Mathematical Biosciences and Engineering

MBE, 21(2): 2943–2969.
DOI: 10.3934/mbe.2024131
Received: 13 October 2023
Revised: 21 January 2024
Accepted: 23 January 2024
Published: 29 January 2024

http://www.aimspress.com/journal/MBE

Research article

A robust framework for enhancing cardiovascular disease risk prediction using an optimized category boosting model

Zhaobin Qiu[1], Ying Qiao[1,2,*], Wanyuan Shi[1] and Xiaoqian Liu[1]

[1] School of Mathematics and Information Sciences, North Minzu University, Yinchuan, China
[2] Ningxia Collaborative Innovation Center for Scientific Computing and Intelligent Information Processing, North Minzu University, Yinchuan, China

* **Correspondence:** Email: 2005045@nun.edu.cn; Tel: +8613895078685.

Abstract: Cardiovascular disease (CVD) is a leading cause of mortality worldwide, and it is of utmost importance to accurately assess the risk of cardiovascular disease for prevention and intervention purposes. In recent years, machine learning has shown significant advancements in the field of cardiovascular disease risk prediction. In this context, we propose a novel framework known as CVD-OCSCatBoost, designed for the precise prediction of cardiovascular disease risk and the assessment of various risk factors. The framework utilizes Lasso regression for feature selection and incorporates an optimized category-boosting tree (CatBoost) model. Furthermore, we propose the opposition-based learning cuckoo search (OCS) algorithm. By integrating OCS with the CatBoost model, our objective is to develop OCSCatBoost, an enhanced classifier offering improved accuracy and efficiency in predicting CVD. Extensive comparisons with popular algorithms like the particle swarm optimization (PSO) algorithm, the seagull optimization algorithm (SOA), the cuckoo search algorithm (CS), K-nearest-neighbor classification, decision tree, logistic regression, grid-search support vector machine (SVM), grid-search XGBoost, default CatBoost, and grid-search CatBoost validate the efficacy of the OCSCatBoost algorithm. The experimental results demonstrate that the OCSCatBoost model achieves superior performance compared to other models, with overall accuracy, recall, and AUC values of 73.67%, 72.17%, and 0.8024, respectively. These outcomes highlight the potential of CVD-OCSCatBoost for improving cardiovascular disease risk prediction.

Keywords: cardiovascular disease; cuckoo search algorithm; opposition-based learning; CatBoost

1. Introduction

CVD is currently the leading cause of death worldwide, creating a significant burden on global healthcare resources. The reliance on doctors' personal experience and a range of tests for diagnosis can lead to misdiagnosis and high testing costs, negatively impacting patients' well-being. The application of data mining and machine learning in this field has proven advantageous, harnessing the power of information mining and learning to predict disease risks beyond doctors' personal experience. Early detection of cardiovascular disease can aid medical professionals in decision-making, reduce the medical burden on patients, and optimize the distribution of healthcare resources.

The Framingham risk score (FRS) model was initially created in the US to predict coronary heart disease risk within a specific timeframe [1]. The European Society of Cardiology (ESC) developed a scoring program to enhance risk assessment accuracy for the European population [2]. The UK introduced the QRISK model, which assesses cardiovascular disease risk over ten years, considering endpoints like myocardial infarction, stroke, transient ischemic attack, and cardiovascular disease [3]. Cardiovascular risk scores derived from traditional biostatistical methods' strict assumptions tend to oversimplify complex relationships and limit applications. Machine learning algorithms (MLA) were able to overcome these statistical drawbacks and improve discriminatory performance over traditional models. In a prospective cohort study, Weng et al. explored the potential of machine learning for improving cerebrovascular disease risk prediction. Their findings suggest that machine learning algorithms can significantly enhance prediction precision and are a viable approach for cardiovascular disease prediction [4]. Dimopoulos et al. explore the potential of employing machine learning methods to predict cardiovascular disease, especially when compared to the established risk tool, HellenicSCORE. The experimental results indicate that the machine learning method demonstrates remarkably high accuracy and sensitivity, making it a suitable prediction tool for cardiovascular disease [5]. Huang et al. employed integrated machine learning algorithms to explore novel data sources for cardiovascular risk prediction, incorporating detailed lifestyle questionnaires and continuous blood pressure monitoring. In comparison to the conventional risk scoring method, FRS, all integrated machine learning algorithms exhibited superior performance in both low and high-risk categories. However, it's important to note that this study was constrained by the relatively small sample size of patients at high risk for CVD [6]. Ordikhani et al. employed genetic algorithms to construct a novel risk assessment model for predicting CVD events. In contrast to classical machine learning and statistical methods, the calibrated XPARS charts demonstrated the capacity to enhance existing models by balancing interpretability and predictive accuracy. This approach offers the advantages of both black and white box models, ensuring high performance and interpretability. However, it is crucial to acknowledge that the coverage of various factors may be limited due to dataset influences. Additionally, training on a large amount of data does not necessarily guarantee optimal prediction time and accuracy [7].

Machine learning classifiers have gained significant popularity in predicting cardiovascular diseases [8–10]. Kanagarathinam et al. curated a hybrid dataset with the objective of facilitating the development of optimal CVD risk prediction models. During the feature selection process, the Pearson correlation method was employed to eliminate redundant features. The risk prediction model was constructed utilizing six machine learning classifiers, and through a rigorous 10-fold cross-validation, the CatBoost ML classifier emerged as the top performer, achieving an impressive average accuracy of 94.34% [11]. Sung et al. showcased the precision of deep learning by contrasting the performance

of Cox risk regression with RNN-LSTM based on survival analysis. They utilized layer-wise relevance propagation (LRP) to extract known risk factors identified in prior clinical studies from the study results. The experimental findings indicated a notable decrease in the predictive power of deep learning methods over time. Furthermore, despite the assessment of risk factors using LRP alone, the specific impact of these factors remained unclear [12]. Pan et al. proposed an enhanced deep learning-assisted convolutional neural network (EDCNN) to aid and enhance outcomes for patients with heart disease. In comparison to traditional methods such as ANN, DNN, EDL-SHS, RNN, and NNE, the designed diagnostic system can accurately and efficiently determine the risk level of heart disease [13]. These advancements demonstrate the promising potential of combining machine learning techniques with other approaches to effectively predict and diagnose cardiovascular diseases. By leveraging the strengths of different algorithms and models, researchers are continuously improving the accuracy and reliability of predictions in this important domain. In Table 1, we show the summary of the latest work that has been done in the field of predicting heart disease.

Table 1. The state-of-the-art on CVD with various methods.

Year	Authors	Research Title	Method
2019	Pandey et al. [14]	Automatic detection of arrhythmia from imbalanced ECG database using CNN model with SMOTE	SMOTE+CNN
2019	Ali et al. [15]	An Automated Diagnostic System for Heart Disease Prediction Based on $\chi 2$ Statistical Model and Optimally Configured Deep Neural Network	$\chi 2$-DNN
2020	Mienye et al. [16]	An improved ensemble learning approach for the prediction of heart disease risk	Randomized decision tree ensemble
2022	Pandya et al. [17]	InfusedHeart: A Novel Knowledge-Infused Learning Framework for Diagnosis of Cardiovascular Events	LBP+LSTM-CNN
2022	Srinivas et al. [18]	hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost	OPTUNA-XGBoost
2023	Baviskar et al. [19]	Optimization using Internet of Agent based Stacked Sparse Autoencoder Model for Heart Disease Prediction	EPO+SSC-AE
2023	Wei et al. [20]	Risk assessment of cardiovascular disease based on SOLSSA-CatBoost model	SOLSSA-CatBoost
2023	Kumar et al. [21]	An improved hawks optimizer based learning algorithms for cardiovascular disease prediction	HO optimizer

In conclusion, studies have predominantly relied on small and medium-sized cohort data, often lacking sufficient and effective validation. While the existing research demonstrates high prediction accuracy, there is a notable oversight in identifying certain risk groups. Improving the predictive accuracy of the model is an immediate and critical task. Furthermore, the application of models in most studies is based on a trial-and-error approach. To address these challenges, we have developed a systematic framework for cardiovascular disease risk prediction based on existing methods. This framework utilizes Lasso regression for feature selection, leveraging its good interpretability and protection against overfitting to select the optimal feature subset for accurate risk prediction.

Additionally, improved classifiers are integrated into the framework to enhance cardiovascular disease risk prediction. Recognizing the high dependence of the classifier on model parameters, we propose an enhanced cuckoo search algorithm by incorporating an opposition-based learning strategy. This modification aims to improve the convergence speed and accuracy for the rapid optimization of CatBoost model parameters. In summary, the systematic framework for cardiovascular disease risk prediction incorporates essential considerations such as data quality management, feature screening, and an improved classifier, resulting in accurate and stable prediction outcomes.

The contribution of the study is summarized as follows:

1) The proposed framework (CVD-OCSCatBoost) presents a systematic approach for predicting the risk of cardiovascular disease. It includes three key steps: outlier processing, feature screening, and the use of an improved classifier for accurate and stable predictions.

2) An improved cuckoo search algorithm (OCS) is proposed, which combines the opposition-based learning strategy with the cuckoo search algorithm. This approach generates high-quality initial populations during the initialization stage, accelerates the convergence rate, promotes exploration of the search area during the position update stage, and improves population diversity.

3) The proposed approach (OCSCatBoost) aims to enhance parameter selection in the CatBoost model through the use of the OCS algorithm, allowing for a more efficient identification of the optimal parameter combination.

4) Comparisons with popular machine learning algorithms and swarm intelligence algorithms validate the effectiveness of the OCSCatBoost algorithm.

2. Materials and methods

To enhance the accuracy of cardiovascular disease risk prediction, we propose the CVD-OCSCatBoost framework based on machine learning. The framework incorporates several techniques, including data outlier handling, feature selection , and classifier improvement using the OCS algorithm. Outlier data is identified by utilizing a box-line plot and examined. Feature selection is carried out using the Lasso regression algorithm to obtain better input data. Finally, the optimal hyperparameter combination of the CatBoost classifier model is identified using the OCS algorithm to enhance predictive outcomes concerning cardiovascular disease risk.

Let $D = (x_1,\ldots,x_m)$ represent a dataset with m examples, where each example is characterized by d attributes. Each example $x_i = (x_{i1},\ldots,x_{id})$ is a vector in a d-dimensional sample space X, where x_{ij} is the value of x_i on the j-th attribute and d is the dimension of the x_i sample. Cardiovascular disease prediction involves establishing a mapping $f: X \rightarrow Y$ from the input space X to the output space Y, where $y = f(x)$. Each (x_i, y_i) represents the i-th sample, where $y_i \in Y$ is the label of example x_i. The set of all labels, Y, is $\{0,1\}$, where $y = 0$ represents a normal condition and $y = 1$ represents a disease.

Based on existing research, we selected the linear regression method (logistic regression), KNN [5], support vector machine [8], two tree-based methods (decision tree [14], XGBoost), and ensemble optimization methods (CatBoost [21]) as the control group and utilized six stages to evaluate the six machine learning methods (as shown in Table 2). These six evaluation stages include: (1) Loading the dataset; (2) preprocessing data; (3) feature selection; (4) running the machine learning model; (5) applying evaluation metrics; and (6) processing classifier performance results.

Table 2. Overview of machine learning classification algorithms.

Model	Description	Advantage	Limitations
Decision Tree	Based on features, it segments data to create a tree-like structure of decision rules.	Easy to interpret	Prone to overfitting
KNN	Based on sample similarity, it predicts based on the class of K nearest neighbor samples to the current sample.	Simple and easy to use	Computationally intensive
Logistic regression	This algorithm utilizes linear methods to classify data into different categories or groups.	Easy to implement	Vulnerable to interference
SVM	Based on support vectors, it finds a linear or non-linear classifier that maximizes class boundary.	Suitable for high-dimensional data	Long training time
XGBoost	A joint-learned decision tree is built to reduce sample loss errors and prevent overfitting.	Efficiently handles large datasets	Sensitive to parameter tuning
CatBoost	It introduces categorical information processing features to avoid class bias and changes in weighted errors.	High accuracy, suitable for categorical data	Slow, memory issues, noise-sensitive

Each method stage is described as follows:

1) **Load the data set**. Select and load data from a data set containing clinical records of CVD patients.

2) **Preprocess the data**. Check the loaded data, understand its content, and deal with missing and abnormal values to ensure the best results of the classification algorithm.

3) **Select attributes or the main influencing factors**. In order to select the most important features that affect the performance of the model, Lasso regression is employed. Additionally, the dataset is split into two subsets: a training set and a test set (80% for training and 20% for testing).

4) **Run the machine learning model**. This stage involves feeding the preprocessed data into the chosen machine learning algorithm and training it to make predictions.

5) **Apply evaluation indicators**. According to five evaluation indexes, the training performance of the model is analyzed: accuracy, precision, recall rate, F1 score, and AUC value.

6) **Processing classifier performance results**. This stage involves analyzing the results of the evaluation metrics, comparing performance across different methods, and drawing conclusions about the usefulness and effectiveness of each method.

2.1. Lasso returns

Lasso regression, proposed by Robert Tibshirani in 1996, adds a penalty term to the least squares method to achieve variable selection on sample data. The penalty term gradually reduces the coefficients of non-significant variables, resulting in only significant variables with non-zero coefficients, enabling data dimensionality reduction.

Assume that the dependent variable is $y=(y_1,...,y_n)^T$, the independent variables are $X=(x_{1j},...,x_{nj})^T$, $j=1,...,p$, and $\beta=(\beta_1,...,\beta_n)^T$ are the coefficient vectors, and the residual term ε satisfies the Gauss-Markov hypothesis. The linear model is as follows:

$$y=X\beta+\varepsilon \tag{1}$$

The choice of variables for the Lasso method is obtained by the following equation, with λ is the canonical term parameter:

$$\beta = argmin\sum_{i=1}^{n}(y_i - \sum_{j=0}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{2}$$

The solution of the above equation can be transformed into an optimization problem with a penalty term, where k is the adjustment parameter, corresponding to λ:

$$\beta = argmin\sum_{i=1}^{n}(y_i - \sum_{j=0}^{p}\beta_j x_{ij})^2 \ s.t.\sum_{j=1}^{p}|\beta_j| \le k \tag{3}$$

The basic idea of Lasso regression is to minimize the sum of squared residuals by setting a penalty term. If the sum of absolute values of regression coefficients is less than a certain value, then some coefficients of variables are compressed to zero. The variables with zero coefficients are considered non-significant variables, and the corresponding key impact variables are identified.

2.2. CatBoost

CatBoost is a new machine learning algorithm framework developed by Russian search giant Yandex in April 2017. It is based on the gradient boosting decision tree (GBDT) algorithm framework and has the ability to handle category-based features more effectively. It can also use a combination of category features to greatly enrich the feature dimension. GBDT is an algorithm proposed by Friedman in 2000 to avoid the problem of overfitting. This problem is caused by integrating multiple decision trees into a single decision tree for regression and classification. GBDT utilizes gradient descent for optimization, constructing a learner in each iteration that reduces loss along the steepest direction of the gradient. This compensates for the shortcomings of the currently constructed model. The algorithmic model can be defined as follows:

$$F(x,\omega) = \sum_{t=0}^{T}\alpha_t h_t(x,\omega_t) = \sum_{t=0}^{T} f_t(x,\omega_t) \tag{4}$$

where: $F(x,\omega)$ is the output of the whole decision tree; x is the input of the sample; ω is the parameter of the whole decision tree; α_t is the weight of the *t*-th tree; *T* is the number of trees; $h_t(x,\omega_t)$ is the output of the *t*-th decision tree; ω_t is the parameter of the *t*-th decision tree; $f_t(x,\omega_t)$ is the output of the *t*-th decision tree after weighting.

The optimal parameters of the model can be obtained by minimizing the loss function, defined as follows:

$$(\alpha_t,\omega_t) = argmin\sum_{i=0}^{N} L(y_i, F(x_i,\omega)) \ t = 1,2,...,T \tag{5}$$

where: $L(y_i, F(x_i,\omega))$ is the loss function, and usually the mean square error or absolute loss can be used as the loss function; y_i is the actual output of sample *I*; x_i is the actual input of sample *I* and

N is the number of samples.

CatBoost uses a more efficient strategy to reduce overfitting and effectively utilize data information during training. First, the algorithm converts categorical features into numerical features based on the statistical value of the prediction target. It uses an oblivious tree as the base predictor, and binarizes the floating-point features, statistical information, and one-hot encoding together. Second, the algorithm reduces the influence of less noisy and low-frequency category-type data on the data distribution by adding prior terms and weight coefficients. This helps to reduce model overfitting.

$$\hat{x}_k^i = \frac{\sum_{j=1}^{N} I_{\{x_j^i = x_k^i\}} y_j + ap}{\sum_{j=1}^{N} I_{\{x_j^i = x_k^i\}} + a} \tag{6}$$

where x_k^i is the i-th category feature of the k-th training sample; y_j is the label of the j-th sample; p is the added prior term; a is the weight coefficient; I is the indicator function, i.e., 1 is taken when the two quantities in parentheses are equal, and 0 is taken vice versa, i.e.,

$$I_{\{x_j^i = x_k^i\}} = \begin{cases} 1 & x_j^i = x_k^i \\ 0 & other \end{cases} \tag{7}$$

Last, the CatBoost algorithm employs a "greedy strategy" to process feature combinations. During the first split of the tree, no feature combinations are performed. However, during the second split, all current tree splits and category-based features are combined with all category-based features in the dataset. The new combined values are instantly converted to numerical features. All splits that are selected in the tree are considered category-based features with two values, and they are combined to generate combinations of numerical and categorical features.

The CatBoost algorithm uses a leaf node calculation that can effectively avoid overfitting and make the model more general. It synchronizes the training dataset with the processing of category-based features, thus greatly improving the efficiency of feature processing. The algorithm also binarizes floating-point features, statistical information, and unique thermal encoding features. It then uses binary features to calculate the model prediction.

For each feature, prediction values change shows how much on average the prediction changes if the feature value changes. The bigger the value of the importance the bigger on average is the change to the prediction value if this feature is changed.

Leaf pairs that are compared have different split values in the node on the path to these leaves. If the split condition is met (this condition depends on the feature *F*), the object goes to the left subtree; otherwise, it goes to the right one.

$$feature\_importance_F = \sum_{tree, leafs_F} (v_1 - avr)^2 \cdot c_1 + (v_2 - avr)^2 \cdot c_2 \tag{8}$$

$$avr = \frac{v_1 \cdot c_1 + v_2 \cdot c_2}{c_1 + c_2} \tag{9}$$

where c_1, c_2 represent the total weight of objects in the left and right leaves, respectively. This weight is equal to the number of objects in each leaf if weights are not specified for the dataset. V_1, v_2 represent

the formula value in the left and right leaves, respectively.

If the model uses a combination of some of the input features instead of using them individually, an average feature importance for these features is calculated and output.

2.3. Cuckoo search algorithm

The Cuckoo search (CS) algorithm is an intelligent optimization algorithm developed by Yang and Deb in 2009 [22]. It is inspired by the breeding behavior of cuckoos and the Levy flight search mechanism. The CS algorithm has gained significant attention from researchers due to its simplicity, minimal parameter requirements, and ease of implementation. The CS algorithm comprises three major components: Best solution preservation: This component involves selecting the best solution and ensuring that it is carried forward to the next generation, akin to preserving the best bird nest. Local random movement: The algorithm incorporates local random movements to explore and search for optimal solutions. Global Levy flight: This part simulates the cuckoo's behavior of finding the best nesting bird's eggs through Levy flight, providing a mechanism for random global search. The CS algorithm is based on three idealized assumptions, and its specific model can be defined as follows:

(1) Local random movement

Local random processes can be defined as:

$$x_i^{t+1} = x_i^t + \alpha s \oplus H(p_\alpha - \varepsilon) \otimes (x_j^t - x_k^t) \tag{10}$$

where, x_j^t and x_k^t are two distinct random sequences. *S* represents the step length. α is the step scale factor. $\oplus$ denotes point-to-point multiplication. *H(u)* represents the Heaviside function. p_α is the switching probability, responsible for balancing local and global search. ε is a randomly selected number from a distribution.

(2) Global Levy flight

The global stochastic process is characterized as a Levy flight.

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \tag{11}$$

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} \frac{1}{s^{1+\lambda}} \quad s \gg s_0 > 0, \ \ 1 < \lambda \le 3 \tag{12}$$

where, x_i^t represents the position of the *i*-th bird's nest in the *t*-th generation, *L* denotes the characteristic range of the problem of interest, and s_0 signifies the minimum step size.

2.4. Evaluation indicators

To evaluate the predictive effectiveness of models for cardiovascular disease classification, five statistical metrics are commonly used: Accuracy, precision, recall, F1 score, and AUC value, as shown in Eqs (13)–(16). Precision measures the proportion of samples accurately predicted as belonging to the positive class. In contrast, recall measures the proportion of samples correctly classified as positive. Recall is also referred to as the true class rate or sensitivity. The prediction results of a binary classification model include TP, FN, FP, and TN. These metrics are shown in Table 3. These indices

provide useful information to evaluate model performance. Additionally, the ROC curve combines recall (sensitivity) and true-negative class rate (specificity) to provide a comprehensive analysis, where different thresholds are tested to obtain recall and true-negative class rates, and then the curve is plotted with recall on the vertical axis and (1–true-negative class rate) on the horizontal axis.

In addition to the aforementioned evaluation metrics, the ROC curve is a composite indicator that reflects the accuracy of the model when distinguishing between positive and negative samples. The ROC curve takes into account the continuous variables of recall (sensitivity) and true-negative class rate (specificity). To obtain the curve, different thresholds are set, and a series of recall and true-negative class rates are calculated. The curve is then plotted with recall as the vertical axis and the false positive rate (1–true-negative class rate) as the horizontal axis.

Table 3. Model evaluation metrics.

Indicators	Description
True Positive (TP)	actual disease and predicted disease
False negative (FN)	actual disease, but predicted normal results
False Positive (FP)	actual normal, but predicted to have disease
True negative (TN)	actual normal and predicted results show normal

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{13}$$

$$recall = \frac{TP}{TP + FN} \tag{14}$$

$$precision = \frac{TP}{TP + FP} \tag{15}$$

$$F1 = \frac{2 \cdot (recall \cdot precision)}{recall + precision} \tag{16}$$

$$FNR = \frac{FN}{FN + TP} \tag{17}$$

where, FNR represents the proportion of actual positive instances that were incorrectly predicted as negative by the model.

3. Proposed methods

3.1. Representation of agents

The swarm intelligence (SI) algorithm is primarily designed for traditional continuous optimization problems. However, to solve optimization problems in various applications, certain components need to be modified accordingly to enhance the algorithm's adaptability and effectiveness. In the case of parameter optimization for the CatBoost algorithm, which is discussed in this paper, the range and meaning of each parameter are often different. Therefore, we adopt a more standardized proxy representation that facilitates algorithm interpretation and optimization.

Assuming that m parameters of the CatBoost algorithm are selected for adjustment, we express the values of these parameters in a proportional manner, as each parameter may have distinct requirements regarding value ranges and types. Specifically, the proportional method involves generating random numbers within the range of [0,1] to represent the proportion of the parameter value range at a given position. Let x_j (j = 1, 2, 3, …, m) denote the value of the proxy at different parameter positions. When evaluating the effect of the parameter combination represented by the proxy, it is necessary to convert the proxy value into the corresponding actual parameter value and input it into the model for calculation.

The formula designed to implement the aforementioned theory is as follows:

$$p_j = low_j + (up_j - low_j) \cdot x_j \tag{18}$$

where p_j as the effective parameter value after conversion, while low_j and up_j represent the lower and upper bounds, respectively, of the default value range for the parameters. By employing the representation described above, the SI algorithm can be applied to the task of finding the optimal parameter combination for the CatBoost algorithm.

3.2. Proposed Cuckoo search algorithm with opposition-based learning (OCS)

Opposition-based learning (OBL) is a method first proposed by Tizhoshz [23], which has proven to be effective in enhancing various meta-heuristic optimization algorithms [24–26]. OBL calculates the opposite solution of a given solution during the evaluation process, providing an additional opportunity to discover a more globally optimal solution. This approach involves evaluating both the feasible solution and its inverse solution. Excellent individuals are selected from the inverse population and the current population to form a new population, thus increasing the diversity of the population. The underlying idea of this strategy is to retain solutions of higher quality while replacing solutions of poorer quality, resulting in the exploration of a larger solution space.

The current population is represented as $X(N) = (X_1, X_2,\ldots,X_n)$, and the opposite population, $OBX(N) = (OBX_1, OBX_2,\ldots, OBX_n)$, based on the OBL strategy, is calculated as follows.

$$OBX_k = (up + low) - X_k \tag{19}$$

The convergence rate of the cuckoo search algorithm is slow, and population diversity diminishes during the later stages of evolution. The OBL strategy is integrated with the CS algorithm to generate an initial population of higher quality. This integration enhances convergence speed, promotes exploration in the search area, and leads to improved population diversity. The proposed strategy is divided into two stages: The initialization stage and the location update stage.

(1) Initialization

The initial population is generated randomly, while the opposition population is created using the OBL strategy. Subsequently, these randomly generated initial individuals and those produced through the OBL strategy are merged into a new population. From this combined population, the top N solutions are selected to form the initial population. A high-quality initial population plays a pivotal role in the convergence and iterative performance of the algorithm, with the OBL strategy guaranteeing its quality.

(2) Updating stage

Utilizing the OBL strategy, we generate opposing nests from the initial nest locations. During the exploration phase, the OBL strategy is employed to expand the search space, enabling comprehensive exploration within it. In the local stochastic process, the OBL strategy can be considered a mutation factor, aiding the algorithm in breaking free from local optimal solutions and facilitating the full exploitation of local space. OBL generates a new nest with a certain probability, denoted as p. First, we generate a random value between 0 and 1. If the random value is less than p, we use OBL to produce an opposing nest based on the existing nest. We then compare and select the nest that retains the best based on fitness values.

For more details, the pseudo code for the OCS algorithm is provided in Algorithm 1.

Algorithm 1 : The proposed OCS Algorithm

Initialize the random nest population $X(N)$
Generate an opposite population $OBX(N)$ by Eq (21)
Calculate the fitness of nest in $\{X(N) \cup OBX(N)\}$
The top N individuals with fitness values are selected as the current population $X(N)$
While ($t<T$)
  for (each nest individual)
    Levy flight Eqs (11) and (12) are used to update the position, replacing it if the new solution is better
    if ($rand < 0.3$)
      Generate an opposite population $OBX(N)$ by Eq (19)
      Calculate the fitness of nest in $OBX(N)$, replacing it if the opposite solution is better
    end if
    Generate a random number, 'r,' following a normal distribution
    if ($r > p_a$)
      Update the position with random walk Eq (10) and replace if the new solution is better
    end if
  end for
  t = t + 1
end while

Suppose the population size is N, the individual dimension is D, and the number of iterations is T. The computational complexity of OCS depends on four processes: Initialization, generation of opposing populations, fitness evaluation, and location updating. The computational complexity of population initialization is O (N × D), that of opposing population generation is O (T × N × D), that of the updating mechanism is O (T × N × D), and that of fitness evaluation is O (T × N). Therefore, the total computational time complexity of the OCS algorithm is O (T × N × D), and the space complexity is O (N × D).

3.3. The cardiovascular disease risk prediction model based on OCSCatBoost

CatBoost is highly regarded by scholars for its strong feature classification ability and high accuracy in various applications. Disease data are often accompanied by classification characteristics, making CatBoost suitable for cardiovascular disease risk prediction. However, it has been observed that CatBoost faces challenges related to parameter setting, which can lead to issues such as falling into local optimal solutions and overfitting. To address this concern, we have introduced the proposed OCS algorithm as an optimization technique to fine-tune the parameters of the CatBoost model.

Several scholars have optimized different model parameters using the improved CS algorithm and verified that the improved CS algorithm's search function can effectively enhance the model's performance [27–30]. By integrating the OCS algorithm into CatBoost, several benefits are achieved. First, the OCS algorithm facilitates faster convergence, allowing the model to reach an optimal solution more quickly. Second, it enhances the global search ability of the model, enabling it to explore a wider range of potential solutions. Finally, the OCS algorithm promotes higher population diversity, reducing the risk of the model getting trapped in local optimal solutions.

Applying the OCS algorithm to optimize CatBoost parameters improves the generalization performance, prediction accuracy, and stability of the model. This optimization technique helps to fine-tune the model's parameters effectively, thereby mitigating the risk of overfitting and improving its ability to handle complex classification tasks.

The OCSCatBoost algorithm is designed to optimize the parameter values of the CatBoost model by employing the OCS algorithm. To accomplish this, the hyperparameters of CatBoost (please refer to Table 4 for information on the function, default value, and value range of the hyperparameters) are mapped to the position matrix of each nest individual in multidimensional space. Furthermore, the fitness function of the OCSCatBoost algorithm model is designed to solve for the individual corresponding to the smallest global value. The fitness function defined in this paper is expressed in Eq (20). Consequently, the location of the nest individual is the global optimal solution, and the optimal parameters of the CatBoost model can be obtained through the mapping relationship (as shown in Eq (18)) between the nest location and corresponding parameters. This leads to the generation of an accurate CatBoost model with optimal parameters. The operating principle of the cardiovascular disease risk prediction model based on OCSCatBoost is depicted in Figure 1.

$$Fitness = -recall \tag{20}$$

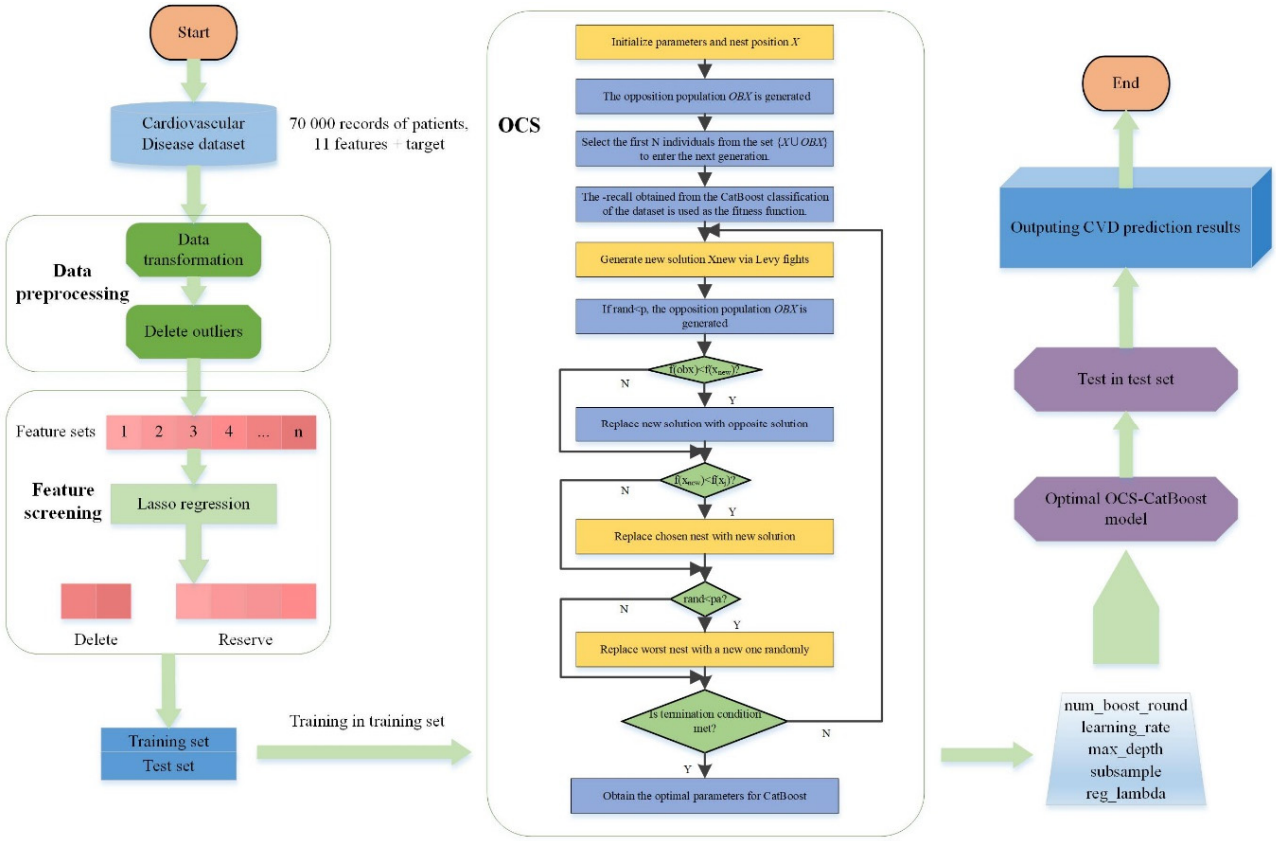

Figure 1. OCSCatBoost flow chart.

The algorithm flow of the OCS algorithm to optimize CatBoost is as follows:

Step 1: Parameter initialization: Set parameters such as the population size N, the search space dimension d, and the maximum number of iterations T. The nest positions are randomly initialized.

Step 2: Define the objective function. Train the CatBoost model to classify the dataset, using -recall as the fitness function to identify the nest individual with the least fitness.

Step 3: Calculate the objective function value for each nest position and compare to obtain the current optimal function value.

Step 4: Levy flight (Eqs (11) and (12)) is employed to update the nest location, followed by the generation of a random number. Determine whether to employ the OBL strategy to generate the opposite nest of the new location based on the magnitude of the random number. If the OBL strategy is employed, the fitness values of the two nests are compared, and the nest with the superior fitness value is selected as the new nest location.

Step 5: A nest location is randomly selected and compared with the fitness value of the new nest location. If the fitness value of the new nest location is better, it is updated and recorded.

Step 6: Generate a random number r. If $r > p_a$, randomly update (Eq (10)) the nest position once; otherwise, keep the nest position unchanged.

Step 7: If the maximum number of iterations is reached, proceed to the next step; otherwise, go back to step 4.

Step 8: Output the global optimal nest position, representing the optimal parameter for the model.

Table 4. Hyperparameters to be optimized.

Hyperparameters	Description	Default value	General range of values
num_boost_round	Number of model integration trees	500	[0,1000]
learning_rate	Model Learning Rate	0.03	[0,1]
max_depth	Maximum depth of the tree in the model	6	[1,16]
subsample	Data proportion in random sampling	0.66	[0.5,0.9]
reg_lambda	L2 regularization coefficient	3	[1,100]

n_estimators: Influences memory usage and training duration. Adjusting the number of trees helps to detect significant overfitting and underfitting issues more effectively.

learning_rate: Reduces the weight of each step to mitigate overfitting.

max_depth: Setting a limit on the tree's depth simplifies the model and decreases the risk of overfitting.

subsample: Controls the sample proportion within each tree. Assigning a value less than 1 to this parameter decreases the tree's variance, thus preventing overfitting.

reg_lambda: Applying L2 regularization to the model's weights reduces model complexity and mitigates the risk of overfitting.

4. Experimental analysis

4.1. Experimental data

The experiment utilized the Kaggle data platform Cardiovascular Disease dataset, which includes 70,000 cases encompassing characteristics like age, gender, height, weight, systolic and diastolic blood pressures, cholesterol, and glucose levels. Additionally, it contains information regarding habits such as smoking, alcohol consumption, physical exercise, and the status of cardiovascular disease, making for a total of 12 characteristics (as illustrated in Table 5).

Table 5. Description of the data set.

Features	Description	Range and symbol description
age	Age (year)	[39,64]
height	Height (cm)	[142.5,186.5]
weight	Body weight (kg)	[39.5,172]
ap_hi	Systolic blood pressure (mmHg)	[60,120]
ap_lo	Diastolic blood pressure (mmHg)	[90,200]
cholesterol	Cholesterol	1 = normal, 2 = above normal, 3 = well above normal
gluc	Blood sugar	1 = normal, 2 = above normal, 3 = well above normal
smoke	Smoking	0 = no smoker, 1 = Smoker
active	Exercise	0 = no exercise, 1 = exercise
alco	Alcohol consumption	0 = no drink alcohol, 1 = drink alcohol
gender	Gender	1 = female, 2 = male
cardio	Cardiovascular disease	0 = normal, 1 = sick

The presented data in Tables 6 and 7 indicate that individuals with cardiovascular disease had higher age, weight, diastolic and systolic blood pressure levels than those without. Additionally, high cholesterol prevalence was highest among individuals with much higher than normal levels, being 28.57% higher than those with normal levels and 16.57% higher than those with higher than normal levels. Blood glucose prevalence was highest among individuals with much higher than normal levels, being 14.05% higher than those with normal levels and 2.88% higher than those with higher than normal levels. Data analysis suggests that smoking and alcohol consumption have an impact on cardiovascular disease, however, the effect is not significant, with a difference of about 2% between those with and without smoking and drinking habits. People who do not exercise are more likely to suffer from cardiovascular disease than those who do.

Table 6. Statistical distribution of continuous fields in the dataset.

Features	Suffering from cardiovascular disease (mean ± std)	Normal (mean ± std)
age	54.47 ± 6.35	51.23 ± 6.77
height	164.31 ± 7.69	164.52 ± 7.51
weight	76.65 ± 14.63	71.57 ± 13.11
ap_hi	133.77 ± 16.94	119.67 ± 12.29
ap_lo	84.52 ± 9.28	78.20 ± 7.99

Table 7. Statistical distribution of discrete fields in the dataset.

Features	Category	Suffering from cardiovascular disease (33568)	Normal (34290)
cholesterol	normal	22172 (43.56%)	28724 (56.44%)
	above normal	5465 (59.58%)	3708 (40.42%)
	well above normal	5931 (76.15%)	1858 (23.85%)
gender	Female	21798 (49.23%)	22476 (50.77%)
	Male	11770 (49.91%)	11814 (50.09%)
gluc	normal	27451 (47.58%)	30248 (52.42%)
	above normal	2935 (58.75%)	2061 (41.25%)
	well above normal	3182 (61.63%)	1981 (38.37%)
smoke	no	30784 (49.72%)	31129 (50.28%)
	yes	2784 (46.83%)	3161 (53.17%)
active	no	7108 (53.37%)	6210 (46.63%)
	yes	26460 (48.51%)	28080 (51.49%)
alco	no	31845 (49.57%)	32403 (50.43%)
	yes	1723 (47.73%)	1887 (52.27%)

4.2. Data pre-processing and feature selection

4.2.1. Missing and abnormal values analysis and processing

By visualizing the raw data, it is possible to observe the potential data distribution, which aids in comprehending the subsequent data processing. Owing to the substantial size of raw data and notable outlier issues, data preprocessing has a considerable impact on the later predictive models. The

framework proposed in this study employs box-line diagrams to detect outliers, as shown in Figure 2. In addition to the age field, all the continuous features in the figure contain outliers. The outliers are discarded based on the physical examination data range of the real population and the criteria of the boxplot.

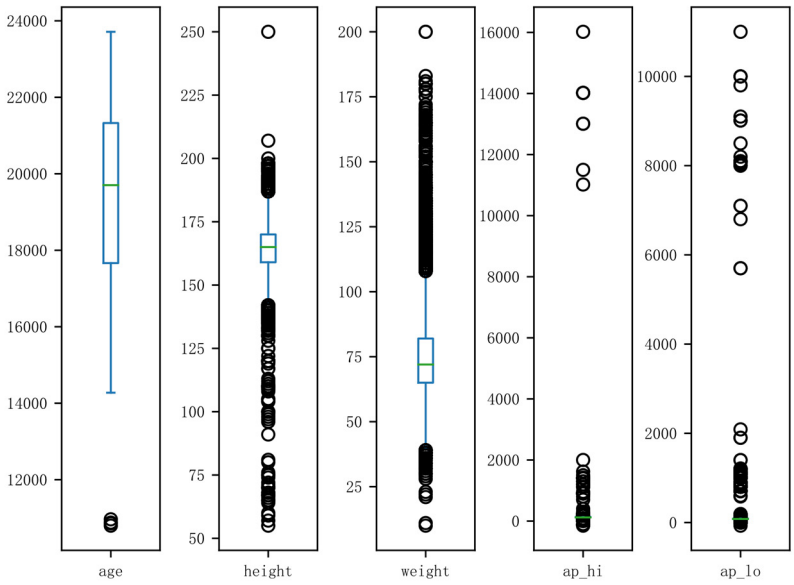

Figure 2. Analysis of outliers in CVD dataset.

4.2.2. Data balance

Following the processing of missing and abnormal data values, the CVD dataset eventually comprised 67,858 samples, including 33,568 cases of cardiovascular disease and 34,290 healthy samples. It is apparent that the number of healthy samples in this dataset is somewhat greater than those afflicted with cardiovascular disease. Therefore, this dataset is considered to be balanced, without further consideration of the balance status required for subsequent experiments.

4.2.3. Feature screening-Lasso regression

The processed data underwent Lasso regression analysis to determine the optimal number of steps responsible for producing the smallest cp value. Thereafter, eight features, including age, weight, ap_hi, ap_lo, cholesterol, smoke, alco, and active, with non-zero coefficients at that step, were retained as critical indicators for cardiovascular disease risk prediction (as illustrated in Table 8).

4.3. Experiment settings

After data preprocessing and feature selection, eight feature variables appearing in Table 7 were designated as input variables. It was decided to use 80% of the data as training set samples and 20% as test set samples. The experimental platform used for this study was Pycharm, and the development language was Python 3.10, with the running environment being Windows 10 Education Edition.

In this paper, PSO [31], SOA [32], and CS are selected for comparison with the OCS algorithm in the experiments. The parameter settings of these SI algorithms are as suggested in the original articles. Additionally, we select KNN, decision tree, logistic regression, grid-search SVM, grid-search XGBoost, the default CatBoost, and grid-search CatBoost as the default machine learning comparison algorithms. The SI algorithms will run independently 30 times with a maximum iteration limit of 50 and a population size of 30. Information about the optimal solutions obtained will be recorded. Conversely, non-heuristic traditional machine learning algorithms only need to run once due to their deterministic nature, resulting in a unique solution.

The crucial hyperparameters for the optimization classification algorithm CatBoost include num_boost_round, learning_rate, max_depth, subsample, and reg_lambda. These parameters have a significant impact on the model's overall performance. Num_boost_round helps expose evident overfitting and underfitting problems, whereas adjusting the learning rate reduces the gradient step size, which affects the training time and helps alleviate the overfitting issue. All SI algorithms aim to find the best parameter combination within the general range of the five key parameters of the CatBoost model.

Table 8. Lasso regression coefficients.

Features	Lasso regression coefficient
age	0.009230188
weight	0.001534384
ap_hi	0.011303342
ap_lo	0.001446262
cholesterol	0.078921746
smoke	–0.00818737
alco	–0.005719433
active	–0.022278004

4.4. Analysis of prediction results

This section will analyze and describe the comparison experiments conducted for the CVD dataset. Figure 3 shows the curve of the iterative process of the heuristic algorithms. It can be seen from Figure 3 that the OCS used in this paper can converge at a faster speed and obtain better results. As seen from the curve in the figure, the improvement of the CS algorithm has essentially met our expectations. In the early stages of iteration, OCS achieves better results with fewer iterations; in the later iterations, the algorithm effectively escapes local optimal solutions, resulting in improved convergence. The parameter combinations of the CatBoost algorithm found by the SI algorithms are shown in Table 9.

Table 9. The parameter combinations found by the SI algorithms.

Hyperparameters	PSO	SOA	CS	OCS
learning_rate	1	0.9933111	1	0.77050522
num_boost_round	988	993	383	524
max_depth	6	9	6	5
subsample	0.812155539	0.747217395	0.69472235	0.89777281
reg_lambda	28	99	85	58

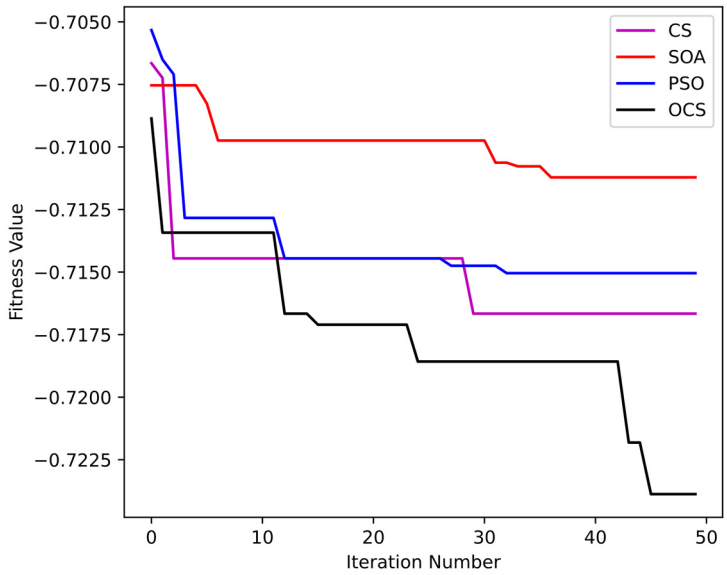

Figure 3. The iteration curves of comparison algorithms.

In addition to the SI algorithms, other ensemble learning methods were selected for comparison in this paper, using the aforementioned evaluation criteria. Table 10 presents the classification outcomes and corresponding evaluation metrics for various classification models applied to the cardiovascular disease dataset. It is important to highlight that in this study, the fitness value of the SI algorithm is calculated based on the training set, while the values presented in Table 10 reflect the results obtained by evaluating the model's performance using the test set. It should be noted that a higher value of evaluation metrics indicates a greater level of effectiveness for the model. Based on the results presented in Table 10, it is evident that the OCSCatBoost model proposed in this paper surpasses the comparative SI algorithm across all indicators. When compared to the default CatBoost and grid search CatBoost, it can be observed that grid search has a certain impact on tuning CatBoost. However, the OCS algorithm yields significant improvements in comparison to the grid search method.

Table 10. Experimental results.

Model	Accuracy	Precision	Recall	F1	AUC	FNR	AUPRC
KNN	69.69%	70.58%	67.65%	69.08%	0.7435	32.38%	0.7708
DT	65.84%	68.65%	58.45%	63.14%	0.6690	41.73%	0.7394
Logistic	72.89%	75.79%	67.35%	71.32%	0.7944	32.65%	0.7974
Grid-search SVM	71.91%	77.34%	62.05%	68.86%	0.7930	37.95%	0.7919
Grid-search XGBoost	73.29%	75.67%	69.75%	72.04%	0.7974	31.25%	0.8003
Default CatBoost	72.78%	74.32%	69.72%	71.95%	0.7861	30.28%	0.7960
Grid-search CatBoost	73.31%	74.63%	70.74%	72.63%	0.8003	29.26%	0.8001
PSO CatBoost	73.44%	74.17%	72.02%	73.08%	0.8013	27.98%	0.8010
SOA CatBoost	73.53%	74.78%	71.12%	72.90%	0.8015	28.88%	0.8018
CS CatBoost	73.56%	74.39%	71.95%	73.15%	0.8018	28.05%	0.8019
OCSCatBoost	73.67%	74.45%	72.17%	73.29%	0.8024	27.83%	0.8027

In comprehensive analysis, SVM achieved a high precision rate of 77.34%, whereas the ensemble learning model ranked second-best with a precision rate of approximately 75%. However, the predictive performance of support vector machines was inconsistent, with a recall rate of only 62.05%. It can be observed from the confusion matrix that the support vector machine exhibits a low proportion of false positive (FP) and a high proportion of false negative (FN), leading to a relatively high precision rate but a relatively low recall rate. In contrast, ensemble learning models exhibit the opposite behavior. However, the integration of the OCS algorithm enhanced CatBoost's prediction results to some extent, reducing FP and FN, thereby increasing recall rates and slightly improving precision. The combined F1 values revealed the ensemble learning model to be more efficient, with relatively high accuracy and recall rates. The F1 values were about 3 percentage points higher compared to KNN and SVM and around 9 percentage points higher than DT. In terms of AUC values highlighted in Table 9, the KNN and DT algorithms had relatively small AUC values, while the integrated algorithm had AUC values hovering around 0.8. This suggests that the model can accurately identify 80% of cardiovascular diseases, indicating improved performance. As a predictive model for cardiovascular disease, the focus was on the model's recall rate, which measures the proportion of affected individuals correctly identified. The OCSCatBoost model had the highest recall rate, with a 4.52, 13.72, 4.82, 10.12, and 2.42 percentage point improvement compared to the KNN, DT, logistic regression, grid-search SVM, and grid-search XGBoost models, respectively. The OCSCatBoost model enhanced the recall rate of CatBoost from 69.72% to 72.17%, indicating an improvement of 2.45 percentage points. Thus, the improved OCSCatBoost model effectively enhanced the recall rate of CatBoost, consequently improving the overall performance and predictive ability of the model. Overall, the OCSCatBoost outperformed other models regarding accuracy, recall, F1 score, AUC, FNR, and AUPRC.

The ROC curves for all algorithms are plotted in Figure 4, where plot (b) provides a partial enlargement of plot (a). It can be observed from Figure 4 that the OCSCatBoost model exhibits the best ROC curve, although its advantages are not as pronounced compared to the integrated algorithm. This suggests that there is room for further improvement in this algorithm. Based on the analysis mentioned above, it is evident that the performance of the OCSCatBoost model proposed in this paper surpasses that of other selected comparison algorithms in most indicators, except for precision. While there is potential for improvement in the proposed algorithm, this experiment successfully demonstrates its superiority and delivers satisfactory results in other metrics. In summary, the experiments conducted in this section validate the effectiveness of OCSCatBoost. Additionally, the analysis of the aforementioned results indicates that the OCS algorithm effectively leverages parameter tuning to enhance the performance of the CatBoost model.

To verify the predictive efficacy and robustness of the classifier, it is essential to employ appropriate evaluation methods. The K-fold cross validation method uses the average performance index as the performance estimation. The experiment in this section adopts a 10-fold cross validation strategy, dividing the data into ten equal subsets. In each iteration, one of the ten subsets is used as the test set, while the remaining subsets are utilized for model learning. This process is repeated until all subsets have been used as test sets once. Such a method provides a reliable estimate of the model's generalization ability. The results are shown in Table 11. As seen in Table 11, OCSCatBoost achieved good results in 10-fold cross-validation, with accuracy, precision, recall, F1 value, and AUC scores of 72.65%, 74.34%, 68.27%, 71.18%, and 0.79 respectively. The corresponding standard deviations were 0.005687, 0.006570, 0.007142, 0.006142, and 0.006438. These values are comparable to the performance observed in the separated test set, indicating a good fit of the model.

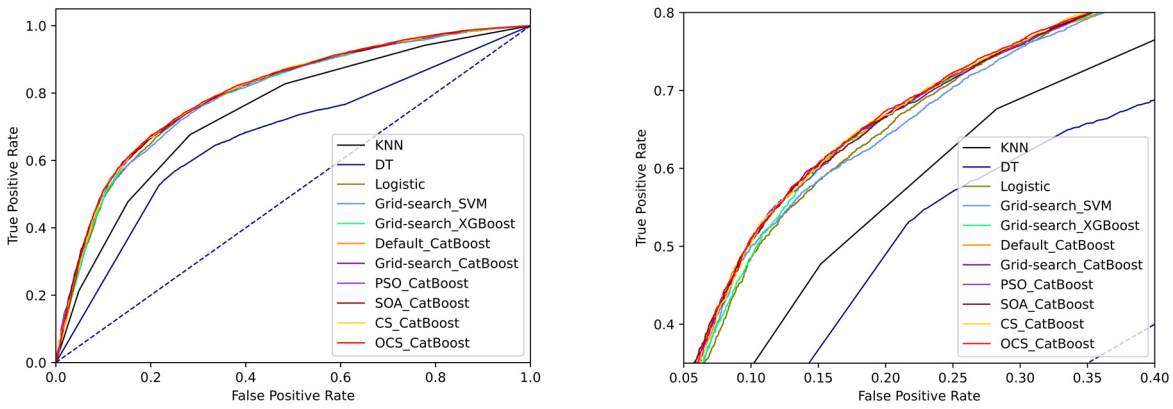

Figure 4. The ROC curves of comparison algorithms.

Table 11. Ten-fold cross-validation results.

	Accuracy	Precision	Recall	F1	AUC
MEAN	72.65%	74.34%	68.27%	71.18%	0.79
STD	0.005687	0.006570	0.007142	0.006142	0.006438

To compare the difference in prediction effect between the proposed algorithm in this paper and the latest algorithm, we compared it with the most recent algorithm using the same dataset, and the experimental results are presented in Table 12. As indicated in the table, the accuracy of the algorithm proposed in this paper surpasses that of other recent algorithms, particularly outperforming deep learning algorithms like ANN and GA-ANN employed by Arroyo et al. In relation to recall and F1 value, it outperforms the XGBH algorithm proposed by Peng et al., thus confirming the effectiveness of this algorithm. Furthermore, since a high recall rate indicates the system's ability to accurately detect the presence of diseases among patients, we place greater emphasis on the algorithm's recall rate and optimizes the model using the recall rate as the fitness function. The experimental results highlight a noticeable improvement in the recall rate, and although the improvement in accuracy is relatively modest, it surpasses other algorithms.

Table 12. Comparison between the proposed method and other state-of-the-art methods.

Authors	Year	Method	Accuracy	Recall	F1
Maiga et al. [33]	2019	RF	73	–	–
Nikam et al. [34]	2020	DT	73.13	–	–
Arroyo et al. [35]	2022	ANN	68.35	–	–
		GA-ANN	73.43	–	–
Mengxiao Peng et al. [36]	2023	XGBH	–	70.4	73
Proposed Method	2023	Lasso+OCS-CatBoost	73.67	72.17	73.29

4.5. Feature importance analysis

In Figure 5, the global importance of each feature of cardiovascular disease is presented. The vertical coordinate represents the eight features, while the horizontal coordinate represents the percentage of each feature's importance to the total importance. According to CVD-OCSCatBoost, ap_hi, age, and cholesterol are considered more important, accounting for 49.93%, 24.51%, and 15.2%, respectively. These three features together account for 90% of the total importance of the feature. On the other hand, smoke and alco were identified as the least important, accounting for 0.94% and 0.81%, respectively.

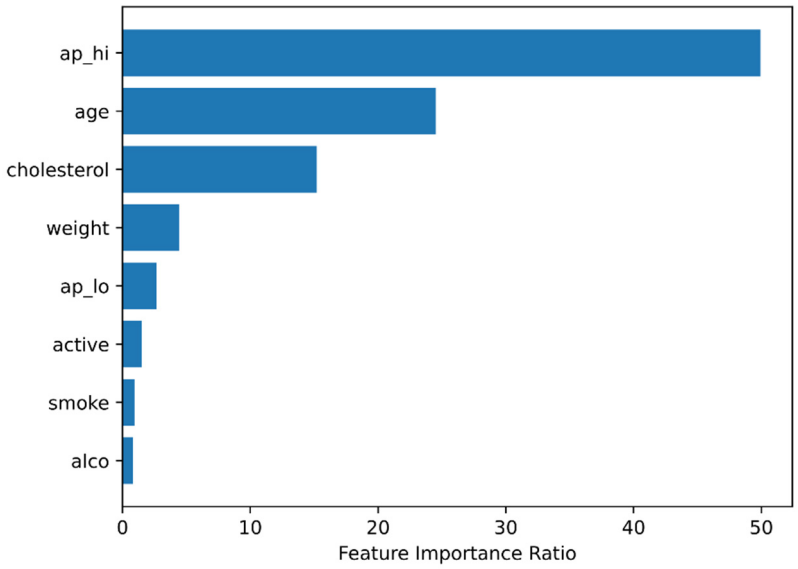

Figure 5. Feature importance.

It is evident from the interpretability of the model that systolic blood pressure, age, and cholesterol levels play a significant role in determining the presence of cardiovascular diseases in the future. These findings align with previous clinical studies that also identified systolic blood pressure, age, and cholesterol levels as important risk factors [10]. It is important to note that these three indicators are considered traditional risk factors utilized in Framingham's 10-year risk score for cardiovascular diseases. Among these factors, systolic blood pressure has the most prominent influence, followed by age and cholesterol levels. In comparison to systolic blood pressure, age, and cholesterol levels, smoking and drinking have a relatively lesser impact on cardiovascular disease risk prediction. However, in real-life scenarios, physical activity, smoking, and drinking are significant factors that can contribute to changes in cardiovascular risk. Therefore, they deserve close attention and control in order to effectively prevent cardiovascular diseases. Additionally, it is worth mentioning that the risk factors identified in the above conclusions have been consistently included in research on cardiovascular diseases in different countries, further supporting the validity of these findings [37–41].

5. Discussion

Although previous studies have achieved higher model accuracy, their datasets have often been

reliant on small and medium-sized queue data, lacking sufficient and valid validation in real-world scenarios. Therefore, we explore the Kaggle cardiovascular disease dataset, consisting of 70,000 instances, to enhance the practicality, efficiency, and robustness of the proposed model.

We have developed a systematic framework for predicting cardiovascular disease risk, aiming for more accurate predictions. The first segment of this framework involves data processing and feature selection. Initially, outliers are identified using a boxplot, and those screened are removed. In the feature selection phase, Lasso regression, chosen for its interpretability in data dimensionality reduction and the ability of the L1 norm to prevent overfitting, is employed. Eight features are selected as the optimal subset for prediction, reducing the model's computational complexity and enhancing prediction effectiveness.

Many scholars have observed that intelligent optimization algorithms can effectively optimize the parameters of machine learning algorithms. In response to the challenge of different parameter combinations affecting the prediction results of the CatBoost model, we propose an enhanced cuckoo search algorithm. In literature [42], an opposition-based cuckoo algorithm is presented, integrating opposition-based learning with cuckoo position updates. This algorithm utilizes either the cuckoo position update mode or opposition-based learning mode, updating the position based on the control parameter CR. The divergence from previous algorithms can be attributed to two main aspects. First, we employ opposition learning to generate the initial population. The best individuals from the two populations are selected as the initial population, enhancing the quality of the initial solution and expediting algorithm convergence. Furthermore, opposition-based learning is applied to the generation of new positions. After the position update of the cuckoo search algorithm, it determines whether to generate the opposition solution of the new position by comparing the generated random number with a set probability. The algorithm selects the superior position as the new one and continues the subsequent selection iteration. This strategy aids the algorithm in breaking free from local optimal positions and fully exploring the solution space.

The second half of the framework employs the OCSCatBoost model for predictions. CatBoost has consistently demonstrated robust feature classification capabilities and high accuracy across various applications, making it particularly suitable for cardiovascular disease risk prediction [11,43]. However, some parameters of the CatBoost model lack high interpretability, and different parameter settings significantly influence prediction outcomes. To attain an optimal state, we utilize the improved cuckoo search algorithm to optimize the model's parameters. The fitness function is defined as the recall, and the optimal parameter combination is determined through continuous iteration. In comparison with three other optimization algorithms, experimental results demonstrate that the OCSCatBoost model exhibits superior predictive accuracy in cardiovascular disease risk prediction. Additionally, we analyze computational complexity, revealing that the proposed model completes training and testing in an average of 2 seconds, with each iteration using OCS parameter optimization taking approximately 360 seconds.

We believe this study holds significant value for predicting CVD risk. For patients, it can effectively guide lifestyle changes, reducing disease risks. Likewise, for doctors, it aids in identifying potential risks and trends, enabling the screening of high-risk groups, and reducing misdiagnosis rates. However, the study does have limitations and suggests directions for future research. First, the absence of clear cholesterol threshold ranges impedes an accurate assessment of specific cholesterol values' impact on cardiovascular disease. Second, cardiovascular disease risk assessment data exists in diverse structured forms, encompassing clinical, imaging, pathological, and multiomics data. Yet, the method

proposed in this paper is solely validated on datasets containing partial types. In future work, we aim to integrate the proposed model into telemedicine platforms or mobile applications, potentially enhancing access to early diagnosis, particularly for individuals facing geographic or financial barriers. The field of cardiovascular disease risk prediction stands to benefit significantly from the ongoing advancements in interaction prediction research within various realms of computational biology. Particularly, the exploration of genetic markers and non-coding RNAs (ncRNAs) interaction predictions, such as miRNA-lncRNA interactions, holds promise for providing deeper insights into the molecular mechanisms underlying cardiovascular diseases. We will incorporate GFPA [44], scAAGA [45], MDA-AENMF [46], and other models [47–51] that reveal potential links between genes and diseases into our study. Embracing these computational approaches and continuously evolving our understanding of genetic and ncRNAs interactions will undoubtedly drive innovation and transformative breakthroughs in the prevention and treatment of cardiovascular diseases.

6. Conclusions

Complex problems like cardiovascular disease risk prediction and diagnosis require physician expertise and medical experience, and machine learning algorithms can assist physicians in making quick and accurate decisions. We propose a CVD-OCSCatBoost framework for predicting cardiovascular disease risk, using the Lasso regression model to determine input features and the OCS algorithm to optimize the CatBoost parameters, improving the accuracy and recall rate of cardiovascular disease risk prediction. The results reveal that the OCSCatBoost model outperforms other algorithms in terms of classification accuracy, recall, and algorithm stability, validating the superiority of the algorithm. The proposed framework provides efficient and accurate classification and has significant application value. An importance analysis of each feature has also been conducted, highlighting the crucial features that need attention for cardiovascular disease prevention and thereby providing effective guidance for preventive measures.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Project of Ningxia Natural Science Foundation (grant number 2021AAC03195), the Construction Project of First-class Subjects in Ningxia Higher Education (grant number NXYLXK2017B09), Nanjing Securities supports basic discipline research projects (grant number NJZQJCXK202201).

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. W. B. Kannel, D. Mcgee, T. Gordon, A general cardiovascular risk profile: The Frmingham study, *Am. J. Cardiol.*, **38** (1976), 46–51. https://doi.org/10.1016/0002-9149(76)90061-8
2. R. M. Conroy, K. Pyoral, A. P. Fitzgerald, S. Sans, A. Menotti, G. De Backer, et al., Estimation of ten-year risk of fatal cardiovascular disease in Europe: The SCORE project, *Eur. Heart J.*, **24** (2003), 987–1003. https://doi.org/10.1016/S0195-668X(03)00114-3
3. C. Hippisley, Derivation and validation of QRISK, a new cardiovascular diseaserisk score for the United Kingdom: Prospective open cohort study, *BMJ*, **335** (2007), 136. https://doi.org/10.1136/bmj.39261.471806.55
4. S. F. Weng, J. Reps, J. Kai, Can machine-learning improve cardiovascular risk prediction using routine clinical data, *PLoS ONE*, **12** (2017), e0174944. https://doi.org/10.1371/journal.pone.0174944
5. A. C. Dimopoulos, M. Nikolaidou, F. F. Caballero, Machine learning methodologies versus cardiovascular risk scores, in predicting disease risk, *BMC Med. Res. Methodol.*, **18** (2018). https://doi.org/10.1186/s12874-018-0644-1
6. W. Huang, T. W. Ying, W. L. C. Chin, Application of ensemble machine learning algorithms on lifestyle factors and wearables for cardiovascular risk prediction, *Sci. Rep.*, **12** (2022), 1033. https://doi.org/10.1038/s41598-021-04649-y
7. M. Ordikhani, M. S. Abadeh, C. Prugger, An evolutionary machine learning algorithm for cardiovascular disease risk prediction, *PLoS ONE*, **17** (2022), e0271723. https://doi.org/10.1371/journal.pone.0271723
8. M. Pal, S. Parija, G. Panda, K. Dhama, R. K. Mohapatra, Risk prediction of cardiovascular disease using machine learning classifiers, *Open Med.*, **17** (2022), 1100–1113. https://doi.org/10.1515/med-2022-0508
9. L. R. Guarneros-Nolasco, N. A. Cruz-Ramos, G. Alor-Hernández, L. Rodríguez-Mazahua, J. L. Sánchez-Cervantes, Identifying the main risk factors for cardiovascular diseases prediction using machine learning algorithms, *Mathematics*, **9** (2021), 2537. https://doi.org/10.3390/math9202537
10. M. M. Ali, B. K. Paul, K. Ahmed, F. M. Bui, J. M. W. Quinn, M. A. Moni, Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison, *Comput. Biol. Med.*, **136**(2021), 104672. https://doi.org/10.1016/j.compbiomed.2021.104672
11. K. Kanagarathinam, D. Sankaran, R. Manikandan, Machine learning-based risk prediction model for cardiovascular disease using a hybrid dataset, *Data Knowl. Eng.*, **140** (2022), 102042. https://doi.org/10.1016/j.datak.2022.102042
12. J. M. Sung, I. J. Cho, D. Sung, S. Kim, Development and verification of prediction models for preventing cardiovascular diseases, *PLoS ONE*, **14** (2019), e0222809. https://doi.org/10.1371/journal.pone.0222809
13. Y. Pan, M. Fu, B. Cheng, X. Tao, J. Guo, Enhanced deep learning assisted convolutional neural network for heart disease prediction on the internet of medical things platform, *IEEE Access*, **8** (2020), 189503–189512. https://doi.org/10.1109/ACCESS.2020.3026214
14. S. K. Pandey, R. R. Janghel, Automatic detection of arrhythmia from imbalanced ECG database using CNN model with SMOTE, *Australas. Phys. Eng. Sci. Med.*, **42** (2019), 1129–1139. https://doi.org/10.1007/s13246-019-00815-9

15. L. Ali, A. Rahman, A. Khan, M. Zhou, A. Javeed, J. A. Khan, An automated diagnostic system for heart disease prediction based on $\chi 2$ statistical model and optimally configured deep neural network, *IEEE Access*, **7** (2019), 34938–34945. https://doi.org/10.1109/ACCESS.2019.2904800
16. I. D. Mienye, Y. Sun, Z. Wang, An improved ensemble learning approach for the prediction of heart disease risk, *Inf. Med. Unlocked*, **20** (2020), 100402. https://doi.org/10.1016/j.imu.2020.100402
17. S. Pandya, T. R. Gadekallu, P. K. Reddy, W. Wang, M. Alazab, InfusedHeart: A novel knowledge-infused learning framework for diagnosis of cardiovascular events, *IEEE Trans. Comput. Soc. Syst.*, **2022** (2022). https://doi.org/10.1109/TCSS.2022.3151643
18. P. Srinivas, R. Katarya, HyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost, *Biomed. Signal Process. Control*, **73** (2022), 103456. https://doi.org/10.1016/j.bspc.2021.103456
19. V. Baviskar, M. Verma, P. Chatterjee, G. Singal, T. R. Gadekallu, Optimization using internet of agent based stacked sparse autoencoder model for heart disease prediction, *Exp. Syst.*, **2023** (2023), e13359. https://doi.org/10.1111/exsy.13359
20. X. Wei, C. Rao, X. Xiao, L. Chen, M. Goh, Risk assessment of cardiovascular disease based on SOLSSA-CatBoost model, *Exp. Syst. Appl.*, **219** (2023), 119648. https://doi.org/10.1016/j.eswa.2023.119648
21. A. S. Kumar, R. Rekha, An improved hawks optimizer based learning algorithms for cardiovascular disease prediction, *Biomed. Signal Process. Control*, **81** (2023), 104442. https://doi.org/10.1016/j.bspc.2022.104442
22. X. S. Yang, Cuckoo search via L´evy flights, in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, (2009), 210–214. https://doi.org/10.1109/NABIC.2009.5393690
23. H. R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in *Proceedings of IEEE International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents*, *Web Technologies and Internet Commerce(CIMCA-IAWTIC06*, (2005), 695–701. https://doi.org/10.1109/cimca.2005.1631345
24. A. A. Ewees, A. E. Mohamed, E. H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, *Exp. Syst. Appl.*, **112** (2018), 156–172. https://doi.org/10.1016/j.eswa.2018.06.023
25. X. Yu, W. Xu, C. Li, Opposition-based learning grey wolf optimizer for global optimization, *Knowl.-Based Syst.*, **226** (2021), 107139. https://doi.org/10.1016/j.knosys.2021.107139
26. M. Khishe, Greedy opposition-based learning for chimp optimization algorithm, *Artif. Intell. Rev.*, **56** (2022), 7633–7663. https://doi.org/10.1007/s10462-022-10343-w
27. M. Imran, S. Khan, H. Hlavacs, Intrusion detection in networks using cuckoo search optimization, *Soft Comput*., **26** (2022), 10651–10663. https://doi.org/10.1007/s00500-022-06798-2
28. B. Jia, B. Yu, Q. Wu, Adaptive affinity propagation method based on improved cuckoo search, *Knowl.-Based Syst.*, **111** (2016), 27–35. https://doi.org/10.1016/j.knosys.2016.07.039
29. S. Chakraborty, K. Mali, Fuzzy and elitist cuckoo search based microscopic image segmentation approach, *Appl. Soft Comput.*, **130** (2022), 109671. https://doi.org/10.1016/j.asoc.2022.109671
30. P. N. Maddaiah, P. P. Narayanan, An improved Cuckoo search algorithm for optimization of artificial neural network training, *Neural Process. Lett.*, **2023** (2023), 1–28. https://doi.org/10.1007/s11063-023-11411-0

31. R. Eberhart, K. James, A new optimizer using particle swarm theory, in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, (1995), 39–43. https://doi.org/10.1109/mhs.1995.494215
32. G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for largescale industrial engineering problems, *Knowl.-Based Syst.*, **165** (2019), 169–196. https://doi.org/10.1016/j.knosys.2018.11.024
33. J. Maiga, G. G. Hungilo, Comparison of machine learning models in prediction of cardiovascular disease using health record data, in *2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, (2019), 45–48. https://doi.org/10.1109/ICIMCIS48181.2019.8985205
34. A. Nikam, S. Bhandari, A. Mhaske, S. Mantri, Cardiovascular disease prediction using machine learning models, in *2020 IEEE Pune Section International Conference (PuneCon)*, (2020), 22–27. https://doi.org/10.1109/PuneCon50868.2020.9362367
35. J. C. T. Arroyo, A. J. P. Delima, An optimized neural network using genetic algorithm for cardiovascular disease prediction, *J. Adv. Inf. Technol.*, **13** (2022), 95–99. https://doi.org/10.12720/jait.13.1.95-99
36. M. Peng, F. Hou, Z. Cheng, T. Shen, K. Liu, C. Zhao, et al., A cardiovascular disease risk score model based on high contribution characteristics, *Appl. Sci.*, **13** (2023), 893. https://doi.org/10.3390/app13020893
37. T. B. Olesen, M. Pareek, The influence of age and sex on the prognostic importance of traditional cardiovascular risk factors, selected circulating biomarkers and other markers of subclinical cardiovascular damage, *Curr. Opin. Cardiol.*, **38** (2023), 21–31. https://doi.org/10.1097/hco.0000000000001005
38. E. Harold, P. R. Bays, E. E. Taub, Ten things to know about ten cardiovascular disease risk factors, *Am. J. Prev. Cardiol.*, **5** (2021), 100149. https://doi.org/10.1016/j.ajpc.2021.100149
39. C. Phanish, B. Radhika, Assessing the risk factors associated with cardiovascular disease, *Eur. J. Prev. Cardiol.*, **25** (2018), 932–933. https://doi.org/10.1177/2047487318778652
40. A. Arafa, H. H. Lee, E. S. Eshak, K. Shirai, K. Liu, J. Li, et al., Modifiable risk factors for cardiovascular disease in Korea and Japan, *Korean Circ. J.*, **51** (2021), 643–655. https://doi.org/10.4070/kcj.2021.0121
41. M. George, K. George, T. Athanasios, Cardiovascular disease in Greece; the latest evidence on risk factors, *Hell. J. Cardiol.*, **60** (2019), 271–275. https://doi.org/10.1016/j.hjc.2018.09.006
42. P. Zhao, H. Li, Opposition-based Cuckoo search algorithm for optimization problems, in *2012 Fifth International Symposium on Computational Intelligence and Design*, (2012), 344–347. https://doi.org/10.1109/ISCID.2012.93
43. N. A. Baghdadi, S. M. F. Abdelaliem, A. Malki, I. Gad, A. Ewis, E. Atlam, Advanced machine learning techniques for cardiovascular disease early detection and diagnosis, *J. Big Data*, **10** (2023). https://doi.org/10.1186/s40537-023-00817-1
44. H. Huan, F. Zhen, L. Hai, J. Cheng, J. Lyu, Y. Zhang, et al., Gene function and cell surface protein association analysis based on single-cell multiomics data, *Comput. Biol. Med.*, **157** (2023), 106733. https://doi.org/10.1016/j.compbiomed.2023.106733
45. R. Meng, S. Yin, J. Sun, H. Hu, Q Zhao, ScAAGA: Single cell data analysis framework using asymmetric autoencoder with gene attention, *Comput. Biol. Med.*, **165** (2023), 107414. https://doi.org/10.1016/j.compbiomed.2023.107414

46. H. Gao, J. Sun, Y. Wang, Y. Lu, L. Liu, Q. Zhao, et al., Predicting metabolite–disease associations based on auto-encoder and non-negative matrix factorization, *Briefings Bioinf.*, **24** (2023), bbad259. https://doi.org/10.1093/bib/bbad259
47. W. Wang, L. Zhang, J. Sun, Q. Zhao, J. Shuai, Predicting the potential human lncRNA–miRNA interactions based on graph convolution network with conditional random field, *Briefings Bioinf.*, **23** (2022), bbac463. https://doi.org/10.1093/bib/bbac463
48. L. Zhang, P. Yang, H. Feng, Q. Zhao, H. Liu, Using network distance analysis to predict lncRNA–miRNA interactions, *Interdiscip. Sci. Comput. Life Sci.*, **13** (2021), 535–545. https://doi.org/10.1007/s12539-021-00458-z
49. F. Sun, J. Sun, Q. Zhao, A deep learning method for predicting metabolite–disease associations via graph neural network, *Briefings Bioinf.*, **23** (2022), bbac266. https://doi.org/10.1093/bib/bbac266
50. T. Wang, J. Sun, Q. Zhao, Investigating cardiotoxicity related with hERG channel blockers using molecular fingerprints and graph attention mechanism, *Comput. Biol. Med.*, **153** (2023), 106464. https://doi.org/10.1016/j.compbiomed.2022.106464
51. Z. Chen, L. Zhang, J. Sun, R. Meng, S. Yin, Q. Zhao, DCAMCP: A deep learning model based on capsule network and attention mechanism for molecular carcinogenicity prediction, *J. Cell Mol. Med.*, **27** (2023), 3117–3126. https://doi.org/10.1111/jcmm.17889

AIMS Press