Brain Stroke Prediction: A PSO Optimized Stacked Ensemble Machine Learning Approach with SMOTEEN Data Balancing

Achin Jain[1], Arun Kumar Dubey[1], Meenakshi Gupta[2], Sarita Yadav[1], Arvind Panwar[3], Roger Atanga[5], Bernardo Lemos[4,5,*], Saurav Mallik[4,5,*]

[1]Department of Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, achin.mails@gmail.com, arudubey@gmail.com, sarita1320@yahoo.co.in

[2]Department Of Electronics And Communications, Manav Rachna University, Sector-43, Aravali Hills, Surajkund Road, Faridabad, Haryana-121001, meenakshigupta@mru.edu.in

[3]Galgotias Multi-Disciplinary Research Development Cell(G-MRDC)), Galgotias University, Greater Noida-201308,UP(India), arvind.nice3@gmail.com

[4]Department of Environmental Health, Harvard T H Chan School of Public Health, Boston, MA, 02115, USA; blemos@hsph.harvard.edu, sauravmtech2@gmail.com, smallik@hsph.harvard.edu

[5]Department of Pharmacology & Toxicology, University of Arizona, Tucson, AZ, 85721, USA; ratanga@arizona.edu, blemos@pharmacy.arizona.edu, smallik@arizona.edu

*Correspondence: Saurav Mallik (sauravmtech2@gmail.com, smallik@hsph.harvard.edu, smallik@arizona.edu), Bernardo Lemos (blemos@hsph.harvard.edu, blemos@pharmacy.arizona.edu)

***Abstract*—Prediction of stroke is a critical challenge in healthcare, where early intervention can significantly reduce risks and improve patient outcomes. Traditional methods often struggle with imbalanced datasets and low prediction accuracy. This paper proposes a novel approach to address these issues by combining PSO-optimized Stacked Ensemble ML Classifiers with SMOTEEN data balancing to predict strokes effectively. The technique optimizes 3 Baseline ML Classifiers(RF, SVM, XGB) using Particle Swarm Optimization (PSO) and Stacked Ensemble approach is applied that significantly improving prediction performance. The proposed methodology achieved an impressive 95.57% accuracy, outperforming conventional models such as SVM, LR, RF, KNN, XGB. Overall, the PSO-optimized stacked ensemble approach outperforms traditional machine learning techniques in stroke prediction, providing a more accurate and robust model for early detection and intervention.**

***Index Terms*—Stroke prediction, Machine learning, Stacking ensemble, Class Imbalance, SMOTEEN, PSO Optimization.**

I. Introduction

Stroke is a serious medical condition and a leading cause of death worldwide, occurring when blood flow to the brain is disrupted. According to the World Stroke Organization, one in four people over 25 is at risk of experiencing a stroke. The widespread impact of the condition underscores the need for accurate and timely prediction methods to improve patient outcomes. As the world population has grown, there has been an alarming rise in brain strokes, which will result in a significant rise in annual deaths by 2023. The need to solve this situation has become more urgent as the number of stroke-related deaths continues to increase. Stroke research is now at the forefront of medical research due to this concerning trend.

By examining large data sets that include demographic data, medical history, and physiological markers such as age, blood pressure, and glucose levels, machine learning algorithms have shown promise in transforming the prediction of strokes [1], [2]. However, there are issues that need to be resolved when implementing these algorithms in clinical contexts. Potential bias in training data is an alarming concern, as it can lead to biased predictions and unequal healthcare outcomes [3]. Disparities in healthcare access, socioeconomic considerations, and insufficient or unrepresentative datasets can all lead to biases. Ensemble learning techniques such as stacking have become a reliable way to address these issues. By integrating several classifiers, stacking might enhance the predictive system's overall generalization capacity and lessen the effects of biases present in individual models.

Furthermore, principal component analysis (PCA) is a powerful dimensionality reduction method. It helps simplify data representations by linearly converting the original features into orthogonal features called principal components. These are arranged according to the variance. This allows PCA to reduce complex datasets into a lower-dimensional space. In this process, it preserves important features.PCA determines the directions of highest variance and their corresponding magnitudes by identifying eigenvectors and eigenvalues from the data covariance matrix [4]- [6]. PCA is used in many fields, such as feature extraction, noise reduction, and data visualization [7]. The suggested research has achieved remarkable predictive accuracy, reaching an astounding 95.57%, through a groundbreaking approach for predictive analysis in ischemic brain stroke using sophisticated machine learning techniques, such as various ML algorithms and ensemble learning procedures.

Because it provides a means of improving prediction accuracy by combining several classifiers, ensemble learning has gained attention in the domains of machine learning and computational intelligence. Ensemble approaches were first developed to increase classification accuracy, but have since been developed to address a variety of real-world prob-

lems, including adjusting to shifting concepts, fixing mistakes, choosing the most pertinent characteristics, gradually learning and assessing confidence levels. Significant progress has been made in recent years as a result of the in-depth exploration by researchers of different fusion strategies and the elements that comprise the ensembles [8]- [11]. Stroke prediction is a complex field that requires careful consideration of challenges such as accuracy, missing data, data imbalance, and interoperability. However, A novel approach can utilize the potential of machine learning. The main highlights of proposed worked are here.

A. Key Contributions

1) To Conduct a background study and provide a review of the literature on brain stroke detection, highlighting the existing techniques used for the assessment of stroke risk.
2) To propose a novel methodology **PSO Optimized Stacked Ensemble ML Classifier using SMOTEEN Data Balancing** to address the problem of data imbalance in stroke prediction. The novelty aspect of this technique lies in using Particle Swarm Optimization (PSO) to optimize 3 Baseline ML Classifiers(RF, XGB, SVM) and then applying the stacking ensemble model for data balancing.
3) To test and validate the proposed methodology using various performance metrics, such as precision, precision (p), recall (r), F1 score, AUC, and AUCPR, ensuring a comprehensive evaluation and robustness of the model.
4) To compare the performance of the proposed methodology against baseline machine learning classifiers, as well as other optimization techniques like Bayesian Optimization (BO), Differential Evolution (DE), and Ant Colony Optimization (ACO), to demonstrate the improvements and benefits of the proposed approach.

The remainder of the paper is organized as follows. Section II reviews the Related Work on stroke prediction models. Section III describes the materials and methods, including the data set, pre-processing, data balancing using SMOTEEN, and optimization with PSO. In Section IV, we present the Proposed System Architecture. Section V outlines the Experimental Setup and evaluation metrics. Section VI discusses the results and discussions, comparing the proposed model with existing methods. Finally, Section VII concludes the paper and suggests future work.

II. Related Work

The application of machine learning (ML) techniques has significantly advanced stroke prediction research, leading to improved accuracy and timely interventions. This literature review synthesizes recent studies on ML-based stroke prediction, focusing on algorithmic performance, feature selection methods, model interpretability, and key outcomes. In [12], a novel stroke detection algorithm integrates classifiers such as Naïve Bayes, logistic regression, XGBoost, and support vector machines (SVM). The SVM algorithm demonstrated superior performance, achieving 98.6% accuracy and 99.9% precision. However, the study lacked detailed discussions on feature selection and data pre-processing techniques. [13] explores an algorithm based on ML that uses hospital presentation data and social determinants of health (SDoH). Inclusion of SDoH variables markedly improved sensitivity, specificity, and AUC scores, increasing from 0.694 to 0.823. This study highlights the consistent outperformance of ML classifiers over logistic regression.

The authors in [14] utilized various datasets and machine learning models, including Naive Bayes, SVM, Decision Tree, Random Forest, and Logistic Regression. This study introduces the minimal genetic folding model (MGF), which achieves 83. 2% precision and outperforms other models in AUC scores. Although the MGF model shows promise in distinguishing stroke recovery stages, more research is needed. A potential limitation is the oversampling method, which may have impacted the classifier's performance.

In [15] , logistic regression with recursive feature elimination (RFE) was applied to predict stroke and transient ischemic attack (TIA), focusing on the symptoms reported by the patient. The models achieved an AUC greater than 0.94 and performance was further enhanced by incorporating follow-up data. [16] highlights the efficacy of stack classification, combining base classifiers such as Nave Bayes and random forests with a logistic regression metaclassifier. This method achieved an AUC of 98.9% and 98% accuracy, demonstrating its robustness across multiple performance metrics. [17] examines the interpretability of the model for stroke prediction using SHAP and LIME. Random Forest emerged as the model with the best performance with 90. 36% accuracy, closely followed by the XGB classifier at 89.02%. [18] applies ML to predict ischemic stroke in emergency settings, focusing on clinical laboratory features. The XGBoost-based model stood out, demonstrating the highest accuracy for ischemic stroke prediction and consistent validation across multiple datasets. The high sensitivity and specificity of the model further affirm its reliability in prescreening patients.

The authors in [20] propose a stroke prediction strategy using logistic regression, achieving 86% precision through techniques such as SMOTE, feature selection, and outlier control. Factors such as blood pressure, age, glucose levels, and previous stroke history are analyzed. Although logistic regression outperforms other models, Naive Bayes achieves 82% accuracy in related studies. The research highlights the potential of machine learning for early stroke detection and prevention. In [21] the proposed model achieves 94% accuracy in stroke prediction, outperforming algorithms like Naive Bayes, Logistic Regression, SVM, and Decision Tree. The authors also used Ensembled Naive Bayes and Ensembled Decision Tree. The study contributes to developing an integrated learning model and refining the fixed structure of the algorithm. The authors in [22] evaluate ten machine learning models for stroke prediction, including Naive Bayes variants, Gradient Boosting, KNN, SVM, Decision Tree, Random Forest, Logistic Regression, and MLP. The study highlights the importance of early stroke diagnosis, and the KNN algorithm reaches the highest accuracy of 94%.

In [23] , deep learning models are used to predict major

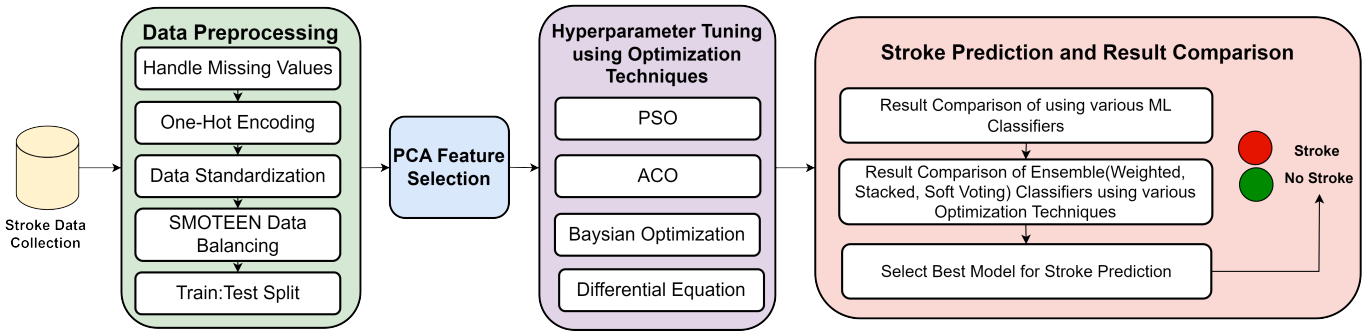

Fig. 1. Stroke Prediction Workflow

adverse cerebrovascular events (MACE) after acute ischemic stroke (AIS). Using clinical and imaging data, models like DeepSurv and Deep-Survival Machines surpassed traditional survival methods. The validation results, including AUC, sensitivity, specificity, and other metrics, showcased the reliability of the models for personalized patient outcome predictions, helping clinical decision-making. The reviewed studies, summarized in Table 1, highlight diverse ML approaches and their substantial contributions to stroke prediction. These advances emphasize the potential of ML to improve stroke risk assessment, enabling proactive interventions and improved patient outcomes.

III. Materials and Methods

In this study, the SMOTEEN [25] data balance technique is used to address the problem of data imbalance and PCA to select the relevant features. Five baseline ML classifiers (RF, KNN, SVM, XGB, and LR) are used for classification. In this paper four different optimization techniques are used for comparison, i.e. PSO, DE, BO, and ACO optimization. To further enhance the prediction of stroke, we have used three different Optimized Ensemble Approaches(Weighted, Soft Voting, and Stacked Ensemble) to find the most efficient solution. Figure 1 represents the stroke detection workflow diagram. and Algorithm 1 presents the data pre-processing and PCA-based feature selection.

A. Dataset

In this paper, we have used the Kaggle data set [24] that includes 5,110 patients and includes variables such as sex, age, hypertension status, history of heart disease, marital status, type of occupation, type of rehabilitation, average glucose level, body mass index (BMI), smoking behavior, and incidence of stroke. The main variable of interest is the prediction of the stroke, represented as a binary result (1 indicating the stroke, 0 indicating that there is no stroke). In this paper, SMOTEEN Data Balancing is used to address data imbalances to improve the accuracy of the model. Most of the samples obtained through various sampling methods were inadequate to provide adequate training data because class 0 had fewer instances, resulting in inaccurate results. Data preprocessing involves removing redundant values, selecting key features, and analyzing the data. A comparison of the classes and data points is illustrated in Figure 2 (classes before SMOTEEN) and Figure 3 (classes after SMOTEEN).

Fig. 2. Class Distribution before Data Balancing

Fig. 3. Class Distribution after SMOTEEN Data Balancing

Algorithm 1 Data Preprocessing with SMOTEEN Balanching and PCA Feature Selection

1: **Input:** Raw dataset $\mathbf{D}$
2: **Output:** Processed dataset $\mathbf{D}'$
3: **Step 1:** Load the dataset $\mathbf{D}$ and Convert categorical columns to numerical using one-hot encoding using below equation:

$$X_{\text{encoded}} = \text{pd.get\_dummies}(X_{\text{categorical}}) \tag{1}$$

Where $X_{\text{categorical}}$ is the set of categorical columns in the dataset.
4: **Step 3:** Impute missing values with the mean values of non-missing values for each column

$$x_{\text{imputed}} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{where } x_i \text{ are non-missing values in the column.} \tag{2}$$

5: **Step 4:** Apply SMOTEEN for balancing the dataset. Let X_{minority} be the minority class, and the synthetic samples $\mathbf{X}_{\text{synthetic}}$ are generated as:

$$X_{\text{synthetic}} = \alpha X_{\text{minority}} + (1-\alpha)X_{\text{neighbors}} \tag{3}$$

where α is a random coefficient between 0 and 1, and $X_{\text{neighbors}}$ are the nearest neighbors of the minority class samples.
6: **Step 5:** Standardize the features by subtracting the mean and dividing by the standard deviation:

$$X_{\text{standardized}} = \frac{X-\mu}{\sigma} \quad \text{where} \quad \mu = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i-\mu)^2} \tag{4}$$

where X is the dataset and μ, σ are the mean and standard deviation of each feature.
7: **Step 6:** Apply PCA to reduce the dimensionality of the data set. The transformation can be represented as:

$$X_{\text{PCA}} = XW \tag{5}$$

where W is the matrix of eigenvectors (principal components) obtained from the covariance matrix of X, and X_{PCA} is the transformed dataset with reduced dimensions.
8: **Return:** Processed dataset $\mathbf{D}' = X_{\text{PCA}}$
=0

B. PCA for Feature Selection

Principal Component Analysis (PCA) [26] is a method used for linear dimensionality reduction, converting data into a new coordinate system where the largest variances by any data projection are situated on the initial coordinates, known as principal components. The principal components are determined through eigenvalue decomposition of the data's covariance matrix. Given a data matrix X with n samples and p characteristics, the covariance matrix Σ is calculated as:

$$\Sigma = \frac{1}{n-1}X^T X \tag{6}$$

The next step is to perform the decomposition of the eigenvalues in the covariance matrix Σ, obtaining the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ and the corresponding eigenvectors $v_1, v_2, \ldots, v_p$, such that:

$$\Sigma v_i = \lambda_i v_i \tag{7}$$

where λ_i represents the eigenvalue and v_i the corresponding eigenvector. The eigenvectors (principal components) are sorted in descending order based on their eigenvalues, and the first k eigenvectors form the new basis for the reduced feature space.

The data is then projected onto the new basis by multiplying the original data X by the matrix of the top k eigenvectors:

$$X_{\text{new}} = XV_k \tag{8}$$

where V_k is the matrix of the top k eigenvectors, and X_{new} is the transformed data in the new k dimensional space. PCA is widely used to reduce the dimensionality of datasets while preserving as much variance as possible.

C. Classifier Ensemble

1) Stacked Ensemble Classifier(S_E): Stacking [27], or stacked generalization, is a hierarchical ensemble methodology that synthesizes multiple base learners into a meta-learner as shown in Figure 4. Formally, let the output of the base models n be represented as $Z = \{P_1, P_2, \ldots, P_n\}$. The metamodel f_{meta} learns a mapping $f_{\text{meta}}(Z) \rightarrow y$, where y is the target variable.

The training process for stacked ensembles typically involves:

- Training base models independently on the dataset (X, y).
- Generating predictions (or transformed feature representations) from these models on either a holdout validation set or via cross-validation.
- Training a meta-model using the outputs of base models as input features and the true labels y as targets.

Depending on the complexity of the problem and the variety of base learners, the meta-model can utilize either linear or

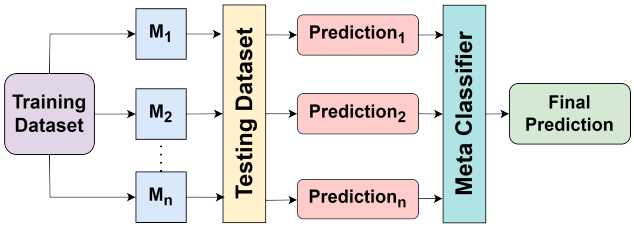

Fig. 4. Stacked Ensemble Classifier Workflow

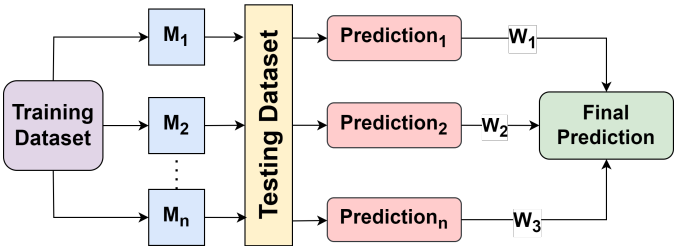

Fig. 5. Weighted Ensemble Classifier Workflow

nonlinear mappings, including methods like logistic regression, gradient boosting, or neural networks. Stacking naturally harnesses the unique advantages of each model, typically achieving better generalization results. The effectiveness of the ensemble is influenced by the diversity among base models as well as the learning capability of the meta-model.

2) Weighted Ensemble Classifier(W_E): The weighted ensemble method [28] is formulated to aggregate predictions from multiple base models by assigning a weight w_i to the output of each model based on its individual performance or importance as shown in Figure 5. The combined prediction P_{ensemble} is expressed mathematically as:

$$P_{\text{ensemble}} = \sum_{i=1}^{n} w_i P_i \tag{9}$$

where P_i denotes the output of the i-th base model, and w_i represents its corresponding weight such that

$$\sum_{i=1}^{n} w_i = 1 \tag{10}$$

This method successfully tackles model variance by highlighting the input of models that perform well in predictions, all while preserving diversity. Nonetheless, it requires a strong approach to assigning weights to avoid dependency on one specific model, thereby maintaining the ensemble's stability.

3) Soft Voting Ensemble Classifier(SV_E): The voting ensemble method [29] consolidates predictions from multiple base models using either majority voting (hard voting) or probability averaging (soft voting) as shown in Figure 6. soft voting averages the probability distributions predicted by the models using the following formula:

$$P_{\text{ensemble}}(y = c) = \frac{1}{n} \sum_{i=1}^{n} P_i(y = c) \tag{11}$$

where $P_i(y = c)$ denotes the predicted probability of class c by model i. Voting ensembles are based on the assumption

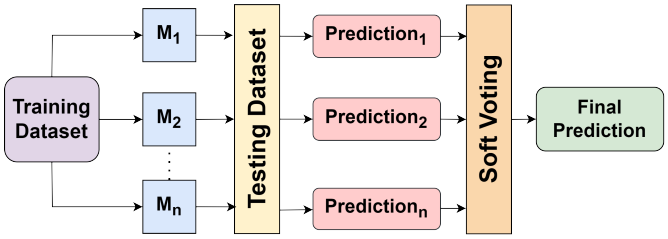

Fig. 6. Soft Voting Ensemble Classifier Workflow

that base models are diverse and uncorrelated, since aggregate prediction smooths out individual model errors. This method avoids model weighting, making it computationally efficient and easy to implement.

D. Optimization Techniques

1) Differential Equation(DE): Differential Evolution (DE) [30] is a strategy for optimization that is based on a population of potential solutions, which are refined through a cycle of mutation, crossover, and selection processes. During each cycle, a fresh candidate solution is produced by adding a scaled disparity between two population members with a third one. The fundamental mutation equation in DE is expressed as:

$$\mathbf{v}_i = \mathbf{x}_r 1 + F \cdot (\mathbf{x}_r 2 - \mathbf{x}_r 3) \tag{12}$$

where $\mathbf{v}_i$ is the mutated vector for the i-th individual, $\mathbf{x}_r 1, \mathbf{x}_r 2, \mathbf{x}_r 3$ are randomly selected population vectors, and F is a scaling factor. The crossover operation creates a trial vector $\mathbf{u}_i$ from the parent vector $\mathbf{x}_i$ and the mutated vector $\mathbf{v}_i$:

$$\mathbf{u}_i = \begin{cases} \mathbf{v}_i & \text{if rand}(j) \leq C_r \text{ or } j = j_{\text{rand}} \\ \mathbf{x}_i & \text{otherwise} \end{cases} \tag{13}$$

where C_r is the crossover rate, rand(j) is a random value between 0 and 1, and j_{rand} is a randomly selected index. The best candidate solution is selected based on the objective function evaluation, ensuring convergence towards an optimal solution.

2) Bayseian Optimization(BO): Bayesian Optimization (BO) [31] is a technique to optimize costly-to-evaluate objective functions using a probabilistic model. It creates a surrogate probabilistic model, often employing a Gaussian process (GP), to represent the objective function. An acquisition function is then utilized to direct the search towards the optimal solution. The GP model is described by:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{14}$$

where $f(x)$ is the objective function, $m(x)$ is the mean function and $k(x, x')$ is the covariance function (kernel). The acquisition function, which is used to decide the next evaluation point, is typically based on the expected improvement (EI). The expected improvement is given by:

$$\text{EI}(x) = \mathbb{E}\left[\max(f(x) - f(x^+), 0)\right] \tag{15}$$

where x^+ is the best observed point in the current, and $f(x)$ is the predicted value of the objective function at the point x. The next point to evaluate is chosen by maximizing the acquisition function.

$$x_{\text{next}} = \arg\max_x \text{EI}(x) \tag{16}$$

By iterating this process, BO efficiently finds the global optimum while minimizing the number of evaluations of objective functions.

3) Particle Swarm Optimization(PSO): Particle Swarm Optimization (PSO) [32] is a population-based optimization algorithm inspired by the collective behavior of bird flocks. In this approach, each particle symbolizes a possible solution, continuously refining its position by considering its own optimal past position as well as the optimal positions discovered by the swarm as a whole. The updates for position and velocity in PSO are defined as follows:

$$v_i^{t+1} = wv_i^t + c_1r_1(p_i^t - x_i^t) + c_2r_2(g^t - x_i^t) \tag{17}$$

where v_i^t is the velocity of the i-th particle at iteration t, w is the inertia weight, c_1 and c_2 are acceleration coefficients, r_1 and r_2 are random values between 0 and 1, p_i^t is the best personal position of the particle i -th particle, and g^t is the best global position of the swarm.

The position of the i-th particle is updated as:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{18}$$

where x_i^t is the position of the i-th particle at iteration t. The particles move through the solution space by adjusting their velocities based on their own best experiences and the best global experience found by the swarm. This process continues until convergence or a stopping criterion is met.

4) Ant Colony Optimization (ACO): ACO is a nature-inspired optimization algorithm based on the foraging behavior of ants [33]. The algorithm simulates how ants deposit pheromones on paths and adjust their movements to find the shortest path to a food source. ACO is used for discrete optimization problems, such as the traveling salesman problem. The core principle of ACO is to iteratively improve solutions by strengthening paths that lead to better solutions, using a pheromone updating rule.

The pheromone update formula is given by:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{19}$$

where: - $\tau_{ij}(t)$ is the pheromone level on edge (i, j) at time t, - ρ is the evaporation rate, - $\Delta\tau_{ij}(t)$ is the pheromone deposited by ants on edge (i, j).

The probability of an ant choosing the next node in the solution is given by:

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha[\eta_{ij}]^\beta}{\sum_{k\in\text{allowed}}[\tau_{ik}(t)]^\alpha[\eta_{ik}]^\beta} \tag{20}$$

where: - $\eta_{ij} = \frac{1}{d_{ij}}$ is the inverse of the distance between nodes i and j, - α and β control the influence of pheromone and distance, respectively.

E. Machine Learning Classifiers

1) Logistic Regression(LR): Logistic regression (LR) is a statistical method used for binary classification tasks. Models the probability of an outcome using a logistic function, ensuring that the output ranges between 0 and 1. The hypothesis for LR is given by:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{21}$$

where $h_\theta(x)$ represents the predicted probability, θ is the parameter vector and x is the input feature vector. The model is trained by minimizing the log-loss function, which measures the difference between predicted and actual labels, ensuring robust performance on classification tasks.

2) Random Forest: Random Forest (RF) is an ensemble learning method that is used primarily for classification and regression tasks. It constructs multiple decision trees during training and outputs the class label based on majority voting or averages the predictions for regression. The prediction of the Random Forest classifier can be expressed as:

$$\hat{y} = \text{majority\_vote}\left(T_1(x), T_2(x), \ldots, T_n(x)\right) \tag{22}$$

where $T_i(x)$ represents the prediction of the i-th decision tree for input x, and n is the total number of trees. By aggregating predictions, RF reduces overfitting and improves model generalization, making it effective for handling high-dimensional and imbalanced datasets.

3) K Nearest Neighbor: The K-Nearest Neighbors (KNN) algorithm is a simple and intuitive supervised learning method that is used for classification and regression tasks. It works by identifying the k closest data points to a query point in the feature space based on a chosen distance metric, such as the Euclidean distance. The Euclidean distance d between two points $\mathbf{x_1} = (x_{11}, x_{12}, \ldots, x_{1n})$ and $\mathbf{x_2} = (x_{21}, x_{22}, \ldots, x_{2n})$ is calculated as:

$$d(\mathbf{x_1}, \mathbf{x_2}) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2} \tag{23}$$

For classification, KNN assigns the query point to the class that is most frequent among its k-nearest neighbors. For regression, the output is typically the mean or median of the neighbors' values. KNN is non-parametric and requires no explicit training phase, making it straightforward to implement but computationally intensive for large datasets.

4) Extreme Gradient Boosting: Extreme Gradient Boosting(XGBoost) is an advanced implementation of gradient boosting, designed for speed and performance in both classification and regression tasks. It combines the predictions of an ensemble of weak learners, typically decision trees, by minimizing a differentiable loss function by using gradient descent. The model is built iteratively and each tree corrects the errors of the previous ones.

The objective function of XGBoost is given as:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{24}$$

where:

- $l(y_i, \hat{y}_i)$ is the loss function that measures the difference between the actual value y_i and the predicted value $\hat{y}_i$.
- $\Omega(f_k)$ is the regularization term to penalize the complexity of the k-th tree f_k, expressed as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{25}$$

where T is the number of leaves in the tree, γ is the penalty for adding a leaf, λ controls the regularization of the weights of the L_2 leaves w.

XGBoost employs the second-order Taylor expansion to approximate the loss function, improving optimization efficiency. Its ability to handle missing data, regularization, and parallel processing makes it highly effective for structured data tasks.

5) Support Vector Machine: Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression tasks. It seeks to find the optimal hyperplane that maximizes the margin between data points of different classes. The hyperplane is defined as:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{26}$$

where $\mathbf{w}$ is the weight vector, $\mathbf{x}$ is the feature vector and b is the bias. For a correctly classified point $(\mathbf{x}_i, y_i)$, the condition is:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \tag{27}$$

The objective of SVM is to minimize the norm $||\mathbf{w}||^2$ subject to the above constraint, which can be formulated as:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \tag{28}$$

For nonlinearly separable data, a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ is used to map the input data into a higher-dimensional space to find a separating hyperplane.

IV. Proposed System Architecture

The proposed system aims to predict stroke by applying the PCA feature reduction technique and the optimized ensemble classification approach. We have first applied the SMOTEEN Data Balancing Approach to balance the dataset and after that Five Baseline ML Classifiers are used. Then, to further enhance model performance, optimization techniques such as Ant Colony Optimization (ACO), Differential Equation(DE), Bayesian Optimization(BO), and Ant Colony Optimization(ACO) are applied to perform optimized ensemble methods (stacked, soft voting, and weighted average). The ensemble model, using optimized parameters, combines the predictions from baseline ML models to improve accuracy.Accuracy, precision, recall, F1 score, and AUC, with visualizations have been used for performance evaluation.ROC curves and confusion matrices are used to provide a comprehensive evaluation of the model's predictive capabilities. This approach ensures robust stroke prediction while addressing class imbalance, making the model more reliable and accurate. Figure 7 and Algorithm 2 show the complete proposed system architecture.

V. Experimental Setup

Stroke dataset is processed over proposed methodology. In this process, imbalance dataset problems are also addressed.In this work a highly imbalanced Kaggle brain stroke dataset with data preprocessing as explained in Algorithm 1. In this experiment, 80% of the data is used to train the ML classifiers, and the remaining 20% is used to test them, ensuring a robust evaluation. The empirical results of the baseline ML classifiers and optimized Ensemble ML classifiers on the SMOTEEN balanced data set have been represented using different visualization techniques based on the evaluation metrics considered, viz. Accuracy, precision, recall, F1 score, and AUC-ROC. We have implemented the experiments on Kaggle Platform with Python3, GPU Support and 30GB RAM. To evaluate the model performance, the following metrics are used, and in equations the following abbreviations are used: TP(True Positives), TN(True Negatives), FP(False Positives), FN(False Negatives).

1) **Accuracy(ACC)**:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{37}$$

2) **Precision(P)**:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{38}$$

3) **Recall(R)**:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{39}$$

4) **F1 Score**:

$$\text{F1 Score} = 2 \cdot \frac{\text{P} \cdot \text{R}}{\text{P} + \text{R}} \tag{40}$$

5) **ROC AUC (Receiver Operating Characteristic Area Under the Curve)**: This metric shows the performance of models based on specificity and sensitivity. The AUC (Area Under the Curve) value ranges from 0 to 1, where a value closer to 1 indicates a better-performing model.

VI. Results and Discussion

A. Results on Imbalanced Dataset

The table I summarizes the performance metrics of various machine learning models (LR, RF, KNN, SVM, and XGB) for two classes (0 and 1). Across all models, the metrics for Class 0 (No Stroke) are significantly better than those for Class 1 (Stroke), highlighting a severe data imbalance problem as the number of class 1 instances is significantly lower than class 0. For Class 0, all models achieve high precision, recall, and F1 scores, with accuracies ranging from 0.936 to 0.942. However, for Class 1, the metrics are notably poor, with precision as low as 0.25, recall close to 0, and F1-scores below 0.08 for most models. The Area Under the Precision-Recall Curve (AUCPR) values are also low, indicating that the models struggle to distinguish the minority class. This disparity underscores the challenge of imbalanced datasets, where models are biased toward the majority class, leading to poor performance in the minority class. Figures 8 and Figure 9 show the comparative ROC and the confusion matrix of the baseline ML classifications.

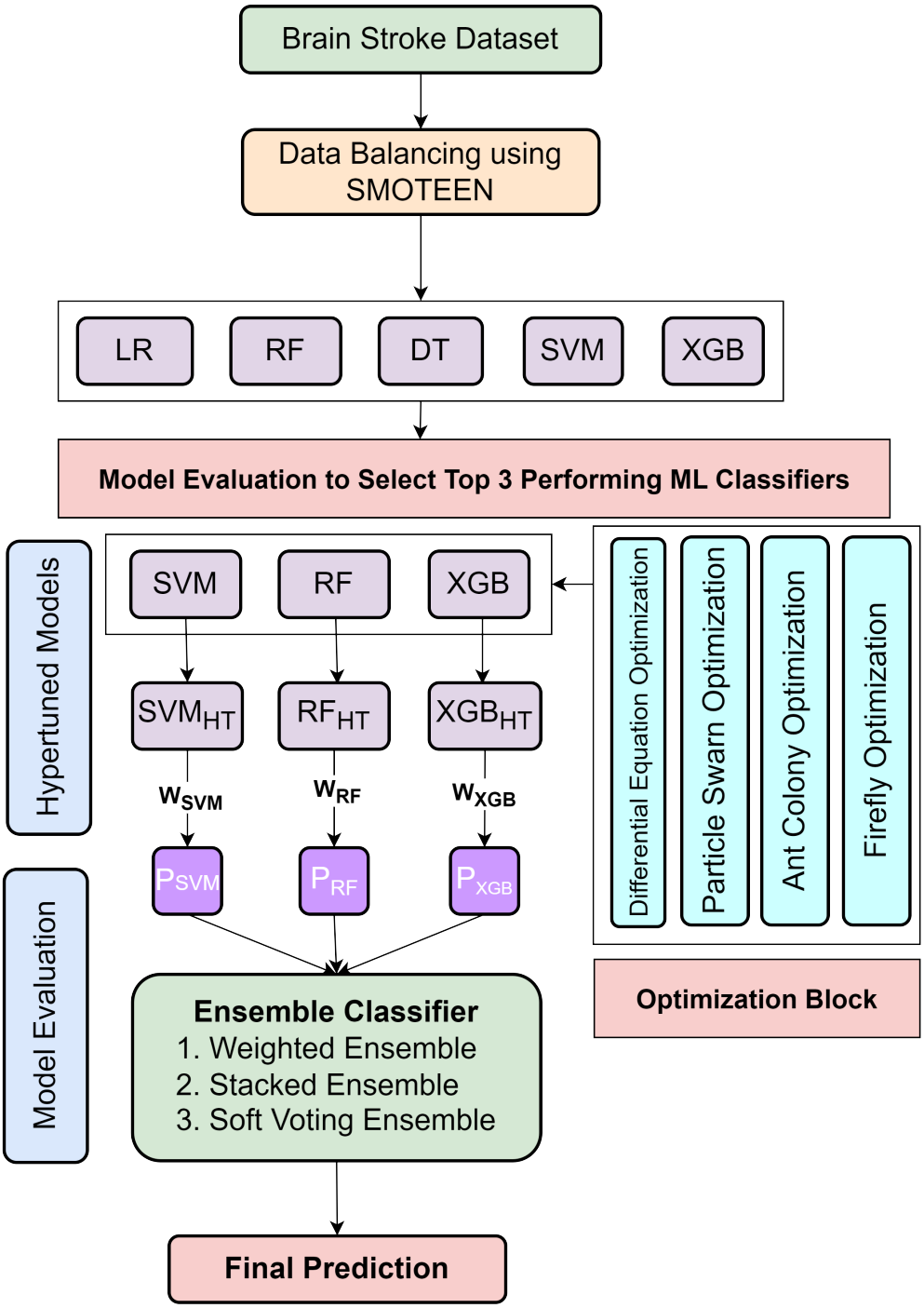

Fig. 7. Proposed System Architecture

B. Results on Balanced Dataset using SMOTEEN and PCA Feature Selection

The table II shows that all models perform well, achieving high precision, recall, and F1 scores for both the majority (Class 0) and minority (Class 1) classes. Random Forest achieves the highest accuracy of 94.4% and an AUCPR of 0.993, with a strong performance in handling the minority class, achieving a precision of 0.98, recall of 0.90, and an F1 score of 0.94. XGB and SVM also perform exceptionally well, with accuracies of 94.2% and 91.7%, respectively. Based on these results, the top three classifiers: Random Forest, XGBoost, and SVM were selected for further optimization using an ensemble approach. Figures 10 and Figure 11 show the comparative ROC and the confusion matrix of the baseline ML classifications.

C. Effect of Data Balancing

The results shown in Figure 12 highlight the significant improvement achieved by applying SMOTEEN to address

Algorithm 2 Proposed System Workflow Steps

Input: Preprocessed SMOTEEN balanced dataset D, baseline ML classifiers $M = \{\text{LR}, \text{RF}, \text{KNN}, \text{SVM}, \text{XGB}\}$, optimization algorithms $\mathcal{A} = \{\text{ACO}, \text{PSO}, \text{BO}, \text{DE}\}$

Output: Best ensemble model from top 3 classifiers (RF, XGB, SVM) with optimized hyperparameters.

Step 1: Load Dataset: Import dataset D with features X and target Y.

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}, \quad X \in \mathbb{R}^{m \times n}, \quad Y \in \mathbb{R}^m \tag{29}$$

Step 2: Train-Test Split: Split dataset (X, Y) into training and testing sets:

$$(X_{\text{train}}, Y_{\text{train}}), (X_{\text{test}}, Y_{\text{test}}), \quad \text{using an 80:20 ratio} \tag{30}$$

Step 3: Hyperparameter Tuning: For each classifier $m \in M$, perform hyperparameter tuning using the optimization algorithms $\mathcal{A}$.

For Random Forest (RF): Define the hyperparameters for RF as:

$$\theta_{\text{RF}} = \{\text{n\_estimators}, \text{max\_depth}, \text{min\_samples\_split}, \text{min\_samples\_leaf}\} \tag{31}$$

Use optimization algorithm $\mathcal{A}$ to find optimal hyperparameters:

$$\theta^*_{\text{RF}} = \arg\min_{\theta_{\text{RF}}} \mathcal{L}(m_{\text{RF}}(X_{\text{train}}, Y_{\text{train}}), X_{\text{test}}, Y_{\text{test}}) \tag{32}$$

For Support Vector Machine (SVM): Define the hyperparameters for SVM as:

$$\theta_{\text{SVM}} = \{\text{C}, \text{kernel}, \text{gamma}\} \tag{33}$$

Use optimization algorithm $\mathcal{A}$ to find optimal hyperparameters:

$$\theta^*_{\text{SVM}} = \arg\min_{\theta_{\text{SVM}}} \mathcal{L}(m_{\text{SVM}}(X_{\text{train}}, Y_{\text{train}}), X_{\text{test}}, Y_{\text{test}}) \tag{34}$$

For XGBoost (XGB): Define the hyperparameters for XGBoost as:

$$\theta_{\text{XGB}} = \{\text{learning\_rate}, \text{n\_estimators}, \text{max\_depth}\} \tag{35}$$

Use optimization algorithm $\mathcal{A}$ to find optimal hyperparameters:

$$\theta^*_{\text{XGB}} = \arg\min_{\theta_{\text{XGB}}} \mathcal{L}(m_{\text{XGB}}(X_{\text{train}}, Y_{\text{train}}), X_{\text{test}}, Y_{\text{test}}) \tag{36}$$

Step 4: Train Models: Train and evaluate each classifier with the optimized hyperparameters as mentioned above and select the top 3 peforming classifiers.

Step 7: Apply three ensemble methods: weighted, stacked, and soft voting to the top 3 models

1) Evaluate the performance of each ensemble method on the test set.
2) Select the ensemble method with the highest performance metric (e.g. accuracy, F1 score) as the final model.

Output: Best Hyperparameter Tuned Ensemble model from weighted, stacked, or soft voting =0

TABLE I
PERFORMANCE METRICS FOR DIFFERENT MODELS.

Model	Class	Accuracy	AUCPR	P	R	F1
LR	0	0.941	0.183515	0.942	0.997	0.969
	1			0.25	0.011	0.021
RF	0	0.942	0.161	0.941	1	0.970
	1			0	0	0
KNN	0	0.940	0.103	0.941	1	0.970
	1			0	0	0
SVM	0	0.940	0.078	0.941	1	0.970
	1			0	0	0
XGB	0	0.936	0.164	0.943	0.991	0.967
	1			0.25	0.044	0.076

TABLE II
PERFORMANCE METRICS FOR DIFFERENT MODELS.

Model	Class	Accuracy	AUCPR	P	R	F1
LR	0	0.836	0.879	0.86	0.78	0.82
	1			0.82	0.89	0.85
RF	0	0.944	0.993	0.98	0.90	0.94
	1			0.92	0.98	0.95
KNN	0	0.910	0.959	0.98	0.85	0.90
	1			0.88	0.98	0.93
SVM	0	0.917	0.980	0.94	0.88	0.91
	1			0.90	0.95	0.92
XGB	0	0.942	0.983	0.98	0.89	0.94
	1			0.91	0.98	0.95

the class imbalance in the dataset. For Class 1 (Stroke), the imbalanced dataset shows very poor performance, with mean precision, recall, and F1-score values of 0.1296, 0.0378, and 0.0476, respectively. However, after SMOTEEN application, these metrics improve substantially to 0.886, 0.956, and 0.920, respectively. This improvement is visually evident in the figure, where the balanced dataset using SMOTEEN outperforms the imbalanced dataset with significant scores. These findings demonstrate SMOTEEN's ability to effectively generate synthetic minority class samples while reducing noise, thereby

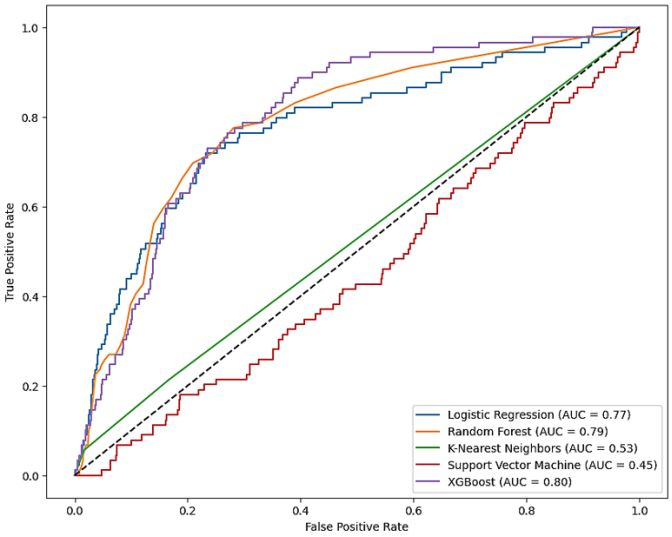

Fig. 8. ROC Comparison for Imbalanced Dataset

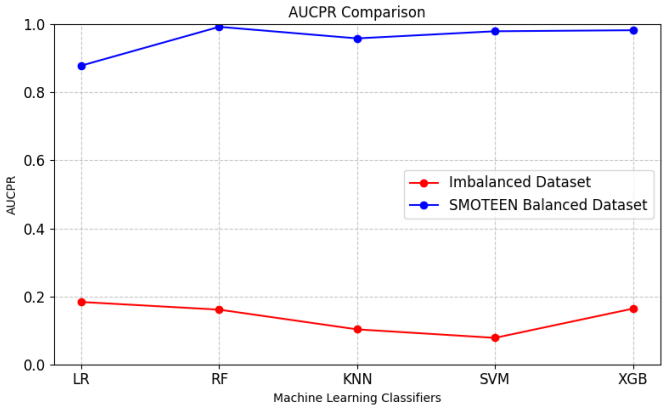

Fig. 13. AUCPR Comparison between Imbalanced and SMOTEEN Balanced Dataset

enhancing the classifier's ability to detect stroke cases and deliver a more balanced performance.

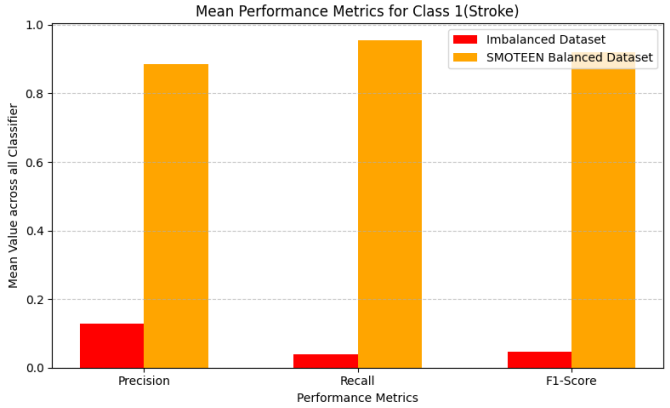

Fig. 12. Mean Performance Metric Comparison between Imbalanced and SMOTEEN Balanced Dataset

Figure 13 highlights the significant impact of data balance using the SMOTEEN technique on the AUCPR values of various models. In the imbalanced dataset, models show poor performance in handling the minority class, with lower AUCPR values, especially for Logistic Regression with an AUCPR of 0.183 and SVM with an AUCPR of 0.078. After applying SMOTEEN, the AUCPR values improve considerably for all models, with Random Forest achieving 0.993 and XGB reaching 0.983. This demonstrates that data balancing effectively reduces bias towards the majority class and enhances the ability of models to classify minority instances accurately.

D. Results on Proposed Model

Table III presents a comparison of the performance of various optimization techniques, including ACO, DE, BO, and PSO, applied to different models: RF, SVC, GB, W_E, S_E, SV_E. Among all the approaches, the PSO-S_E combination achieves the best results, with the highest accuracy of 0.956, precision of 0.956, recall of 0.956, F1 score of 0.992, and AUC of 0.993, highlighting its superior ability to balance classification performance and effectively handle imbalanced data. In contrast, the ACO-XGB combination performs the poorest, with the lowest accuracy of 0.897, precision of 0.878, and AUC of 0.895, indicating weak generalization and classification capabilities. Overall, PSO proves to be the most effective optimization technique, consistently delivering high performance across models, whereas ACO shows significant under performance, particularly with the GB model. The comparison of the confusion matrix for all 12 different optimized ensemble models is shown in Figure 14.

We have also verified our proposed(PSO Optimized Stacked Ensemble) Model performance against well established baseline ML Classifiers on SMOTEEN Balanced Dataset with PCA Feature selection technique. The results shown in Figure 15 showed that our proposed model consistently achieves the highest values, with an accuracy of 0.956, precision of 0.956, recall of 0.956 and F1-Score of 0.992, outperforming all other models. The proposed model achieves the highest F1 score of 0.992, significantly outperforming all other models. Compared to the second-best model, RF, which has an F1-Score of 0.945, the proposed model shows a notable improvement of 4.97%, demonstrating its superior ability to balance Precision and Recall. RF performs strongly, followed by KNN, SVM, and XGB, all of which achieve an F1 score of 0.915, indicating competitive but slightly lower performance. However, LR lags behind, with the lowest F1 score of 0.835, making it the weakest model in terms of balanced classification performance.

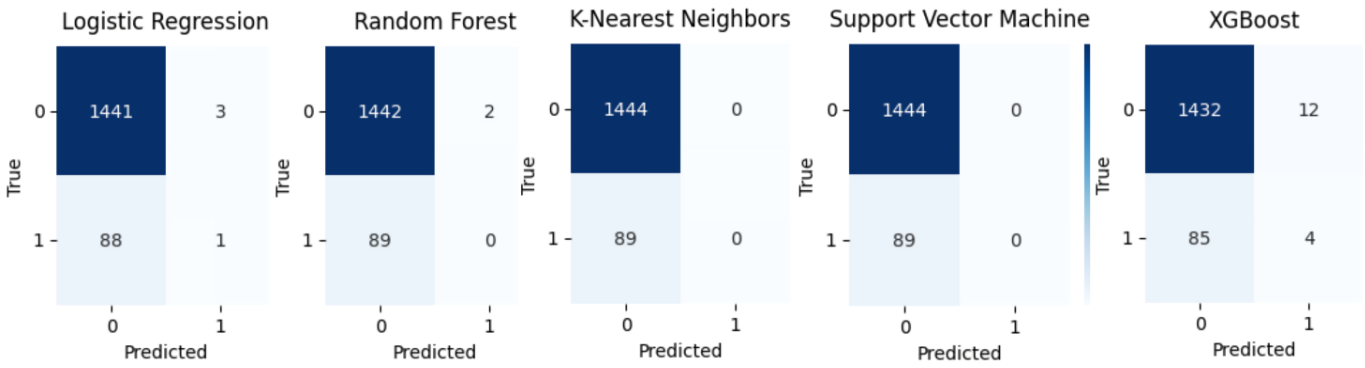

Fig. 9. Confusion Matrix for Imbalanced Dataset

TABLE III
PERFORMANCE METRICS FOR DIFFERENT OPTIMIZATION TECHNIQUES

Optimization	Model	Accuracy	Precision	Recall	F1-Score	AUC
ACO	RF	0.943	0.914	**0.986**	0.949	0.992
	SVC	0.922	0.914	0.942	0.928	0.978
	GB	0.897	0.878	0.938	0.907	0.895
	W_E	0.941	0.913	0.983	0.947	0.987
	S_E	0.945	0.921	0.981	0.950	0.987
	SV_E	0.951	0.940	0.970	0.955	0.992
DE	RF	0.938	0.943	0.935	0.937	0.988
	GB	0.930	0.933	0.927	0.929	0.979
	SVC	0.937	0.938	0.935	0.936	0.987
	W_E	0.948	0.952	0.945	0.947	0.990
	S_E	0.945	0.948	0.942	0.944	0.989
	SV_E	0.946	0.949	0.944	0.945	0.990
BO	RF	0.942	0.947	0.939	0.941	0.989
	GB	0.934	0.937	0.932	0.934	0.980
	SVC	0.937	0.938	0.935	0.936	0.987
	W_E	0.949	0.951	0.949	0.949	0.780
	S_E	0.949	0.950	0.949	0.949	0.989
	SV_E	0.947	0.948	0.947	0.947	0.990
PSO	RF	0.941	0.946	0.938	0.940	0.987
	GB	0.923	0.925	0.920	0.922	0.972
	SVC	0.936	0.937	0.934	0.935	0.987
	W_E	0.953	0.954	0.953	0.953	0.992
	S_E	**0.956**	**0.956**	0.956	**0.992**	**0.993**
	SV_E	0.952	0.951	0.951	0.991	0.992

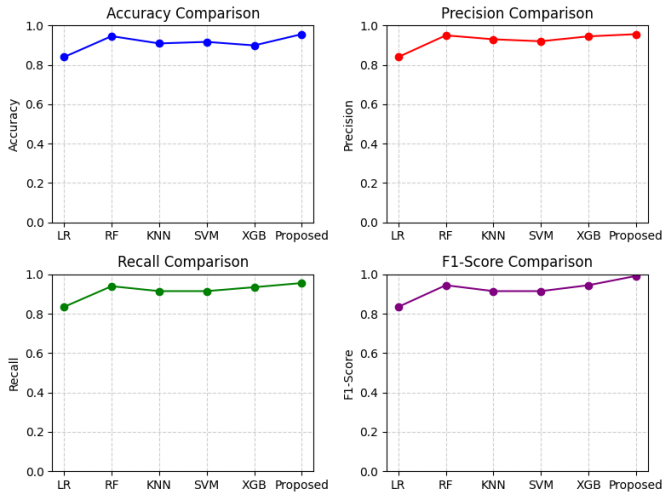

Fig. 15. Evaluation Metrics Comparison of Baseline ML Classifiers with Proposed Model

Table IV compares the results of the proposed approach with previous work done on the same data set. The results show that our proposed approach outperforms other existing work which shows the effectiveness of PSO Optimized Stacked Ensemble ML Classifier over baseline ML Classifiers.

E. Statistical Analysis

To evaluate the strength and robustness of the optimization techniques and different ensemble methods used in this paper, we carried out Statistical Analysis. In this we have compared the difference between Best Accuracy(B_A) and Mean Accuracy(M_A). The standard deviation (SD) is also calculated from the accuracy distribution for all proposed models. The formulas are shown in the following equations.

$$B_A(m_i) = \max_{j=1}^{N} A_{ij} \tag{41}$$

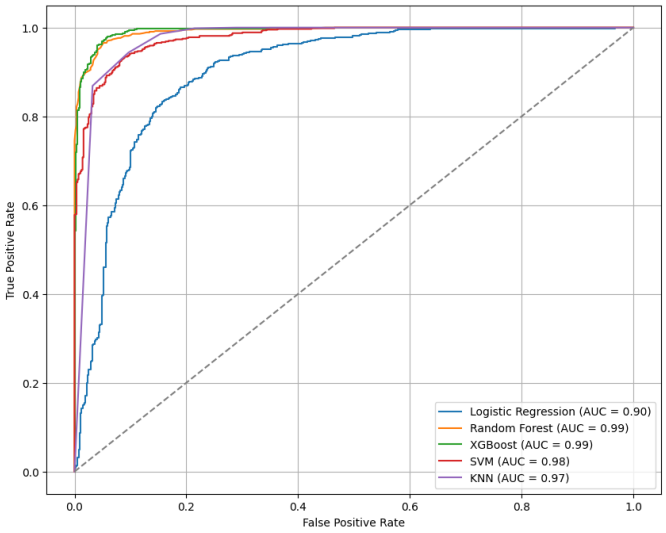

Fig. 10. ROC Comparison for SMOTEEN Balanced Dataset with PCA Feature Selection

TABLE IV
COMPARISON OF PROPOSED APPROACH WITH EXISTING WORK

Source	Methodology	Results
[14]	Minimal genetic folding	Accuracy: 83.20%
[20]	Logistic Regression	Accuracy: 86.00%
[34]	Naïve Bayes	Accuracy: 82.00%
[22]	K-nearest neighbours	Accuracy: 94.00%
[35]	Logistic regression, SVM, random forest, decision tree	Accuracy: 90%
[36]	LSTM	Accuracy: 94%
[37]	DNN-based AutoHPO	Sensitivity: 71.6%
[38]	Naïve Bayes	Accuracy: 82%
[39]	Random Forest	Accuracy: 90%
Proposed	PSO Optimized Stacked Ensemble	Accuracy: 95.6%

Where: $\max_{j=1}^{N} A_{ij}$: Maximum accuracy for model m_i across all models.

$$M_A(m_i) = \frac{1}{N}\sum_{j=1}^{N} A_{ij} \tag{42}$$

Where: N: Total number of ETL models, A_{ij}: Accuracy of model m_i for ETL model j, $M_A(m_i)$: Mean accuracy of model m_i.

$$SD(m_i) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(A_{ij} - M_A(m_i))^2} \tag{43}$$

Where: A_{ij}: Accuracy of model m_i for ETL model j, $M_A(m_i)$: Mean accuracy of model m_i, The sum of squared deviations is divided by N and the square root is taken to compute the standard deviation.

The table V presents the performance metrics of three ensemble models (W_E, S_E and SV_E) based on Mean Accuracy (M_A), Mean Precision (M_P), Mean Recall (M_R), Mean F1 score (M_F1), Best Accuracy (B_A), the difference between Best Accuracy and Mean Accuracy (B_A-M_A), and Standard Deviation (STD). Among the models, SE achieves the highest M_F1 (0.959) and B_A (0.956), indicating its superior ability to balance precision and recall while achieving the best classification performance. However, SV_E demonstrates the highest consistency, with the lowest B_A-M_A (0.003) and STD (0.003), making it the most stable model. W_E performs slightly lower across all metrics, with an M_F1 of 0.949 and B_A of 0.953, but still delivers competitive results. Overall, S_E is the best performing model in terms of overall accuracy and F1 score, while SV_E is the most reliable due to its minimal variability.

The table VI compares the performance of four optimization techniques: ACO, DE, BO, and PSO, in all the ensemble models used in this paper. From the result, it is clear that PSO achieves the best overall performance with the highest mean accuracy of 0.954, mean F1-Score of 0.979 and best accuracy of 0.956, making it the most effective technique. ACO, while excelling in Mean Recall with a value of 0.978, has the lowest Mean Precision at 0.924 and shows greater variability with a B_A-M_A of 0.005 and a standard deviation of 0.005, indicating an imbalance between precision and recall. In contrast, DE and BO exhibit the highest stability, both having the lowest B_A-M_A and a standard deviation of 0.001, making them the most consistent techniques. BO slightly outperforms DE in Mean Accuracy, with values of 0.949 compared to 0.946, along with other metrics. PSO is the best choice for maximizing overall performance, while BO or DE are more suitable for scenarios that require stability and consistency.

Figures 16, 17 show that S_E and PSO emerge as the top performers in their respective categories, delivering the best overall precision and the F1 score. However, SV_E and BO or DE are preferable in scenarios where stability and minimal variability are critical. ACO, despite its strong recall performance, shows limitations due to its imbalance between precision and recall.

VII. CONCLUSION AND FUTURE SCOPE

The research aims at improving stroke prediction using machine learning techniques, notably through a combination of SMOTEEN data balancing and the PSO optimized stacked ensemble model. Our Proposed Approach demonstrated a high prediction accuracy of 95.6%, highlighting its effectiveness in managing imbalanced datasets. A key focus of the study is to address the bias in machine learning predictions caused by unbalanced training data and disparities in healthcare access. The study argues that SMOTEEN Data Balancing in conjunction with ensemble methods, such as stacking, can help mitigate these biases by improving the resilience and accuracy of the system. We have also compared our proposed approach with Other Ensemble Methods and optimization techniques and found that PSO with Stacked Ensemble is outperforming all other combinations. Future work can explore the integration of deep learning models, such as CNNs or RNNs, to automatically extract complex features and improve prediction accuracy. Combining deep learning with PSO-optimized ensemble methods could further improve performance. Furthermore, deploying and validating these models in real-world settings and developing explainable AI techniques for deep learning would ensure robustness, transparency, and fairness across diverse populations.

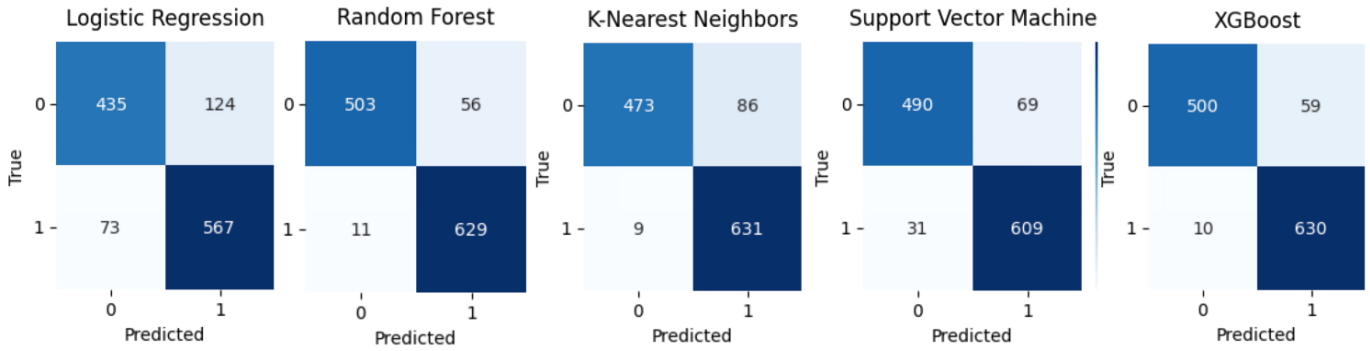

Fig. 11. Confusion Matrix for SMOTEEN Balanced Dataset with PCA

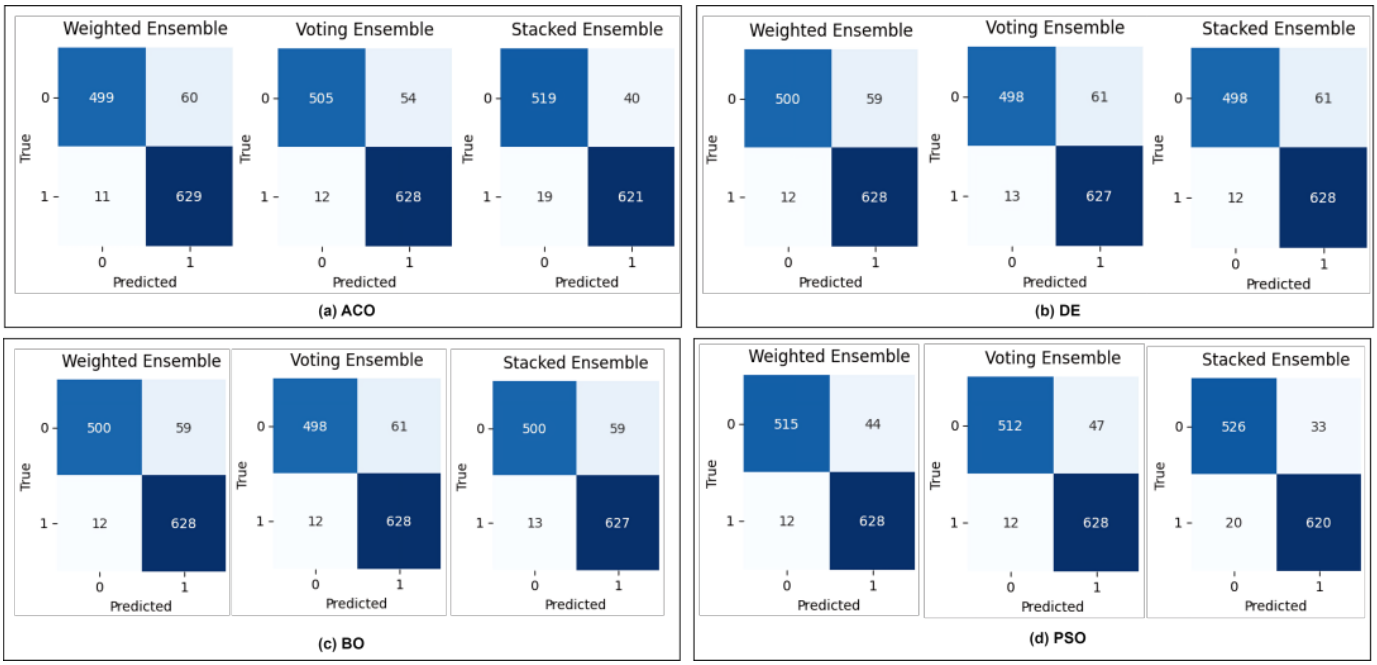

Fig. 14. Confusion Matrix: (a) Ant Colony Optimization Based Ensemble Models, (b) Differential Equation Based Ensemble Models, (c) Bayesian Optimization based Ensemble Models, (d) Particle Swarm Optimization based Ensemble Models

TABLE V
STATISTICAL ANALYSIS OF ENSEMBLE MODELS ACROSS ALL OPTIMIZERS

Model	M_A	M_P	M_R	M_F1	B_A	B_A-M_A	**STD**
W_E	0.948	0.942	0.958	0.949	0.953	0.005	0.005
S_E	0.949	0.944	0.957	0.959	0.956	0.007	0.005
SV_E	0.949	0.947	0.953	0.960	0.952	0.003	0.003

TABLE VI
STATISTICAL ANALYSIS OF OPTIMIZATION TECHNIQUES ACROSS ALL ENSEMBLE MODELS

Optimizer	M_A	M_P	M_R	M_F1	B_A	B_A-M_A	**STD**
ACO	0.946	0.924	0.978	0.950	0.951	0.005	0.005
DE	0.946	0.949	0.944	0.946	0.948	0.001	0.001
BO	0.949	0.950	0.949	0.949	0.949	0.001	0.001
PSO	0.954	0.954	0.953	0.979	0.956	0.002	0.002

DECLARATION

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

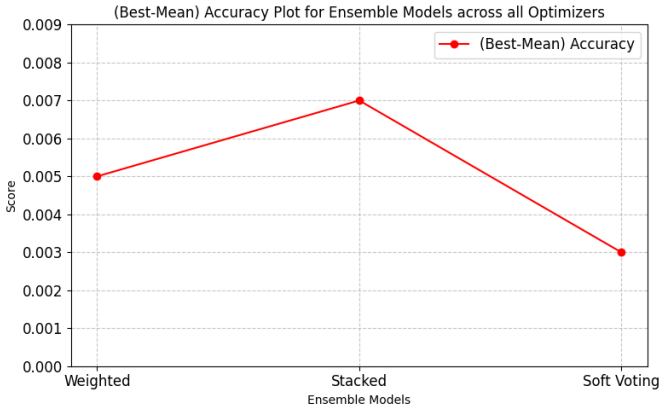

Fig. 16. (Best-Mean) Accuracy Comparison for Ensemble Models

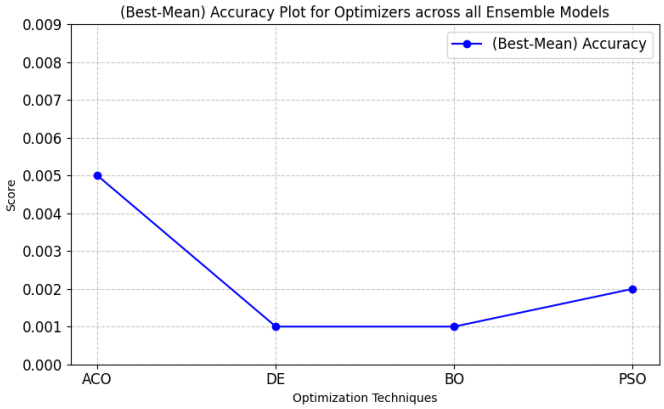

Fig. 17. (Best-Mean) Accuracy Comparison for Optimization Techniques

COMPETING INTERESTS

The authors declare no competing interests.

Author Contributions

A.J., A.K.D., M.G., S.Y., A.P. worked on Writing – original draft, Validation, Project administration, Methodology, Data curation, Conceptualization, while R.A., B.L., S.M. conducted Writing – review & editing, Visualization, Project administration, Data curation.

AVAILABILITY OF DATA AND MATERIALS

The datasets used during the current study are available in Kaggle repository, https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset.

REFERENCES

[1] Kogan, E., Twyman, K., Heap, J., Milentijevic, D., Lin, J. H., & Alberts, M. (2020). Assessing stroke severity using electronic health record data: a machine learning approach. BMC medical informatics and decision making, 20, 1-8.

[2] Wang, W., Rudd, A. G., Wang, Y., Curcin, V., Wolfe, C. D., Peek, N., & Bray, B. (2022). Risk prediction of 30-day mortality after stroke using machine learning: a nationwide registry-based cohort study. BMC neurology, 22(1), 195.

[3] Campagnini, S., Arienti, C., Patrini, M., Liuzzi, P., Mannini, A., & Carrozza, M. C. (2022). Machine learning methods for functional recovery prediction and prognosis in post-stroke rehabilitation: a systematic review. Journal of NeuroEngineering and Rehabilitation, 19(1), 54.

[4] Polikar, R. (2012). Ensemble learning. Ensemble machine learning: Methods and applications, 1-34.

[5] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. Wiley interdisciplinary reviews: data mining and knowledge discovery, 8(4), e1249.

[6] Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. Frontiers of Computer Science, 14, 241-258.

[7] Firoozbakhsh, K. K., Kunkel, C. F., Scremin, A. E., & Moneim, M. S. (1993). Isokinetic dynamometric technique for spasticity assessment. American journal of physical medicine & rehabilitation, 72(6), 379-385.

[8] Singh, T., Ninkovic, B. M., Tasic, M. S., Stevanovic, M. N., & Kolundzija, B. M. (2022). 3-D EM modeling of medical microwave imaging scenarios with controllable accuracy. IEEE Transactions on Antennas and Propagation, 71(2), 1640-1653.

[9] Taylor, R. A., & Sansing, L. H. (2013). Microglial responses after ischemic stroke and intracerebral hemorrhage. Journal of Immunology Research, 2013(1), 746068.

[10] Schiff, L., Hadker, N., Weiser, S., & Rausch, C. (2012). A literature review of the feasibility of glial fibrillary acidic protein as a biomarker for stroke and traumatic brain injury. Molecular diagnosis & therapy, 16, 79-92.

[11] Frey, S., & Ertl, T. (2016). Progressive direct volume-to-volume transformation. IEEE transactions on visualization and computer graphics, 23(1), 921-930.

[12] Mushtaq, S., Saini, K. S., & Bashir, S. (2023, May). Machine Learning for Brain Stroke Prediction. In 2023 International Conference on Disruptive Technologies (ICDT) (pp. 401-408). IEEE.

[13] Chen, M., Tan, X., & Padman, R. (2023). A Machine Learning Approach to Support Urgent Stroke Triage Using Administrative Data and Social Determinants of Health at Hospital Presentation: Retrospective Study. Journal of Medical Internet Research, 25, e36477.

[14] Mezher, M. A. (2022). Genetic folding (GF) algorithm with minimal kernel operators to predict stroke patients. Applied Artificial Intelligence, 36(1), 2151179.

[15] Khatri, I., Fraser, H., Bacher, I., & Madsen, T. (2023). Abstract TMP53: Prediction Of Acute Cerebrovascular Events Based On Patient Reported Symptoms. Stroke, 54(Suppl_1), ATMP53-ATMP53.

[16] Dritsas, E., & Trigka, M. (2022). Stroke risk prediction with machine learning techniques. Sensors, 22(13), 4670.

[17] Mridha, K., Ghimire, S., Shin, J., Aran, A., Uddin, M. M., & Mridha, M. F. (2023). Automated stroke prediction using machine learning: an explainable and exploratory study with a web application for early intervention. IEEE Access, 11, 52288-52308.

[18] Zheng, Y., Guo, Z., Zhang, Y., Shang, J., Yu, L., Fu, P., ... & Global Health Epidemiology Reference Group (GHERG). (2022). Rapid triage for ischemic stroke: a machine learning-driven approach in the context of predictive, preventive and personalised medicine. EPMA Journal, 13(2), 285-298.

[19] Kaur, M., Sakhare, S. R., Wanjale, K., & Akter, F. (2022). [Retracted] Early Stroke Prediction Methods for Prevention of Strokes. Behavioural Neurology, 2022(1), 7725597.

[20] Guhdar, M., Melhum, A. I., & Ibrahim, A. L. (2023). Optimizing accuracy of stroke prediction using logistic regression. Journal of Technology and Informatics (JoTI), 4(2), 41-47.

[21] Paul, D., Gain, G., Orang, S., Das, P., & Chaudhuri, A. K. (2022). Advanced random forest ensemble for stroke prediction. Training, 66, 34.

[22] Uma, S. K., & Rakshith, S. R. (2022). Stroke analysis using 10 ml comparison. Int. J. Res. Appl. Sci. Eng. Technol, 10, 3857-3862.

[23] Kim, D. Y., Choi, K. H., Kim, J. H., Hong, J., Choi, S. M., Park, M. S., & Cho, K. H. (2023). Deep learning-based personalised outcome prediction after acute ischaemic stroke. Journal of Neurology, Neurosurgery & Psychiatry, 94(5), 369-378.

[24] Fedesoriano. Stroke prediction dataset. Kaggle. https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

[25] Swain, K., Tak, T. K., Upreti, K., Kshirsagar, P. R., Bala, S., Krishnan, R. C. P., ... & Mohanty, M. N. (2024). Enhancing Stroke Prediction Using LightGBM With SMOTE-ENN and Fine-Tuning: A Comprehensive Analysis. Cureus, 16(12).

[26] Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). Computers & Geosciences, 19(3), 303-342.

[27] Mondal, S., Ghosh, S., & Nag, A. (2024). Brain stroke prediction model based on boosting and stacking ensemble approach. International Journal of Information Technology, 16(1), 437-446.
[28] Alhatemi, R. A. J., & Savaş, S. (2024). A weighted ensemble approach with multiple pre-trained deep learning models for classification of stroke. Medinformatics, 1(1), 10-19.
[29] Mondal, S., Ghosh, S., & Nag, A. (2024). Brain stroke prediction model based on boosting and stacking ensemble approach. International Journal of Information Technology, 16(1), 437-446.
[30] Storn, R. (1996, June). On the usage of differential evolution for function optimization. In Proceedings of North American fuzzy information processing (pp. 519-523). Ieee.
[31] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE, 104(1), 148-175.
[32] Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. Soft computing, 22(2), 387-408.
[33] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.
[34] Sailasya, G. & Kumari, G. L. Analyzing the performance of stroke prediction using ml classification algorithms. Int. J. Adv. Comput.Sci. Appl. https:// doi. org/ 10. 14569/ IJACSA. 2021. 01206 62 (2021).
[35] Ali, A. A. Stroke prediction using distributed machine learning based on Apache spark. Stroke 28(15), 89–97 (2019).
[36] Singh, M. S., & Choudhary, P. Stroke prediction using artificial intelligence. In 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), 158–161 (IEEE, 2017).
[37] Liu, T., Fan, W. & Wu, C. A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical dataset. Artif. Intell. Med. 101, 101723 (2019).
[38] S.K. Mamatha, H.K. Krishnappa, N. Shalini, Graph theory based segmentation of magnetic resonance images for brain tumor detection, Pattern Recognit. Image Anal. 32 (1) (2022) 153–161.
[39] G.N. Ahmad, H. Fatima, A.S. Saidi, Efficient medical diagnosis of human heart diseases using machine learning techniques with and without GridSearchCV, IEEE Access (2022).