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

Cardiac Condition Anticipation and Prognostication via Integrated WOA and Bagging-GBDT

Javvaji Venkatarao[1], V. Deeban Chakravarthy[2]

Submitted: 18/09/2023 **Revised**: 19/11/2023 **Accepted**: 30/11/2023

Abstract: Cardiac disease is a significant health concern that leads to more than 17 million deaths every year. The integration of Internet of Medical Things (IoMT) and Artificial Intelligence (AI) in healthcare has made considerable improvements in patient outcomes. However, the recent approach of using the Bagging-Fuzzy-GBDT classifier for anticipating and prognosticating Cardiac disease may not be suitable for all cases due to membership function-based data fuzzification and Grid Search (GS) based hyperparameter selection limitations. This study presents a novel method combining optimization with ensemble learning techniques, specifically Whale Optimization Algorithm (WOA), with Bagging-GBDT classifiers to anticipate and prognosticate Cardiac disease more effectively. This new approach employs membership functions to capture data uncertainty and vagueness, uses ensemble learning techniques to generate multiple random subsamples from the original dataset, and finally utilizes the Bagging-WOA-GBDT classifier to build an accurate prognostication model based on enhanced data representation. The results of the experiment conducted on a publicly available Cardiac disease dataset show that the proposed approach performs better than traditional classifier methods. It provides more reliable and accurate prognostication for Cardiac disease. These findings suggest that the suggested approach could be a valuable tool for healthcare practitioners in diagnosing and preventing Cardiac disease.

Keywords: Cardiac disease predication and diagnosis, Machine learning, Bagging-GBDT, WOA, Optimal hyperparameter selection.

1. Introduction

Cardiac disease, a major global cause of death, requires accurate prognostication for early diagnosis and treatment. It affects the Cardiac and blood arteries, causing over 17 million deaths annually worldwide. Cardiac disease is a global public health concern due to risk factors like high blood pressure, cholesterol, smoking, diabetes, obesity, and a sedentary lifestyle. Symptoms include chest pain, shortness of breath, palpitations, fatigue, dizziness, and swelling. Early detection and treatment are crucial. Machine learning uses binary classification and multi-classification for Cardiac disease prognostication, requiring accurate models. The choice depends on the dataset and desired accuracy level, with both methods offering advantages and disadvantages [1-2]. AI is revolutionizing the way Cardiac disease is diagnosed, by utilizing Internet of Medical Things IoMT data to make more accurate and efficient prognostications in E-healthcare [3]. IoMT is a network of medical devices, sensors, and wearable technologies connected to the internet. These devices track physiological parameters, health metrics, and ECG readings, providing real time patient data to healthcare professionals. This data aids in improving patient care and enhancing treatment effectiveness [4]. IoMT improves patient care and healthcare delivery by enabling remote monitoring, eliminating hospital visits, and improving access for rural residents. However, widespread adoption raises privacy and security concerns [5]. Ensuring the protection of this data is crucial, as breaches potentially have detrimental effects, such identity theft, financial fraud, and unauthorized access to medical records. Overall, the Internet of Medical Things represents a major step forward in use of technology in the healthcare industry, offering the possibility to greatly improve patient outcomes, increase efficiency, and reduce costs, while also requiring vigilant attention to data privacy and security issues [6].

In medical research, classifiers play a crucial role in identifying important indicators of illnesses and developing effective treatment strategies. These classifiers assign scores to prognosticator variables, which help guide further research. Additionally, classifiers can be used to personalize medicine by prognosticating the progression of an illness or the response to treatment based on patient specific data [7-9]. This information can then be used to create personalized treatment plans that are more effective than generic approaches. Classifiers are widely utilized in disease prognostication within medical applications. For instance, Tang et al. employed an SVM classifier for estimating the risk of type-2 diabetes, surpassing decision trees and logistic regression methods. Random forests were

[1,2] Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur Campus - 603203, Chennai, India.
[1]ORCID ID : 0009-0007-3604-2121
[2]ORCID ID : 0000-0003-3880-5181
jv1562@srmist.edu.in[1]
deepanv@srmist.edu.in[2]

used to identify the presence of breast cancer, while both logistic regression and decision trees were employed for managing non-linear interactions related to medical conditions such as breast cancer diagnosis [10-15], lung cancer diagnosis [16], and diabetes prognostication [17]. Random forests have found wide application in prognosticating diseases across various medical fields, including breast cancer diagnosis [18], Alzheimer's disease diagnosis [19], and diabetes prognostication [20]. On the other hand, deep neural networks (DNNs) are a type of classifiers that can automatically learn relevant features from high dimensional data and build models capable of accurately prognosticating the presence or absence of a disease. Composed of multiple layers interconnected nodes, DNNs process input data to extract important attributes. Deep neural networks have bee-n successfully utilized in disease prognostication for a range of medical applications such as skin cancer diagnosis [21], diabetic retinopathy detection [22], and Cardiac disease prognostication [23]. The choice of classifier depends on data characteristics and application needs.

The remaining of this research paper is structured as follows. A thorough assessment of the literature is presented in Section 2. Bagging-GBDT,standard WOA algorithm and Section 3. Describes the hyperparameter optimization challenge of the Bagging-GBDT. Section 4 describes the proposed. Section 5 contains the optimal hyperparameter tuning and innovative outcomes. In Section 6, findings are presented.

2. LITERATURE REVIEW

Many researchers have been focused on enhancing the precision of models for prognosticating cardiac illness. Machine learning methods include naive bayes, XGBoost, decision trees, artificial neural networks, K-nearest neighbours, logistic regression, and artificial neural networks and ensemble learning models like GBDT, Bagging-GBDT have been utilized while creating the classification system [24-29]. Of these, the Bagging-GBDT trained on fuzzified data has been found to be particularly productive and beneficial as a prognostication method for Cardiac disease categorization. It can handle a variety of data types, includes continuing and discrete values and capable of determining the disease's degree of risk as well as solving the binary classification problem of Cardiac disease prognostication. According to the angiographic findings, there are five stages of severity for cardiac disease, ranging from 0 (absent) to 4. Author of [30] prognosticated the risk for everyone. However, there is still scope for increased accuracy compared to the initial multiclassification ensemble model for identifying various cardiac risk factors.

A potent machine learning algorithm called Bagging GBDT uses groups of decision trees to create prognostications. The algorithm involves combining the prognostications from many decision trees trained on various subsets of the training data to produce a final prognostication. One of the key challenges in using bagging GBDT is selecting the optimal parameters for the algorithm. Grid search is one method which entails giving each parameter a specific set of values before training the algorithm with all conceivable combinations of these values [28, 33]. The optimal parameters in bagging GBDT are the highest performance obtained from an examination set given certain parameters. The choice of optimal parameters depends on the specific problem and available resources. Methods for parameter tuning include grid search, random search, and Bayesian optimization. Grid search involves setting possible values for every parameter and searching for matches among all possible combinations. Random search is faster and more efficient when the search space is large and the number of iterations is limited. Bayesian optimization uses a probabilistic model to estimate the objective function and determine the set of parameters based on the current model's outcomes, achieving better results with fewer iterations. Several tools and libraries are available for parameter tuning in GBDT, including scikit-learn, XGBoost, and LightGBM.

Several optimization algorithms, including Evolutionary Algorithms (EA), have been proposed to find the best hyperparameters for Bagging-GBDT. EA is a class of metaheuristic optimization algorithms that have been successfully used to choose the Bagging-GBDT parameter. EA generates a population of potential solutions and evaluates them in accordance with a fitness function that represents the objective of the optimization problem, typically the performance metric of the Bagging-GBDT model. Several studies have proposed different EA-based approaches for Bagging-GBDT parameter selection. For instance, Roshan, S. E., &Asadi, S. proposed a new approach for improving the effectiveness of Bagging, a popular ensemble learning algorithm, for classification of imbalanced datasets. The proposed method uses multiple evolutionary optimization objectives, which is a powerful optimization method that can balance the compromise between classification accuracy and class similarity Includes scores obtained with given decision trees development and use of Bayesian hyper-parameter optimization A novel strategy Xia et al. proposed in the research article " Using Bayesian hyper-parameter boosted decision tree technique for credit scoring" by Credit scoring is an important task for financial institutions, as it facilitates the evaluation of borrowers' credit assessment and associated risk management The authors first outline the challenges facing credit scoring models, including the importance of accuracy; it is complex and interpretable along with. An enhanced decision tree model is then proposed, which mixes different decision trees to increase

the robustness and accuracy of the model. Furthermore, the authors use Bayesian over-parameter optimization to modify the over-parameters of the model and thus improve its performance. Title "Using a gradient enhancing decision tree with a multi verse optimizer for breast cancer detection",Tabrizchi et al.[31] proposed a new breast cancer screening method using GBDT algorithm, based on Multi Verse Optimizer (MVO) The unique contribution of this research is the MVO method a will be used to improve GBDT parameters eg. This approach allows the model to be optimized for accuracy, sensitivity, specificity, and AUC, making it more robust and accurate than conventional breast cancer screening methods. Parameter selection is a crucial aspect of optimizing algorithm parameters in search field optimization.

EA-based approaches are effective in scouring the search field and identifying sound solutions. The choice of algorithm parameters, population size, mutation rate, and crossover rate significantly impact the effectiveness of these approaches. To achieve optimal results, these parameters must be precisely tuned. The literature survey emphasizes the trade-off between computational complexity and optimization effectiveness when using EA-based methods for parameter selection in Bagging-GBDT models. Some EA-based approaches, like GA and PSO, can be computationally expensive, especially when dealing with large datasets and complex models. Therefore, it is essential to balance computational requirements and optimization performance to select an approach that strikes a good balance. Further research is needed to explore hybrid approaches that combine multiple EA-based approaches or other optimization algorithms, as well as objective functions that incorporate additional performance metrics like model interpretability or robustness.

The authors of [32] advocate a singular method for precisely figuring out volcanic lithology that combines the GBDT with optimized parameters. The proposed technique is carried out to a real-international dataset from the Jilin Oilfield in the Songliao Basin of Northeast China. One of the unique contributions of this research is the usage of a parameter-optimized GBDT for volcanic lithology identification. The authors optimized the key parameters of the GBDT set of rules to achieve high accuracy quotes, demonstrating the effectiveness of this approach. The research paper titled ‘A strong AI-based binary and forecast of many classes of cardiac ailment version for IoMT’ with the aid of Yuan et al. [33] makes a specialty of developing a solid and accurate Cardiac sickness prognostication version for the Internet of Medical Things IoMT the usage of AI techniques. The authors propose a unique technique for developing a cardiac ailment prognostication model that mixes assist vector gadget SVM, KNN, and GBDT algorithms. The approach suggested by way of the authors [34] is implemented to a actual-world dataset which include affected person facts from hospitals. The findings indicate that the advised version performs higher existing fashions in step with stability, accuracy, and performance. The version as it should be categorizing coronary Cardiac contamination into binary and a couple of instructions, making it a precious device for scientific practitioners. This accomplishment is creating a reliable and correct coronary Cardiac sickness prognostication model. The model is constructed on aggregate of different AI strategies, making it a sturdy and reliable tool for healthcare professionals. The findings of this research have critical implications for the healthcare industry, as the proposed coronary Cardiac ailment prognostication version can assist healthcare professionals make knowledgeable choices and enhance patient outcomes. The model can be integrated into IoMT systems to permit real-time tracking and early detection of Cardiac ailment, doubtlessly increasing affected person best of existence while decreasing healthcare expenses.

The optimization algorithms provide numerous advantages for optimal parameter selection in GBDT. They can improve efficiency via automating the parameter search system; enhance effectiveness by way of exploring a extensive range of parameter values, and robustness to noise and uncertainty inside the statistics. They also provide flexibility and adaptability to specific problem domain names and dataset characteristics, and permit automation and reproducibility, which might be critical for research, experimentation, and model deployment. By leveraging the strength of optimization algorithms [37], practitioners can efficiently and correctly tune the hyperparameters of GBDT models, main to progressed version performance and higher usage of system getting to know strategies in real-global packages.

3. BAGGING-GBDT, WOA AND HPRERPARAMETER OPTIMIZATION PROBLEM

A. Bagging-GBDT

Bagging can be formulated as an averaging process over multiple models. Let us assume N independent models, each trained using a distinct subset of the training data. The average of all N models' prognostications serves as the Bagging ensemble's forecast for the latest sample x. It is mathematically expressed as shown in Eq. 1.

$$F(x) = \frac{1}{M} * \sum f_i(x) \quad (1)$$

$$f(x) = \sum[\alpha_t * f_t(x)] \quad (2)$$

Where $f_i(x)$ is the prognostication of the i^{th} model for sample x. This averaging process reduces the variance of the prognostications and increases the stability of the

ensemble, as a result of the combination of the separate models' forecasts into a single prognostication. GBDT can be formulated as a boosting process over multiple decision trees. Let us assume T trees in the ensemble, and the prognostication of the t^{th} tree is given by $f_t(x)$. The final prognostication of the GBDT ensemble is given by the average weighted of all T trees' prognostications as shown in Eq. 2. Where α_t is the weight assigned to the t^{th} tree. In GBDT, the weights α_t are determined by minimizing the loss function, that calculates the discrepancy between the actual labels and the anticipated labels. In each iteration, the GBDT algorithm builds a new tree that focuses on the samples that were misclassified by the previous trees.

This boosting process corrects the mistakes of the previous trees and improves the performance of the ensemble. Bagging and GBDT are both ensemble methods that can be formulated mathematically. Bagging is an averaging process over multiple models, while GBDT is a boosting process over multiple decision trees [35]. The mathematical formulation helps to understand the underlying principles and how they work, and can be used to optimize the parameters of the algorithms and to evaluate their performance.

B. WOA

This optimization algorithm inspired by the foraging circular and helical bubble release behaviour of humpback whales is a meta-heuristic optimization algorithm that takes its inspiration from the feeding behaviour of humpback whales. In 2016, this method was proposed by Mirzalili et al. and is designed to solve optimization problems in engineering and science [36]. This algorithm's main purpose is to mimic the foraging behaviour of humpback whales, who release bubbles in a circular or helical pattern to create a wall of bubbles that concentrates their prey, making it easier to catch. There are three major phases of the WOA, namely exploitation phase through bubble-net, prey encircling, and exploration phase.

Prey encircling: The best search agent is selected based on its capacity to find the intended prey. Following this search agent, along with others then modify their places, as described by equations (3) and (4). You may figure out the coefficients R and Q by using equations (3) and (4), respectively. To move search agents into the best place, the numerical values of 'P' and 'R' are modified, encircling the prey in multiple dimensions.

$$\vec{P} = \left|\vec{Q}\vec{Y}^{*}(s) - \vec{Y}(s)\right| \quad (3)$$

$$\vec{Y}(s+1) = \vec{Y}^{*}(s) - \vec{R}.\vec{P} \quad (4)$$

The best performing solution's position vector is denoted by Y^{*}, while $\vec{Y}$ represents another position vector. The current iteration is represented by s and j denotes the absolute value. The dot symbol (•) is used to symbolize the multiplication of elements. The coefficients $\vec{R}$ and $\vec{P}$ can be computed in the following way as shown in Eq. 5 & 6.

$$\vec{R} = 2\vec{p}.\vec{r} - \vec{p} \quad (5)$$

$$\vec{Q} = 2\vec{r} \quad (6)$$

The vector $\vec{p}$ linearly decreases from 2 to 0 with each iteration, and the value of $\vec{r}$ falls within the range of [0, 1]. Changes to the values of vectors $\vec{P}$ and $\vec{R}$, search agents could move into a better position from their current position. It also helps to surround the prey in an n-dimensional space as a result of this process of updating agent positions in the neighborhood direction.

Exploitation phase through bubble-net: As p falls, P's value changes, representing the shrinking behavior of search agents. In order to update its position, the humpback whale selects P values at random between [-1,1], spiraling towards the prey. This shrinking spiral movement can be modelled mathematically as shown in equations (7) and (8). The position update process can take two forms, i.e., the spiral mechanism or the shrinking mechanism, with a 50% chance that both mechanisms would act simultaneously. Equation (4) is a representation of the two behaviors' combined equation.

$$\vec{Y}(s+1) = \vec{Q}'(s).e^{al}.\cos(2\pi l) + \vec{Y}^{*}(s) \quad (7)$$

$$\vec{Q}'(s) = \vec{Y}^{*}(s) - \vec{Y}\,(s) \quad (8)$$

The constant component $\vec{a}$ determines how the spirals are shaped, while *l* is a random number with a range of [-1, 1]. Whales can use either the spiral mechanism or the shrinking mechanism during the position update phase. Throughout the optimization process, it is assumed that there is a 50% chance of employing both mechanisms at once. Both behaviours' combined equations may be written as shown in Eq. 9 and 10.

$$\vec{Y}(s+1) = f(x) = \begin{cases} \vec{Y}^{*}(s) - \vec{R}.\vec{P}, & p < 0.5 \quad (9) \\ \vec{Q}'(s).e^{al}.\cos(2\pi l) + \vec{Y}^{*}(s), & p \geq 0.5 \quad (10) \end{cases}$$

Exploration phase: P is set up in this phase so that it can only have a value of larger than or less than -1, allowing the humpback whales to migrate apart and accelerate their exploration pace. In mathematics, this phenomenon is represented by equations (11) and (12).

$$\vec{Q} = |\vec{R}.\vec{Y}rand - Y| \quad (11)$$

$$\vec{Y}(s+1) = \vec{Y}_{rand} - \vec{P}.\vec{Q} \quad (12)$$

A random whale's position is indicated by $\vec{Y}_{rand}$.The optimization process terminates once the termination criteria have been met. The WOA algorithm is distinguished by its utilization of both the spiral path and circular shrinking dual theories, which enhance the

exploitation process of identifying the ideal location close to the prey. The optimization process ends when the termination criteria are met. The WOA algorithm is characterized by its twin idea of spiral path and circular shrinkage, which improves the process of finding the optimal location around the prey during exploitation, followed by the exploration phase, which expands the by using a random selection of data, do a search in the area of $|\vec{A}|$. The mathematical framework of the WOA algorithm provides a clear understanding of its working principles. The three-step model encompasses the processes of prey encircling, exploitation phase through bubble-net, and exploration phase.

In the first step, the humpback whale's ability to find the location of prey is modelled through a mathematical equation that search agents' positions are updated based on the best solution obtained. The coefficients used in the equation are calculated using linear and random values. In the second step, the exploitation phase through bubble-net is modelled using spiraling movements of humpback whales. An equation with random values and a constant factor simulates the mathematical behavior of spirals decreasing in a helix shaped movement. The final step of the WOA algorithm, the exploration phase, is modelled through random movement of humpback whales. To update the location of search agents, the calculation makes advantage of a random whale's location, increasing the exploration rate. The optimizing procedure inspired by the foraging circular and helical bubble release behavior of humpback whales has been applied to various optimization problems, including function optimization, constraint optimization, and issues with multi-objective optimization. The algorithm is well known for fast convergence speed, high precision, and ability to find the global optimum in complex search spaces. WOA has limitations such as a limited ability to perform global searches and an inconsistent convergence speed. When faced with complex optimization problems, WOA may struggle to escape local optimum.

C. Parameter optimization problem of Bagging-GBDT

Bagging GBDT has several parameters that can be optimized to improve its performance. The mathematical expression for parameter optimization of Bagging-GBDT includes identifying the parameter values that minimize a specific loss function. The loss function for Bagging-GBDT may be written as shown in Eq. 13.

$$L(y, F) = \sum_i \ell(y_i, f_i) \tag{13}$$

where L is the loss function, y is the target variable, F is the prognosticateed output, and ℓ is the loss function for each individual observation. Minimising the loss function is the aim of parameter optimization by finding the optimal values for the parameters. The parameters of Bagging-GBDT that can be optimized include:

1) Number of estimators (n_estimators): This parameter determines the number of trees in the ensemble. The mathematical expression for optimizing this parameter involves determining n_estimators' value that minimizes the loss function. There are estimators in simply the value of the parameter, denoted as n_estimators. For example, if it is set the value of n_estimatorsto 100, then the Bagging-GBDT algorithm will build an ensemble of 100 decision trees. The choice of n_estimators is an important parameter that can affect the performance of the algorithm, with larger values typically resulting in better efficiency, while also raising the price of calculation. To determine the optimal value of n_estimators for a given dataset and problem, various techniques for parameter optimization can be used, as discussed in the previous answer. The optimization process involves finding the value of n_estimators that minimizes the loss function, which is typically evaluated using cross-validation or another suitable evaluation metric.

2) Maximum depth of the trees (max_depth): max_depth is an important parameter in determining the performance of the ensemble. It dictates the depth of each individual decision tree within the Bagging-GBDT algorithm. By setting it at a specific value, such as 5, we ensure that all trees in the group operate independently with a maximum depth limit of 5. The choice of max_depth is crucial as it can impact the algorithm's performance. Larger values may result in overfitting, while smaller values may lead to underfitting. Finding the optimal value involves minimizing the loss function through mathematical calculations and experimentation. To determine the optimal value of max_depth for a given dataset and problem, various techniques for parameter optimization can be used, as discussed in the previous answer. The optimization process involves finding the value of max_depththat minimizes the loss function, which is typically evaluated using cross-validation or another suitable evaluation metric.

3) Learning rate (learning_rate): This parameter determines the step size at which the algorithm updates the weights of the trees. The mathematical expression for optimizing this parameter involves finding the value of learning_rate that minimizes the loss function. The mathematical expression for the value of the hyperparameter, learning_rate, is all that determines the learning rate. The pace of learning is a crucial hyperparameter that can have a substantial effect on execution of the Bagging-GBDT algorithm. Faster convergence during training may be the result of a quicker learning rate, but might potentially result in overfitting. On the other hand, superior generalisation performance can be obtained with a lower learning rate, but may require more iterations during training. The update equation for the weights of the trees in the Bagging-GBDT algorithm can be expressed as shown in Eq. 14.

$$w_{i,j} \leftarrow w_{i,j} - learning_{rate} * \frac{\partial \mathrm{L}(y_i, f_i)}{\partial f_i} \tag{14}$$

Where $w_{i,j}$ is the weight of the t^{th} tree for the i^{th} training instance, L is the loss function, y_i is the target value for the i^{th} training instance, f_i is the ensemble's anticipated outcome, and $\frac{\partial L(y_i,f_i)}{\partial f_i}$ is the loss function's gradient with regard to the anticipated result. The choice of *learning_rate* can be critical for achieving good performance in Bagging-GBDT. The algorithm may converge too rapidly and overfit to the training data if the learning rate is high enough. The approach may need too many iterations to converge if the learning rate is insufficient, resulting in a longer training time. To determine the optimal value of *learning_rate* for a given dataset and problem, various techniques for hyper parameter optimization can be used, as discussed in the previous answers. The optimization process involves finding the value of *learning_rate* that minimizes the loss function,which is typically evaluated using cross-validation or another suitable evaluation metric.

4) Subsample ratio (subsample): The parameter that determines the fraction of data used to train each tree is important in optimizing the Bagging-GBDT method. This parameter, called "subsample," can significantly impact training times and generalization performance. By finding the right value for subsample, we can minimize loss and achieve faster training times with improved generalization performance. However, choosing a higher subsample ratio increases the risk of overfitting. On the other hand, a lower subsample ratio can result in slower training times, but may also decrease the risk of overfitting. The subsample ratio determines the fraction of the training instances that are randomly selected for each iteration of the Bagging-GBDT algorithm. For example, if the subsample ratio is set to 0.8, then 80% of the training instances will be used for each iteration, and the remaining 20% will be left out. The update equation for the weights of the trees in the Bagging-GBDT algorithm with subsampling can be expressed as shown in Eq. 15.

$$w(i,j) \leftarrow w(i,j) - learning_{rate} * \frac{\partial L(y_i,f_i)}{\partial f_i} \qquad (15)$$

where $w_{i,j}$ represents the weight of the j^{th} the i^{th} training instance tree, L is the loss function, y_i is the desired value for the i^{th} training example, f_i is the ensemble's anticipated result, and $\frac{\partial L(y_i,f_i)}{\partial f_i}$ is the loss function's gradient with regard to the anticipated result. The choice of subsample can be critical for achieving good performance in Bagging-GBDT If the subsample Ratio is too high the algorithm may overfit to the training data. If the subsample ratio is too low, the algorithm may not have enough data to learn from, resulting in poor performance. Hyperparameter optimization is a process used to find the optimal subsample value for a dataset and problem. It involves minimizing the loss function using cross-validation or other metrics. For Bagging-GBDT parameters, techniques like Bayesian optimization, random search, and grid search can be used These methods involve defining a search space, evaluating model performance, and selecting parameter values that minimize the loss function. However, finding the global minimum of the loss function can be challenging for high-dimensional parameter spaces.

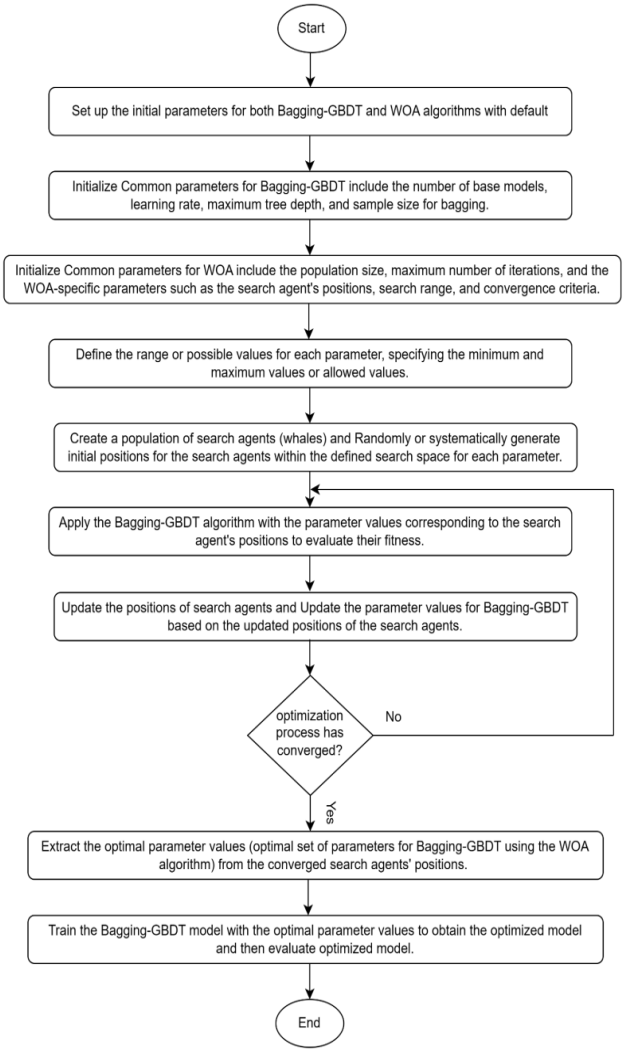

Fig. 1: Flowchart representation of finding optimal parameters for Bagging-GBDT using WOA

.

4. PROPOSED SYSTEM MODEL: BAGGING -WOA-GBDT

Parameter optimization is an essential aspect of designing a Bagging-GBDT classifier model,as it determines the efficacy of the model's prognosticative ability. Among the optimization techniques, the WOA is a powerful algorithm that can be applied to find the optimal parameters for Bagging GBDT models. The WOA algorithm aims to optimize two types of parameters of Bagging-GBDT models: hyperparameters and model parameters. The number of trees, subsample ratio, learning rate, and maximum level of the decision trees are hyperparameters. On the other hand, the design parameters include weights of decision trees and residuals. The optimization of hyperparameters involves selecting the best set of values that can provide the most accurate and robust model. Using the WOA algorithm, the

hyperparameters of Bagging-GBDT models can be optimized by iteratively updating the candidate solutions based on the model's effectiveness. This procedure seeks to reduce overfitting and increase the model's accuracy. The model parameters are optimized by adjusting the weights of decision trees and residuals. The weights of decision trees are updated using the WOA algorithm in a way that emphasizes the best performing trees, while the residuals are adjusted to account for the misclassified instances. To optimize parameters using WOA, Bagging-GBDT models are trained on a dataset and cross-validation is utilized to evaluate how well they function. The most effective models are then used as a candidate solution to the optimization process, and their hyperparameters and model parameters are adjusted iteratively using the WOA algorithm. The number of trees (n_estimators), the maximum depth of every tree (max_depth), the learning rate (learning_rate), and the size of the subsample for each tree (subsample) are among the variables of the hyperparameters that are intended to be optimized. Then define the mean squared error (MSE) between the anticipated values and the actual target values is the objective function that must be optimized. Among the actual target values and the prognosticated values, the mean squared error (MSE) is defined as shown in Eq 16:

$$\text{MSE} = \frac{1}{n} * \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (16)$$

where n is the dataset's sample count, y_i represents the actual goal value for the i^{th} sample, and $\hat{y}$ is the estimated value for that sample. The MSE calculates the average squared variation between the prognosticated and true values, where a lower MSE shows that the anticipated and real values match more closely together. The WOA algorithm starts by initializing a population of whales, where each whale represents a set of hyper parameters. The position vector of each whale represents the current set of hyper parameters, and the velocity vector represents the change in hyper parameters. A range of values for each hyper parameter defines the search space. The fitness of each whale is calculated by applying the bagging GBDT algorithm to the training data with the set of hyper parameters represented by the whale's position vector. The resulting MSE is used as the fitness value for the whale. The whale with the lowest MSE is set as the global best whale. The WOA update rules are used to update the position and velocity of each whale in the population. Using a random integer and the distance to the next whale, each whale's position and velocity are updated.

Algorithm 1: Bagging-WOA-GBDT

1: Initialize the WOA parameters:
max_iter, n_pop, a, c_max, c_min, l
2: Initialize the bagging GBDT parameters:
n_trees, max_depth, learning_rate
3: Initialize the search agent's positions
4: Create a list of n_pop search agents
5: For each search agent,create a list of n_trees
Decision trees with a max_depth
6: Initialize the parameter c for each search agent to 0
7: Start the main loop for the WOA algorithm:
8: For each iteration t in range (max_iter):
9: Update the value of the parameter c:
10: c = c_max - t * ((c_max - c_min) / max_iter)
11: For each search agent i in range(n_pop):
12: For each tree j in range(n_trees):
13: Generate three random vectors: r1, r2 and r3
14: Select a random tree x_rand from the population
15: Compute the a_dist based on the parameter a:
a_dist = a* abs(r1[j] * x_rand.tree_)
16: Compute c_dist: c_dist = c * (r2[j] *
x_rand.tree_ - pop[i]['trees'][j].tree_)
17: Compute l_dist based on the parameter l:
l_dist = l *(r3[j] - 0.5)
18: Compute delta_x: delta_x = a_dist * np.sin
(2 * np.pi *l_dist) + c_dist
19: Create a new decision tree new_tree with a
Maximum depth of max_depth
20: Update weights of the decision tree:
new_tree.tree_ = pop[i]['trees'][j].tree_ + delta_x
21: Replace old tree with new in search agent's list
of trees: pop[i]['trees'][j] = new_tree
22: Update value of the parameter c for the search agent:
pop[i]['c'] = c
23: Evaluate the fitness of each search agent:
24: Create a list of n_pop fitness values
25: For each search agent i in range(n_pop):
26: Compute y_pred: y_pred += tree.prognosticate(X)
27: Compute the fitness value: fitness[i] =
mean_squared_error(y, y_pred)
28: Sort the search agents based on their fitness:
29: Sort the search agents using argsort() function
and store them in a new list sorted_pop.
30: Select the best search agent:
31: Select search agent with lowest fitness value:
best_agent = sorted_pop[0]
32: Update the bagging GBDT model:
33: Create a new list of decision trees with a maximum
depth of max_depth
34: For each tree j in range(n_trees):
35: Compute weight of each decision tree by averaging
the weights of the corresponding trees:
36: Compute the weight of the j^{th} tree: $w_j = 0$
37: For each search agent i in range (n_pop):
38: Compute the weight of the corresponding tree in the
i^{th} search agent: $w_{(i,j)}$= abs(best_agent['trees'][j].tree_
- pop[i]['trees'][j].tree_)
39: Add the weight to the total weight of the j-th tree:
w_j+= $w_{(i,j)}$
40: Normalize the weight of the j^{th} tree: $w_j = w_j / \text{sum}(w_j)$
41: Create a new_tree with a maximum depth of

max_depth and A weight of w_j

42: Add the new tree to the list of decision trees: new_trees.append(new_tree)

43: Replace the old list of decision trees with the new one in the bagging GBDT model: model.estimators_ = new_trees

44: Return the bagging GBDT model with optimized parameters.

The search operator is used to ensure that the position vector of each whale stays within the search space. After applying the WOA update rules and the search operator to each whale, then evaluate the fitness of the new population. If a whale's fitness is lower than its previous fitness and update its position vector and fitness. This process can be repeated until convergence criteria are met, for instance, a minimal increment in the objective function or a maximum number of iterations. The flowchart representation of finding optimal parameters for Bagging-GBDT using WOA is shown in Fig. 1 and Algorithm 1 presents pseudo code.

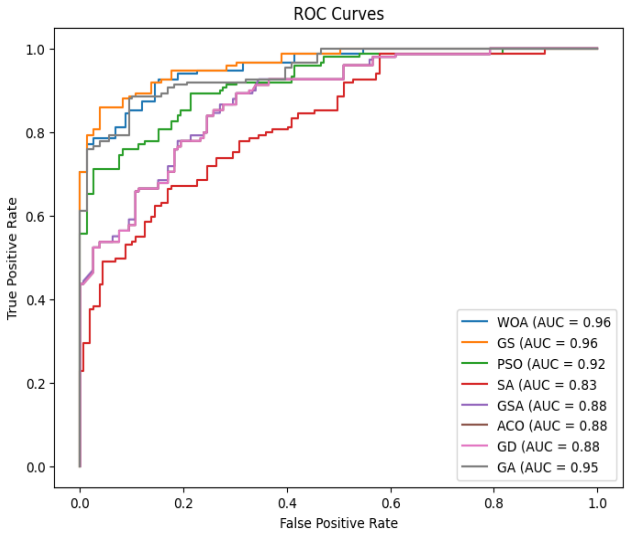

Fig. 2: ROC Curves

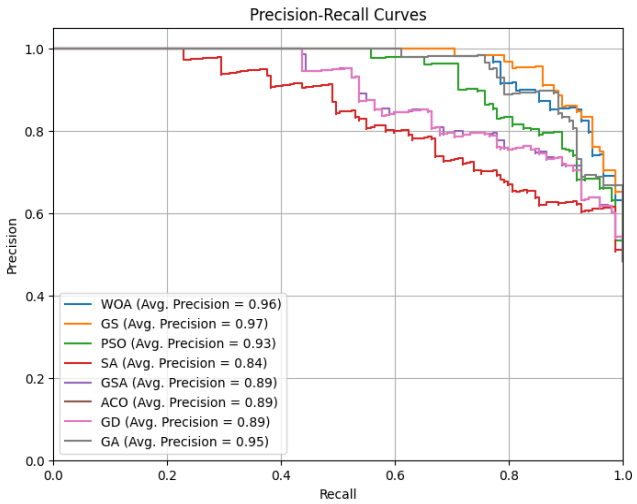

Fig. 3: Precision-Recall Curves

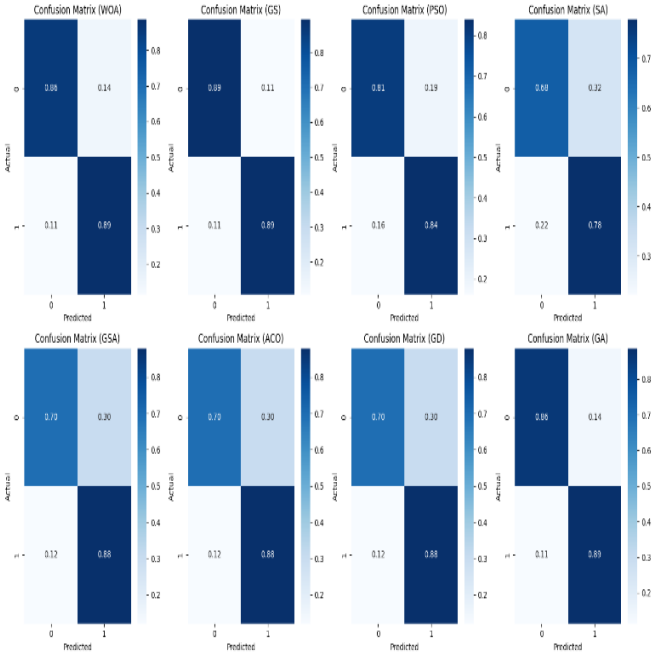

Fig. 4: Confusion Matrix

5. PARAMETER SETTINGS AND EXPERIMENTAL OUTCOMES

A. Parameter Settings

Bagging-GBDT is a powerful algorithm for prognosticating outcomes, but it requires careful selection of six key parameters to be able to achieve the improved results. The prognostication model's consistency and precision are significantly influenced by these variables, and finding optimal values for each of them is a critical challenge.

The first parameter, M, refers to how many decision trees the algorithm produced during the growth process. Increasing M can improve the accuracy of the model on training sets, however, picking the wrong value could result in overfitting. This parameter is selected through testing. The second parameter, MD, specifies each decision tree's maximum depth. If MD is too big, it can lengthen the duration of each tree's training, while if it is too little, the resultant algorithm may be ill-fitting. Finding an appropriate MD value is critical to ensure a well-fitting prognostication model. The third parameter, MS, provides the fewest samples required to split an internal node. This value is determined based on whether it is an int or a float, and (MS $*$ MD). Selecting the appropriate MS value is critical for the accurate splitting of internal nodes. The fourth parameter, ML, the least number of samples that can be present at each leaf node. Only if each of the right and left offshoots of a dividing point have at least ML training samples remaining will it be taken into consideration. This parameter is particularly important in regression, where it ensures a smooth fit. The fifth parameter, m, specifies the number of sub-datasets generated through sampling for bagging. The effect of this parameter on the

prognostication model's precision can be significant, so selecting an appropriate m value is essential for obtaining the most accurate results. Finally, the sixth parameter, l, refers to the learning rate used to manage one decision tree's contribution to the model. This parameter is a regularization method that prevents overfitting of the Bagging-GBDT algorithm.

B. Determination of Hyperparameters for Bagging-WOA-GBDT Algorithm

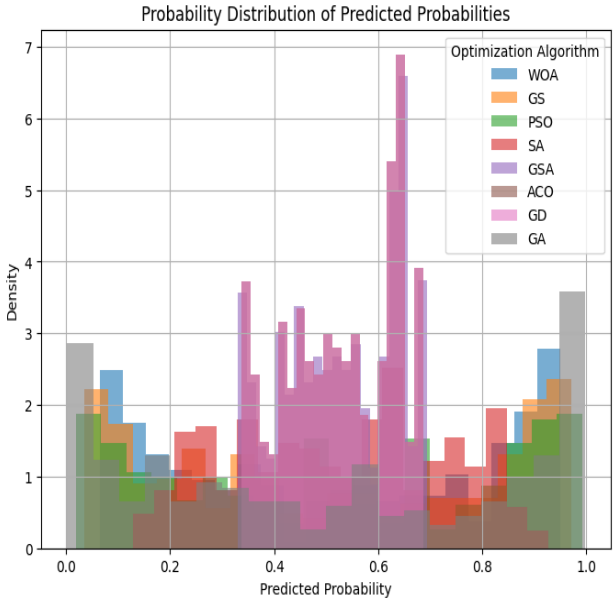

Fig. 5: Probability Distribution of Predicted Probabilities

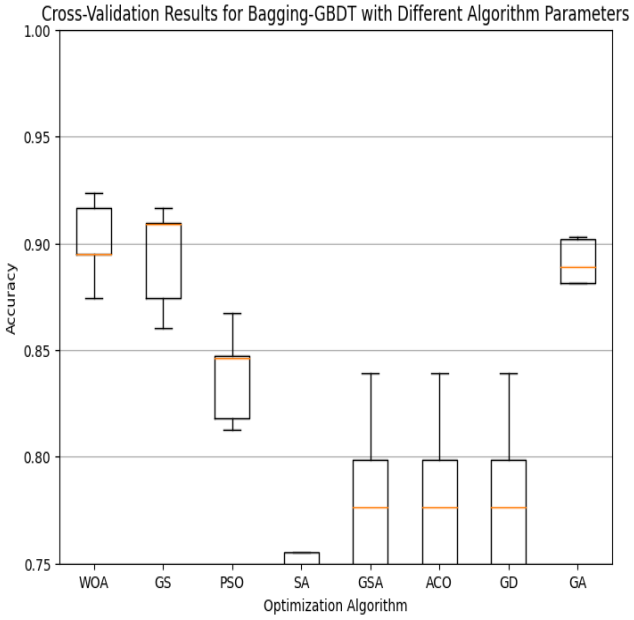

Fig. 6: Cross-Validation Results for Bagging-GBDT with Different Algorithm Parameters.

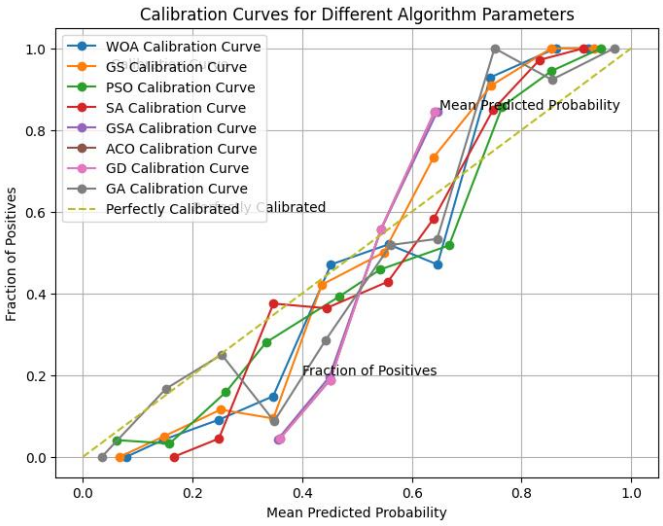

Fig. 7: Calibration Curves for Different Algorithm Parameters

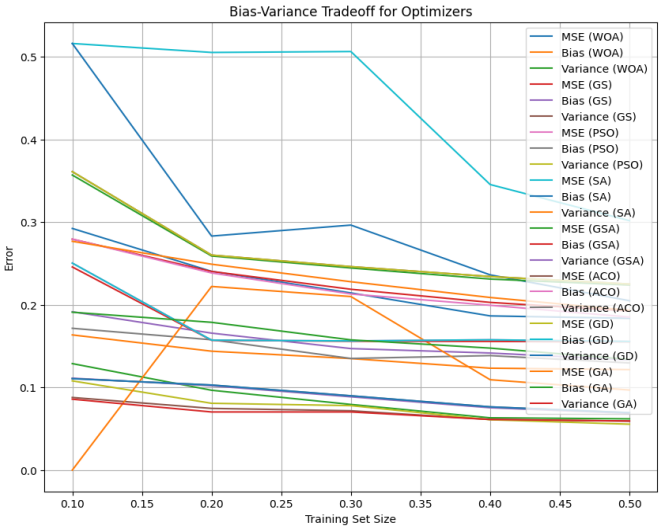

Fig. 8: Bias-Variance Tradeoff for Optimizers

This paper proposes bagging as a performance improvement method for GBDT models, combining multiple models trained on different data sets. Hyperparameters like tree number, depth, and learning rate are optimized using a Latin hypercube sampling method. The model's fitness is evaluated using validation data and relevant hyperparameters. The WOA algorithm is applied to update the position of whales, enlarging the search area, and achieving an optimal response while allowing exploratory search.

C. Experimental Outcomes

The proposed method has been evaluated on Cardiac disease datasets. A detailed comparison is made between the performance of the method and other optimization algorithms commonly used for GBDT parameter adjustment, like grid search, random search. The outcome of the experiment demonstrate that suggested approach

works better the baseline techniques for prognosticative accuracy and generalization ability. The Table 1 provided in the research paper demonstrates assessment measures for various machine learning models. These metrics consist of accuracy, F1 score, recall, and precision. Precision is the proportion of accurate positive prognostications among all positive forecasts. Recall calculates the proportion of correctly foreseen positive outcomes among all instances of positive conduct. The F1-score represents the harmonic mean of recall and accuracy, and it generates one score by combining the two measures. Measured as the percentage of accurate prognostications among all prognostications, accuracy. The Measured as the proportion of accurate prognostications among all prognostications and accuracy. The results of our cardiovascular prediction study reveal valuable insights into the performance of different optimization algorithms. In terms of accuracy scores, WOA stands out as the most efficient algorithm, achieving an impressive accuracy of 0.8929. The GA follows closely behind, with an admirable accuracy score of 0.8734. The GS also delivers strong performance, boasting an accuracy score of 0.8701. On the other hand, PSO exhibits a respectable performance with an accuracy of 0.8214. In contrast, SA follows with the lowest accuracy score of 0.7273. GSA, ACO, and Gradient Descent (GD) all have an accuracy score of 0.7857, indicating similar performance. Going deeper into the results, the confusion matrices as shown in Fig. 4 provide insight into the predictive ability of the algorithm.

For example, the confusion matrix of WOA exhibits a balanced distribution of true positives and true negatives, highlighting the skill of the algorithm in terms of sensitivity and specificity while the confusion matrix of GS exhibits effective prediction with few false positives and false negatives. In contrast, PSO, SA, GSA, ACO, and GD exhibit different misclassifications, which are reflected in their confusion matrices. The classification report provides a detailed analysis of the accuracy, recall, and F1 scores of each algorithm. WOA and GA consistently exhibit high accuracy, indicating good and accurate prediction capabilities. WOA exhibits a balanced performance in terms of accuracy and recall. PSO, while exhibiting respectable accuracy overall, exhibits some imbalances the analysis of various performance metrics and assessment criteria famous precious insights into the effectiveness of optimization algorithms for coronary heart disorder prediction the use of Bagging-WOA-GBDT. ROC and PR curves as shown Fig. 2 and 3, offer a comprehensive understanding of the exchange off between real nice fee and false effective rate and precision and recall, respectively. From our results, WOA reveals the very best region beneath the ROC curve (AUC-ROC) and vicinity underneath the PR curve (AUC-PR).

This indicates that WOA presents a most reliable stability among sensitivity and specificity, as well as precision and recollect, making it the best preference for applications that require a well-rounded predictive version. On the alternative hand, SA suggests the lowest AUC-ROC and AUC-PR values, suggesting that it may now not be appropriate for eventualities where specific class is crucial. Analysing the opportunity distribution of predicted probabilities as shown in Fig. 5, throughout distinctive algorithms reveals precious records about their prediction self-assurance. While WOA and GA always generate properly separated and extra concentrated distributions of possibilities, SA produces much less defined distributions. This indicates that GS and GA's predictions are greater confident and distinguishable, making them properly-proper for applications where self-belief in predictions is crucial. The cross-validation as shown in Fig. 6, effects provide insights into the generalization abilities of the Bagging-WOA-GBDT version with numerous algorithm parameters. GS and GA consistently yield the highest move validation rankings, reinforcing their robustness and reliability throughout distinct folds and datasets. In comparison, SA indicates lower cross-validation ratings, indicating a discounted potential to generalize to unseen records efficiently. These effects emphasize the significance of selecting optimization algorithms that promote strong version generalization. Calibration curves as shown in Fig. 7, provide an assessment of the version's predictive reliability. In our evaluation, WOA and GA exhibit well-calibrated models, as their expected possibilities align intently with the real consequences. On the alternative hand, SA's calibration curve shows deviations from the appropriate line, suggesting that its predictions may require recalibration to beautify their reliability. The bias-variance trade-off analysis as shown in Fig. 8, offers a vital attitude on version complexity and the effect of different optimization algorithms. WOA, with its high accuracy and balanced precision and recall, demonstrates an most effective model complexity that strikes a stability between underfitting and overfitting. On the opposite, SA, despite its low accuracy, well-known shows surprisingly low bias, indicating that it may no longer be overfitting the records. However, it suffers from excessive variance, which may additionally result in inconsistent overall performance on special datasets.

6. CONCLUSION AND FUTURE OUTLINE

In conclusion, the cardiac condition is a serious health concern and one of the leading causes of death globally. The integration of IoT and AI in healthcare services has revolutionized the healthcare industry and improved patient outcomes. The Bagging-Fuzzy-GBDT classifier is a popular ensemble learning method that was utilized to prognosticate Cardiac disease, but it may not always be the most suitable technique for a given problem. Using a mix of ensemble learning methods, this study offered a novel method for prognosticating cardiac disease. WOA and Bagging-GBDT classifiers. The proposed method

enhances data representation by capturing uncertainty and vagueness in the data and generating multiple random sub-samples from the original dataset. The Bagging-WOA-GBDT classifier is then utilized to build an accurate prognostication model. The outcomes of the study on a publicly available Cardiac disease dataset Show that the suggested strategy performs better than the traditional classifier techniques, providing a more robust and accurate prognostication of Cardiac disease. Our analysis of ROC curves, PR curves, chance distributions, cross-validation consequences, calibration curves, and the bia-variance tradeoff underscores the significance of choosing the proper optimization algorithm for coronary heart disorder prediction the use of Bagging-GBDT. WOA and GA consistently outperform different algorithms in more than one element, making them suitable for numerous healthcare packages. SA, whilst showing a few weaknesses in our evaluation, might also nonetheless have capability in particular eventualities if calibrated and optimized efficaciously. These insights are pivotal for researchers and practitioners aiming to maximize the performance of predictive fashions for healthcare programs. The proposed method can be valuable equipment for healthcare practitioners in diagnosing and preventing Cardiac disease.

In order to further improve the method, it would be valuable to conduct research on larger and more diverse datasets. Additionally, comparing its performance with other ensemble learning techniques could provide valuable insights. Expanding the application of this proposed method beyond Cardiac disease to other healthcare fields like cancer prognostication or diagnosis holds potential as well. Furthermore, exploring alternative-s to the current WOA algorithm such as Bagging-WOA-GBDT or Bagging-enhanced WOA-GBDT might result in improved optimization and enhanced performance of the GBDT model. In summary, combining AI and IoT in healthcare has the potential to enhance patient outcomes while reducing costs, and this study presents a promising approach for prognosticating Cardiac disease.

References

1. Azmi, J., Arif, M., Nafis, M. T., Alam, M. A., Tanweer, S., & Wang, G. (2022). A systematic review on machine learning approaches for cardiovascular disease prediction using medical big data. *Medical Engineering & Physics*, 103825.
2. Kibria, H. B., & Matin, A. (2022). The severity prediction of the binary and multi-class cardiovascular disease– A machine learning-based fusion approach. *Computational Biology and Chemistry*, *98*, 107672.
3. Kishor, A., & Chakraborty, C. (2022). AI and internet of things based healthcare 4.0 monitoring system. *Wireless personal communications*, *127*(2), 1615-1631.
4. Ashfaq, Z., Rafay, A., Mumtaz, R., Zaidi, S. M. H., Saleem, H., Zaidi, S. A. R.,. & Haque, A. (2022). A review of enabling technologies for Internet of Medical Things IoMT Ecosystem. *Ain Shams Engineering Journal*, *13*(4), 101660.
5. Kibria, H. B., & Matin, A. (2022). The severity prediction of the binary and multi-class cardiovascular disease– A machine learning-based fusion approach. *Computational Biology and Chemistry*, *98*, 107672.
6. Ajagbe, S. A., Awotunde, J. B., Adesina, A. O., Achimugu, P., & Kumar, T. A. (2022). Internet of Medical Things IoMT: Applications, Challenges, and Prospects in a Data-Driven Technology. *Intelligent Healthcare: Infrastructure, Algorithms and Management*, 299-319.
7. El Hechi, M. W., Eddine, S. A. N., Maurer, L. R., &Kaafarani, H. M. (2021). Leveraging interpretable machine learning algorithms to predict postoperative patient outcomes on mobile devices. *Surgery*, *169*(4), 750-754.
8. Huang, Y. P., Vadloori, S., Chu, H. C., Kang, E. Y. C., Wu, W. C., Kusaka, S., & Fukushima, Y. (2020). Deep learning models for automated diagnosis of retinopathy of prematurity in preterm infants. *Electronics*, *9*(9), 1444.
9. Zhu, H., Wang, G., Zheng, J., Zhu, H., Huang, J., Luo, E., ... & He, X. (2022). Preoperative prediction for lymph node metastasis in early gastric cancer by interpretable machine learning models: A multicenter study. *Surgery*, *171*(6), 1543-1551.
10. Elgedawy, M. N. (2017). Prediction of breast cancer using random forest, support vector machines and naïve Bayes. *International Journal of Engineering and Computer Science*, *6*(1), 19884-19889.
11. Feres, M., Louzoun, Y., Haber, S., Faveri, M., Figueiredo, L. C., & Levin, L. (2018). Support vector machine-based differentiation between aggressive and chronic periodontitis using microbial profiles. *International dental journal*, *68*(1), 39-46.
12. Lucas, M., Jansen, I., van Leeuwen, T. G., Oddens, J. R., de Bruin, D. M., &Marquering, H. A. (2022). Deep learning–based recurrence prediction in patients with non–muscle-invasive bladder cancer. *European Urology Focus*, *8*(1), 165-172.
13. Raja, M. S., Anurag, M., Reddy, C. P., &Sirisala, N. R. (2021, January). Machine learning based Heart disease prediction system. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE.

14. Islam, J., & Zhang, Y. (2017). A novel deep learning based multi-class classification method for Alzheimer's disease detection using brain MRI data. In *Brain Informatics: International Conference, BI 2017, Beijing, China, November 16-18, 2017, Proceedings* (pp. 213-222). Springer International Publishing.
15. Tarawneh, O., Otair, M., Husni, M., Abuaddous, H. Y., Tarawneh, M., &Almomani, M. A. (2022). Breast cancer classification using decision tree algorithms. *International Journal of Advanced Computer Science and Applications*, *13*(4).
16. Patil, L., Sirsat, A., Kamble, D., & Pawar, M. Y. (2017). Lung cancer detection using decision tree algorithm. *International Research Journal of Engineering and Technology*, *4*(2), 1885-1888.
17. Sisodia, D., & Sisodia, D. S. (2018). Prediction of diabetes using classification algorithms. *Procedia computer science*, *132*, 1578-1585.
18. Hosseinpour, M., Ghaemi, S., Khanmohammadi, S., &Daneshvar, S. (2022). A hybrid high-order type-2 FCM improved random forest classification method for breast cancer risk assessment. *Applied Mathematics and Computation*, *424*, 127038.
19. Gorji, H. T., &Haddadnia, J. (2015). A novel method for early diagnosis of Alzheimer's disease based on pseudo Zernike moment from structural MRI. *Neuroscience*, *305*, 361-371.
20. Malik, S., Harous, S., & El-Sayed, H. (2021). Comparative analysis of machine learning algorithms for early prediction of diabetes mellitus in women. In *Modelling and Implementation of Complex Systems: Proceedings of the 6th International Symposium, MISC 2020, Batna, Algeria, October 24-26, 2020 6* (pp. 95-106). Springer International Publishing.
21. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., &Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, *542*(7639), 115-118.
22. Qummar, S., Khan, F. G., Shah, S., Khan, A., Shamshirband, S., Rehman, Z. U., ... &Jadoon, W. (2019). A deep learning ensemble approach for diabetic retinopathy detection. *Ieee Access*, *7*, 150530-150539.
23. Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzel, R. (2015). Learning to diagnose with LSTM recurrent neural networks. *arXiv preprint arXiv:1511.03677*.
24. Anggoro, D. A., &Kurnia, N. D. (2020). Comparison of accuracy level of support vector machine (SVM) and K-nearest neighbors (KNN) algorithms in predicting Heart disease. *International Journal*, *8*(5), 1689-1694.
25. Tayefi, M., Tajfard, M., Saffar, S., Hanachi, P., Amirabadizadeh, A. R., Esmaeily, H., ... & Ghayour-Mobarhan, M. (2017). hs-CRP is strongly associated with coronary Heart disease (CHD): A data mining approach using decision tree algorithm. *Computer methods and programs in biomedicine*, *141*, 105-109.
26. Jeyaranjani, J., Rajkumar, T. D., & Kumar, T. A. (2021). WITHDRAWN: Coronary Heart disease diagnosis using the efficient ANN model.
27. Yuan, X., Chen, J., Zhang, K., Wu, Y., & Yang, T. (2021). A stable AI-based binary and multiple class Heart disease prediction model for IoMT. *IEEE Transactions on Industrial Informatics*, *18*(3), 2032-2040.
28. Mary, M. M. A. (2020). Heart disease prediction using machine learning techniques: A survey. *International Journal For Research In Applied Science And Engineering Technology*, *8*(10), 441-447.
29. Jiang, H., Mao, H., Lu, H., Lin, P., Garry, W., Lu, H.,.& Chen, X. (2021). Machine learning-based models to support decision-making in emergency department triage for patients with suspected cardiovascular disease. *International Journal of Medical Informatics*, *145*, 104326.
30. Yuan, X., Chen, J., Zhang, K., Wu, Y., & Yang, T. (2021). A stable AI-based binary and multiple class Heart disease prediction model for IoMT. *IEEE Transactions on Industrial Informatics*, *18*(3), 2032-2040.
31. Tabrizchi, H., Tabrizchi, M., &Tabrizchi, H. (2020). Breast cancer diagnosis using a multi-verse optimizer-based gradient boosting decision tree. *SN Applied Sciences*, *2*, 1-19.
32. Yu, Z., Wang, Z., Zeng, F., Song, P., Baffour, B. A., Wang, P., ... & Li, L. (2021). Volcanic lithology identification based on parameter-optimized GBDT algorithm: A case study in the Jilin Oilfield, Songliao Basin, NE China. *Journal of Applied Geophysics*, *194*, 104443.
33. Yuan, X., Chen, J., Zhang, K., Wu, Y., & Yang, T. (2021). A stable AI-based binary and multiple class Heart disease prediction model for IoMT. *IEEE Transactions on Industrial Informatics*, *18*(3), 2032-2040.
34. Bisht, K., & Kumar, A. (2023). A method for fuzzy time series forecasting based on interval index number and membership value using fuzzy c-means clustering. *Evolutionary Intelligence*, *16*(1), 285-297.
35. Louk, M. H. L., & Tama, B. A. (2023). Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. *Expert Systems with Applications*, *213*, 119030.
36. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51-67.

37. Saidala, R. K., Devarakonda, N., Rao, T. B., Syam, J., & Kumar, S. (2023). A Novel Optimum Clustering Method Using Variant of NOA. In *Computational Intelligence in Medical Decision Making and Diagnosis* (pp. 221-237). CRC Press.

.

Mr. JAVVAJI VENKATARAO, a well-known Research scholar and Pursuing (PhD) in SRM UNIVERSITY, CHENNAI. His area of Interest includes Machine Learning, AI, Soft computing, data mining, cloud computing and IoT.

Dr. V. Deeban Chakravarth, Associate Professor, SRM Institute of Science & Technology, Chennai, a well knowledge Research Guide. His area of Interests Networking, Cloud Computing & IOT.