and Optimized GlobalBestPSO-RF classifiers is assessed. The comparison of test results shows that the optimized technique outperforms than the basic classifiers and classifiers with GridsearchCV in terms of performance measures, making it a more effective classification method. The analysis's goal was to determine the performance metrics of classifiers that had been improved to an accuracy score of almost above 95% along with other satisfied parameters.

Mathematical Formulation of the GlobalBestPSO-based Machine learning Classification Model

The dataset can be defined as shown in equation 4.

$$\mathrm{D} = (\mathrm{x_i}, \mathrm{y_i})\mathrm{for}\ \ \mathrm{i} = 1,2, \dots, \mathrm{N} \qquad (4)$$

where $x_i \in R^d$ represents the d-dimensional feature vector of the $\mathrm{i^{th}}$ customer, and $y_i \in \{0,1,2,3,4\}$ represents the corresponding class label. The machine learning classifier can be expressed as a mapping function: $\hat{y} = f(x; \theta)$, where θ represents the set of hyperparameters of the classifier. The objective of the learning process is to minimize the classification loss function: $\mathrm{L}(\theta) = 1 - \mathrm{Accuracy}(\theta)$ and the cross-validation accuracy can be used to define the fitness function as in equation 5.

$$\mathrm{F}(\theta) = 1 - (1/\mathrm{K})\Sigma\mathrm{Accuracy_k}(\theta) \qquad (5)$$

In Particle Swarm Optimization, each particle i is characterized by: Position vector: $\theta_\mathrm{i}(\mathrm{t})$ and Velocity vector: $v_i(t)$. The velocity update equation is given by the equation 6.

$$\mathrm{v_i}(\mathrm{t}+1) = \mathrm{wv_i}(\mathrm{t}) + \mathrm{c1r1}\big(\mathrm{pbest_i} - \theta_\mathrm{i}(\mathrm{t})\big) + \mathrm{c2r2}\big(\mathrm{gbest} - \theta_\mathrm{i}(\mathrm{t})\big) \quad (6)$$

The position update equation is: $\theta_\mathrm{i}(\mathrm{t}+1) = \theta_\mathrm{i}(\mathrm{t}) + \mathrm{v_i}(\mathrm{t}+1)$ where, w is the inertia weight, c1 and c2 are cognitive and social learning coefficients, r1 and r2 are random values uniformly distributed in [0,1], pbest_i is the personal best position of particle i, and gbest is the global best position among all particles. The optimization problem can be formulated as: $\theta *= \arg \min\theta\mathrm{F}(\theta)$ subjected to: $\theta_{\mathrm{min}} \leq \theta \leq \theta_{\mathrm{max}}$. The algorithm repeats the process of updating the position of particles until the convergence criteria are satisfied. A final classifier is then trained using the optimum set of hyperparameters θ* and assessed using unseen test data.

EVALUATION METRICS

The model evaluation is done by measuring the performance factors such as accuracy, Cohen's Kappa, F-score, Precision, Recall, ROC-AUC score, TPR, and FPR. The percentage of accurate classifications is what is used to calculate accuracy score given by equation 7 and precision & recall as in equation 8 & 9 respectively. A higher accuracy rating means that more of the model's predictions were accurate. Cohen's Kappa is suggested as a way to gauge how well two classifiers agree. The chance of agreement (total accuracy) minus the possibility of disagreement (random accuracy) divided by 1 minus the probability of disagreement is used to determine the Kappa coefficient as shown in equation 12. This score is positive, indicating that the parties have reached some sort of understanding. Higher values of the Kappa coefficient signify stronger agreement or better performance, with values ranging from -1 to 1.

The F1- score as in equation 10, which is a single score formed by combining recall and precision. Consequently, a higher F1 score is preferred since it denotes a better balance between recall and precision. Precision measures how well a model can forecast a particular classification. A greater precision indicates that a model provides more relevant results than irrelevant ones. Recall quantifies the proportion of pertinent data points that the model accurately identifies. As it shows a reduced rate of false negatives and an outstanding ability to record all positive cases, a higher recall value is typically recommended. ROC-AUC weighed the performance of a multi-class classification model using a characteristic curve, which is a probability graph, demonstrate the curve between two parameters tpr (true positive rate) and fpr (false positive rate) at various threshold levels. A classification model's accuracy in accurately classifying actual positive events is measured by a performance parameter called TPR. The ratio of true positives to the total of true positives and false negatives is used to compute it. FPR, shown in equation 11 is a performance indicator that quantifies the percentage of real negative instances that classification models mistakenly classify as positive. The mathematical formulations of the evaluation metrices are

$$\mathrm{Accuracy} = \frac{\mathrm{T\_P} + \mathrm{T\_N}}{\mathrm{T\_P} + \mathrm{T\_N} + \mathrm{F\_P} + \mathrm{F\_N}} \qquad (7)$$

$$\text{Precision} = \frac{\text{T\_P}}{\text{T\_P} + \text{F\_P}} \quad (8)$$

$$\text{Recall (TPR)} = \frac{\text{T\_P}}{\text{T\_P} + \text{F\_N}} \quad (9)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

$$\text{FPR} = \frac{\text{F\_P}}{\text{F\_P} + \text{T\_N}} \quad (11)$$

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (12)$$

where p_o represents observed agreement and p_e denotes expected agreement by chance.

RESULT ANALYSIS AND DISCUSSION

The Table 2 shows the comparison of performance metrics of simple classifier SVM, tuned SVM with gridsearch and optimized SVM with gbestPSO. Table 3 displays the comparison of performance metrics of simple classifier KNN, tuned KNN with gridsearch and optimized KNN with gbestPSO. The Table 4 reveals the comparison of performance metrics of simple classifier DTC, tuned DTC with gridsearch and optimized DTC with gbestPSO. The Table 5 outlines the evaluation of performance metrics of simple classifier RF, tuned RF with gridsearch and optimized RF with gbestPSO.

All experiments in this study were implemented using Python in Jupyter Notebook. The machine learning models were developed using the Scikit-learn library. It has the efficient implementations of Support Vector Machines, K-Nearest Neighbours, Decision Tree, and Random Forest classifiers. Data preprocessing operations such as handling missing values, normalization, encoding categorical variables, and dataset splitting were carried out using NumPy and Pandas libraries.

Table 2. Performance evaluation metrics of the simple SVM, tuned with GridSearchCV, optimized with gbestPSO

Algorithm		**Precision**	**Recall**	**F1-Score**	**TPR**	**FPR**	**Kappa Value**	**AUC**	**Accuracy**
Simple SVM	Class0	1.00	0.89	0.94	0.891	0.001	0.95	0.997	0.96
	Class1	0.98	1.00	0.99	1.000	0.006			
	Class2	0.97	1.00	0.99	1.000	0.008			
	Class3	0.90	0.93	0.91	0.920	0.026			
	Class4	0.98	1.00	0.99	1.000	0.006			
SVM with GridsearchCV	Class0	0.99	0.91	0.95	0.977	0.002	0.96	0.997	0.97
	Class1	0.97	1.00	0.99	0.997	0.000			
	Class2	1.00	1.00	1.00	1.000	0.000			
	Class3	0.91	0.91	0.91	0.981	0.039			
	Class4	0.96	1.00	0.98	0.981	0.000			
SVM with GlobalBestPSO	Class0	0.97	1.00	0.98	0.970	0.000	0.99	0.999	0.99
	Class1	1.00	1.00	1.00	1.000	0.000			
	Class2	1.00	1.00	1.00	1.000	0.000			
	Class3	1.00	0.97	0.99	1.000	0.008			
	Class4	1.00	1.00	1.00	1.000	0.000			

In the provided Table 2 evaluation metrics for different Support Vector Machine (SVM) models, we can observe distinct performance characteristics across the three models: Simple SVM, SVM with GridsearchCV, and SVM with GlobalBestPSO. These models are evaluated across multiple classes (Class0 to Class4) based on precision, recall, F1-score, true positive rate (TPR), false positive rate (FPR), Kappa value, area under the ROC curve (AUC), and overall accuracy. The Simple SVM demonstrates high precision and recall for most classes, resulting in impressive F1-scores and TPR values. It is particularly effective in Class1 and Class2, with F1-scores of 0.99 and perfect recall. However, its FPR is relatively high for some classes, showing the false positives to modest level. On the other hand, the SVM with GridsearchCV offers improved performance with even higher precision, recall, and F1-scores across most classes, resulting in a superior TPR. It also minimizes the FPR, indicating a better balance between true and false positives. The SVM with GlobalBestPSO takes it a step further, achieving near-perfect precision, recall, and F1-scores for all classes, with almost no false positives (FPR close to zero). It particularly excels in Class3 with a perfect precision-recall balance.

While all three SVM models show strong performance in terms of precision and recall, the SVM with GlobalBestPSO stands out as the most robust and reliable classifier with consistently high metrics across all classes. The SVM with GridsearchCV also performs admirably and is a significant improvement over the Simple SVM. The choice of the best model should consider the specific trade-offs between precision, recall, and FPR, depending on the application's priorities, but overall, the SVM with GlobalBestPSO seems to offer an excellent compromise between these factors and should be considered for scenarios where high accuracy and minimal false positives are essential.

Table 3. performance evaluation metrics of the simple KNN, tuned with grid search, optimized with gbestPSO

Algorithm		**Precision**	**Recall**	**F1-Score**	**TPR**	**FPR**	**Kappa Value**	**AUC**	**Accuracy**
KNN	Class0	0.94	0.86	0.90	0.860	0.014	0.94	0.991	0.96
	Class1	1.00	1.00	1.00	0.998	0.000			
	Class2	1.00	1.00	1.00	1.000	0.000			
	Class3	0.84	0.94	0.89	0.938	0.040			
	Class4	1.00	0.99	0.99	0.990	0.000			
KNN with GridsearchCV	Class0	0.98	0.88	0.93	0.877	0.004	0.96	0.997	0.97
	Class1	1.00	1.00	1.00	0.996	0.000			
	Class2	1.00	1.00	1.00	0.999	0.000			
	Class3	0.84	0.98	0.91	0.981	0.039			
	Class4	1.00	0.98	0.99	0.981	0.000			
KNN with GlobalBestPSO	Class0	1.00	0.96	0.96	0.933	0.005	0.98	0.988	0.98
	Class1	1.00	1.00	1.00	1.000	0.000			
	Class2	1.00	1.00	1.00	1.000	0.000			
	Class3	0.93	0.98	0.95	0.980	0.018			
	Class4	1.00	1.00	1.00	0.996	0.000			

The evaluation metrics provided in Table 3 shed light on the performance of three K-Nearest Neighbors (KNN) models, namely KNN, KNN with GridsearchCV, and KNN with GlobalBestPSO, across multiple classes. The basic KNN model exhibits commendable precision and recall for the majority of classes with noteworthy F1-scores in Class1 and Class2. Nevertheless, it suffers from a relatively elevated false positive rate (FPR) in certain classes. In contrast, the KNN with GridsearchCV presents a notable enhancement in various metrics across various classes, coupled with a substantial reduction in FPR. It particularly shines in Class1 and Class2, boasting impeccable precision and recall. Meanwhile, the KNN with GlobalBestPSO consistently performs well across all classes, boasting high precision, recall, and F1-scores, all while keeping a tight lid on the FPR. It excels in Class1 and Class2, mirroring the strengths of the KNN with GridsearchCV.

To summarize, both KNN models with GridsearchCV and GlobalBestPSO exhibit substantial improvements in precision, recall, and F1-scores when compared to the basic KNN model. However, the KNN with GlobalBestPSO emerges as the most well-balanced and dependable classifier, demonstrating consistently impressive metrics across all classes. For applications where high accuracy and minimal false positives are of paramount importance, the KNN with GlobalBestPSO stands out as the optimal choice.

The provided Table 4 offers a comprehensive comparison of three Decision Tree (DT) models, which include the basic DT, DT with GridsearchCV, and DT with GlobalBestPSO. These models have been assessed based on various evaluation metrics, and compare across different classes. In terms of precision and recall, the basic DT model demonstrates exceptional performance in Class0, achieving 100% precision and recall. However, it struggles in Class1 and Class3, with relatively low recall values, indicating that it misses a significant portion of true positives. The F1-Score, which combines precision and recall, reflects this performance. The DT with GridsearchCV makes notable improvements, particularly in Class1 and Class2, achieving higher recall and F1-Scores. However, in Class3 and Class4, its FPR values are still relatively high. The DT with GlobalBestPSO further enhances precision, recall, and F1-Scores across the board, especially excelling in Class3 and Class4 with high precision and recall, and maintaining a low FPR, making it a well-balanced classifier. In summary, while the basic DT model demonstrates excellent performance in some classes, it falls short in others, with relatively high FPR values. The DT with GridsearchCV represents an improvement, notably in Class1 and Class2, but still faces challenges in Class3 and Class4. The DT with GlobalBestPSO stands out as the most balanced and reliable classifier across all classes. For applications where maintaining a low FPR and achieving high precision and recall are crucial, the DT with GlobalBestPSO is the preferred choice, offering a well-rounded solution.

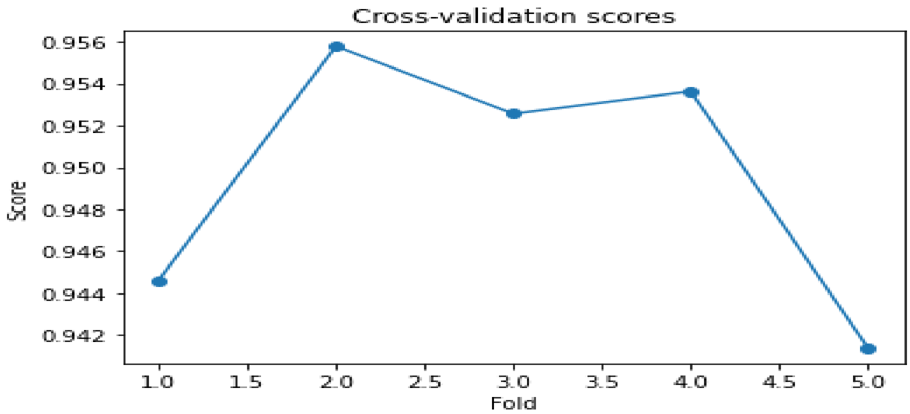
Table 4. Performance evaluation metrics of the simple DT, tuned with gridsearch, optimized with gbestPSO

Algorithm		Precision	Recall	F1-Score	TPR	FPR	Kappa Value	AUC	Accuracy
DT	Class0	1.00	1.00	1.00	0.999	0.000	0.77	0.963	0.81
	Class1	0.90	0.62	0.74	0.623	0.029			
	Class2	0.88	0.84	0.86	0.837	0.031			
	Class3	0.62	0.91	0.74	0.912	0.088			
	Class4	0.67	0.81	0.74	0.812	0.078			
DT with GridsearchCV	Class0	1.00	1.00	1.00	0.999	0.000	0.94	0.992	0.95
	Class1	0.98	0.90	0.94	0.901	0.006			
	Class2	1.00	0.94	0.97	0.941	0.001			
	Class3	0.84	1.00	0.91	0.997	0.039			
	Class4	0.94	0.92	0.93	0.921	0.016			
DT with GlobalBestPSO	Class0	1.00	1.00	1.00	0.999	0.000	**0.99**	**0.995**	**0.99**
	Class1	1.00	0.99	0.99	0.986	0.000			
	Class2	1.00	1.00	1.00	0.995	0.000			
	Class3	0.96	1.00	0.98	1.000	0.011			
	Class4	1.00	0.98	0.99	0.997	0.000			

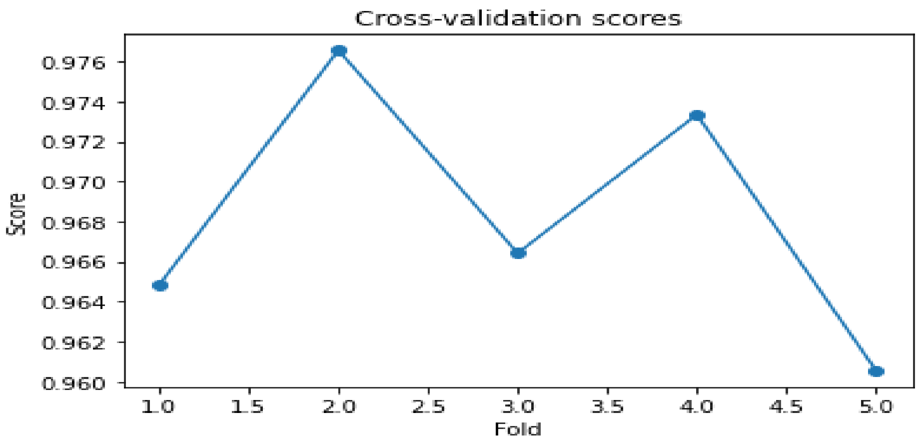
Table 5. Performance evaluation metrics of the simple RF, tuned with gridsearch, optimized with gbestPSO

Algorithm		Precision	Recall	F1-Score	TPR	FPR	Kappa Value	AUC	Accuracy
RF	Class0	0.95	1.00	0.98	0.954	0.000	0.95	0.999	0.96
	Class1	1.00	0.89	0.94	1.000	0.031			
	Class2	1.00	1.00	1.00	1.000	0.001			
	Class3	0.94	0.95	0.95	0.938	0.011			
	Class4	0.91	0.97	0.94	0.907	0.007			
RF with GridsearchCV	Class0	0.99	0.91	0.95	0.911	0.002	0.96	1.000	0.97
	Class1	0.97	1.00	0.99	1.000	0.007			
	Class2	1.00	1.00	1.00	1.000	0.001			
	Class3	0.91	0.91	0.91	0.914	0.022			
	Class4	0.96	1.00	0.98	1.000	0.011			
RF with GlobalBestPSO	Class0	1.00	0.97	0.99	0.973	0.000	**0.99**	**1.000**	**0.99**
	Class1	0.98	1.00	0.99	1.000	0.004			
	Class2	1.00	1.00	1.00	1.000	0.000			
	Class3	0.97	0.98	0.98	0.985	0.007			
	Class4	1.00	1.00	1.00	1.000	0.000			

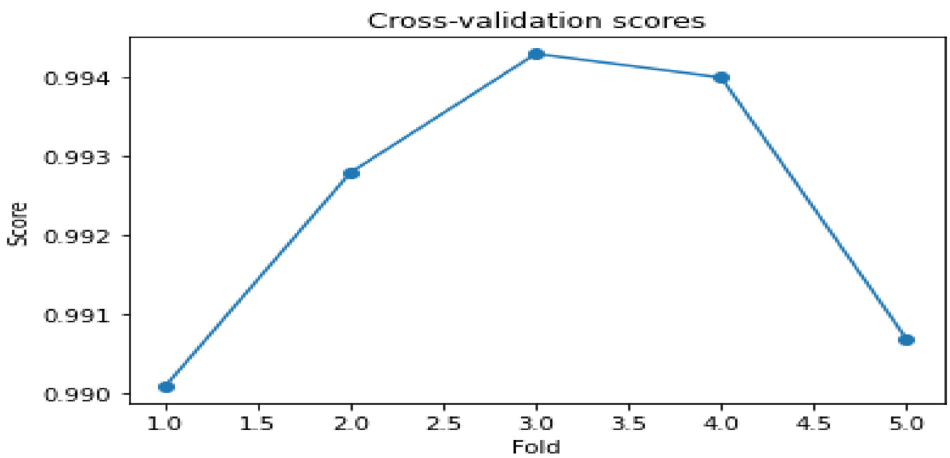
In the comparison of Random Forest (RF) models, including the basic RF, RF with GridsearchCV, and RF with GlobalBestPSO are shown in Table 5, we observe notable variations in performance across different classes. The basic RF model demonstrates high precision and recalls in most classes, with exceptional F1-scores in Class1 and Class2. However, it exhibits a relatively high false positive rate (FPR) in Class3 and Class4. The RF with GridsearchCV improves upon the basic RF model by enhancing precision, recall, and F1-scores across several classes, achieving a better balance between true and false positives. The RF with GlobalBestPSO maintains high precision and recall across all classes while minimizing FPR. It excels in Class1 and Class2, offering a robust and well-rounded classification. In summary, the RF models with GridsearchCV and GlobalBestPSO provide significant improvements in precision, recall, and F1-scores when compared to the basic RF model. The RF with GlobalBestPSO stands out as the most balanced and reliable classifier, demonstrating consistently high metrics across all classes. For applications where high accuracy, precision, and recall are essential, while keeping false positives to a minimum, the RF with GlobalBestPSO proves to be the optimal choice.



(a)

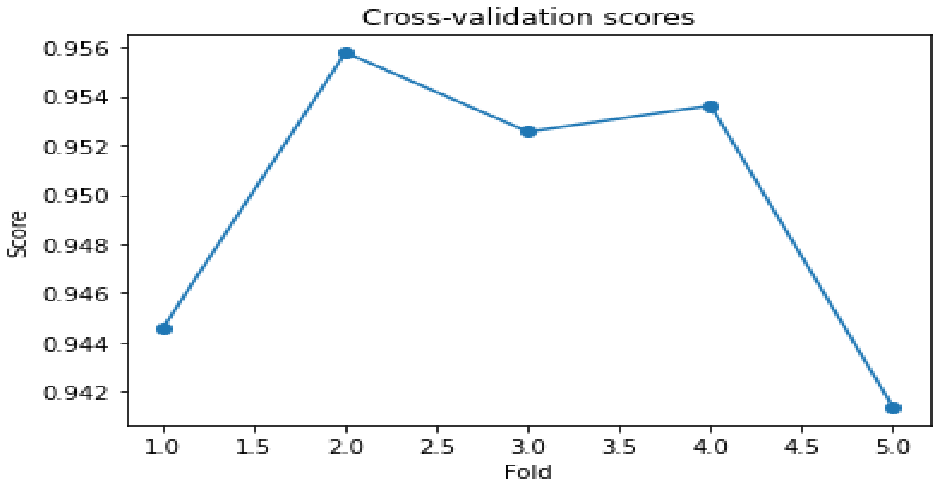


(b)

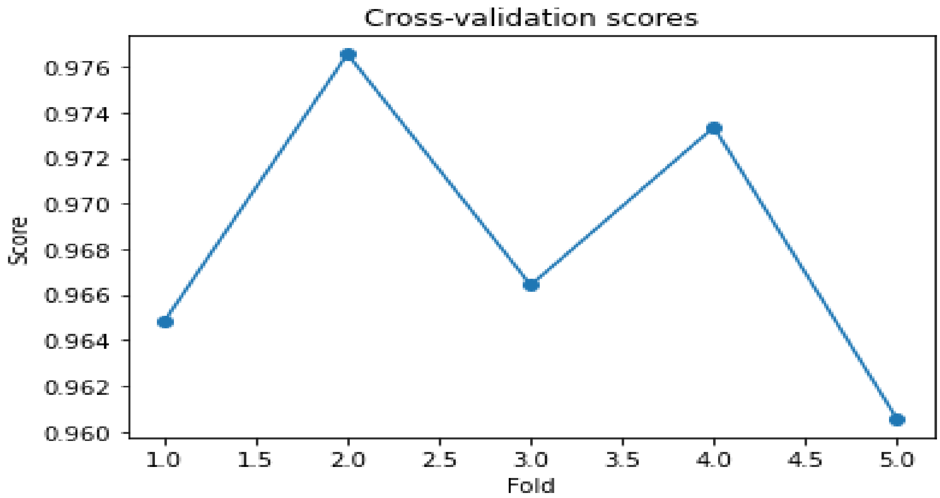


(c)

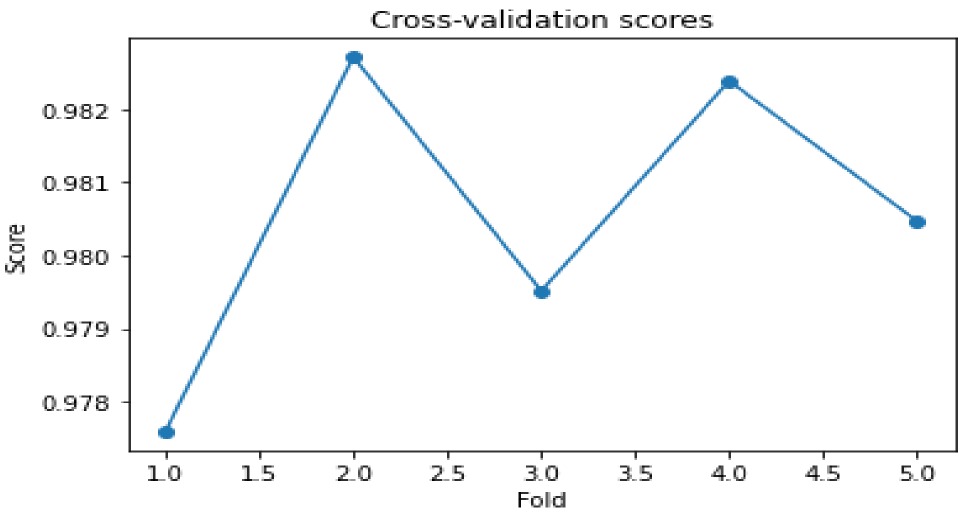
Figure 2. a) Cross validation score of simple SVM, b) SVM tuned with GridSearch, c) SVM optimized with gbestPSO



(a)

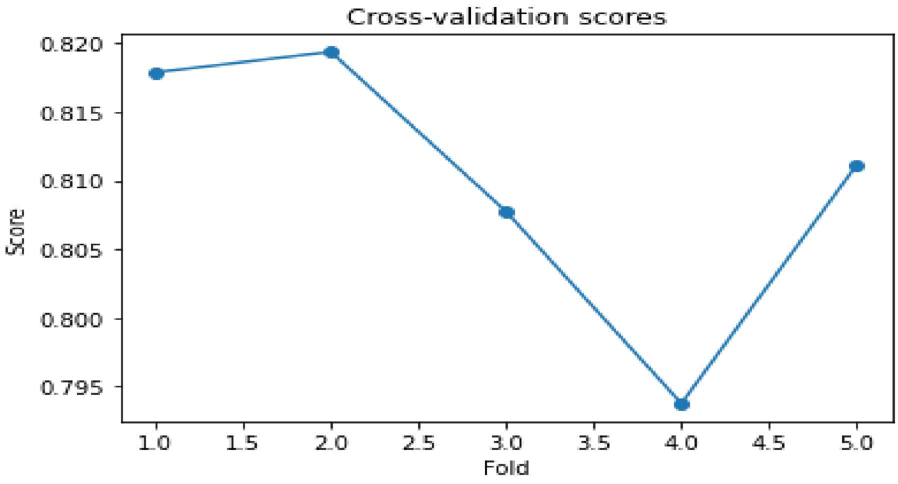


(b)

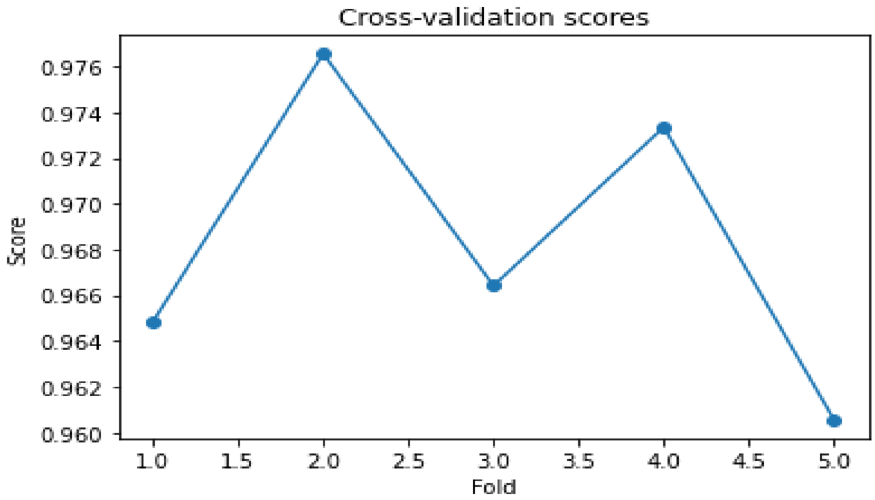


(c)

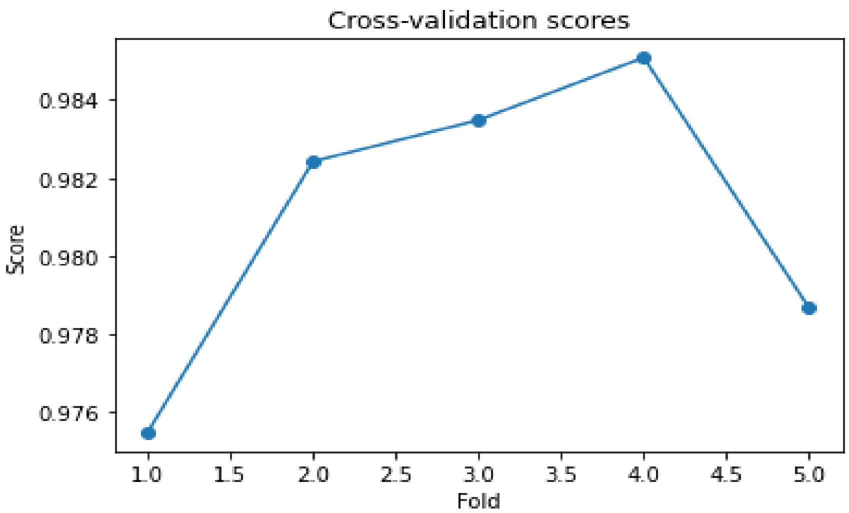
Figure 3. a) Cross validation score of simple KNN, b) KNN tuned with gridsearch, c) KNN optimized with gbestPSO



(a)

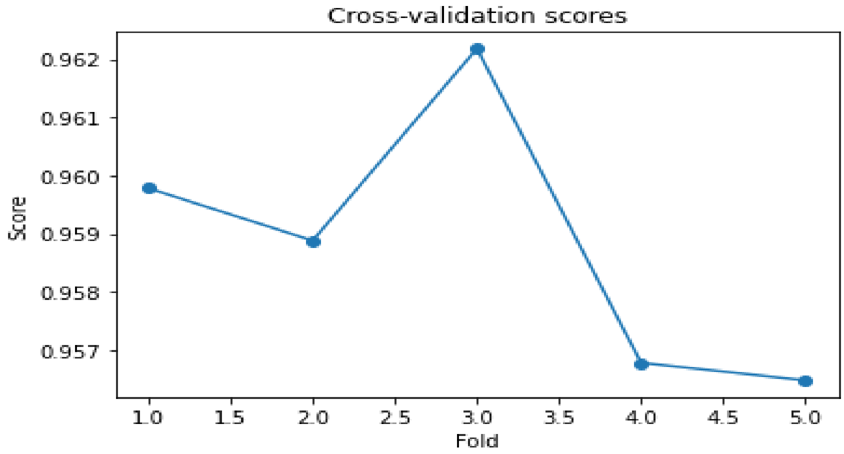


(b)

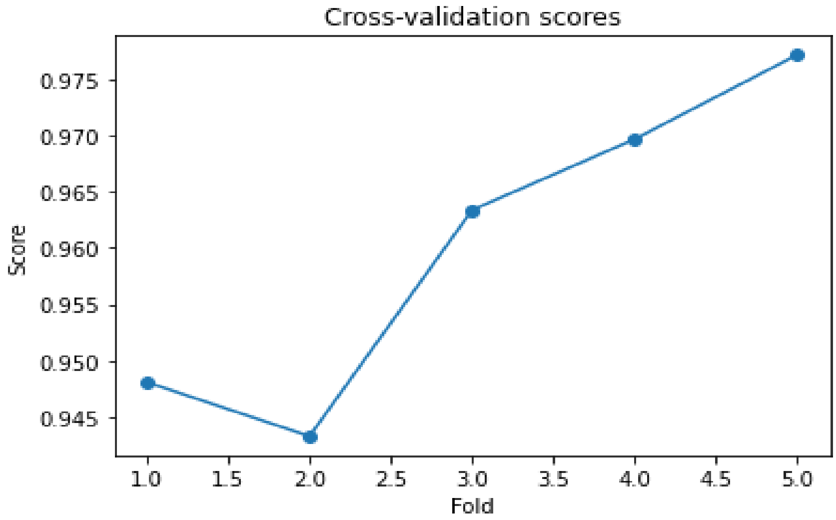


(c)

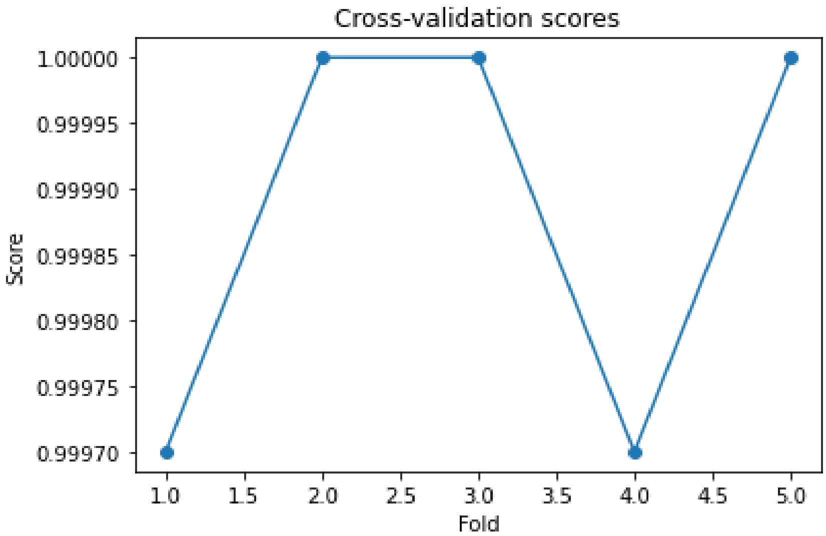
Figure 4. a) Cross validation score of simple DT, b) DT tuned with gridsearch, c) DT optimized with gbestPSO



(a)



(b)



(c)

Figure 5. a) Cross validation score of simple RF, b) RF tuned with GridSearch, c) RF optimized with gbestPSO

During the analysis phase, the model evaluation involves k-fold cross-validation estimation. This technique was applied to the basic, tuned, and optimized models, and the results were depicted in the provided figures. Cross-validation is a vital aspect of model assessment, as it helps in ascertaining a model's performance while mitigating the potential effects of data partitioning. The results presented here, with their high mean cross-validation score and low standard deviation, suggest that both the tuned and optimized models are well-suited for the given task.

The impact of hyperparameter optimization on SVM and cross-validation performance is shown in Figure 2. Figure 2(a) indicates that the baseline SVM. With the application of Grid Search in Figure 2(b), cross-validation scores are stabilized and moderately improved. The maximum and most stable validation performance of the SVM optimized with using gbest-PSO in Figure 2(c). The cross-validation performance of the KNN classifier is depicted in figure 3 at varying tuning strategies. Simple KNN model in Figure 3(a) has its variation in accuracy in validations because it is sensitive to the selection of k and distance values. Figure 3(b) provides the best average score through grid search. Conversely, the gbest-PSO optimized KNN in Figure 3(c) provides better and more consistent cross-validation performances and this shows that PSO can effectively optimize the KNN hyperparameters.

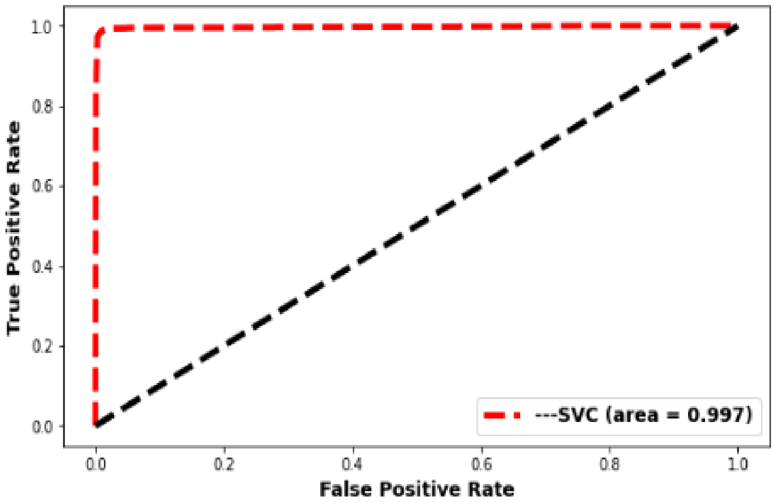
In Figure 4, the comparison of the cross-validation score of the DT classifier is presented. The baseline DT model of Figure 4(a) has the disadvantage of being unstable in terms of performance. The instance of Figure 4(b) with Grid Search tuning generates better scores and consistent within the generalization of suitable parameters of the trees. The DT optimized with gbest-PSO in Figure 4(c) further enhances the performance by balancing between model complexity and accuracy. Figure 5 is the comparison of the cross-validation of the Random Forest (RF) classifier. RF in Figure 5(a) is a reasonably performing model that is however variable as a result of non-optimal parameters of number of trees and feature selection. The optimization of Figure 5(b) with grid search results in better accuracy and lower variance by optimizing these parameters. The gbest-PSO optimized RF in Figure 5(c) has the best overall and most stable cross-validation performance.

Evaluation Of Classifiers Using ROC-AUC Curve

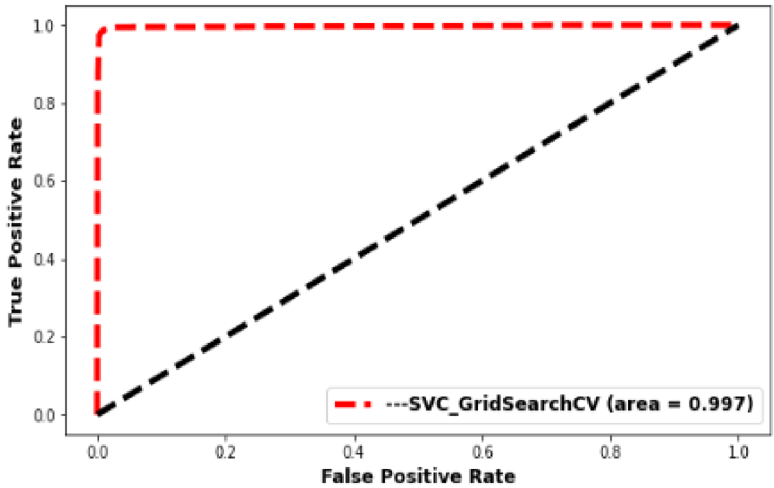
The is a critical metric in the field of machine learning, used to assess the performance of classification models, particularly in scenarios where class imbalances exist or focuses on how well the model ranks positive instances higher than negative ones is crucial. When an ROC AUC score exceeds 99%, it signifies an exceptional level of performance and efficiency in a classification model. A score this high indicates that the model is not only capable of accurately distinguishing between different classes but does so with an extremely high degree of precision. A ROC AUC score exceeding 99% indicates near-flawless separation of classes. An ROC AUC score above 99% reflects exceptional model capabilities for complex classification tasks (Table 6).

Table 6. Implemented models with AUC score

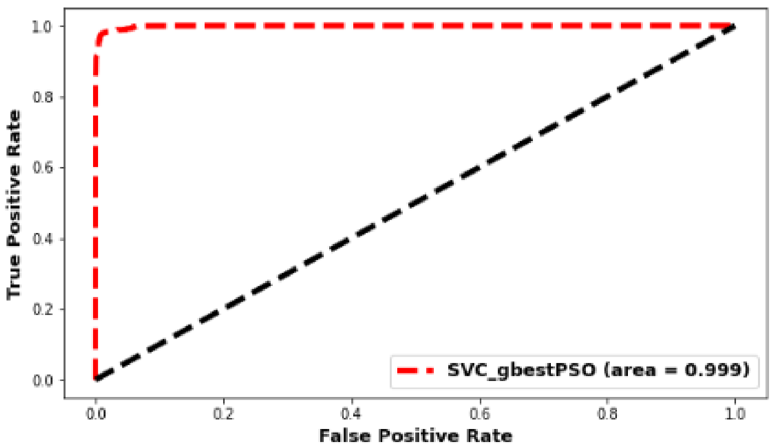
Models	AUC Score	Models	AUC Score
SVC	0.997	Decision Tree (DT)	0.963
SVC with GridSearchCV	0.997	DT with GridSearchCV	0.992
SVC with gbestPSO	0.999	DT with gbestPSO	0.995
KNN	0.991	Random Forest (RF)	0.999
KNN with GridSearchCV	0.997	RF with GridSearchCV	1.000
KNN with gbestPSO	0.988	RF with gbestPSO	1.000



(a)

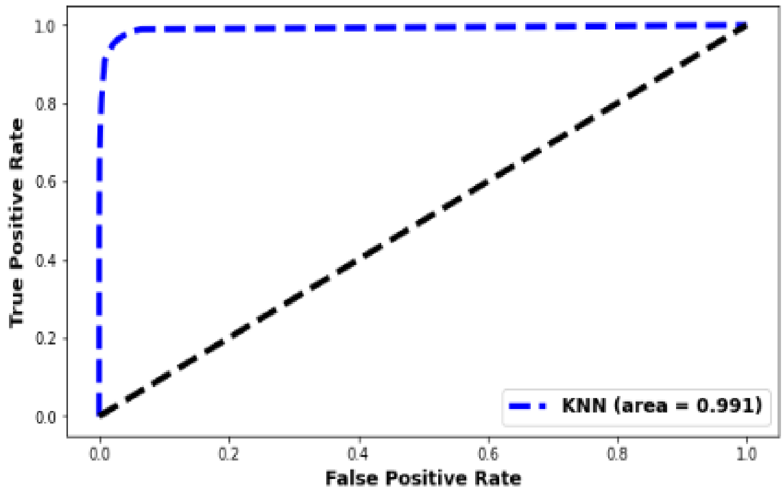


(b)

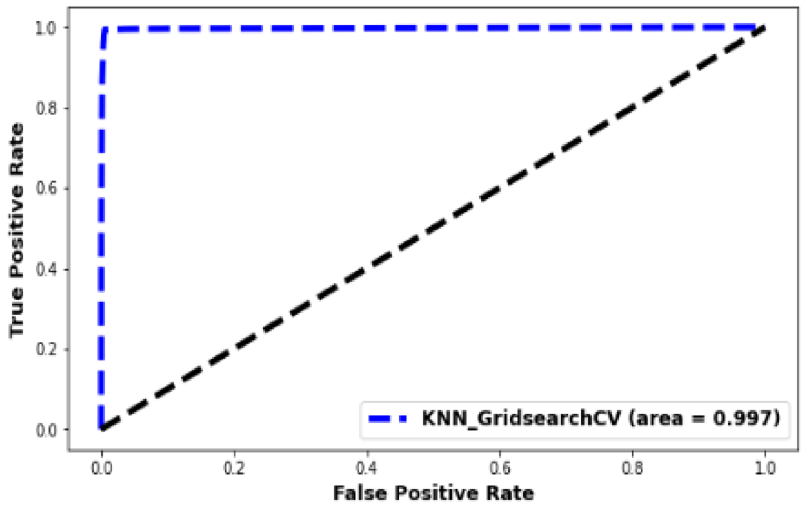


(c)

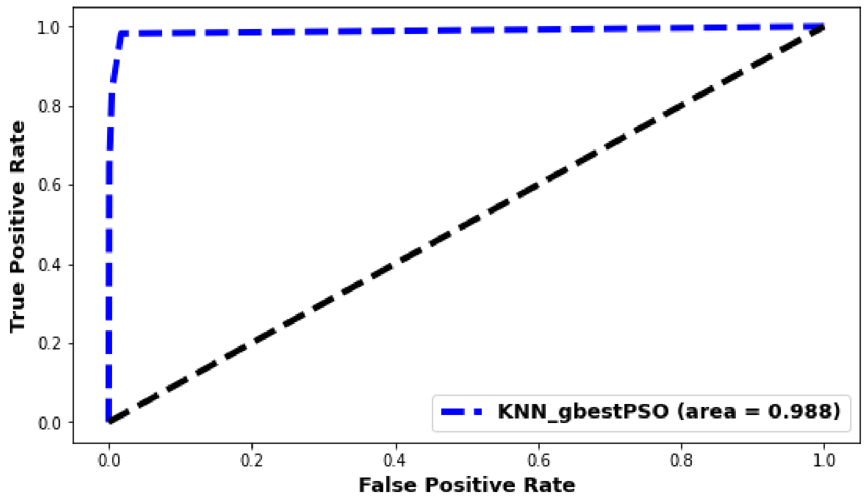
Figure 6. ROC curve of (a) Simple SVM, b) SVM tuned with Grid Search, c) SVM optimized with gbestPSO)**.**



(a)

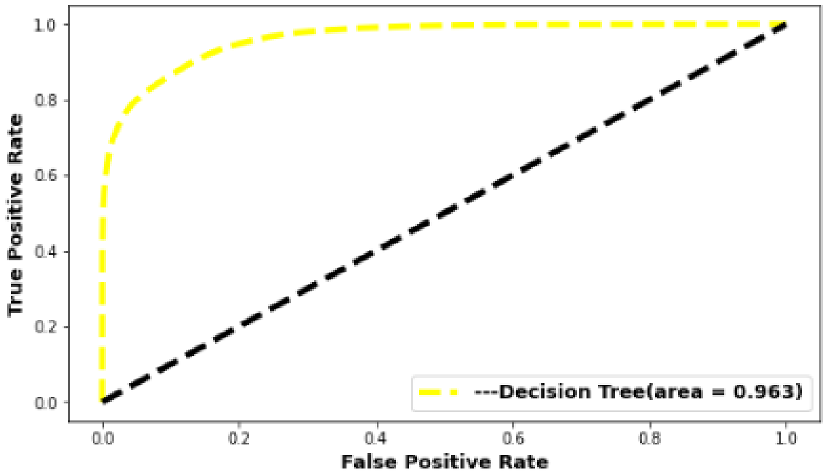


(b)

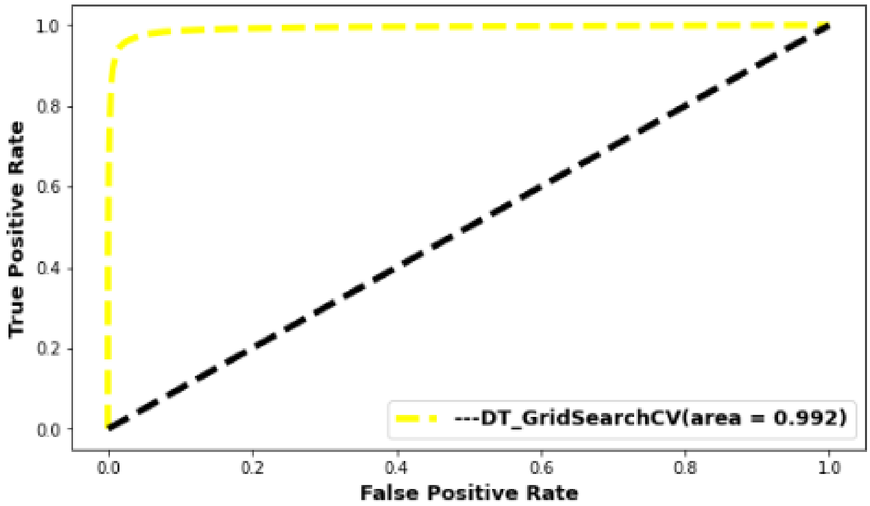


(c)

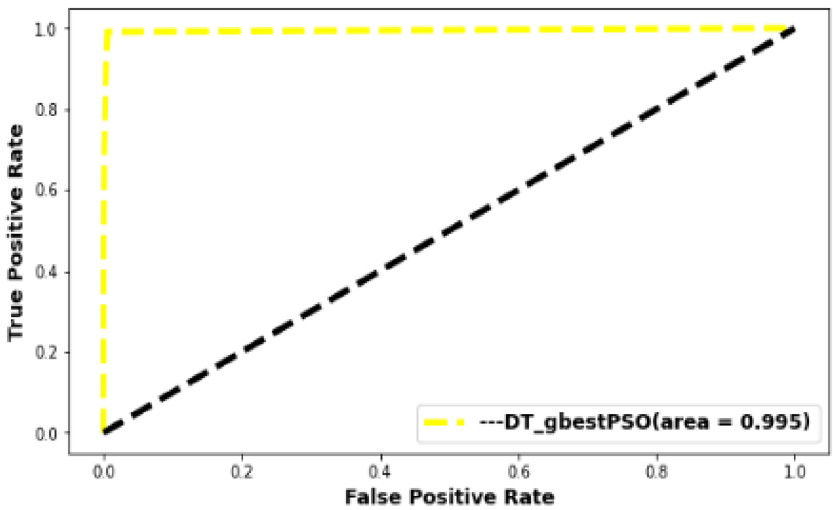
Figure 7. ROC curve of (a) Simple KNN, b) KNN tuned with Grid Search, c) KNN optimized with gbestPSO



(a)

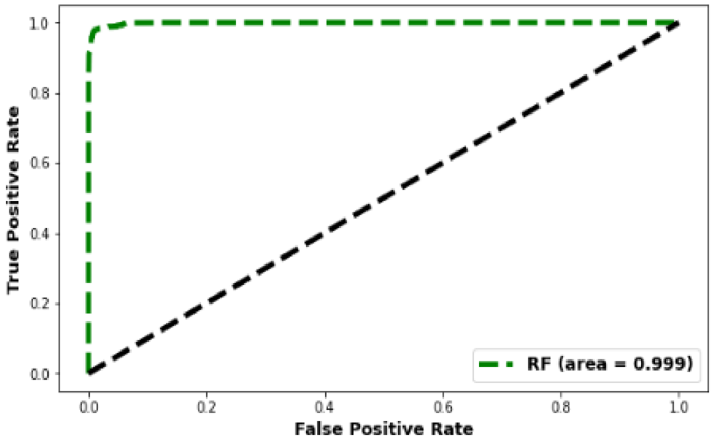


(b)

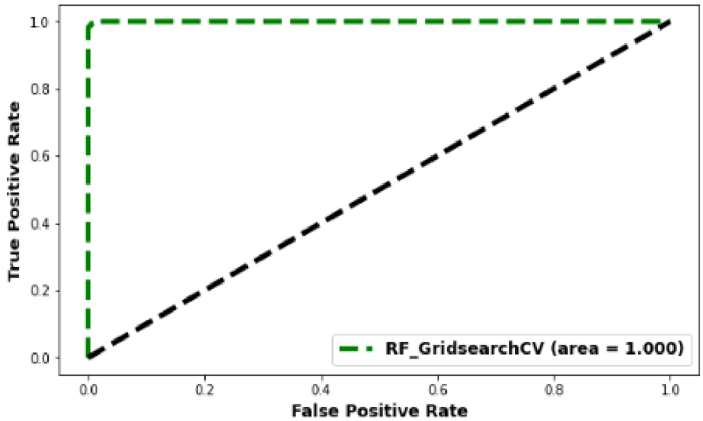


(c)

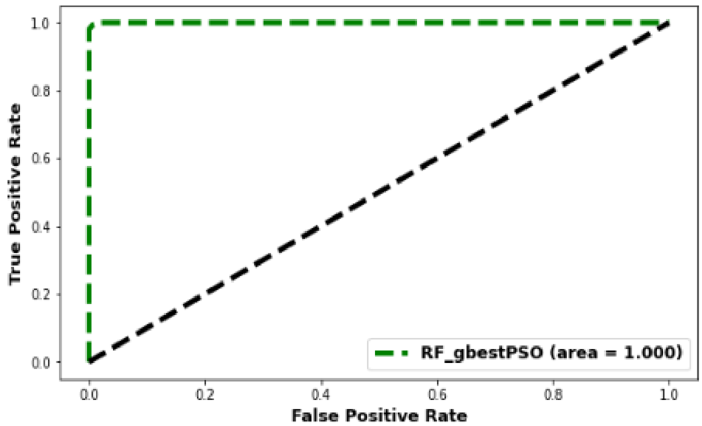
Figure 8. ROC curve of (a) simple DT, b) DT tuned with grid search, c) DT optimized with gbestPSO



(a)



(b)



(c)

Figure 9. ROC curve of (a) simple RF, b) RF tuned with Grid Search, c) RF optimized with gbestPSO

The simple SVM illustrated in Figure 6(a) has a relatively smaller area under the curve (AUC), meaning that it has limited discrimination ability using default hyperparameters. Following the tuning of Figure 6(b) using the Grid Search, the ROC curve moves towards the upper-left corner which indicates better sensitivity-specificity trade off. Figure 6(c) shows that the gbest-PSO-optimized SVM has the best AUC. The Figure 7 shows ROC of KNN classifier. Moderate classification performance can be observed in the baseline KNN of Figure 7(a). The ROC characteristics are enhanced through grid Search tuning of Figure 7(b) which chooses the number of neighbors and distance metrics as well. The results of applying

KNN to the optimization via gbest-PSO in Figure 7(c) further improve the ROC. In Figure 8(a) the simple DT model shows relatively poorer performance. The classifier discriminative strength is enhanced in Figure 8(b) through a grid search approach which optimizes the depth of the tree, as well as the split parameters. Figure 8(c) shows that the gbest-PSO optimized DT gives the highest ROC performance compared to the other two. The comparison of ROC curves of the Random Forest (RF) classifier is presented in Figure 9. The default settings used in the baseline RF in Figure 9(a) offer a decent level of discrimination but fail to use the full potential of the model. Figure 9(b) with optimization of the grid search maximizes the ROC curve and maximizes the AUC through optimization of ensemble parameters. The RF optimized with gbest-PSO in Figure 9(c) has the best AUC and the nearest to the top-left corner which proves the efficiency of PSO in order to maximize the classification ability of the ensemble-based models.

Table 7 shows the parameters used for training in all the cases. All models were first trained using default parameter settings and subsequently tuned using GridSearchCV and GlobalBestPSO. Parameter ranges were selected based on prior literature and empirical validation to ensure proper training within the computational availability.

Table 7. Initialization parameter of classification models and globalbestPSO

Model	Parameter Name	Description	Initial Value	Optimized / Search Range
SVM	C	Regularization parameter	1.0	{0.1, 1, 10, 100, 1000}
	γ (Gamma)	Kernel coefficient	scale	{1, 0.1, 0.01, 0.001, 0.0001}
KNN	n_neighbors	No. of nearest neighbors	5	{5, 7, 9, 11}
Decision Tree	Criterion	Split quality measure	Gini	{Gini, Entropy}
	Max Depth	Maximum tree depth	None	{4, 5, 6, 7, 8, 9, 10, 11, 12, 15}
Random Forest	n_estimators	Number of trees	10	{2, 4, 6, 8}
	Max Depth	Maximum tree depth	2	{3, 5, 7, 9}
	Random State	Seed for reproducibility	0	{10, 20}
GlobalBestPSO	Swarm Size	Number of particles	20	Fixed
	Max Iterations	Optimization iterations	50	Fixed

DISCUSSIONS

The experimental findings reveal that the combination of GlobalBest Particle Swarm Optimization with conventional machine learning classifiers contributes to a high level of multiclass customer classification in banks. In all the models tested, the GlobalBestPSO-optimized versions always performed better in accuracy, in terms of F1-score and the values of the ROC-AUC and also significantly lowering the rates of false positives.

Such results demonstrate the usefulness of swarm intelligence in finding the way through complicated hyperparameter space and enhancing model generalization on large, non-linear datasets. The other key observation is that the optimized models are robust in all the five classes of customers. In contrast to most of the existing literature, which deals with binary or even constrained multiclass contexts, the paper proves that it is indeed possible to have fine-grained multiclass segmentation when highly sophisticated optimization methods are utilized. This observation is an invitation to the future banking analytics models to go beyond the crude classification models and use more detailed customer classification models to aid in strategic planning and interventions.

In the view of classifiers, the findings indicate that the swarm-based optimization is of significant benefit to the ensemble-based learning models in the form of Random Forest, enabling the models to achieve close-to-perfect classification in all categories of customers. In the same vein, the trained SVM with the best margin separation across the five categories of customers shows that PSO is very useful in the process of adjusting the kernel related parameters. The fact that decision Tree models that are normally vulnerable to overfitting and unstable are very robust when optimized with GlobalBestPSO undergoes testament to the fact that intelligent setting of hyperparameters can help address structural constraints of simpler models. The optimized KNN classifier also exhibits better neighborhood selection and distance weighting which results in better class discrimination and low sensitivity to noise and outliers.

CONCLUSION

In order to categories customers in the banking data, firstly the research studies the performance of four machine learning classifiers and then tuned the hyperparameter using gridsearchCV and applied the advanced swarm optimization methodology gbestPSO. The simple classifiers result the accuracy measures to 0.81 to 0.96, and while tuning the parameters with traditional gridsearchCV the subtle variations are perceived in the four classifiers. However, the optimized technique Global Best PSO showed optimum advances in the accuracy metrics approximately greater than 0.97 for four models. This states that all the classifiers performed well to figure out the optimum solution, while using the swarm intelligence technique gbestPSO. Similarly, the gbestPSO with the classifiers achieved the highest precision, recall, f-score and tpr rate while comparing to gridsearchCV. Moreover, the gbestPSO exhibited exceptional results by significantly reducing the false positive rate (FPR) rate for the four classifiers. These outcomes proved that Global Best PSO efficacy in enhancing model's performance and point to its potential for use in a different domain. Future research should concentrate on classifier tuning exclusively in various datasets to achieve performance and also focus on ensemble learning classification algorithms with improved tuning strategies to overcome the challenge.

REFERENCES

[1] Veeralagan J, Manju Priya S. Hyper tuning using GridSearchCV on machine learning models for prognosticating dementia.2022 Dec 8. doi:10.21203/rs.3.rs-2316713/v1

[2] Etaiwi W, Biltawi M, Naymat G. Evaluation of classification algorithms for banking customer's behavior under Apache Spark Data Processing System. Procedia computer science. 2017 Jan 1; 113:559-64. https://doi.org/10.1016/j.procs.2017.08.280

[3] Sarker IH. Machine learning: Algorithms, real-world applications and research directions. SN computer science. 2021 May;2(3):160. https://doi.org/10.1007/s42979-021-00592-x

[4] Smeureanu I, Ruxanda G, Badea LM. Customer segmentation in private banking sector using machine learning techniques. Journal of Business Economics and Management. 2013 Nov 1;14(5):923-39. https://doi.org/10.3846/16111699.2012.749807

[5] Zeinulla E, Bekbayeva K, Yazici A. Comparative study of the classification models for prediction of bank telemarketing. In2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT) 2018 Oct 17 (pp. 1-5). IEEE. https://doi.org/10.1109/ICAICT.2018.8747086

[6] Dawood EA, Elfakhrany E, Maghraby FA. Improve profiling bank customer's behavior using machine learning. Ieee Access. 2019 Aug 12; 7:109320-7 https://doi.org/10.1109/ACCESS.2019.2934644

[7] Charbuty B, Abdulazeez A. Classification Based on Decision Tree Algorithm for Machine Learning. JASTT. 2021 Mar. 24 [cited 2026 Feb. 10];2(01):20-8. https://doi.org/10.38094/jastt20165

[8] Gupta G. A self-explanatory review of decision tree classifiers. InInternational conference on recent advances and innovations in engineering (ICRAIE-2014) 2014 May 9 (pp. 1-7). IEEE. https://doi.org/10.1109/ICRAIE.2014.6909245

[9] Patel HH, Prajapati P. Study and analysis of decision tree-based classification algorithms. International Journal of Computer Sciences and Engineering. 2018 Oct 31;6(10):74-8.

[10] Chicho BT, Abdulazeez AM, Zeebaree DQ, Zebari DA. Machine learning classifiers-based classification for IRIS recognition. Qubahan Academic Journal. 2021 May 4;1(2):106-18. https://doi.org/10.48161/qaj.v1n2a48

[11] Khorshid SF, Abdulazeez AM. Breast cancer diagnosis based on k-nearest neighbors: a review. PalArch's Journal of Archaeology of Egypt/Egyptology. 2021 Feb;18(4):1927-51.

[12] Zebari DA, Zeebaree DQ, Abdulazeez AM, Haron H, Hamed HN. Improved threshold based and trainable fully automated segmentation for breast cancer boundary and pectoral muscle in mammogram images. Ieee Access. 2020 Nov 5;8:203097-116. https://doi.org/10.1109/ACCESS.2020.3036072

[13] Ahmed NS, Sadiq MH. Clarify of the random forest algorithm in an educational field. In2018 international conference on advanced science and engineering (ICOASE) 2018 Oct 9 (pp. 179-184). IEEE. https://doi.org/10.1109/ICOASE.2018.8548804

[14] Priyanka, Kumar D. Decision tree classifier: a detailed survey. International Journal of Information and Decision Sciences. 2020;12(3):246-69. https://doi.org/10.1504/IJIDS.2020.108141

[15] Pandey AK, Singh P. A systematic survey of classification algorithms for cancer detection. Int. J. Data Informatics Intell. Comput. 2022 Dec 21;1(2):34-50. https://doi.org/10.5281/zenodo.7464708
[16] Wang S, Lu H, Khan A, Hajati F, Khushi M, Uddin S. A machine learning software tool for multiclass classification. Software Impacts. 2022 Aug 1; 13:100383. https://doi.org/10.1016/j.simpa.2022.100383
[17] Belete DM, Huchaiah MD. Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. International Journal of Computers and Applications. 2022 Sep 2;44(9):875-86. https://doi.org/10.1080/1206212X.2021.1974663
[18] Religia YR, Pranoto GT, Suwancita IM. Analysis of the use of particle swarm optimization on naïve bayes for classification of credit bank applications. JISA (Jurnal Informatika dan Sains). 2021 Dec 26;4(2):133-7. https://doi.org/10.31326/jisa.v4i2.946
[19] Chopard B, Tomassini M. Particle swarm optimization. InAn introduction to metaheuristics for optimization 2018 Nov 3 (pp. 97-102). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-93073-2_6
[20] Vardhini KK, Sitamahalakshmi T. A review on nature-based swarm intelligence optimization techniques and its current research directions. Indian Journal of Science and Technology. 2016 Mar 16;9(10):1-3. https://doi.org/10.17485/IJST/2016/V9I10/81634
[21] Li J, Ding L, Li B. A novel naive bayes classification algorithm based on particle swarm optimization. The Open Automation and Control Systems Journal. 2014 Dec;6(1):747-53. https://doi.org/10.2174/1874444301406010747
[22] Lamba A, Kumar D. Survey on KNN and its variants. Int. J. Adv. Res. Comput. Commun. Eng. 2016 May;5(5):430-5.
[23] Ibrahim I, Abdulazeez A. The role of machine learning algorithms for diagnosing diseases. Journal of Applied Science and Technology Trends. 2021 Mar 19;2(01):10-9. https://doi.org/10.38094/jastt20179
[24] Ray S. A quick review of machine learning algorithms. In2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) 2019 Feb 14 (pp. 35-39). IEEE. https://doi.org/10.1109/COMITCon.2019.8862451
[25] Almasi ON, Rouhani M. Fast and de-noise support vector machine training method based on fuzzy clustering method for large real-world datasets. Turkish Journal of Electrical Engineering and Computer Sciences. 2016;24(1):219-33. https://doi.org/10.3906/elk-1304-139
[26] Cheushev V, Simovici DA, Shmerko V, Yanushkevich S. Functional entropy and decision trees. InProceedings. 1998 28th IEEE International Symposium on Multiple-Valued Logic (Cat. No. 98CB36138) 1998 May 29 (pp. 257-262). IEEE. https://doi.org/10.1109/ISMVL.1998.679467
[27] Molala R. Entropy, Information Gain, Gini Index—The Crux of a Decision Tree. Medium. 2020 Mar.
[28] Sarica A, Cerasa A, Quattrone A. Random Forest algorithm for the classification of neuroimaging data in Alzheimer's disease: a systematic review. Frontiers in aging neuroscience. 2017 Oct 6;9:329. https://doi.org/10.3389/fnagi.2017.00329
[29] DeCastro-García N, Munoz Castaneda AL, Escudero Garcia D, Carriegos MV. Effect of the sampling of a dataset in the hyperparameter optimization phase over the efficiency of a machine learning algorithm. Complexity. 2019;2019(1):6278908. https://doi.org/10.1155/2019/6278908
[30] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. InMHS'95. Proceedings of the sixth international symposium on micro machine and human science 1995 Oct 4 (pp. 39-43). Ieee. https://doi.org/10.1109/MHS.1995.494215
[31] Kennedy J, Eberhart R. Particle swarm optimization. InProceedings of ICNN'95-international conference on neural networks 1995 Nov 27 (Vol. 4, pp. 1942-1948). ieee. https://doi.org/10.1109/ICNN.1995.488968