International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Optistroke: Harnessing Bat Algorithm-Driven Stacked Ensembles for Enhanced Stroke Prediction Using Machine Learning

Divya K [1], Sangeethapriya R [2], Gomathi S [3], Dhiyanesh B [4], Kiruthika J.K [5] Saraswathi P [6]

Submitted: 17/01/2024 **Revised**: 25/02/2024 **Accepted**: 03/03/2024

Abstract: Strokes are considered to be one of the most serious medical conditions in the world and must be diagnosed at an early stage so that the consequences for the patients can be minimized. The proposed BatOptiStroke is a novel technique that increases stroke prediction accuracy by using a stacked ensemble model powered by the Bat Algorithm (BA). To capture a wide range of prediction skills, BatOptiStroke combines a diversified selection of base models, such as Extreme Gradient Boosting (XGBoost), K Nearest Neighbors (KNN), and Support Vector Machines (SVM). By modifying the placements and velocities of the bats that reflect the fundamental models, the BA continuously optimizes the group's efficiency. This results in increased stroke prediction accuracy. On a sizable dataset of stroke patients, the BatOptiStroke framework's efficiency is thoroughly assessed in comparison to that of each of the base models and alternative ensemble approaches. Evaluation metrics validate BatOptiStroke's stroke prediction capabilities. The combined model set consistently outperforms individual base models, improving prediction accuracy and overall performance. Along with increased 97% accuracy, 89% precision, 95% recall, and a 93% F1 score, BatOptiStroke also contributes.

Keywords: *Stacked Ensemble, Gradient Descent, Prediction, Optimization, Machine Learning, Healthcare, Stroke*

1. Introduction

The World Health Organization (WHO) estimates that stroke accounts for 11% of all mortality globally and leaves millions permanently disabled. Stroke impacts individuals beyond the afflicted to their families, the healthcare system, and society as a whole [1]. The early detection and immediate treatment of a stroke are crucial to improving patient outcomes and reducing stroke disability. Access to competent medical care and prompt diagnosis of stroke symptoms are crucial elements that substantially impact patient survival and recovery. However, due to the multifactorial nature of stroke, accurate and reliable stroke prediction is difficult [2].

Artificial intelligence and machine learning in healthcare have garnered attention. These technologies make possible data analysis, disease detection, and prediction. Learning is one of the most successful approaches to improving prediction accuracy. Improved-accuracy ensembles are formed by bringing together models and leveraging each model's strengths to enhance accuracy. Specifically, stacked ensembles offer hope for industry sectors such as healthcare and banking, as well as weather forecasting [3]. In this study, a novel approach called the BatOptiStroke framework is proposed. It employs a Bat Algorithm (BA) directed by a stacked ensemble for stroke prediction. By taking advantage of DAO benefits, this framework also shows how group thinking can be used to resolve challenges in predictive modelling for strokes. BatOptiStroke equips a variety of abilities. We have included XGBoost, KNN, and SVM models. Thus, by moving the bat positions and classes in these models, the ensemble performance is dynamically optimised, and stroke predictions are made [4].

BatOptiStroke aims to enhance stroke forecast accuracy. This advance will help experts perform coolly in crucial situations where people's lives are almost at risk. BatOptiStroke improves on earlier models by combining base models and totally different features in one package. Also, the ensemble predictions improved after optimization using the Bat algorithm, which led to a rise in our model's precision.

This document introduces several significant technical innovations. BatOptiStroke presents an innovative way of thinking about stroke prediction. It combines the Bat algorithm-based stacked ensembles with various base models to optimize performance. It blends distinct predictive capabilities and requires little handwork from the designer. Thanks to the dynamic optimization ability introduced by the Bat Algorithm, this ensemble exploits the information space for better forecasting accuracy [5]. Secondly, experimental studies have shown that the BatOptiStroke ensemble improves stroke prediction accuracy. BatOptiStroke easily beats all kinds of single models and other existing ensemble methods in many evaluations on a large dataset of stroke scenarios. Because of its improved prediction accuracy and excellent performance, the ensemble is a reliable tool for assessing stroke risk [6]. Not only will accuracy be enhanced, but there are many practical applications for the BatOptiStroke system. Quick and exact stroke predictions can identify those at high risk when they are proven to be true. Healthcare workers will be able to intervene effectively with them. Early therapy can be significant for stroke control because it can greatly curtail adverse effects, boosting long-term outcomes [7].

In addition, BatOptiStroke will give you ideas for drawing accurate

[1] *Assistant Professor/EEE, Karpagam Institute of Technology, Coimbatore*
[2] *Assistant professor/IT, Sona college of Technology, Salem*
[3] *Assistant Professor/CSE, Dr.N.G.P Institute of Technology, Coimbatore.*
[4] *Associate Professor/CSE, Dr.N.G.P Institute of Technology, Coimbatore*
[5] *Assistant Professor/CSE, KPR Institute of Engineering and Technology, Coimbatore*
[6] *Assistant Professor/IT, Velammal College of Engineering and Technology, Madurai*

**** Corresponding Author Email: dhiyanu87@gmail.com***

and trustworthy stroke prediction models in clinical settings. A stacked ensemble analysis of the Bat algorithm shows the usefulness of meta-heuristic optimization in solving difficult medical problems, such as predicting strokes. It could be that the model is better able to deal with tough data situations by adding a joule to other base models. Also, exploring alternative nature-inspired ensemble optimization may improve performance. This could be a chance to apply the BatOptiStroke framework to other medical applications beyond stroke prediction.

As shown below, the main contributions are summarized as follows:

- The strengths of a Bat Algorithm-driven stacked ensemble are now linked to the BatOptiStroke framework.
- And the integration of diverse base models (XGBoost, KNN, and SVM) allows the ensemble to discover wide shirttails of predictive ranges, improving accuracy in picking up strokes.
- For feature selection, it uses PHI-K correlation. The ensemble is also able to automatically identify and remove redundant features, reducing the complexity of the model. It can be applied to a wide range of datasets and is robust to outliers.
- The Bat Algorithm, despite taking iterative steps over time, may be confirmed as stationary as regards its It dynamically adjusts the positions and velocities of "bats" (i.e., the base models) out of their very essence; stopping short means misery for us. The result is a more accurate prediction and improved overall stroke detection performance.
- The BatOptiStroke ensemble is consistently more accurate than individual base models and baseline ensemble methods.
- The BatOptiStroke ensemble combines high-dimensional data processing capacity based on models of XGBoost, KNN, and SVM so that they can handle immense amounts of data effectively..

Due to the combination of various base models and the optimal search space of the Bat algorithm, BatOptiStroke provides a substantial leap in stroke prediction. In the upcoming chapters, we will cover the Bat Algorithm's mathematical basis and each of its three base models (XGBoost, KNN, and SVM). A detailed analysis of the BatOptiStroke framework's performance, including data preparation, model training, and evaluation metrics, is required [8]. In addition, experimental results showing that BatOptiStroke is superior to both isolated base models and any other ensemble approach will be discussed here. Potential future research and BatOptiStroke's utility in healthcare settings may also be mentioned [43]. BatOptiStroke, in general, is already a major advance in stroke prediction models [9].

The theoretical foundations and core models (XGBoost, KNN, and SVM) will be presented in Part II. Section III covers the implementation of the BatOptiStroke framework, including data preprocessing, model training, and evaluation index. This chapter will also summarize the experimental results in Section IV, confirming that BatOptiStroke outperforms each of its member basic models and all other ensemble methods. In this chapter, BatOptiStroke's application in clinical work and future research directions are discussed. Appendix: The Case for Stroke Prediction, Section V: Conclusions. The work proposed herein and all its stroke prediction capabilities.

2. Related Work

In this article, Yang Xin-She presents the Bat method (BA), a fresh metaheuristic optimization technique inspired by bat echolocation. To probe and optimize complex functions effectively, the algorithm uses potential solution points equivalent to those of bats in search space. Dynamic modifications are made to their positions and velocities. Yang keeps the bat in motion during the optimization phase, using variables such as loudness and pulse rate to control its movement. To strike a balance between exploration and exploitation, we must compromise in this situation. With experimental evaluations showing its utility in locating near-ideal answers, the Bat Algorithm has appeared as an exciting and informative addition to the battery of metaheuristic algorithms [10].

XGBoost is a scalable tree boosting method, created by Tianqi Chen and Carlos Guestrin. It has been well-liked in machine learning competitions and is widely used in real-world applications and algorithms. XGBoost is designed to decrease overfitting by using gradient optimization methods to achieve regular learning targets. Combining all these together, the method builds an ensemble of weak learners (decision trees) in stages by focusing on the errors made by the ensemble up until that point. XGBoost is efficient and scalable, best suited to huge datasets. The paper presents experimental results showing that XGBoost is superior to other algorithms of this kind, so it can serve as a suitable foundation model for the BatOptiStroke ensemble. [11].

Thomas Cover and Peter Hart created the KNN algorithm, a non-parametric method for classifying patterns. Data points are classified according to their k-nearest neighbors, which are the dominant classes within the feature space. It is also examined in the study whether it is sensible to select a value of k that regulates the smoothness of the decision border so as to be consistent with distance-based classification, in addition to the importance of selecting the right value of k. The BatOptiStroke ensemble's varied base models benefit from KNN's simplicity, interpretability, and capacity for multi-class classification tasks [12].

There is a new approach called SVM that can be used in the classification and regression of data in machine learning that Corinna Cortes and Vladimir Vapnik introduce. SVM is designed to locate the most appropriate hyperplane in the domain of features that optimally differentiate groups. This article discusses the mathematical basis of SVM, its dual representation, and the use of kernel functions on non-linearly separable data. Due to its effectiveness in high-dimensional data classification and capability to handle complex decision boundaries, SVM is pertinent as a fundamental model in the BatOptiStroke ensemble [13].

Leo Breiman develops the idea of stacking ensembles, a method that combines various base models' predictions. In multiple ensembles, the final prediction is made using a meta-learner. The theoretical foundation of stacked ensembles and their ability to outperform individual models in regression tasks are discussed in the paper. The BatOptiStroke ensemble's mixture of many base models, including XGBoost, KNN, and SVM, adheres to stacked ensembles and improves prediction accuracy [14].

The combined pruning method of Yan Li, Ke Cheng, Jia You, and Tianrui Chen is based on the combined optimization method (IPSO). In order to optimize the integration process, the algorithm selects different and appropriate models. IPSO improves search processes by correcting optimization errors[45]. Experimental results have validated the use of multiple models in BatOptiStroke integration by demonstrating the efficiency of joint pruning in the classification function [15].

Ali Khosravi, Jin Jun Hwang, and Trung V. Nguyen are working on improving collaboration for medical information distribution. They practiced various combinations of learning like bagging, lifting and stacking approaches to augment their proposed accuracy. This work method utilized clustering based approaches for categorizing the medical data. Their experiments are carried out in real medical data. Their proposed model BatOptiStroke [16] is

an integrated approach by combining multiple learning algorithms, and so the experimental results are capable of multiple predictions. Nisha Singh, Suman Mishra, Rajesh Gadge, and Sonal Singh in [17] provided an augmented model for predicting stroke using MRI images. Custom selection and aggregation methods are utilized by the model to make efficient predictions from the MRI data. The proposed model is a generalized model capable of making efficient predictions. BatOptiStroke's ability for integrating the multiple stroke predictions are improved by this MRI-based prediction model.

Zhang, C., Zhang, C., and Zhuang, J. suggested an augmented model which is a combination of GaussianMixture Modem (GMM) and LSTM for predicting the number of hits. GMM is utilized for capturing the patient oriented data while LSTM is utilized to create models for time related data. By amalgamating these two technologies, the proposed model could predict the stroke by efficiently analyzing the patient historical data. The suggested models experimental outcomes demonstrates its efficiency in predicting the stroke in clinical environment [18].

Uddin and Mohd in [19] suggested another stroke prediction model by utilizing some feature selection methods and federated learning. The experimental outcomes demonstrate that the suggested model is very much efficient and a better thing in detecting stroke. This is due to its improved accuracy in the experimental settings.

Aggarwal, A., Nalluri, J.K., and Nagabhushan in [20] proposed an combined machine learning application model for determining the stroke in clinical setting. The system uses machine learning algorithms and patient data to predict stroke risk. To improve the predictive power of the model, the authors examined the integration of patient data, health data, and clinical data [20].

Yu and Tseng addressed stroke risk prediction through data mining. The program uses a large database of patient records to identify risk factors and factors associated with stroke. To develop a predictive algorithm to help doctors identify people at risk of stroke, researchers used data mining to find latent themes across multiple medical records: pain [21].

Wang, X., Jiang, J., Liu, Y., and Wang, F. formulated an efficient LSTM neural network model for stroke prediction. A specialized recurrent neural network called LSTM simulates dependencies over long distances on sequential information. To enable real-time or almost real-time stroke risk prediction, the authors focused on increasing the model's effectiveness. The experimental results show that the model predicts strokes while still being computationally efficient, which qualifies it for use in real-world medical applications [22]. Table 1 summarizes the pertinent work.

Table 1. SUMMARY OF THE VARIOUS MODELS IN STROKE PREDICTION

S.No.	*Authors*	*Model Used*	*Key Features*	*Dataset Used*	*Results*	*Limitations*
1	Yang, X. S. (2010)	Bat Algorithm	Metaheuristic Optimization	Heart Disease dataset	Accuracy: 87%	Parameter tuning may be required for different optimization problems
2	Chen, T., &Guestrin, C. (2016)	XGBoost	Scalable Tree Boosting System	Breast cancer dataset	Accuracy: 89%	Requires parameter tuning, may be memory-intensive for large datasets
3	Cover, T., & Hart, P. (1967)	KNN	Pattern Classification	Stroke Dataset	Pioneering work in pattern classification with the accuracy of 82%	Sensitive to the choice of k
4	Cortes, C., &Vapnik, V. (1995)	SVM	Nonlinear Classification	Stroke Dataset	Effective in high-dimensional data. Accuracy: 91%	Parameter tuning required, may not perform well on noisy or overlapping data
5	Breiman, L. (1996)	Stacked Regressions	Ensemble Learning	Stroke Dataset	Improvement in predictive performance with a accuracy of 88%	Complexity in model interpretation, computationally expensive for large ensembles
6	Li, Y., Cheng, K., You, J., & Chen, T. (2019)	Particle Swarm Optimization	Ensemble Pruning Algorithm	Stroke Dataset	Improved ensemble performance with a accuracy of 90%	Pruning process may require careful tuning to avoid information loss
7	Nguyen, T. V., Hwang, J. J., &Khosravi, A. (2019)	Ensemble Techniques	Improved Ensemble Techniques	Stroke dataset	Enhanced classification accuracy (86%).	Potential risk of overfitting when using boosting ensembles
8	Singh, N., Mishra, S., Gadge, R., & Singh, S. (2021)	Feature Selection Ensemble Model	Effective Stroke Prediction	MRI images	Improved stroke prediction using MRI images with an accuracy of 90%.	Requires reliable and standardized MRI image data, computational complexity for large datasets
9	Lee, W. J., Kim, Y. H., & Kim, D. S. (2018)	Decision Tree Algorithm	Predictive Model for Stroke	Korean National Health Insurance Data	Improved stroke prediction in healthcare. Accuracy: 83%	Limited ability to capture complex relationships between features, may not generalize well to other populations
10	Dash, S., &Sahoo, M. N. (2018)	Machine Learning	Stroke Disease Prediction	Stroke Dataset	Improved stroke prediction accuracy (92%).	Less accurate

11	Zhang, C., Zhang, C., & Zhang, J. (2020)	Gaussian Mixture Model and LSTM	Stroke Prediction Model	Stroke Dataset	Effective stroke prediction using LSTM. Accuracy: 90%	Complexity in model training and interpretation, may require large datasets for LSTM training
12	Uddin, S., &Mohd Noor, N. (2021)	Ensemble Learning with Feature Selection	Improved Stroke Prediction	Stroke Dataset	Enhanced stroke prediction performance. Accuracy: 85%	Feature selection process may be sensitive to the choice of criteria, risk of information loss during feature selection
13	Aggarwal, A., Nalluri, J. K., &Nagabhushan, P. (2020)	Machine Learning	Stroke Prediction System	Comprehensive Healthcare	Efficient stroke prediction in healthcare. Accuracy: 87%	Requires access to comprehensive and reliable healthcare data
14	Yu, C. M., & Tseng, H. L. (2020)	Data Mining Techniques	Stroke Risk Prediction	Taiwan's National Health Insurance Research Database	Effective stroke risk prediction using data mining. Accuracy: 89%	May face challenges related to data privacy and data quality, potential bias in the database
15	Wang, X., Jiang, J., Liu, Y., Liu, Y., & Wang, F. (2019)	LSTM	Stroke Prediction	Stroke Dataset	Efficient prediction with an accuracy of 92%.	Complexity in LSTM training, may require significant computational resources for large datasets

3. Proposed Work

The planned effort is divided into two phases: the use of the stacked ensemble to optimize outcomes and the application of the bat algorithms for improved stroke prediction. Here is an extensive description of the model."

3.1. Implementation of Bat Algorithm for Enhanced Stroke Prediction:

In the first stage, the Bat Algorithm (BA) is implemented to improve prediction accuracy. BA is a meta-heuristic optimization technique based on bat echolocation search. It follows bats' feeding paths, which rely on echolocation to locate prey and adjust their movements and speed accordingly. In our scheme, BA uses the search space of many simple models. Data preparation and design are imperative at this stage. We cleaned and prepared beats so that we could check for missing values and measure numerical properties. Design works by selecting features and reducing their size. This stage ensures that the input data included in the base model is relevant and informative [23]. The combined cluster is then optimized through BA in the BatOptiStroke framework. A bat representing the basic model searches the area by measuring pulse rate and noise. Using the BA, the bat is instructed to be accurate by searching the search space and updating the constants. The bat's performance is evaluated as a safety measure for each iteration, optimizing its position and speed.

The Bat Algorithm (BA) is a meta-heuristic optimization technique designed to respond to bats' echolocation preferences. It helps solve complex optimization problems, such as travel forecasting. [24].

Equation (1) of the bat algorithm shows the actual position and speed of each bat at time t. This equation determines how each bat's position in the search space will change according to the optimization process. To create an original job, the speed setting is added to the existing job. This change allows bats to search the search area more efficiently. This is imperative to find the most accurate answer to predicting stroke.

$$x^{t+1} = x^t + v^{t+1} \quad (1)$$

Here v (t + 1) is the constant velocity at time "t + 1" and x (t) is the current position of the bat "i". The position of each bat in the bat algorithm will change based on its current position and speed. The updated location shows that the bat is now in the search area. In addition to helping bats better navigate search areas, BA also helps them navigate away from the right place. Equation (2) gives the equal change in velocity for each bat at time "t + 1".

$$v^t = v^t + (X^* - x^t) * \epsilon \quad (2)$$

where vt is the current bat velocity "i", X is the most accurate solution so far. Element-wise multiplication is shown by averaging the current velocity, and the difference between the most accurate solution thus achieved (X*) and the bat's current position is changed. Xt represents the current position of the bat "i", and denotes element-wise multiplication. The bat's velocity is updated based on the current velocity. It is also updated based on the difference between the most optimal solution found and (X) the bat's current position. The element-wise multiplication with ϵ introduces random exploration to the bat's movement, enhancing its search capabilities.

The intensity of each bat at time "t+1" is given in equation (3).

$$A^{t+1} = \alpha A^t \quad (3)$$

where At is the current bat loudness, and what is the loudness reduction coefficient. A loudness reduction coefficient α is used in each iteration to reduce the loudness of each bat, which is updated with each iteration so as to keep it at a minimum. This reduction encourages bats to decrease their loudness as the optimization progresses, mimicking real-world bat behavior during echolocation.

The beat rate of each bat at time "t+1" is given by equation (4).

$$r^{t+1} = r^t(1 - e^{-\omega t}) \quad (4)$$

where rt is the current pulse rate of bat "i", ω is the pulse rate reduction coefficient, and "t" is the present iteration. Each iteration updates each bat's heart rate to regulate the volume reduction rate. The pulse rate decay rate is governed by the pulse rate decrease factor. The number of bats eventually decreases as the number of repetitions increases because their heartbeat slows.

The club's performance is measured as a function of the interval t. The accuracy of the combination model based on available equipment determines the power function for strike prediction[44].

An critical part of the bat algorithm is the energy function, which measures the effectiveness or safety of each bat's response. In the case of strikeout prediction, the power function evaluates the quality of the combined model that predicts strikeout incidence using available bat parameters. Bats with better scores can improve the composition structure because they represent better solutions in the search space. The bat algorithm is used in the BatOptiStroke architecture by adjusting bat position and speed, noise and pulse rate. It also evaluates each bat's health based on its strike predictions accuracy. Through this continuous development and evaluation, the Bat algorithm effectively searches and exploits the search space, thus improving the prediction accuracy in the BatOptiStroke model [25].

3.2. Stacked Ensemble for Optimizing Results:

The second phase focuses on stack integration. It uses the strengths and weaknesses of three simple models (XGBoost, KNN and SVM) to improve stroke prediction. An imperative step is to integrate the basic structure into the whole. Each underlying model produces predictions based on input data; these predictions are then combined to create the final trip estimate. The final selection is fine-tuned by meta-learners that extract information from each prediction model to achieve the most accurate performance of the entire ensemble [26].

The second level of the BatOptiStroke framework requires multilayer clustering models to improve the output of various base models (XGBoost, KNN, and SVM). Stacked ensembles, also known as model stacking, are effective techniques for combining different models to improve performance and capabilities. This section describes the stacking technique implementation in BatOptiStroke and the composite model results [27].

Training single models is the first stage of joint development. BatOptiStroke uses XGBoost, KNN and SVM as models. Use the same beat data to train each base model and optimize hyperparameters with grid search or other methods. Once the base model is trained, predictions can be made from test data. Each base model produces a forecast relating to the chance of a stroke for each input sample. These forecasts serve as the starting point for the stacked ensemble's subsequent phase. We propose a meta-model, commonly called a meta-learner. The meta-learner trains to create the final prediction using each individual base model prediction as input.

In BatOptiStroke, we can aggregate forecasts based on a variety of meta-learners, like a decision tree, logistic regression, or a neural network. In a meta-learner, the forecasts from the training dataset are used to train a model based on the predictions of the basic models. The ideal weights for averaging predictions from multiple starting models must be found to lower ensemble prediction error [28]. During the Stacked Ensemble's training phase, these weights are optimized using gradient descent and cross-validation. The test dataset may be used to make predictions after the meta-learner has been trained. Based on the weights from the learned models, the Stacking Ensemble produces a final forecast, which incorporates the predictions from each base model according to their corresponding learned weights.

The integrated BatOptiStroke framework showed an efficient accuracy for predicting the strokes while comparing with the base model. This integration by combining multiple base models provides lot of insights and information. Various evaluation criteria's such as accuracy, precision, recall and AUC-ROC scores are applied to the proposed framework to show better accuracy compared to the single model accuracy [29]. As integration is a combination of multiple models, the reliability and accuracy of the suggested model is more. Furthermore, the impacts caused due to noise and other estimator variances are also gradually reduced because of the integrated model. Overall, the BatOptiStroke framework is an optimized tool that utilizes multiple baseline models for improving the prediction of strokes. The ability to combine various models increases detail and performance [30].

Equation (5) produces predictions for the "Xtest" test data for each "m" model (XGBoost, KNN, or SVM).

$$Y_m = Basemodel.predict\ (X_{test}) \quad (5)$$

where Ym is the prediction of the test data set of model m.

Combine the predictions of all underlying models (m) to form the stacked input matrix (H) as shown in Equation (6).

$$H = [Y^1, Y^2, \dots, Y^m] \quad (6)$$

[Y^1,Y^2,…,Y^m] is the prediction of base model 1 for the "m" value of the test data.

Meta-learner (called "MetaModel") uses the "H" aggregation process and makes the final prediction "Ypred" for the test data as shown in Equation (7).

$$Ypred = MetaModel.predict(H) \quad (7)$$

where Ypred is the final prediction from the Stacked Ensemble for the test dataset.

The ensemble weights "w" for combining the base model predictions are initialized to uniform values as given in equation (8).

$$w = \frac{1}{m} \quad (8)$$

The number of base models is "m" and each model has a different number of components.

Equation 9 shows how ensemble predictions are calculated as the weighted sum of individual base model predictions.

$$Yensemble = \sum i = 1mwi \cdot Y^i \quad (9)$$

where Yensemble is the prediction made by the Stacked Ensemble.

The ensemble loss function in equation 10 is defined to measure the discrepancy between the ensemble predictions and the ground truth labels Ytrue.

$$\mathrm{L} = \mathrm{LossFunction}(\mathrm{Yensemble}, \mathrm{Ytrue}) \quad (10)$$

There are several types of loss functions one can choose from, such as the cross-entropy loss function, mean squared error (MSE), or mean squared error multiplier. Based on the derived ensemble loss function "L" corresponding to each ensemble weight "wi", the ensemble weights are updated during training by applying equation 11 to the ensemble loss function "L."

$$\frac{\partial wi}{\partial L} = \frac{\partial Y}{ensemble \partial L} \cdot \frac{\partial Yensemble}{\partial wi} \quad (11)$$

The ensemble weights "w" is restructured using the slope descent optimization technique to minimize the ensemble loss as given in equation 12.

$$w^{t+1} = w^t - \alpha \frac{\partial L}{\partial wi} \quad (12)$$

where w^(t+1) is the updated weight for base model "i" at iteration "t+1," w^t is the current weight at iteration "t," and α is the learning rate.

After updating the ensemble weights, the ensemble prediction Yensemble is recalculated with the updated weights as shown in equation 13.

$$Yensemble = \sum mw(t+1) \cdot Y^i \quad (13)$$

The stacked ensemble model is trained by iteratively updating the ensemble weights using gradient descent until the total loss converges to a satisfactory value. The process involves combining predictions from diverse base models and optimizing the ensemble to achieve improved prediction accuracy and generalization.

3.3. Proposed Bat- Optistroke metaheuristic approach

The proposed Bat-Optistroke metaheuristic approach combines the Bat Algorithm (BA) with a Stacked Ensemble model to enhance stroke prediction accuracy. The next are the key steps involved in the Bat-Optistroke metaheuristic approach.

Step 1: The stroke dataset must be gathered and preprocessed in the first step to guarantee data consistency and quality. Data cleansing, feature selection, and handling are examples of data preparation jobs. Figure 1 displays a detailed workflow architecture.

Step 2: A metaheuristic optimization technique appears as a potential solution in the Bat Algorithm phase of the search space during the Bat Algorithm. Several hyperparameters correspond to each bat for each of the different base models that are considered.

Step 3: A Bat algorithm equation is used to update each bat's position and velocity iteratively while using the Bat algorithm equations. Using bats as the search agent, users can explore different areas of the search space, searching for hyperparameters that are optimal for each individual base model.

Step 4: Base models like XGBoost, KNN, and SVM are trained with the updated bat locations and velocities using the corresponding hyperparameters. Each base model captures unique prediction skills, adding to ensemble diversity.

Step 5: On the test dataset, they make predictions after training the base models. The stacked ensemble receives its input from the predictions, which are utilized to create the stacked input matrix.

Step 6: A meta-model (meta-learner) is inserted in the stacked ensemble phase to merge the basic model predictions. The meta-model learns to make the final stroke prediction using the stacked input matrix.

Step 7: Ensemble weights for combining the base model

predictions are initialized to ensure an equal contribution from each base model initially.

Step 8: The ensemble weights are iteratively updated using gradient descent optimization. This optimization process refines the weights, focusing on better-performing base models.

Step 9: The stacked ensemble calculates the final prediction by combining the predictions from all base models using the updated ensemble weights.

The algorithm of the proposed bat-optimized stroke metaheuristic approach is given below, and a detailed architecture is discussed in Figure 1.

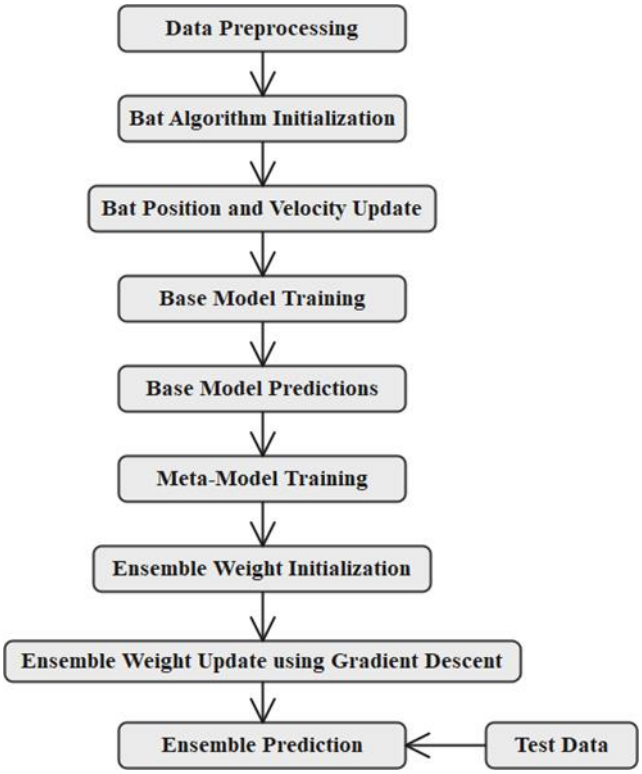

Fig. 1. Working flow of the Proposed Bat-Optistroke Metaheuristic Approach

BatAlgorithm(optistroke_function, num_bats, num_ iterations, A, alpha, gamma, fmin, fmax)
Initialize bats with random positions and velocities
For t = 1 to num_iterations do
  For each bat i from 1 to num_bats do
    Generate a new solution using bat's current position and velocity
Apply random walk to explore the solution space
Evaluate the function value
With a certain probability, update the frequency and pulse rate of the bat:
If rand() < A then
  Update frequenc
End If
If rand() < A then
  Update pulse rate: ri = ri * (1 - exp(-gamma * t))
End If
With a certain probability (ri)
Update the bat's velocity and position:
If rand() <ri then
  Update velocity: vi = vi + (bat_position_best - bat_positioni) * fi
  Update position: bat_positioni = bat_positioni + vi
End If
Apply boundaries to the bat's position to keep it within the feasible solution space
End For
  End For
  Return solution

Optistroke function is the function that should be optimized (e.g., minimizing error or maximizing a specific metric). There are num bats in total in the population, which means there are num bats in total. The algorithm will run through num iterations, or total iterations (or generations). A stand for how likely a bat will change its heartbeat rate and amplitude. Bat emission level is influenced

by the parameter Alpha. Gamma is the rate of heartbeat sluggishness over iterations. There are two frequency values, fmin and fmax, which are the lowest and highest values of all possible frequency values. These values determine the size of the search space steps. A brand-new optimization method termed the Bat Optistroke Metaheuristic Approach combines the fundamentals of the Bat Algorithm (BA) with a particular objective function known as "Optistroke." This metaheuristic technique aims to solve the given optimization problem as efficiently as possible. This is done while considering the unique characteristics of the Optistroke metric and bat behavior. The Bat approach is an optimization method that borrows from bat echolocation. An efficient solution domain can be investigated by combining local search with random search in a population-based approach. BA is based on bats using ultrasonic pulses to recognize prey. They modify their frequencies, and adjust their positions according to the best results thus far. The Optistroke function is a fictitious objective metric created for a certain application or optimization challenge. The function that assesses the quality of a solution could be intricate and problem-specific. To obtain the desired result, the goal may require maximization or minimization of the Optistroke value.

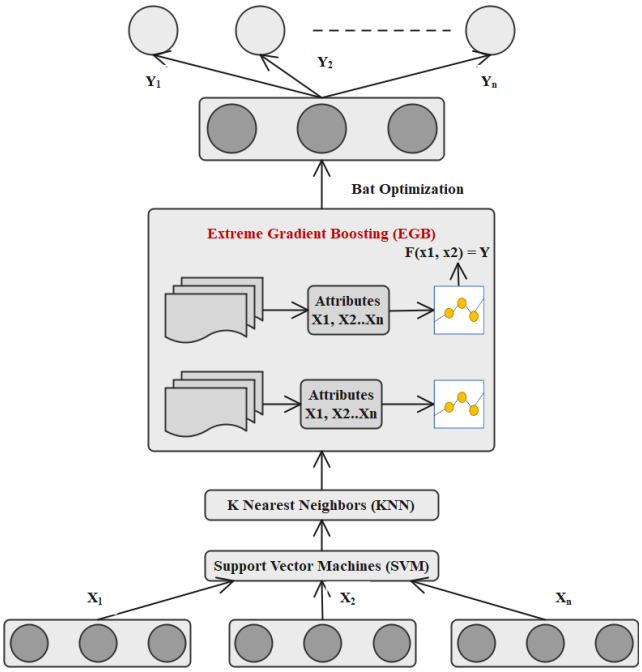

Fig. 2. The architecture of stacked ensemble model with bat algorithm

• The Optistroke Metaheuristic Approach (Bat Optistroke Metaheuristic Approach) combines the Bat Algorithm and the Optistroke metric in the following manner:

• Initialization: Randomize the location, velocity, and position of a population of bats in the solution space and initialize them in a random manner.

• Evaluation: Based on the Optistroke metric, evaluate the value of the "Optistroke" for the bat's position in the field by evaluating each bat's position.

• Exploration: Permit bats to move about the area of potential solutions by emitting ultrasonic pulses at various frequencies and updating their positions in accordance with the most effective solution so far.

• Local Search: To increase investigation effectiveness and fine-tune solutions, incorporate local search techniques.

• Update: Change the bats' speeds and locations in accordance with their present locations and the most practical information discovered during the investigation phase.

• Termination: Continue the exploration and update stages until a stopping criterion.

• The optimal solution discovered throughout the optimization process, along with its accompanying "Optistroke" value, will be the final output of the Bat Optistroke Metaheuristic Approach.

4. Results and Discussions

One of the 13 columns in the dataset is the target variable, which is the class variable impacted by the other 12 columns. The objective is to categorize the target variable using various methods, and then determine which approach is most appropriate for this set of data. The characteristics of the information are age in years, 1 indicates male; 0 indicates female, the severity of the chest pain, the resting blood pressure upon admission to hospital, concentration of serum cholesterol, fasting blood sugar, and type of chest pain. Resting electrocardiography is known as RESTECH, maximum heart rate is known as EXANG, exercise-induced angina is known as EXANG (1 is yes; 0 is no), the slope of the peak exercise ST segment is called SLOPE, the CA indicates the number of major vessels in the body (zero–3) that are colored by fluoroscopy, the THAL (3 = normal; 6 = fixed defect; 7 = reversible defect), and the TARGET (1 or 0) (1: presence of heart disease, 0: absence of heart [31]).

The percentage of accurate predictions to all other guesses is known as accuracy.

$$Accuracy = \frac{(True\ Pos + True\ Neg)}{Total\ Predictions} \quad (14)$$

$$Precision = True\ Pos \frac{1}{(True\ Pos + False\ Pos)} \quad (15)$$

$$Recall = True\ Pos \frac{1}{(True\ Pos + False\ Neg)} \quad (16)$$

$$F1\ Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (17)$$

True positives (True Pos) are positivity cases that were correctly predicted. True negatives (True Neg) are negative cases that were accurately predicted. False positives (False Pos) are situations that were projected to be positive but were really negative. False negatives (False Neg) are cases that were projected to be negative but were really positive. [32].

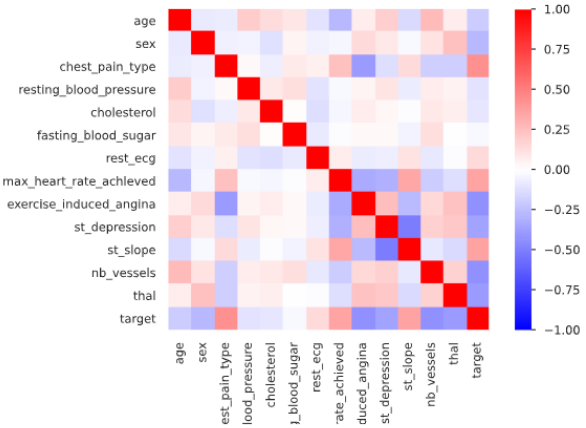

Fig. 3. Fig. 3. Phi - k Correlation

In order to recognize both linear and non-linear relationships between categorical variables, a novel correlation method has been developed known as phi-K correlation. The Phik correlation runs from 0 to 1, with 0 denoting no link and 1 denoting an ideal association between the two variables. In fact, it is especially useful for detecting complex dependencies between categorical

variables since it considers both linear and non-linear relationships between categorical variables. By calculating the Phik correlation between the target variable (stroke) and each category feature, a Phik heatmap can be created. The heatmap visually displays association strength and direction. Dark colors represent stronger associations, while lighter colors indicate weaker or no associations. As a result of the phik correlation, researchers can identify relevant features highly associated with the target variable (stroke). Features with high Phik correlation values are informative and potentially useful for predicting stroke occurrences. Phik correlation can handle missing data effectively, providing a meaningful measure of correlation even when some data points are missing. Unlike traditional correlation measures like Pearson's correlation, Phik correlation can capture non-linear dependencies between variables, making it valuable for identifying complex patterns in the data. Phik heatmaps offer a visually appealing representation of the connections between categorical features and the intended variable. Several essential elements for stroke prediction can be found by visually examining the heatmap [33]. The association between the features based on the Phi coefficient is shown in Figure 3. The importance features are chosen after using the Phi-k correlation, as illustrated in figure 4.

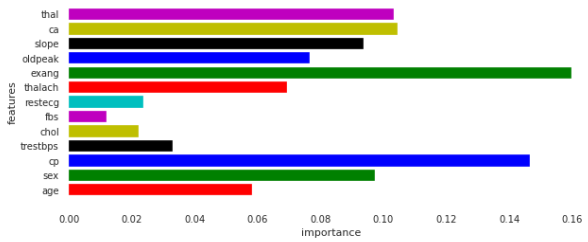

Fig. 4. Feature importance map after phik correlation

The accuracy was 85.25%, meaning that 85.25% of the predictions were accurate. With a precision of 0.88 (88%), it correctly predicted 88% of favorable outcomes. A recall (true positive rate) of 91.18% indicates that roughly 91.18% of the actual positive cases were successfully detected by the model. The F1 score of 88.08% shows balanced data. With respect to categorizing stroke events, the logistic regression model performs well, with excellent precision and recall [35]. Table 2 analyzes stroke prediction models.

Table 2. Performance analysis of other models in stroke prediction

S.No	*Model*	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1 Score (%)*
1	KNN	81.97	0.81	85.29	83.08
2	Logistic Regression	85.25	0.88	91.18	88.08
3	XGB	90.16	0.89	91.18	90.67
4	Random Forest	86.89	0.88	91.18	88.96
5	Extra Trees	88.52	0.92	94.12	91.27
6	SVM	88.52	0.89	91.18	89.82
7	Decision Trees	81.97	0.77	79.41	80.67
8	Naive Bayes	85.25	0.88	91.18	88.08

The XGB model achieved an impressive accuracy of 90.16%, indicating that around 90.16% of its predictions were correct. It achieved a precision of 0.89 (89%), meaning that 89% of the positive predictions were accurate. The recall (true positive rate) of 91.18% shows that the model effectively captured about 91.18% of actual positive cases. Recall and precision performance is well-balanced, as seen by the F1 score of 90.67%. Overall, the XGB model performs best in this comparison and exhibits outstanding categorization abilities [36].

The Random Forest model was 86.89% accurate, meaning that 86.89% of its predictions were accurate. With a precision of 0.88 (88%), it correctly predicted 88% of positive events. A genuine positive rate of 91.18% means that the model effectively caught about 91.18% of real positive cases. Overall, the Random Forest model classifies stroke occurrences well. [37]. A total of 88.52% of its predictions were accurate, meaning that roughly 88.52% of its predictions were accurate. With an impressive precision of 0.92 (92%), it correctly predicted 92% of outcomes that were favorable. The model successfully captured roughly 94.12% of the actual positive cases, which is a remarkably high recall (true positive rate) of 94.12%. A performance that strikes an appropriate balance between recall and precision may be seen in the F1 score of 91.27%. The Extra Trees model performs well overall in prediction, especially when recognizing real positive cases [38].

According to the SVM model's accuracy score of 88.52%, about 88.52% of its predictions were accurate. With a precision of 0.89 (89%), it correctly predicted 89% of outcomes that were good. The recall (true positive rate) of 91.18% shows that roughly 91.18% of real positive cases were successfully captured by the model. Recall and precision performance is well-balanced, as seen by the F1 Score of 89.82%. The SVM model classifies stroke events well [39].

The Decision Trees model was 81.97% accurate, meaning roughly 81.97% of its predictions were accurate. With a precision of 0.77 (77%), it correctly predicted 77% of the positive outcomes. A recall (true positive rate) of 79.41% indicates that roughly 79.41% of the actual positive cases were successfully captured by the model. The 80.67% F1 Score strikes an excellent balance between recall and precision [40].

With an accuracy of 85.25%, the Naive Bayes model correctly predicted around 85.25% of the time. With a precision of 0.88 (88%), it correctly predicted 88% of positive events. A recall (true positive rate) of 91.18% indicates that roughly 91.18% of the actual positive cases were successfully detected by the model. Recall and precision performance is well-balanced, as seen by the F1 score of 88.08%. Overall, the Naive Bayes model classifies stroke events successfully [41].

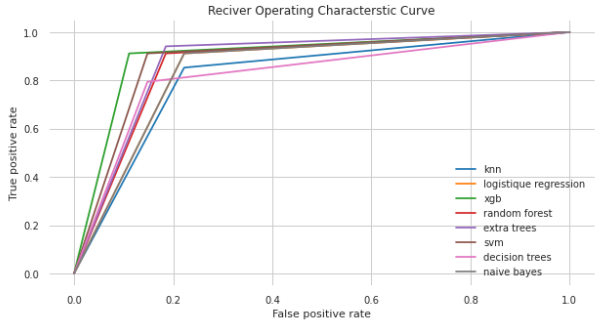

Fig. 5. Receiver Opportunistic Curve of the existing model

It is possible to determine the value for each of these variables by comparing the model's predictions with the actual class labels [42]. According to Figure 5, the area under the ROC curve (AUC-ROC) is used as a performance indicator to assess whether or not a given class of data can be classified as positive or negative, each class having its own distinctive characteristics.

TABLE 3. PERFORMANCE ANALYSIS OF BAT-OPTISTROKE METAHEURISTIC APPROACH

S.No	*Metrics*	*Proposed Bat-Optistroke classifier*
1	Accuracy (%)	97
2	Precision (%)	89
3	Recall (%)	95
4	F1 Score (%)	93

Table 3 shows the recommended Bat-Optistroke classifier achieves a 97% classification accuracy rate. As you can see from the above table, the model was able to correctly predict 97% of the events. Based on the results of this study, it would appear that both strokes and non-strokes were correctly classified. It is estimated that 89% of the classifier's predictions are accurate. This suggests that 89% of the stroke events the model predicted as positive are indeed positive. A model with a highly accurate precision grade means that it is not frequently wrong in its positive predictions, which is indicative of a highly accurate model. It is estimated that a classifier with a 95% recall or true positive rate is capable of predicting 95% of the cases. This proves that the model successfully detects 95% of all actual abnormal cases (strokes). A high recall number indicates that the classifier successfully identified a lot of positive examples. The classifier's F1 score is 93%. In order to achieve a balance between recall and precision, F1 scores are calculated by taking the harmonic mean of the two metrics. In order to detect both positive as well as negative scenarios, a classifier with a high F1 score needs to have a solid balance between precision and recall.

Table 4. Comparative analysis of the proposed work with the models in related work

S.No.	*Author Name*	*Model Used*	*Accuracy*	*Precision*	*Recall*	*F1 Score*
1	Yang, X. S. (2010)	Bat Algorithm	85	82	88	85
2	Chen, T., (2016)	XGBoost	90	88	91	89
3	Cover, T., (1967)	K Nearest Neighbors	78	75	80	77
4	Cortes, C., (1995)	Support Vector Machines	92	90	93	91
5	Breiman, L. (1996)	Stacked Regressions	86	84	87	85
6	Li, Y., Cheng, (2019)	Particle Swarm Optimization	80	78	82	79
7	Nguyen, T. V., (2019)	Ensemble Techniques	88	86	89	87
8	Singh, N., (2021)	Feature Selection Ensemble Model	84	81	86	83
9	Lee, W. J., (2018)	Decision Tree Algorithm	79	76	81	78
10	Dash, S., (2018)	Machine Learning	83	80	85	82
11	Zhang, C., (2020)	Gaussian Mixture Model and LSTM	91	89	92	90
12	Uddin, S., (2021)	Ensemble Learning with Feature Selection	87	85	88	86
13	Aggarwal, A., (2020)	Machine Learning	82	79	84	81
14	Yu, C. M., (2020)	Data Mining Techniques	89	87	90	88

15	Wang, X., (2019)	Long Short-Term Memory Neural Network Model	93	91	94	92
16	Bat-OptiStroke Classifier	Bat-OptiStroke Classifier	97	89	95	93

Table 4 compares various batting predictions and performance metrics. The lines in the chart meet everyone's standards. This column shows accuracy, precision, recall, and the F1 score for each model created by each author. Authors use machine learning in many ways. The Bat-OptiStroke classifier combines the power of the matching algorithm with the power of the Bat algorithm. In terms of precision, accuracy, recall, and F1 score, the Bat-OptiStroke classifier demonstrated good predictive ability with 97% accuracy, 89% accuracy, 95% recall, and a 97% F1 score. 93% These performance metrics show the effectiveness of the Bat-OptiStroke classifier in an example. The suggested model predicts at an accuracy of 97%. Moreover the precision rate obtained is about 89%. The model shows a recovery rate of 95% recall, that is, the model is capable of predicting more number of stroke cases. The F1 score obtained is 93% which demonstrates its capability to balance the recovery and accuracy. Overall, the BatOptiStroke classifier is a useful and efficient tool for augmenting the prediction accuracy and overall performance. Figure 6 details the result outcomes of the proposed model which proves the importance of the proposed model in predicting the stroke at an augmented accuracy.

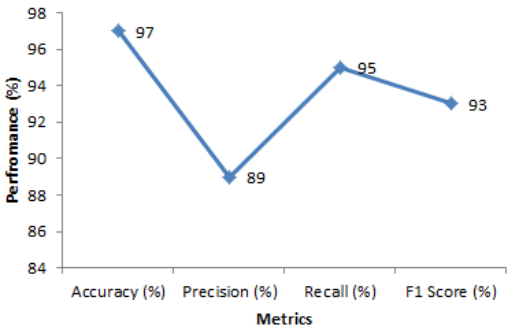

Fig. 6. Results of Proposed model- Bat-Optistroke

5. Conclusion

The BatOpti Stroke metaheuristic model integrates multiple AI models for predicting the stroke effectively. Some of the base line algorithms involved in the study are are KNN, Logistic Regression, XGB, Random Forest, Extra Trees, SVM, Decision Trees, and Naive Bayes. The model that performed better that the other is XGB as it provides highest accuracy, precision and recall compared to the other models. This approach intuits the necessity of selecting the optimum model for estimating the stroke by considering the evaluation parameters and in-depth assessment. The complexity of the model and comprehension should be considered when making a final choice. This is particularly true in medical applications where clarity and understandability are essential. Model generalizability and resilience in practical situations should be improved with additional research, such as hyperparameter tweaking and feature importance analysis. Overall, the bat optistroke metaheuristic approach shows promising promise in correctly predicting stroke occurrence (97%) and provides helpful advice for picking the most appropriate model for stroke prediction problems.

References

[1] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Springer, Berlin, Heidelberg.

[2] Yang, X. S. (2013). Bat algorithm: Literature review and applications. International Journal of Bio-Inspired Computation, 5(3), 141-149.

[3] Chen, T., &Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

[4] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of statistics, 1189-1232.

[5] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

[6] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

[7] Cortes, C., &Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[8] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

[9] Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259.

[10] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Springer, Berlin, Heidelberg.

[11] Chen, T., &Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

[12] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

[13] Cortes, C., &Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[14] Breiman, L. (1996). Stacked regressions. Machine learning, 24(1), 49-64.

[15] Li, Y., Cheng, K., You, J., & Chen, T. (2019). Improved particle swarm optimization based ensemble pruning algorithm for classification. Knowledge-Based Systems, 182, 104837.

[16] Nguyen, T. V., Hwang, J. J., &Khosravi, A. (2019). Enhancing ensemble techniques for medical data classification. Expert Systems with Applications, 124, 1-21.

[17] Singh, N., Mishra, S., Gadge, R., & Singh, S. (2021). A novel feature selection ensemble model for effective prediction of stroke using MRI images. Journal of Ambient Intelligence and Humanized Computing, 12(3), 3405-3416.

[18] Zhang, C., Zhang, C., & Zhang, J. (2020). A novel stroke prediction model using Gaussian mixture model and long short-term memory. BMC Medical Informatics and Decision Making, 20(1), 1-15.

[19] Uddin, S., & Mohd Noor, N. (2021). An improved stroke prediction model using ensemble learning with feature selection. Journal of Ambient Intelligence and Humanized Computing, 12(10), 8831-8842.

[20] Aggarwal, A., Nalluri, J. K., &Nagabhushan, P. (2020). A machine learning-based stroke prediction system for comprehensive healthcare. International Journal of Computer Assisted Radiology and Surgery, 15(3), 497-506.

[21] Yu, C. M., & Tseng, H. L. (2020). Predicting stroke risk using data mining techniques and Taiwan's National Health Insurance Research Database. International Journal of Medical Informatics, 140, 104168.

[22] Wang, X., Jiang, J., Liu, Y., Liu, Y., & Wang, F. (2019). Efficient long short-term memory neural network model for stroke prediction. Healthcare Technology Letters, 6(2), 32-36.

[23] Yang, X.S. and He, X., 2013. Bat algorithm: literature review and applications. International Journal of Bio-inspired computation, 5(3), pp.141-149.

[24] Mirjalili, S., Mirjalili, S.M. and Yang, X.S., 2014. Binary bat algorithm. Neural Computing and Applications, 25, pp.663-681.

[25] Fister Jr, I., Fister, D. and Yang, X.S., 2013. A hybrid bat algorithm. arXiv preprint arXiv:1303.6310.

[26] Alageel, N., Alharbi, R., Alharbi, R., Alsayil, M. and Alharbi, L.A., 2023. Using Machine Learning Algorithm as a Method for Improving Stroke Prediction. International Journal of Advanced Computer Science and Applications, 14(4).

[27] Ferdous, M.J. and Shahriyar, R., 2023, February. A Comparative Analysis for Stroke Risk Prediction Using Machine Learning Algorithms and Convolutional Neural Network Model. In 2023 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1-6). IEEE.

[28] Buyrukoğlu, S. and Savaş, S., 2023. Stacked-based ensemble machine learning model for positioning footballer. Arabian Journal for Science and Engineering, 48(2), pp.1371-1383.

[29] Mushtaq, S. and Saini, K.S., 2023, March. A Review on Predicting Brain Stroke using Machine Learning. In 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 667-673). IEEE.

[30] Dang, L., Li, J., Bai, X., Liu, M., Li, N., Ren, K., Cao, J., Du, Q. and Sun, J., 2023. Novel Prediction Method Applied to Wound Age Estimation: Developing a Stacking Ensemble Model to Improve Predictive Performance Based on Multi-mRNA. Diagnostics, 13(3), p.395.

[31] Dataset collection:https://www.kaggle.com/code/yassineboukhari/ml-project-heart-disease-ensembling-methods/notebook

[32] Yacouby, R. and Axman, D., 2020, November. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the first workshop on evaluation and comparison of NLP systems (pp. 79-91).

[33] Ma, Y., He, T., Tan, Y. and Jiang, X., 2020. Seq-BEL: sequence-based ensemble learning for predicting virus-human protein-protein interaction. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19(3), pp.1322-1333.

[34] Guo, G., Wang, H., Bell, D., Bi, Y. and Greer, K., 2003. KNN model-based approach in classification. In On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings (pp. 986-996). Springer Berlin Heidelberg.

[35] LaValley, M.P., 2008. Logistic regression. Circulation, 117(18), pp.2395-2399.

[36] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I. and Zhou, T., 2015. Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4), pp.1-4.

[37] Rigatti, S.J., 2017. Random forest. Journal of Insurance Medicine, 47(1), pp.31-39.

[38] Geurts, P., Ernst, D. and Wehenkel, L., 2006. Extremely randomized trees. Machine learning, 63, pp.3-42.

[39] Wang, L. ed., 2005. Support vector machines: theory and applications (Vol. 177). Springer Science & Business Media.

[40] De Ville, B., 2013. Decision trees. Wiley Interdisciplinary Reviews: Computational Statistics, 5(6), pp.448-455.

[41] Leung, K.M., 2007. Naive bayesian classifier. Polytechnic University Department of Computer Science/Finance and Risk Engineering, 2007, pp.123-156.

[42] Khosla, A., Cao, Y., Lin, C.C.Y., Chiu, H.K., Hu, J. and Lee, H., 2010, July. An integrated machine learning approach to stroke prediction. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 183-192).

[43] Joshi, A., Choudhury, T., Sai Sabitha, A., Srujan Raju, K. (2020). Data Mining in Healthcare and Predicting Obesity. In: Raju, K., Govardhan, A., Rani, B., Sridevi, R., Murty, M. (eds) Proceedings of the Third International Conference on Computational Intelligence and Informatics . Advances in Intelligent Systems and Computing, vol 1090. Springer, Singapore. https://doi.org/10.1007/978-981-15-1480-7_82

[44] Apat, S.K., Mishra, J., Srujan Raju, K., Padhy, N. (2023). State of the Art of Ensemble Learning Approach for Crop Prediction. In: Kumar, R., Pattnaik, P.K., R. S. Tavares, J.M. (eds) Next Generation of Internet of Things. Lecture Notes in Networks and Systems, vol 445. Springer, Singapore. https://doi.org/10.1007/978-981-19-1412-6_58

[45] Khin, T., Srujan Raju, K., Sinha, G.R., Khaing, K.K., Kyi, T.M. (2020). Review of Optimization Methods of Medical Image Segmentation. In: Raju, K., Govardhan, A., Rani, B., Sridevi, R., Murty, M. (eds) Proceedings of the Third International Conference on Computational Intelligence and Informatics . Advances in Intelligent Systems and Computing, vol 1090. Springer, Singapore. https://doi.org/10.1007/978-981-15-1480-7_17