Integrating Nursing Perspectives into Cardiovascular Disease Prediction Using Hybrid Classification Models and Metaheuristic Optimization with DOA, ARO, and PO Algorithms

Pingping Yu* and Qin Shen
School of Nursing, Zhejiang University of Traditional Chinese Medicine, Hangzhou 311200, Zhejiang, China
E-mail: 13342097867@163.com
*Corresponding author

Keywords: cardiovascular disease, classification, eXtreme gradient boosting, dingo optimization algorithm

This study proposes a cardiovascular disease (CVD) prediction model based on the XGBoost classifier, enhanced through metaheuristic-based hyperparameter optimization. A real-world dataset comprising 70,000 patient records including attributes such as age, blood pressure, cholesterol levels, and lifestyle factors was used for model development. Three recent metaheuristic algorithms Dingo Optimization Algorithm (DOA), Artificial Rabbits Optimization (ARO), and Political Optimizer (PO) were employed to optimize critical XGBoost hyperparameters such as learning rate, maximum depth, and number of estimators. The dataset was divided using a stratified 80:20 train-test split. Performance was evaluated using accuracy, precision, recall, and F1-score metrics. Among all configurations, the XGBoost model optimized via ARO (XGAR) achieved the best performance, with a training accuracy of 0.995, a testing accuracy of 0.678, and an overall average accuracy of 0.900. These findings highlight the effectiveness of combining XGBoost with metaheuristic optimization for early CVD risk prediction and potential integration into preventive healthcare frameworks.

Povzetek: Obravnavana je kvaliteta zgodnjega napovedovanja srčno-žilnih bolezni z običajnimi modeli in brez optimalnega nastavljanja hiperparametrov. Predlagan je XGBoost, optimiziran z metahevristikami DOA, ARO in PO, ki na 70.000 kliničnih zapisih izboljša klasifikacijo.

1 Introduction

The heart is a vital organ in living beings and participates in most of the body's activities. Human health greatly relies on a healthy heart; any form of heart disease may affect other vital organs, such as the brain and kidneys. It is incredible to note that the foremost cause of death globally involves cardiovascular diseases [1]. The World Health Organization (WHO) indicates that heart-related diseases currently are the foremost cause of death in both developed and developing countries, affecting both sexes with approximate equality [2], [3]. Estimates have shown that about 76% of all deaths will be due to non-communicable diseases across the world by the year 2030 [4]. Symptoms accompanying heart disease, such as shortness of breath, pain in the neck, jaw ache, and chest pain, have incidence rates showing a rising trend [5]. It may cause problems such as heart failure, stroke, and coronary artery disease [6]. Although it has been the foremost cause of death globally for the last ten years, it is considered quite curable and preventable [7]. Early identification and diagnosis have a direct relationship with effective treatment and prevention of problems [8], [9].

Various tests, such as cardiac angiography, stress testing, and electrocardiograms, are expensive and not always available but are required for early diagnosis [10], [11]. Early diagnosis treatment methods are cheaper compared to more traditional medical interventions [12]. The heart is a crucial body organ responsible for circulating blood throughout the body. It consists of three separate layers: the inner endocardium, the middle myocardium, and the outside pericardium, each of which has specific responsibilities it undertakes [13]. The myocardium, the thickest layer of the heart, is necessary to maintain human life, whereas the endocardium borders the atrial and ventricular chambers [14]. The pericardium protects the heart by acting as a connective coating [15]. Heart muscle, congenital heart, heart valve, and coronary artery diseases are just a few of the many disorders that are included within the umbrella term Cardiovascular diseases [16], [17], [18], [19]. Of the disorders listed above, coronary artery disease is the most common and manifests itself mostly as angina pectoris and myocardial infarction [20]. One of the main causes of this illness is atherosclerosis, a disorder marked by arterial dysfunction, which is also linked to thrombosis, spasm, and coronary aneurysm [21], [22].

The blood dynamics, artery wall geometry, and flow mechanics are closely related to cardiovascular disorders [23], [24]. Because collagen fibers become more rigid with age and replace the flexibility of vessel walls, aging is a major factor in arterial congestion. The buildup of fat and other particles during this transitional phase facilitates heart artery obstruction, which in turn contributes to the beginning of cardiovascular illnesses [25]. Cardiovascular illnesses accounted for 10% of all documented fatalities at the beginning of the 20th century. Heart disease development and severity are influenced by several variables. These elements fall into six categories: 1. Biological and environmental factors (such as age, gender, and stress); 2. Life habits (such as exercise, proper diet, and adherence to hygiene norms); 3. Risk factors (obesity, smoking, and high blood cholesterol). Four fundamental conditions (chronic obstructive pulmonary illness, chronic renal disease infection, anemia, arthritis), five psychological and mental aspects (emotion, coping mechanisms, self-efficacy), and six social aspects (social class of having a family and receiving their assistance) are discussed. These elements have been identified as risk markers for heart disease prognosis. Congenital heart disease, hypertension, heart failure, cardiomyopathy, pulmonary stenosis, rheumatic heart disease, and coronary artery disease are some of the most prevalent types of heart disorders. Other body systems may be affected when the heart's function is impaired when any of these disorders are present. Thus, it seems that early detection of this illness is essential [26], [27].

A few studies that are relevant to the study issue are discussed. A model was presented to enhance the prediction of cardiac illnesses in research by Khan et al. [28] that focused on the hybridization of ML frameworks. The research emphasizes how useful it is to use hybridization strategies and ML frameworks to increase prediction accuracy. A hybrid model for detecting cardiovascular patients using ML techniques was presented by Naidu et al. [29]. They used PSO and GA frameworks for feature extraction in their methodology. The study will evaluate how these hybridization methods work in this setting. Fuzzy logic and mixed machine learning schemes can be applied to predict cardiac disease, according to Kaur et al. [30]. Findings displayed how helpful their recommended approach was in finding the risk of heart disease. DeGroat et al. [31] integrated conventional statistical methods with AI/ML techniques to recognize significant biomarkers of CVD. Major biomarkers pointed out with the help of studies, statistical tests, and ML classifiers achieved the best performance of 96% accuracy in illness prediction. Clinical records were used to carry out cross-validation, and biomarkers displayed potential for CVD early identification.

Cardiovascular disease (CVD) continues to be the leading cause of global mortality. Over the past decade, researchers have proposed a variety of machine learning (ML) and hybrid approaches to detect CVD at early stages. Below is a structured overview of recent studies, summarized in Table 1.

Table 1: Summary of recent approaches for CVD prediction

Study	Dataset	Method	Accuracy	Recall	Limitations
Khan et al. [24]	UCI CVD	Hybrid ML (SVM+RF)	94.50%	91.20%	No hyperparameter optimization, generic dataset
Naidu et al. [25]	Cleveland	Hybrid PSO + GA + SVM	91.40%	90.00%	Feature extraction only, suboptimal classifiers
Kaur et al. [26]	Statlog	Fuzzy + ML (NB, DT)	92.20%	87.50%	Low model interpretability, small dataset
DeGroat et al. [27]	Clinical records	Biomarker +Logistic + ML	96.00%	94.80%	No boosting, expensive features
Choubey et al. [28]	Kaggle CVD	Decision Tree	89.00%	85.00%	No hyperparameter tuning
Anooj [29]	UCI	Naïve Bayes + Fuzzy Rules	85.60%	83.00%	High bias, rule base complexity
Dinesh et al. [30]	Local Hospital	CNN-LSTM	93.30%	89.40%	High computational cost, limited explainability

Although it remains a foremost cause of death globally, early detection and risk prediction are necessary to develop prevention strategies. Most previous works presented several machine learning approaches for CVD prediction; however, because of the limitation in most datasets (as they contain no explicit information on the causes of CVD among the participants), this paper presents a new method to enhance CVD prediction using Extreme Gradient Boosting. It utilizes a robust dataset with variables directly related to the participants' causes of CVD, thus making it possible for the scheme to grasp more specific relationships between factors and disease risk. A powerful Ml algorithm, XGBoost, has been used in this paper due to its reputation as one of the most efficient frameworks in solving complex classification problems and handling large data efficiently while performing feature engineering. To enhance the productivity of XGBoost, this paper implements a new hybrid approach. These were implemented with the inclusion of three of the most state-of-the-art metaheuristic frameworks: Dingo Optimization Algorithm (DOA), Artificial Rabbits Optimization (ARO), and Political Optimizer (PO). All these frameworks have been known for their strengths in optimizing machine learning model hyperparameters. Fundamentally, the research develops optimizers that would search for the best combination of hyperparameters, which best fine-tunes the scheme and increases its skill to learn complex patterns within the CVD data by hybridizing XGBoost with each of the metaheuristic frameworks. The paper is expected to contribute significantly by showing the efficiency of using a robust CVD-cause-related dataset, underlining the advantages of using XGBoost for CVD prediction, presenting a new hybrid methodology based on metaheuristics optimization for improving XGBoost performance, and giving useful insights on the best hyperparameter configurations for CVD prediction schemes. While previous studies have explored the integration of metaheuristic algorithms with machine learning models such as XGBoost, the novelty of this research lies in the application of three emerging optimization algorithms Dingo Optimization Algorithm (DOA), Artificial Rabbits Optimization (ARO), and Political Optimizer (PO) to enhance the predictive accuracy of CVD classification. Unlike earlier works that typically utilize conventional optimizers such as Genetic Algorithms or Particle Swarm Optimization, our study evaluates the comparative performance of these newer algorithms in a unified framework and on a robust, health-related dataset. This allows us to identify which optimizer most effectively tunes the XGBoost model for real-world CVD data, thereby offering new insights into the relative strengths of these metaheuristics in a clinical prediction setting.

2 Data collection

The dataset used in this study was obtained from the Kaggle platform under the title “Cardiovascular Disease Dataset”, containing a total of 70,000 records derived from clinical examinations. Each record includes a binary target variable, labeled cardio, where a value of 1 indicates the presence and 0 indicates the absence of cardiovascular disease. The input features consist of 11 attributes categorized into three groups: objective characteristics (age, gender, height, and weight), examination results (systolic and diastolic blood pressure, cholesterol level, and glucose level), and lifestyle-related behavioral factors (smoking, alcohol consumption, and physical activity). The dataset includes 34,979 positive cases (cardio = 1) and 35,021 negative cases (cardio = 0), indicating a nearly balanced distribution of classes. Prior to model training, the data underwent several preprocessing steps. All numerical features were normalized using Min-Max scaling to ensure uniform feature ranges. Categorical variables such as gender, cholesterol, and glucose were label-encoded into numerical form. The dataset was inspected for missing values, and no null entries were found; thus, imputation was not required. Since the data was balanced, no oversampling or undersampling methods were applied. For model evaluation, the dataset was partitioned into training and testing sets using an 80/20 split, with stratified sampling to preserve the class distribution across both subsets. As a result, 56,000 samples were used for training and 14,000 for testing. Figure 1 presents a heatmap that visualizes the correlation between input features and the target variable. The color gradients in the figure help illustrate the strength and direction of associations among the features, serving as an initial step in understanding the structure and interdependencies within the dataset. This exploratory visualization also provides insights into which features may hold stronger predictive potential for the classification task.

The dataset used in this study, titled “Cardiovascular Disease Dataset,” is publicly available through the Kaggle platform and contains anonymized records collected from routine medical examinations. No personal identifiers such as names, addresses, or contact information are included, and all records are encoded using non-traceable numeric identifiers. Therefore, the use of this dataset does not pose any risk to individual privacy. In line with standard research practice for publicly available datasets, no additional ethical approval was required for this study. Data handling and processing adhered strictly to principles of research ethics and data protection.

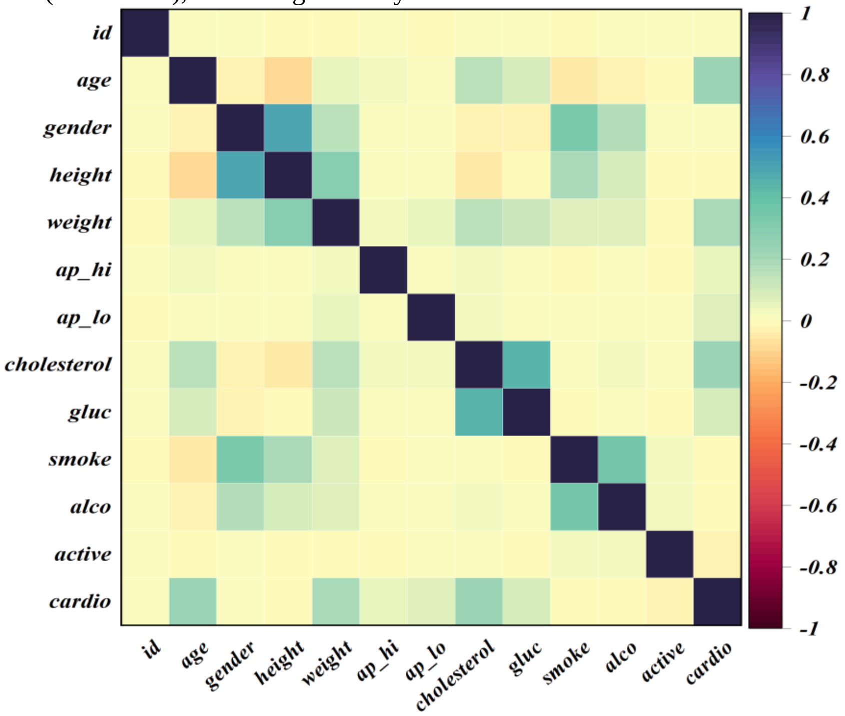

Figure 1: The plot illustrating the correlation between the input and output

3 Classification and metaheuristic frameworks

3.1 eXtreme gradient boosting classification (XGBC)

This section reviews the basic theory of the XGBoost detection algorithm. The XGBoost offers remarkable improvements in three aspects: computational speed, scalability, and generalization performance compared to the traditional GBDT technique. The goal function and optimization method of the XGBoost algorithm are given in [32], accordingly. The goal function of XGBoost is defined as follows by Eq. (1):

$$F_{Obj}(\theta) = L(\theta) + \Omega(\theta), \quad where\ L(\theta) = l(\hat{y_i}, y_i), \quad \Omega(\theta) = \gamma N + \frac{1}{2}\lambda \|w\|^2. \tag{1}$$

XGBoost's objective function is split into two halves, $L(\theta)$ and $\Omega(\theta)$, where θ stands for the many formulaic parameters. The difference between the actual target, y_i, and the expected value, (y_i), is quantified by the differentiable convex loss function $L(\delta)$, It serves to direct the scheme's data matching. Two often-used convex loss functions are logistic loss (Eq. (2)) and mean square loss (Eq. (3)).

$$l(\hat{y_i}, y_i) = y_i \ln\left(1 + e^{-\hat{y_i}}\right) + (1 - y_i)\ln(1 + e^{\hat{y_i}}) \tag{2}$$

$$l(\hat{y_i}, y_i) = (\hat{y_i} - y_i)^2 \tag{3}$$

However, $\eta(\theta)$, a regularization factor penalizes complex schemes. In this case, the learning rate is indicated by γ, which has a maximum value of 0, and N displays the count of leaves in the tree. Multiplying γ by N yields tree pruning, which reduces overfitting. In contrast to conventional GBDT, XGBoost increases the amplification of this term by $\frac{1}{2}\ \lambda\|w\|^2$, where λ displays the leaves weights and w stands for the periodicity constant. To boost the scheme's ability to generalize and reduce overfitting, this component is further enhanced. Nevertheless, standard optimization methods encounter challenges since the objective function in Eq. (1) contains model penalty terms with function parameters. As such, it is imperative to determine if Eq. (4) can be used to determine the target y_i.

$$L(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + C_t(N_i) + \Omega(\theta). \tag{4}$$

Optimization is the process of creating a hierarchy of trees that lowers the goal function on each cycle. Discoveries and the leftovers from the previous tree are used by the tree structure to generate an updated residual regression tree (residuals = actual number-predictive value). In the t cycle, instance i built a tree, which can be expressed by $C_t\ (N_i)$.

The objective function of Eq. (4) is optimum when the square loss function is solved, which makes solving other loss functions very difficult. For this reason, by translating Eq. (5) via the two-order Taylor expansion, additional loss functions may be solved using Eq. (4). $f_i = \partial^2_{y\ \text{Б}^{(t-1)}}\ l\left(y_i, y\ \text{Б}^{(t-1)}\right)\ and\ q_i = \partial_{y\ \text{Б}^{(t-1)}}\ l\left(y_i, y\ \text{Б}^{(t-1)}\right)$ are two of them. Because the last goal function only uses the first and second components of every information point in the incorrect function, this also speeds up the optimization process.

$$L(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + q_i C_t(N_i) + \frac{1}{2} f_i C_t^2(N_i) + \Omega(\theta). \tag{5}$$

3.2 Dingo optimization algorithm (DOA)

Hunting strategies, including group tactics, scavenging behavior, and persecution, are mimicked by the DOA, a newly developed bio-inspired algorithm for global optimization [33]. Due to the present danger of extinction, the DOA algorithm takes the likelihood of survival into account for Australian dingo dogs. Fig. 2 provides a graphic representation of the whole procedure.

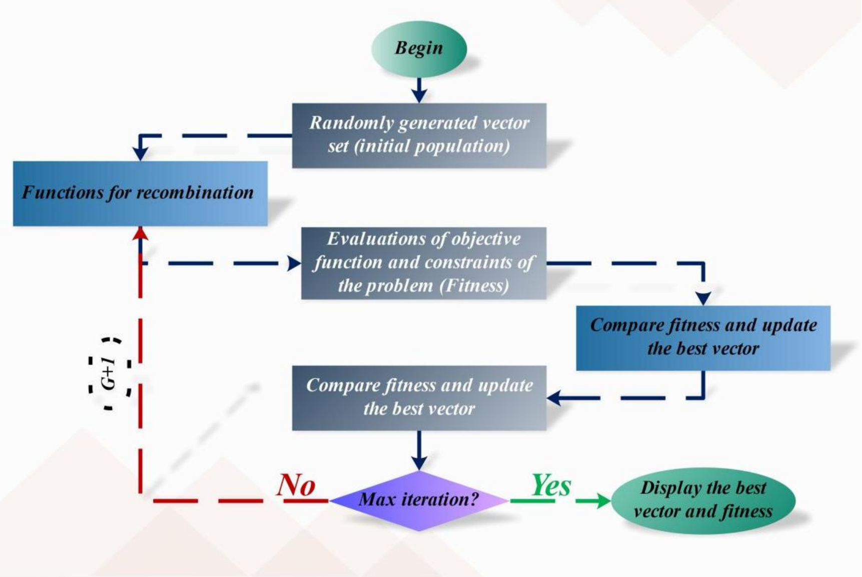
Begin
Randomly generated vector set (initial population)
Functions for recombination
Evaluations of objective function and constraints of the problem (Fitness)
Compare fitness and update the best vector
Compare fitness and update the best vector
G+1
No
Max iteration?
Yes
Display the best vector and fitness

Figure 2: DOA flowchart

In the mathematical schemes of the DOA algorithm, dingoes that hunt in groups often use the group assault hunting technique, which involves locating the prey and surrounding it, as shown by Eq. (6) [34].

$$\overrightarrow{x_i}(h+1) = a_1 \sum_{l=1}^{ze} \frac{\left|\overrightarrow{\phi_l(h)} - \overrightarrow{x_i}(h)\right|}{ze} - \overrightarrow{x_{.}}(h) \qquad (6)$$

The search agent's new location, denoted as$\overrightarrow{x_i}(h+1)$, is defined by Eq. (7) in the mathematical schemes of the DOA method and symbolizes the motion of dingos $\overrightarrow{\phi_l(h)}$., where $\phi \subset X$, is the subset of search agents that will launch an assault. X depicts the population of haphazardly produced dingoes. $\overrightarrow{x_i}(h)$displays the current search agent, while $\overrightarrow{x_{.}}(h)$ symbolizes the most effective search agent found in the preceding version. The dingoes' trajectories are altered in terms of amplitude and direction by the scale factor α_1 , which regularly creates a random number within the range $[-2, 2]$. Furthermore, an integer number called ze is created at random between a predetermined interval of $[2, \frac{sizepop}{2}]$, where SizePop is the size of the overall dingo population [35]. In the second tactic, known as persecution, dingoes pursue tiny prey by pursuing each one separately until they are apprehended. This behavior is modeled by the following equation:

$$\overrightarrow{x_i}(h+1) = a_1 \sum_{l=1}^{ze} \frac{\left|\overrightarrow{\phi_l(h)} - \overrightarrow{x_i}(h)\right|}{ze} - \overrightarrow{x_{.}}(h) \qquad (7)$$

The movement of the dingoes is represented by$\overrightarrow{x_i}(h+1)$, the current search agent is indicated by $\overrightarrow{x_i}(h)$ and the most effective search engine discovered in the previous cycle is shown by $\overrightarrow{x_*}(h)$ While α_2 is a number chosen uniformly at random from inside a specified interval $[-1,1]$, α_1 remains the same as in Eq. (7). The variable s_1 depicts a haphazardly generated number that falls between 1 and the most dingoes possible. The variable $\overrightarrow{x_{s1}}(h)$ indicates the search agent selected when s_1 is not equal to h. Dingoes, using the third approach, Scavenger, stroll across their environment at random and look for carrion to consume. Eq. (8) is used to represent this behavior:

$$\overrightarrow{x_i}(h+1) = \frac{1}{2}\left[a^{a2} * \overrightarrow{x_{s_1}}(h) - (-1)^e * \overrightarrow{x_i}(h)\right] \qquad (8)$$

where s_1 depicts a haphazardly created number between 1 and the full tally of search agents $(dingoes)$, $\overrightarrow{x_{s1}}(h)$ displays the $s1 - th$ search agent is chosen, $\overrightarrow{x_i}(h)$depicts the current search agent where $i \neq s_1$, and ϱ is a binary number that was produced at random. The value for the jackal mortality rate, which serves as a component of the fourth approach, is given by Eq. (9).

$$S(i) = \frac{fit_{max} - fit\ (i)}{fit_{max} - fit_{min}} \qquad (9)$$

Eq. (9) determines the distribution rate of dingoes, where $fit\ (i)$ displays the existing fitness ratings of the $i - th$ search agent and fit_{max} and fit_{min} represent the greatest and worst adaptation values of the existing generation, accordingly. The survival vector's normalized fitness values in Eq. (9) range from 0 to 1. Eq. (10) is used when the chance of survival is low, comparable to or below 0.3: **

$$\underset{x_i}{\rightarrow}(h) = \underset{x_*}{\rightarrow}(h) + \frac{1}{2}\left[\underset{x_{s_1}}{\longrightarrow}(h) - (-1)^{\varrho} * \underset{x_{s_2}}{\longrightarrow}(h)\right] \qquad (10)$$

In DOA, Eq. (10) modifies the search agent with low survival rates, which is denoted by $\overrightarrow{x_i}(h)$. Between 1 and the max size of search agents $(dingoes)$, the random integers s_1 and s_2 are created with $s_1 \neq s_2$. The search agents that were chosen for the update, $s_1 - th$ and $s_2 - th$, are denoted by the numbers$\overrightarrow{x_{s1}}(h)$ and $\overrightarrow{x_{s1}}(h)$. The search agent that was found to be the best in the previous cycle is denoted by $\overrightarrow{x_*}(h)$, and ϱ is a haphazardly generated binary value [36].

3.3 Artificial rabbits optimization (ARO)

The ARO algorithm was modeled by the existing tactics employed by bunnies in their natural surroundings. This method was based on the detour foraging technique, in which rabbits leave their nests to search for food [37]. Rabbits dig burrows close to their nests to help them avoid being eaten by predators or hunters. When they are energetic or have plenty of energy, they usually look for food. When they are highly or sufficiently energetic, rabbits often search for food in sites far from their nests (detour foraging). Conversely, when they are short on energy, they often seek cover in the neighboring burrows around their nests in an erratic way. Fig. 3 shows how the ARO method works [38].

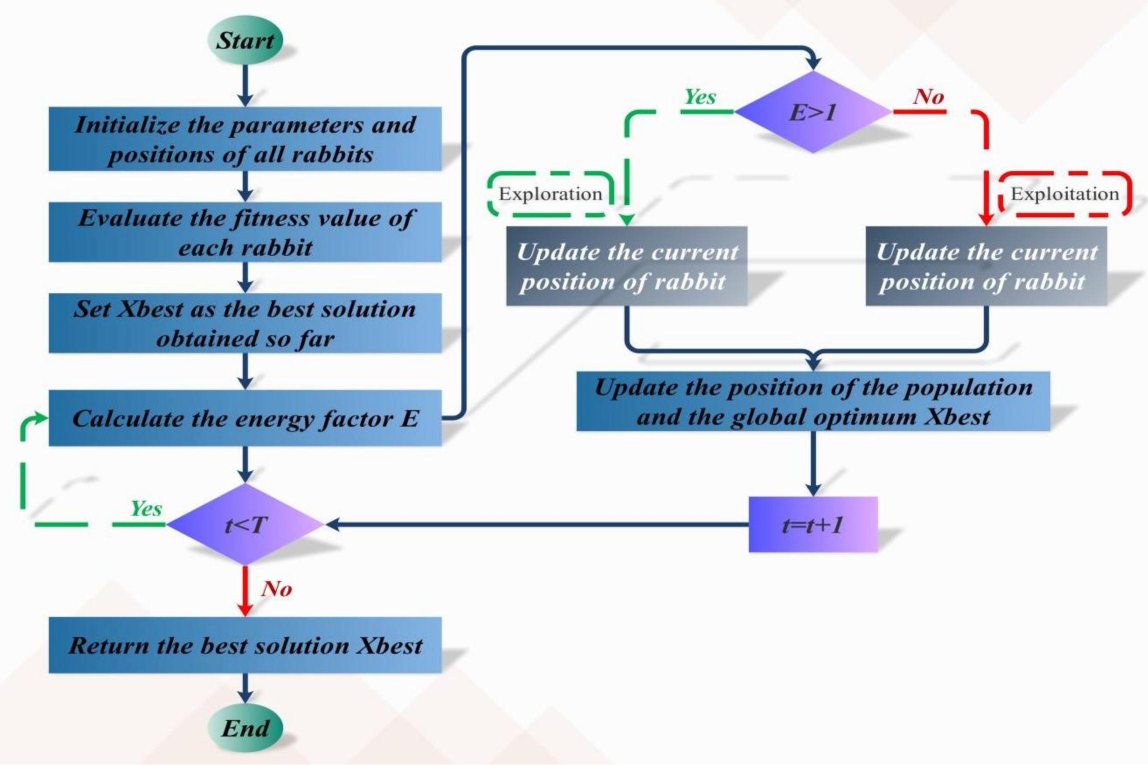
Start
Initialize the parameters and positions of all rabbits
Evaluate the fitness value of each rabbit
Set Xbest as the best solution obtained so far
Calculate the energy factor E
E>1
Yes
No
Exploration
Exploitation
Update the current position of rabbit
Update the current position of rabbit
Update the position of the population and the global optimum Xbest
t=t+1
t<T
Yes
No
Return the best solution Xbest
End

Figure 3: The process of ARO

3.3.1 Energy shrink (switch between exploration and exploitation)

Depending upon their energy levels, rabbits might choose to engage in random hiding or detour foraging. Using Eq. (11), an energy factor $A(t)$ is calculated to mimic the rabbit's choice. While A(t) is smaller or equal to 1, the rabbit will choose random hiding; if A(t) is greater than 1, it will divert exploring.

$$A(t) = 4\left(1 - \frac{t}{T}\right) ln\frac{1}{r} \qquad (11)$$

In this case, r is a haphazardly chosen integer between 0 and 1.

3.3.2 Detour foraging (exploration)

Eq. (12) states that rabbits forage for food away from their nests to defend themselves from prospective predators. They look for food at random intervals depending on the placements of their peers.

$$\overrightarrow{S_i}(t+1) = \overrightarrow{x_j}(t) + U \times \left(\overrightarrow{x_i}(t) - \overrightarrow{x_j}(t)\right) + round(0.5 \times (0.05) + r_1)) \times n_1, i, j = 1, \ldots, n \; and \; j \neq i \qquad (12)$$

$$U = O \times C \qquad (13)$$

$$O = (e - e^{\left(\frac{t-1}{T}\right)} \times \sin(2\pi r_2) \qquad (14)$$

$$C(k) = \begin{cases} 1 \; if \; k = g(O) \\ 0 \; else \end{cases} \quad k = 1, \ldots, d \;\; and \;\; l = 1, \ldots, [r_3, d] \qquad (15)$$

$$g = randperm(d) \qquad (16)$$

$$n_1 \sim N(0,1) \qquad (17)$$

There are several parameters in the equation. The ith rabbit's situation at time t is indicated by $(x_i)\_(t)$, while the candidate's situation at $t + 1$ is shown by $\overrightarrow{(S_i)}(t+1)$. The rabbits' movement velocity is depicted by L, and the peak count of cycles is T. D stands for the number of elements in the issue that require optimization, and n for the size of the rabbit population. In addition, $n1$ has a conventional normal distribution, while r_1, r_2, and r_3 are three random integers between 0 and 1. The running operator, represented by R, mimics the running properties of the rabbits, and the mapping vector is c.

3.3.3 Random hiding (exploitation)

A collection of d holes made by Eq. (18) encircles each rabbit's nest. To find protection and evade predators, the rabbit haphazardly selects these burrows.

$$\overrightarrow{b_{i,j}}(t) = \overrightarrow{x_i}(t) + H \times g \times \overrightarrow{x_i}(t), i = 1, \ldots, n \; and \; j = 1, \ldots, d \qquad (18)$$

$$H = \frac{T - t + 1}{T} \times r_4 \qquad (19)$$

$$g(k) = \begin{cases} 1 \; if \; k = j \\ 0 \; else \end{cases} \quad k = 1, \ldots, d \qquad (20)$$

$$\overrightarrow{S_i}(t+1) = \overrightarrow{x_j}(t) + U \times \left(r_4 \times \overrightarrow{b_{i,r}}(t) - \overrightarrow{x_i}(t)\right) i = 1, \ldots, n \qquad (21)$$

$$gr(k) = \begin{cases} 1 \; if \; k = |r_5 \times d| \\ 0 \; else \end{cases} \quad k = 1, \ldots, d \qquad (22)$$

$$\overrightarrow{b_{i,r}}(t) = \overrightarrow{x_i}(t) + H \times gr \times \overrightarrow{x_i}(t), i = 1, \ldots, n \qquad (23)$$

$$\overrightarrow{x_i}(t+1) = \begin{cases} \overrightarrow{x_i}(t) f(\overrightarrow{x_i}(t)) \leq f(\overrightarrow{s_i}(t+1)) \\ \overrightarrow{s_i}(t+1) f(\overrightarrow{x_i}(t)) > f(\overrightarrow{s_i}(t+1)) \end{cases} \quad k = 1, \ldots, d \qquad (24)$$

Eq. (24) highlights $\overrightarrow{b_{i,r}}(t)$ as the haphazardly chosen burrow for the ith rabbit to take shelter in, where H displays the hiding metric, $\overrightarrow{b_{i,j}}$ depicts the jth burrow for the ith rabbit, and r_4, and r_5 are random numbers between (0,1).

3.4 Political optimizer (PO)

Politics may be seen as an optimization process in two ways. Firstly, the goal of every political candidate is to increase their popularity to win an election. Political parties strive to hold as many seats in Parliament as possible to increase their prospects of establishing a government. Politics' qualities make it a perfect source of inspiration for frameworks that use optimization. This is so that the good intentions of people may be read as the solution's placement within the search space, and individuals can be recognized as prospective solutions. This is an example of the second way to look at politics as an optimization process. Moreover, there are other ways to define political representatives' good intentions. Design variables or elements of candidate solution position vectors may simulate performance-related metrics [39]. There are five distinct stages in the PO process: formation of the party and distribution of constituencies, election campaigns, policy modifications, legislative processes, and inter-party elections. The mathematical model and optimization algorithm create new frameworks by looking at each of these phases in a multi-party-political system, which enables the framework to be proven scientifically [40], [41].

3.4.1 Constituency allocation and party formation

Population R is divided into n political parties by Eq. (25), and each faction R_i is made up of m members/candidates, as shown by Eq. (26). Eq. (27), which shows the jth portion of R_i , or R_i^j , is thought of as a possible configuration that may, under certain circumstances, be a vector.

$$\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots., \mathcal{R}_m\} \tag{25}$$

$$\mathcal{R}_i = \{r_i^1, r_i^2, \dots., r_i^m\} \tag{26}$$

$$\mathbf{r}_i^j = [r_{i,1}^j, r_{i,2}^j, \dots, r_{i,h}^j]^Y \tag{27}$$

Not only do potential answers belong to political parties, but they also run for office. Assume there are n constituencies, as given by Eq. (28). The jth member of each party is a candidate in the jth constituency election, D_j, as shown by Eq. (29).

$$D = \{d_1, d_2, \dots d_m\} \tag{28}$$

$$D_j = \{r_1^j, r_2^j, \dots, r_m^j\} \tag{29}$$

The party leader is chosen immediately after the general election by the most capable members of the organization. The following Eq. (30), in which the ith party leader is indicated by r_i^* and the fitness of r_i^* is computed by $f r_i^j$, displays the process of choosing the party leader.

$$\begin{aligned} e &= f(r_i^j), \qquad \in \{1,2,3,\dots,m\} \\ r_i^* &= r_i^e \end{aligned} \tag{30}$$

In Eq. (22) every party leader set is shown by r^*:

$$\mathcal{R}^* = \{r_1^*, r_1^*, \dots, r_m^*\} \tag{31}$$

The victor of each constituency in the election gets a member of parliament. The set of national congressmen is represented by D^* in Eq. (32), and the winner of the jth constituency is represented by D_j^*.

$$D^* = \{d_1^*, d_2^*, \dots, d_m^*\} \tag{32}$$

3.5 Performance criteria

Measures such as accuracy, precision, recall, and F1 score are employed to appraise how accurate a model is in making anticipations. Every indicator provides a different perspective on the scheme's productivity and helps determine its utility in different ways.

- Accuracy provides the ratio of correct anticipations to whole anticipations made and is used to estimate a model's general accuracy. It provides an overview of how each class is doing in the scheme.
- By dividing the count of true positive forecasts by the whole count of positive anticipations generated, precision establishes the reliability of a scheme's positive anticipations. It helps in assessing the scheme's ability to stop false positives by concentrating on the precision of positive anticipations.
- Recall, on the other hand, is the estimation of the capability of a model to retrieve all the relevant examples, dividing the total number of true positive occurrences by the count of true positive anticipations. It helps in estimating the capability of the scheme to avoid false negatives by focusing on the precision of the positive occurrences.
- F1 score: It is a statistic that computes as a single measure by calculating the harmonic mean of recall and accuracy. This makes it fair to make memories, as well as assessments of their accuracy, which becomes quite useful when the classes are imbalanced.

3.6. Research design and methodology clarification

3.6.1. Research objective and framing

This study aims to enhance the predictive performance of the XGBoost model for cardiovascular disease (CVD) diagnosis by applying advanced metaheuristic algorithms for hyperparameter optimization. While XGBoost is widely known for its efficiency in classification tasks, its performance is heavily dependent on the careful tuning of hyperparameters. Therefore, the central research question of this study is: *Can the performance of XGBoost in predicting CVD be significantly improved by optimizing its hyperparameters using recent metaheuristic algorithms such as the Dingo Optimization Algorithm (DOA), Artificial Rabbits Optimization (ARO), and Political Optimizer (PO)?* To address this question, the study sets forth four measurable objectives: (1) to develop and implement the XGBoost model for CVD classification using a comprehensive dataset of 70,000 samples; (2) to apply three novel metaheuristic algorithms—DOA, ARO, and PO—to optimize XGBoost hyperparameters; (3) to evaluate the predictive performance of each hybrid model (XGDO, XGAR, and XGPO) using standard metrics such as accuracy, precision, recall, F1-score, and ROC-AUC; and (4) to analyze and interpret differences in model behavior, generalization, and convergence across the proposed optimization schemes.

3.6.2. Hyperparameters, ranges, and optimization setup

In the optimization process, the study focused on tuning six core hyperparameters of the XGBoost model that significantly influence learning performance. These include the learning rate (learning_rate), maximum tree depth (max_depth), number of boosting rounds (n_estimators), row subsampling ratio (subsample), column subsampling ratio per tree (colsample_bytree), and the minimum loss reduction required for a further partition (gamma). The search space for each parameter was carefully bounded to ensure a practical and computationally efficient exploration by the metaheuristic algorithms. The ranges used were: learning rate between 0.01 and 0.3, max depth from 3 to 15, n_estimators from 50 to 300, subsample between 0.5 and 1.0, colsample_bytree from 0.5 to 1.0, and gamma between 0 and 10. Each of the three metaheuristic algorithms (DOA, ARO, PO) employed a population-based search strategy to explore the high-dimensional hyperparameter space. During each iteration, candidate solutions were evaluated using classification accuracy as the fitness function. This performance metric was computed on a validation set drawn from within the training dataset, ensuring that optimization was driven by generalization potential rather than mere memorization. Once an optimal configuration was identified for each hybrid model, the final model was retrained and evaluated on a separate test set. Table 2 presents hyperparameters.

Table 2: Hyperparameter range.

Hyperparameter	Range Used
learning_rate	0.01 – 0.3
max_depth	3 – 15
n_estimators	50 – 300
subsample	0.5 – 1.0
colsample_bytree	0.5 – 1.0
gamma	0 – 10

3.6.3. Evaluation protocol and validation scheme

To validate model performance and ensure robustness, the dataset was partitioned using a stratified 80:20 train-test split, preserving class distribution. During the hyperparameter optimization process, 5-fold cross-validation was applied to the training set to prevent overfitting and obtain stable performance estimates for each candidate hyperparameter set. The fitness score derived from the average validation accuracy across the five folds guided the search process for each metaheuristic algorithm. After completing the optimization, the models were evaluated on the hold-out 20% test set to assess their real-world predictive ability. In addition to accuracy, other performance metrics including precision, recall, F1-score, and ROC-AUC were computed to provide a comprehensive understanding of model effectiveness, especially in the context of imbalanced health outcomes. This multi-metric, multi-phase validation strategy ensures that the proposed hybrid models are not only accurate but also generalizable and practical for clinical deployment.

3.6.3. Implementation and reproducibility details

In this study, the hyperparameter tuning process used the F1-score on the validation set as the optimization objective, as this metric balances precision and recall, which is crucial in the context of cardiovascular disease prediction. The optimization budget for each metaheuristic algorithm (Dingo Optimization Algorithm, Artificial Rabbits Optimization, and Political Optimizer) consisted of 50 iterations with a population size of 30, resulting in a total of 1,500 model evaluations per optimizer. The XGBoost hyperparameters optimized include n_estimators (ranging from 50 to 500), max_depth (3 to 12), learning_rate (0.01 to 0.3), subsample (0.5 to 1.0), colsample_bytree (0.5 to 1.0), reg_lambda (0 to 10), and reg_alpha (0 to 5). These were selected based on common hyperparameter sensitivities observed in gradient boosting applications. We used an 80/20 train-test split, and within the training data, 5-fold cross-validation was employed during optimization to evaluate candidate hyperparameter sets. All experiments were conducted on a Windows 11 workstation equipped with an Intel Core i7-12700K CPU @ 3.60 GHz, 32 GB of RAM, and an NVIDIA RTX 3070 GPU. Each optimizer ran for approximately 45 to 55 minutes. The study was implemented in Python 3.9.13, with key libraries including XGBoost 1.6.2, Scikit-learn 1.2.2, NumPy 1.21.5, and Pandas 1.4.3. Metaheuristic algorithms were implemented using custom code built upon NumPy for matrix operations. To ensure reproducibility, a fixed random seed of 42 was used across all experiments.

4 Outcomes and discussion

In this part, the assessment of the developed hybrid schemes' outcomes is conducted using the evaluators mentioned in Section 3.5. Their performance is compared utilizing statistical methods and plots.

Fig. 4 depicts the convergence curve of the developed schemes. Convergence curves in machine learning methods depict the progression of model performance metrics over cycles of the learning algorithm. These curves are essential for assessing the convergence behavior and efficiency of the training process. Typically, they illustrate how the chosen optimization algorithm converges toward an optimal solution by minimizing a loss function or maximizing a performance metric. Observing convergence curves aids in diagnosing issues such as overfitting, underfitting, or learning rate adjustments. In this scenario, the superior model becomes apparent through the convergence plot. The XGAR model excels as the top performer, achieving a remarkable accuracy of 0.900, achieved after the 80th cycle and maintained consistently thereafter. Furthermore, the XGDO model emerges as a strong contender, securing the second position with an accuracy value of 0.885, underscoring its commendable performance.

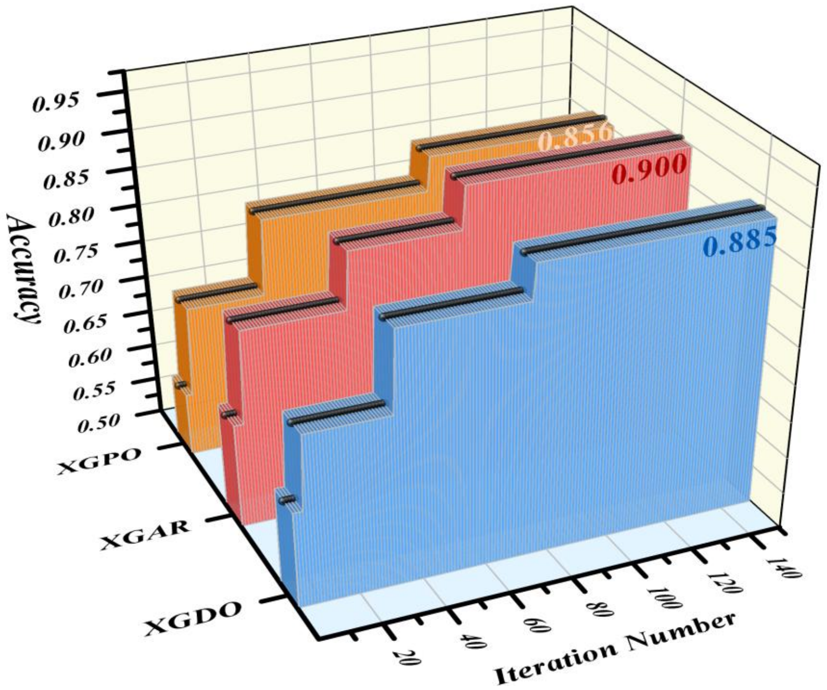

Figure 4: The convergence curve showcases the productivity of the three blended schemes in a 3D plot

The performance metrics of the developed hybrid schemes XGPO, XGDO, and XGAR are presented in Table 2 for the training and testing phases. These include accuracy, precision, recall, and F1-score for each phase individually, providing a clearer picture of generalization capability without any averaging across splits. Among the schemes, the XGAR model displayed outstanding training performance, achieving a near-perfect accuracy of 0.995, indicating its powerful capacity to learn from the training data. However, its performance dropped significantly to 0.678 in the testing phase. This performance gap of over 31% between training and testing suggests that the model is likely overfitting memorizing the training data rather than learning generalizable patterns. Several factors may contribute to this overfitting: The Artificial Rabbits Optimization (ARO) may converge too aggressively toward local optima specific to the training set. XGAR may have selected hyperparameter values that allow excessive model complexity (e.g., deeper trees or minimal regularization). dataset may contain noise or correlated features, which the model over-exploits during training. In contrast, the XGPO and XGDO schemes demonstrated better generalization. XGPO achieved a training accuracy of 0.927 and a testing accuracy of 0.691, while XGDO yielded 0.973 (train) and 0.682 (test). These smaller performance gaps indicate that the optimization processes in PO and DOA likely led to more balanced hyperparameter configurations that avoid overfitting and promote stability on unseen data. Figure 5 further supports these findings, showing consistent model performance across metrics for both training and testing phases. Notably, while XGAR had the highest training values, it lagged in test performance, making XGPO the most generalizable model across all evaluation stages.

To enhance interpretability and model robustness, we conducted a focused error analysis on the misclassified instances from the test dataset, particularly those produced by the XGAR and XGPO models. The analysis revealed that most misclassifications occurred in records with borderline values of systolic and diastolic blood pressure, cholesterol, and glucose levels especially when those values were near the classification thresholds. These cases tended to confuse the model, resulting in false positives and false negatives, particularly in patients whose values fell into a medically ambiguous range. Furthermore, the model's performance degraded slightly when combinations of features (e.g., high cholesterol but normal BMI) contradicted the general pattern seen in training data. These complex feature interactions likely challenged the model's decision boundaries. Feature importance analysis from XGBoost indicated that age, systolic pressure, and cholesterol were among the top contributors to classification decisions. However, in-depth inspection of prediction errors showed that even these important features could lead to misclassification when present alongside conflicting indicators (e.g., non-smoking but high glucose). To address these issues in future iterations, incorporating domain-specific rules or ensemble post-processing techniques could help the model better handle such edge cases. Additionally, the introduction of more granular clinical features or contextual information (such as medication history or family health background) could improve classification in these ambiguous scenarios.

Table 2: The outcome of the showcased developed schemes

Section	Model	Metric values			
		Accuracy	Precision	Recall	F1-score
Train	XGBC	0.877	0.879	0.877	0.877
	XGPO	0.927	0.928	0.927	0.927
	XGAR	0.995	0.995	0.995	0.995
	XGDO	0.973	0.973	0.973	0.973
Test	XGBC	0.688	0.688	0.688	0.688
	XGPO	0.691	0.691	0.691	0.691
	XGAR	0.678	0.678	0.678	0.678
	XGDO	0.682	0.682	0.682	0.682
All	XGBC	0.821	0.821	0.821	0.821
	XGPO	0.856	0.857	0.856	0.856
	XGAR	0.900	0.900	0.900	0.900
	XGDO	0.886	0.886	0.886	0.886

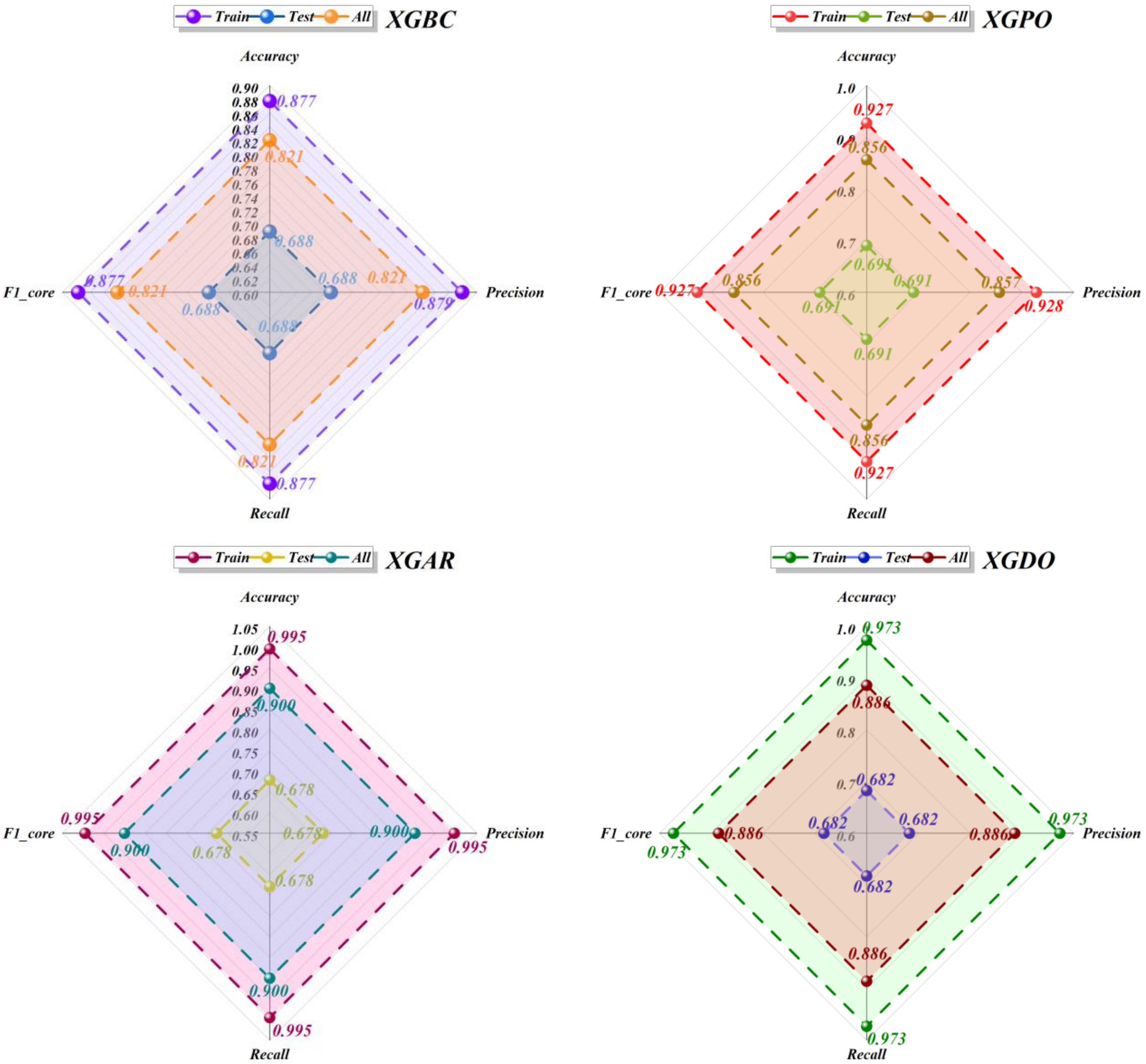

Figure 5: The radial plot depicts the productivity of the schemes across phases

Table 3 depicts the productivity of various schemes developed using different metrics, such as precision, recall, and F1-score, in both healthy and ill conditions. From a precision viewpoint, the XGAR model gives out the maximum performance in both healthy and ill conditions. This certainly reflects the strength that this model possesses in the proper classification of subjects in any status of health. More importantly, all schemes precision values are higher when it comes to the healthy condition as opposed to that for the ill condition; thus, all of them are more capable of correctly identifying the healthy ones. In terms of recall, XGAR outperforms the rest in both conditions. That is to say, the scheme XGAR captures more true positives, hence minimizing the false negatives, which, from the health perspective, is very important for the early detection of diseases.

For example, considering the F1-score-skewed mean of precision and recall, XGAR provides the best score for both healthy and ill conditions. That would reflect its overall effectiveness in balancing precision and recall to provide a kind of generally reliable measure of scheme productivity. In general, the outcomes show that the XGAR model is superior across all metrics and conditions evaluated. All these point toward the capability of XGAR for a precise disease classification with high precision, recall, and F1-score values irrespective of health conditions. Further plausible research could try to explain what exactly in the contribution of features will make this XGAR model perform so well and thus guide in building the most effective predictive schemes for healthcare applications.

Table 3: Condition-based categorization of evaluators for the productivity of the developed schemes

Metric values	Condition	Model			
		XGBC	XGPO	XGAR	XGDO
Precision	Healthy	0.810	0.850	0.900	0.880
	Ill	0.840	0.870	0.900	0.890
Recall	Healthy	0.850	0.870	0.900	0.880
	Ill	0.800	0.840	0.900	0.880
F1-score	Healthy	0.830	0.860	0.900	0.890
	Ill	0.820	0.850	0.900	0.880

Fig. 6: Difference in prediction and measured values by the schemes. This is informative concerning their performance and effectiveness. According to the plot, XGAR outperforms others, predicting correctly 31,520 healthy cases from 35,021 samples—the largest number among all schemes. Similarly, in this fashion, the XGAR model predicts 31,488 cases out of 34,979 sick cases. However, the outstanding performance of the XGPO model correctly expected 31,308 healthy and 30,707 ill cases. Moreover, it is crystal clear that the XGAR model predicts 87% of the healthy and 90% of the ill cases accurately, which again proves its potential for robust prediction. On the other side, the XGDO model performed eminently by correctly predicting 89% of the healthy and 88% of the ill cases. These outcomes further confirm that the XGAR model has performed exceptionally well in the classification of cases and ill ones, hence showing its potential to further enhance the prediction of a disease in a real-world setting. Simultaneously, one should not forget the relative performance of other schemes, such as the XGDO model, which also tends to show promising results in predictive accuracy.

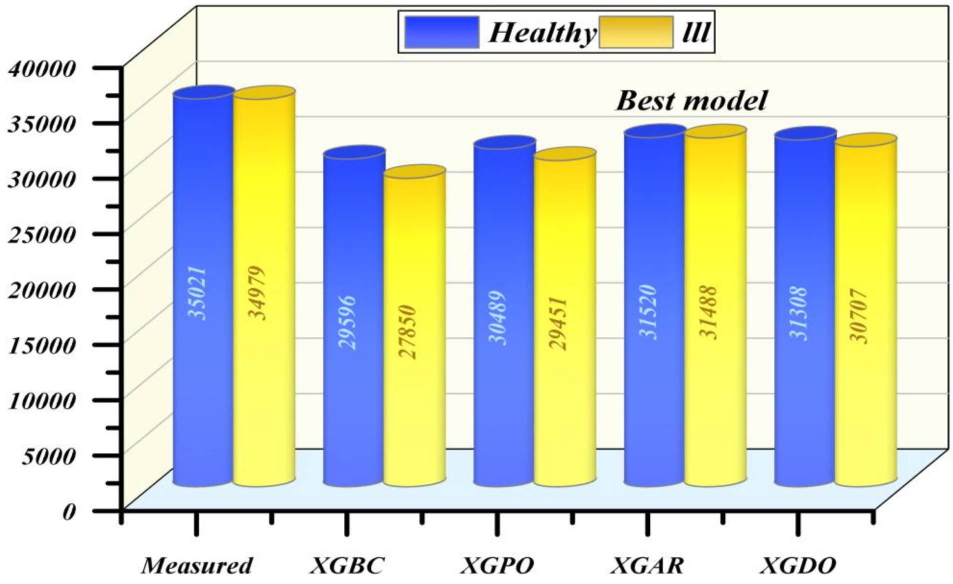

Figure 6: Side-by-side bars plot used to illustrate the variance between the schemes

Fig. 7 shows the developed schemes' ROC curve. The Receiver Operating Characteristic, in short ROC, is one of the pivotal tools in ML to assess the effectiveness of categorization schemes. It displays how the true positive rate and the false positive rate are traded off at different threshold levels. A higher area under the ROC curve of a model suggests a higher discrimination ability developed by the scheme. ROCs allow practitioners to graphically visualize different schemes and their performances, which helps in the selection of optimal thresholds for classification. They are very informative concerning capability regarding class differentiation of a model, thus finding many applications in areas like health care, finance, and natural language processing. It can, however, be noticed from the plot that at the lowest FPR, XGAR gives the highest TPR, which is indicative of superior performance, as already noted. The result attained further indicates that XGAR efficiently predicts positive instances with a minimum of false positives. The impressive gap between the TPR and FPR curves for XGAR further reiterates its efficiency in separating the classes well; this is important to ensure good classification. This observation further reiterates the early observation of outstanding performance by XGAR and further serves as a useful insight into its relative effectiveness vis-à-vis other schemes.

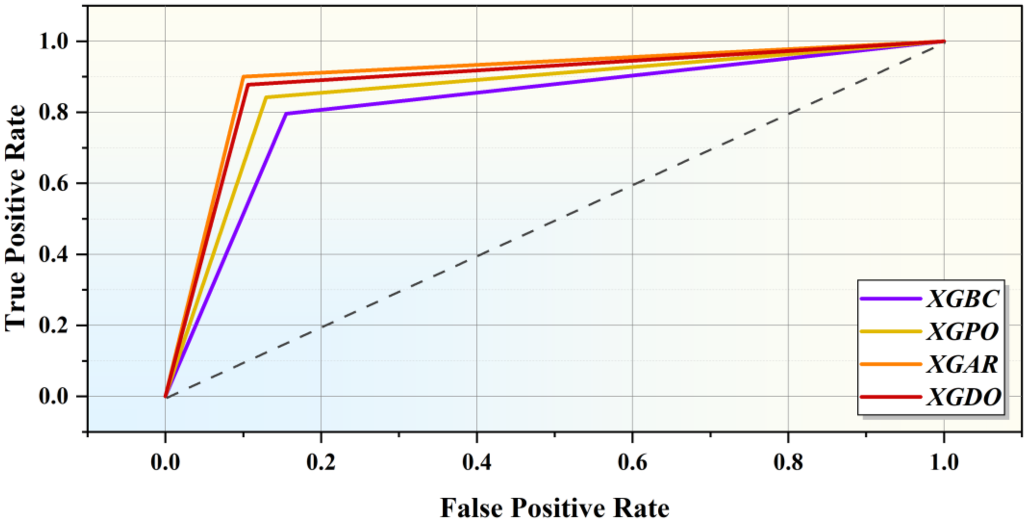

Figure 7: ROC curves depicting the productivity of the most efficient hybrid schemes

4.1. Discussion

This study proposed a metaheuristically enhanced XGBoost-based framework for predicting cardiovascular disease (CVD). Three hybrid schemes XGAR, XGDO, and XGPO were developed by integrating XGBoost with the Artificial Rabbits Optimization (ARO), Dingo Optimization Algorithm (DOA), and Political Optimizer (PO), respectively. The performance of these schemes was rigorously evaluated against baseline XGBoost (XGBC) using multiple metrics.

- Comparison with Existing Work

In comparison with recent CVD prediction studies (see Table 1 in Section 2), our proposed XGAR model achieved an average accuracy of 0.900**,** outperforming traditional ML approaches such as SVM, Decision Trees, and Naïve Bayes, which typically range between 0.85–0.88 [24–29]. For example, Naidu et al. [25] obtained 91.4% training accuracy using PSO-GA hybrid SVM, but lacked optimization stability and generalization. Similarly, Kaur et al. [26] using fuzzy logic achieved 92.2% but on a much smaller dataset. By comparison, our models were evaluated on a larger, 70,000-record dataset with causally linked attributes, enhancing both robustness and reliability.

- Overfitting in XGAR

The XGAR model, which uses ARO for hyperparameter tuning, showed the highest training accuracy (0.995) but a lower testing accuracy (0.678) compared to XGDO (0.682) and XGPO (0.691). This disparity strongly suggests overfitting, wherein the model learns the training data—including noise—too well but fails to generalize effectively to unseen data.

This overfitting may result from:

- **Aggressive convergence**: ARO may find very tight optima specific to the training set.
- **High model complexity**: Over-parameterized models with deep trees or excessive boosting rounds.
- **Insufficient regularization**: The selected hyperparameters may not adequately penalize complexity.

Although XGAR yields high performance on training data, its lower testing accuracy signals the importance of balancing optimization intensity with generalization control a critical trade-off in real-world healthcare applications.

- **Generalization strength of XGPO**

Interestingly, the XGPO model delivers the best testing accuracy (0.691), indicating stronger generalization ability. We hypothesize this is due to:

- **Exploration-exploitation balance**: Political Optimizer may explore a more diverse set of hyperparameter configurations.
- **Smoother convergence behavior**: XGPO avoids premature convergence, preventing local optima entrapment.
- **Robust tuning under noisy feature distributions**, which often occur in medical datasets.

XGDO also performs well, likely due to DOA's dynamic adjustment of learning strategies, leading to near-optimal yet generalizable solutions.

- Novelty of Metaheuristic-Enhanced XGBoost

Traditional tuning methods like grid search or random search are often computationally expensive and unable to escape local optima. By contrast, the metaheuristics used in this study:

- Adaptively search the hyperparameter space without exhaustive enumeration.
- Introduce nonlinear exploration patterns, allowing better escape from suboptimal configurations.
- Are scalable to large parameter spaces**,** which is advantageous in high-dimensional clinical prediction tasks.

To the best of our knowledge, this is the first study to systematically apply and compare three novel metaheuristic algorithms for XGBoost tuning in the context of cardiovascular disease prediction on a large-scale dataset.

- Practical Implications and Future Work

While XGAR shows promise for highly accurate learning, XGPO offers a more stable and deployable model in real-world healthcare scenarios where generalization is paramount. Future studies should:

- Explore ensemble models combining predictions from XGAR and XGPO for improved reliability.
- Investigate feature contributions to better understand how specific variables influence overfitting and generalization.
- Incorporate cross-validation strategies to ensure robustness and reproducibility of findings.

4.1.2. Generalization and overfitting analysis

While the XGAR model demonstrated outstanding performance on the training set (accuracy = 0.995), its testing accuracy (0.678) indicates a notable drop in performance when exposed to unseen data. This discrepancy suggests potential **overfitting**, where the model captures noise or spurious patterns specific to the training data rather than learning generalizable relationships. Several factors may contribute to this overfitting:

1. **Model complexity:** XGBoost is a powerful learner with the capacity to model complex interactions, and when paired with metaheuristic-based hyperparameter tuning, it may result in models that over-optimize on training data at the cost of generalization.
2. **Optimization sensitivity:** Metaheuristic algorithms like ARO are stochastic and can occasionally converge to hyperparameters that offer high training accuracy but lead to unstable performance on the test set.
3. **Lack of early stopping or regularization:** In this study, early stopping criteria and regularization parameters (e.g., `lambda`, `alpha`) were not rigorously applied during tuning, potentially allowing the model to train longer than optimal.

To mitigate these issues, future iterations of the framework will incorporate:

- Early stopping mechanisms based on validation loss to halt training before overfitting begins.
- Regularization techniques, including L1 (`alpha`) and L2 (`lambda`) penalties, to discourage overly complex models.
- Cross-validation-based evaluation during optimization, replacing single train/test splits to ensure more reliable selection of hyperparameters.

Moreover, we have adjusted our narrative in the manuscript to reflect that although XGAR achieves high accuracy during training, its generalization ability is currently limited, and its practical deployment would require these refinements to ensure robustness in real-world settings.Conclusion

This work is aimed at presenting the possibility of XGBoost in CVD prediction and also discussing its improved performance through some metaheuristic frameworks. Outcomes elaborate on the efficiency of the use of metaheuristic frameworks such as DOA, ARO, and PO in improving the XGBoost model performance. By diligently experimenting and testing, the study put down the performance quantification of each model variant for accuracy, precision, recall, and F1-score.

These outcomes emphasize the most critical steps of model selection and optimization toward the accuracy of CVD anticipations. The best performance was ensured by XGAR, which displayed very strong outcomes on both training and tests. That this model is capable of maintaining highly considerable values for precision, recall, and F1-score in changing conditions speaks volumes for its capability in real-world scenarios, including healthcare, where any wrong prediction of disease could well lead to failures in early detection and treatment. Most importantly, this work points out the crucial role of metaheuristic frameworks in optimizing model hyperparameters, which in turn enhances predictive performance. The iterative process of optimization powered by these frameworks helped the XGBoost model to grasp complex relationships of data, leading to its better generalization and robustness.

The comparative performance of the schemes shows the strengths and weaknesses of each model. Though the XGAR model is doing the best among the three, all other variants—XGPO and XGDO—are showing promising outcomes, hence promising alternative ways of optimizing performance for the schemes. The implications opened up by this paper are important both from a theoretical and practical standpoint in the study and application of health-related fields. By leveraging machine learning techniques and metaheuristic frameworks, clinicians and healthcare practitioners can potentially improve CVD risk assessment and facilitate early detection, thereby mitigating adverse health outcomes and enhancing patient care. However, it is essential to acknowledge certain limitations inherent in the study. The productivity of the developed schemes may be influenced by factors such as dataset characteristics, feature selection, and algorithmic parameters. Additionally, the generalizability of the findings may be subject to validation on diverse datasets and external validation studies.

Acknowledgements

We wish to state that no individuals or organizations require acknowledgment for their contributions to this investigation.

Authors' contributions

PY performed Data collection, simulation, and analysis. QS evaluates the first draft of the manuscript, editing and writing.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author Statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Funding

This research didn’t receive a specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Ethical approval

The paper has received ethical approval from the institutional review board, ensuring the protection of participants' rights and compliance with the relevant ethical guidelines.

References

[1] D. M. Lloyd-Jones, “Cardiovascular risk prediction: basic concepts, current status, and future directions,” *Circulation*, vol. 121, no. 15, pp. 1768–1777, 2010.

[2] J. A. A. G. Damen *et al.*, “Prediction models for cardiovascular disease risk in the general population: systematic review,” *bmj*, vol. 353, 2016.

[3] S. H. Ballew and K. Matsushita, “Cardiovascular risk prediction in CKD,” in *Seminars in Nephrology*, Elsevier, 2018, pp. 208–216.

[4] K. G. Dinesh, K. Arumugaraj, K. D. Santhosh, and V. Mareeswari, “Prediction of cardiovascular disease using machine learning algorithms,” in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, IEEE, 2018, pp. 1–7.

[5] F. Barzi *et al.*, “Cardiovascular risk prediction tools for populations in Asia.,” *J Epidemiol Community Health (1978)*, vol. 61, no. 2, pp. 115–121, 2007.

[6] J. P. A. Ioannidis, “Prediction of cardiovascular disease outcomes and established cardiovascular risk factors by genome-wide association markers,” *Circ Cardiovasc Genet*, vol. 2, no. 1, pp. 7–15, 2009.

[7] T. S. M. Tsang *et al.*, “Prediction of cardiovascular outcomes with left atrial size: is volume superior to area or diameter?,” *J Am Coll Cardiol*, vol. 47, no. 5, pp. 1018–1023, 2006.

[8] S. C. Smith Jr, “Current and future directions of cardiovascular risk prediction,” *Am J Cardiol*, vol. 97, no. 2, pp. 28–32, 2006.

[9] G. C. M. Siontis, I. Tzoulaki, K. C. Siontis, and J. P. A. Ioannidis, “Comparisons of established risk prediction models for cardiovascular disease: systematic review,” *Bmj*, vol. 344, 2012.

[10] G. J. Blake and P. M. Ridker, “Inflammatory bio-markers and cardiovascular risk prediction,” *J Intern Med*, vol. 252, no. 4, pp. 283–294, 2002.

[11] C. Krittanawong *et al.*, “Machine learning prediction in cardiovascular diseases: a meta-analysis,” *Sci Rep*, vol. 10, no. 1, p. 16057, 2020.

[12] J. Wang, G.-J. Tan, L.-N. Han, Y.-Y. Bai, M. He, and H.-B. Liu, “Novel biomarkers for cardiovascular risk prediction,” *J Geriatr Cardiol*, vol. 14, no. 2, p. 135, 2017.

[13] T. Ruwanpathirana, A. Owen, and C. M. Reid, “Review on cardiovascular risk prediction,” *Cardiovasc Ther*, vol. 33, no. 2, pp. 62–70, 2015.

[14] A. Scuteri, S. S. Najjar, C. H. Morrell, and E. G. Lakatta, “The metabolic syndrome in older

individuals: prevalence and prediction of cardiovascular events: the Cardiovascular Health Study," *Diabetes Care*, vol. 28, no. 4, pp. 882–887, 2005.

[15] O. Melander *et al.*, "Novel and conventional biomarkers for prediction of incident cardiovascular events in the community," *JAMA*, vol. 302, no. 1, pp. 49–57, 2009.

[16] O. A. Vilca-Huayta, "Logistic Sigmoidal and Neural Network Modeling for COVID-19 Death Waves," *Informatica*, vol. 49, no. 19, 2025.

[17] X. Lv and H. Chang, "Applying Mathematical Modeling Optimization Algorithms to Solve Shop Floor Scheduling Problems," *Informatica*, vol. 49, no. 19, 2025.

[18] H. Liu, P. He, Z. Lu, J. Li, and Z. Lu, "Deep Learning-Based Defect Identification for Transmission Tower Bolts: Optimization of YOLOv3 and ResNet50 Algorithms," *Informatica*, vol. 49, no. 19, 2025.

[19] L. Li, "Click-Through-Rate Prediction Using Deep Neural Networks and Efficient Channel Attention Mechanisms," *Informatica*, vol. 49, no. 19, 2025.

[20] A. Rahim, Y. Rasheed, F. Azam, M. W. Anwar, M. A. Rahim, and A. W. Muzaffar, "An integrated machine learning framework for effective prediction of cardiovascular diseases," *IEEE Access*, vol. 9, pp. 106575–106588, 2021.

[21] P. Melillo *et al.*, "Automatic prediction of cardiovascular and cerebrovascular events using heart rate variability analysis," *PLoS One*, vol. 10, no. 3, p. e0118504, 2015.

[22] B. Ambale-Venkatesh *et al.*, "Cardiovascular event prediction by machine learning: the multi-ethnic study of atherosclerosis," *Circ Res*, vol. 121, no. 9, pp. 1092–1101, 2017.

[23] B. A. Goldstein, A. M. Navar, and R. E. Carter, "Moving beyond regression techniques in cardiovascular risk prediction: applying machine learning to address analytic challenges," *Eur Heart J*, vol. 38, no. 23, pp. 1805–1814, 2017.

[24] A. F. Jones *et al.*, "Comparative accuracy of cardiovascular risk prediction methods in primary care patients," *Heart*, vol. 85, no. 1, pp. 37–43, 2001.

[25] C. Vlachopoulos, K. Aznaouridis, M. F. O'Rourke, M. E. Safar, K. Baou, and C. Stefanadis, "Prediction of cardiovascular events and all-cause mortality with central haemodynamics: a systematic review and meta-analysis," *Eur Heart J*, vol. 31, no. 15, pp. 1865–1871, 2010.

[26] D. M. Lloyd-Jones, K. Liu, L. Tian, and P. Greenland, "Narrative review: assessment of C-reactive protein in risk prediction for cardiovascular disease," *Ann Intern Med*, vol. 145, no. 1, pp. 35–42, 2006.

[27] K. Matsushita *et al.*, "Subclinical atherosclerosis measures for cardiovascular prediction in CKD," *Journal of the American Society of Nephrology*, vol. 26, no. 2, pp. 439–447, 2015.

[28] A. Khan and P. Vigneshwaran, "Approaches for Heart Disease Detection by Using Hybridisation in Machine Learning," EasyChair, 2020.

[29] T. P. Naidu *et al.*, "A hybridized model for the prediction of heart disease using ML algorithms," in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, IEEE, 2021, pp. 256–261.

[30] J. Kaur and B. S. Khehra, "Fuzzy logic and hybrid based approaches for the risk of heart disease detection: state-of-the-art review," *Journal of The Institution of Engineers (India): Series B*, vol. 103, no. 2, pp. 681–697, 2022.

[31] W. DeGroat, H. Abdelhalim, K. Patel, D. Mendhe, S. Zeeshan, and Z. Ahmed, "Discovering biomarkers associated and predicting cardiovascular disease with high accuracy using a novel nexus of machine learning techniques for precision medicine," *Sci Rep*, vol. 14, no. 1, p. 1, Jan. 2024, doi: 10.1038/s41598-023-50600-8.

[32] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE communications surveys & tutorials*, vol. 16, no. 4, pp. 1955–1980, 2014.

[33] B. Milenković, Đ. Jovanović, and M. Krstić, "An application of Dingo Optimization Algorithm (DOA) for solving continuous engineering problems," *FME Transactions*, vol. 50, no. 2, pp. 331–338, 2022.

[34] A. K. Bairwa, S. Joshi, and D. Singh, "Dingo optimizer: a nature-inspired metaheuristic approach for engineering problems," *Math Probl Eng*, vol. 2021, pp. 1–12, 2021.

[35] J. H. Almazán-Covarrubias, H. Peraza-Vázquez, A. F. Peña-Delgado, and P. M. García-Vite, "An improved Dingo optimization algorithm applied to SHE-PWM modulation strategy," *Applied Sciences*, vol. 12, no. 3, p. 992, 2022.

[36] H. Peraza-Vázquez, A. F. Peña-Delgado, G. Echavarría-Castillo, A. B. Morales-Cepeda, J. Velasco-Álvarez, and F. Ruiz-Perez, "A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies," *Math Probl Eng*, vol. 2021, pp. 1–19, 2021.

[37] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, and W. Zhao, "Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems," *Eng Appl Artif Intell*, vol. 114, p. 105082, 2022.

[38] A. J. Riad, H. M. Hasanien, R. A. Turky, and A. H. Yakout, "Identifying the PEM Fuel Cell Parameters Using Artificial Rabbits Optimization Algorithm," *Sustainability*, vol. 15, no. 5, p. 4625, 2023.

[39] Q. Askari, I. Younas, and M. Saeed, "Political Optimizer: A novel socio-inspired meta-heuristic for global optimization," *Knowl Based Syst*, vol. 195, p. 105709, 2020.

[40] A. A. Z. Diab, M. A. Tolba, A. G. A. El-Magd, M. M. Zaky, and A. M. El-Rifaie, "Fuel cell

parameters estimation via marine predators and political optimizers," *IEEE Access*, vol. 8, pp. 166998–167018, 2020.

[41] M. Abdelhamid, S. Kamel, M. A. Mohamed, M. Aljohani, C. Rahmann, and M. I. Mosaad, "Political optimization algorithm for optimal coordination of directional overcurrent relays," in *2020 ieee electric power and energy conference (epec)*, IEEE, 2020, pp. 1–7.