electronics

MDPI

Article

A Novel Framework for Risk Warning That Utilizes an Improved Generative Adversarial Network and Categorical Boosting

Yan Peng, Yue Liu, Jie Wang and Xiao Li *

School of Management, Capital Normal University, Beijing 100056, China; pengyan@cnu.edu.cn (Y.P.); 2222902038@cnu.edu.cn (Y.L.); wangjie@cnu.edu.cn (J.W.)
* Correspondence: lixiao@cnu.edu.cn; Tel.: +86-132-6165-0877

Abstract: To address the problems of inadequate training and low precision in prediction models with small-sample-size and incomplete data, a novel SALGAN-CatBoost-SSAGA framework is proposed in this paper. We utilize the standard K-nearest neighbor algorithm to interpolate missing values in incomplete data, and employ EllipticEnvelope to identify outliers. SALGAN, a generative adversarial network with a self-attention mechanism of label awareness, is utilized to generate virtual samples and increase the diversity of the training data for model training. To avoid local optima and improve the accuracy and stability of the standard CatBoost prediction model, an improved Sparrow Search Algorithm (SSA)–Genetic Algorithm (GA) combination is adopted to construct an effective CatBoost-SSAGA model for risk warning, in which the SSAGA is used for the global parameter optimization of CatBoost. A UCI heart disease dataset is used for heart disease risk prediction. The experimental results show the superiority of the proposed model in terms of accuracy, precision, recall, and F1-values, as well as the AUC.

Keywords: small-sample datasets; data augmentation; improved sparrow search algorithm; novel risk warning; GAN

Citation: Peng, Y.; Liu, Y.; Wang, J.; Li, X. A Novel Framework for Risk Warning That Utilizes an Improved Generative Adversarial Network and Categorical Boosting. *Electronics* **2024**, *13*, 1538. https://doi.org/10.3390/electronics13081538

Academic Editor: Ping-Feng Pai

Received: 2 February 2024
Revised: 29 March 2024
Accepted: 10 April 2024
Published: 18 April 2024

1. Introduction

In reality, many application scenarios contain very few labeled samples, and also many datasets are incomplete with missing information. For example, in the field of medical diagnoses, doctors may only obtain data from a few patients, which is particularly common in the diagnoses of rare or emerging diseases. In the financial field, especially in personal credit scoring or fraud detection, it is necessary to use limited-sample data to predict credit risks or identify fraudulent activity. Therefore, developing methods for learning from small and incomplete samples is an urgent need. Model fine-tuning, data augmentation, and transfer learning are the mainstream technologies used to solve such problems [1]. Meanwhile, a significant body of research has demonstrated the potential of machine learning models in risk prediction on small-sample datasets. However, several challenges still need to be addressed.

Fine-tuning and transfer learning methods have problems with model overfitting when the target dataset and the source dataset are dissimilar. Data augmentation approaches may introduce noise or alter features. The learning ability of a single machine learning model varies across different datasets, resulting in inconsistent prediction performance and limited generalization capability [2]. Although deep learning models can achieve promising prediction accuracy, they require a significant amount of data and a complex training process, and are prone to issues such as gradient vanishing or exploding and poor interpretability [3]. On the other hand, ensemble learning has achieved good results in multi-class prediction tasks, and the appropriate selection of hyperparameter tuning methods can improve the prediction accuracy of the model. However, there is still room for system optimization.

Thus, in this paper, we analyze small-size data and construct a risk-warning framework called SALGAN-CatBoost-SSAGA, which consists of two main parts: SALGAN data augmentation and an improved CatBoost prediction model (i.e., CatBoost-SSAGA). The method of the label-aware self-attention mechanism-based generative adversarial network (SALGAN) learns the various implicit correlations and dependencies of different types of labeled data, contributing to the generation of highly realistic data instances. The prediction model is constructed by CatBoost, which is integrated with an enhanced SSAGA for global parameter optimization. We conduct experiments on a small-sample dataset of heart disease, which demonstrates the effectiveness of the proposed model.

The main contributions of this paper are as follows:

(1) We introduce a novel SALGAN-CatBoost-SSAGA framework for small-sample risk warning.
(2) We propose a SALGAN that generates virtual data according to label types, effectively enhancing small-sample data.
(3) We present a hybrid algorithm, the SSAGA, which combines the SSA and GA to optimize the parameters of the standard CatBoost model, which could improve the prediction accuracy of the CatBoost model.
(4) We conduct small-sample prediction experiments using the UCI heart disease dataset, which demonstrates the advantages of the proposed model in terms of its classification accuracy, recall, precision, F1-score, and AUC, indicating its effectiveness in predicting small-sample data.

2. Related Work

2.1. Few-Shot Learning

Few-shot learning [4] aims to construct machine learning models that can solve real-world problems using a limited amount of training data. In few-shot learning, there are typically two main challenges: inter-class variance and intra-class variance [5]. Currently, few-shot learning primarily includes methods based on model fine-tuning, transfer learning, and data augmentation.

The method of model fine-tuning [6] usually involves pretraining the network model on a large dataset, then fixing some parameters and fine-tuning specific parameters of the network model on a small-sample dataset to obtain a fine-tuned model. Transfer learning [7] helps to train reliable decision functions in the target domain by transferring knowledge from auxiliary sources. This approach addresses the learning problem when the sample data in the target domain are either unlabeled or consist of only a limited number of labeled samples. However, since the target sample set and the source sample set may not be similar, the two methods may lead to overfitting problems of the machine learning model on the target sample set.

Data augmentation [8] includes methods based on unlabeled data, feature enhancement and data synthesis. Methods based on unlabeled data involve using large amounts of unlabeled data to expand the original small-sample dataset, such as semi-supervised learning [9] and transductive learning [10]. Feature enhancement involves adding features in the feature space of the original sample to increase the diversity of features for classification. Schwartz et al. [11] proposed the Delta encoder, which synthesizes new samples for unseen categories by observing a small number of samples and uses these synthetic samples to train a classifier. Data synthesis refers to the synthesis of new labeled data for small-sample categories to augment the training data, and a commonly used method for this is the use of generative adversarial networks (GANs) [12]. A GAN is a deep learning model proposed by Ian J. Goodfellow and his colleagues in 2014. It consists of two mutually competitive networks: a generator G and a discriminator D. Y Kataoka et al. [13] reported image generation that leverages the effectiveness of attention mechanisms and the GAN approach. N Park et al. [14] proposed table-GAN, which uses GANs to synthesize fake tables that are statistically similar to the original tables but do not cause information leakage.

2.2. CatBoost Algorithm

Categorical Boosting (CatBoost) [15] is an enhanced framework of Gradient Boosted Decision Tree (GBDT), which is a commonly used classification algorithm. It is based on a symmetric decision tree as the base learner, and effectively suppresses the gradient bias and prediction bias existing in the gradient decision tree by introducing a rank-boosting strategy. Moreover, CatBoost is characterized by its robust support for categorical variables and exceptional predictive accuracy. Li et al. [16] developed a weather prediction model combining wavelet denoising and CatBoost, which is faster and more accurate than LSTM and Random Forest. PS Kumar et al. [17] developed a CatBoost ensemble technique based on GBDT, specifically for the prediction of early-stage diabetes. Comparative experiments with other machine learning methods have demonstrated that CatBoost excels in various performance metrics. Wang et al. [18] investigated the efficacy of CatBoost in predicting severe hand-foot-and-mouth disease, finding it to have an accuracy rate of 87.6%, higher than other algorithms.

However, despite CatBoost's effectiveness, its extensive hyperparameter space can significantly impact classification results. Therefore, it is crucial to employ parameter optimization algorithms to fine-tune the hyperparameters, enhancing CatBoost's full potential in diverse applications. Cheng et al. [19] used the grid search method with cross-validation to optimize the super parameters of catboost, respectively, and the model showed the highest accuracy in a suspended solids experiment. Jin et al. [20] trained CatBoost, Random Forest, and other models through cyclic training and adjusting the given parameters, and then used the cross-validation method to conduct a grid search for secondary adjustments. Their experimental results show that the prediction effect of CatBoost after two rounds of optimization was significantly higher than that of other models.

2.3. Hyperparameter Optimization Algorithm

The hyperparameters should be determined before the model runs, and they have a relatively important impact on the performance of the model. Currently, there are many optimization methods available, such as the grid search (GS) method [21] and the Bayesian optimization algorithm (BOA) [22]. Some studies also employ swarm intelligence optimization algorithms such as Grey Wolf Optimization (GWO) [23], the Genetic Algorithm (GA) [24], and the Sparrow Search Algorithm (SSA) [25]. However, there are still some deficiencies in these parameter optimization methods, such as the fact that cross-validation and grid search methods do not consider super parameters or only consider a small number of common super parameters; the BOA and GWO do not grasp the global trend of the prediction performance of the model, and are prone to falling into local optimization; the results of the GA are affected by the initial advantages and disadvantages, and cannot eliminate the randomness of the optimization results.

Among them, the SSA is a preferable choice due to the advantages of its simple structure and flexibility, but its optimization ability and convergence speed still need to be improved [26]. Therefore, many studies have focused on optimizing the SSA. Ou et al. [27] improved the SSA by using the good point set method and reducing nonlinear inertia weights to prevent the SSA from falling into local optima. Wang et al. [28] employed a multi-sample learning strategy to assist the SSA in achieving a better optimization capability and convergence speed.

3. Preliminaries

3.1. CatBoost

In the GBDT algorithm, a commonly used method for dealing with categorical features is to replace them with the average value of the category feature label, which can be expressed as Equation (1):

$$\overset{\wedge}{\mathrm{x}}_k^i = \frac{\sum_{j=1}^{p-1} [x_{j,k} = x_{i,k}] \bullet Y_i}{\sum_{j=1}^{n} [x_{j,k} = x_{i,k}]} \tag{1}$$

where, x_k^i represents the ith category feature value of the kth training sample. If a feature has fewer category values, converting it to a numerical value is equivalent to assigning the label value of that record. This scenario commonly leads to overfitting issues.

In response to this, the CatBoost algorithm addresses a specific value within the categorical features. When converting each feature to a numerical type for each sample, the algorithm calculates the average based on the category label preceding the sample, incorporating prior knowledge and weight coefficients. This approach aims to reduce the noise caused by low-frequency features in the categorical features, as shown in Equation (2):

$$\hat{x}_k^i = \frac{\sum_{j=1}^{p-1} [x_{\varepsilon_{j,k}} = x_{\varepsilon_{j,k}}] Y_{\varepsilon_j} + \mathrm{ap}}{\sum_{j=1}^{p-1} [x_{\varepsilon_{j,k}} = x_{\varepsilon_{p,k}}] + \mathrm{a}} \tag{2}$$

where, $\hat{x}_k^i$ represents the statistical target variable, $x_{\varepsilon_{j,k}}$ denotes the categorical feature, Y_{ε_j} corresponds to the label value of the feature, a denotes the weight coefficient, and p represents the prior term.

3.2. SSA

The SSA is a swarm intelligence optimization algorithm that simulates the behavior of sparrows foraging and avoiding predators. In this algorithm, the sparrow population is divided into two categories: discoverers and followers.

Discoverers are responsible for searching for food and providing information about foraging areas to the entire population. The position of the discoverer is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp(\frac{i}{\alpha \cdot iter_{\max}}), R < S \\ X_{i,j}^t + Q \cdot L, R \geq S \end{cases} \tag{3}$$

where $X_{i,j}^t$ represents the position of the ith sparrow in dimension j at iteration t; α is a random number in the range of (0, 1]; $iter_{\max}$ is the maximum number of iterations, a constant value; $\exp(x)$ denotes the exponential function with base e; and Q is a random number following a normal distribution. R is the alert value, and if R is smaller than the safety value S, it indicates that the sparrow's environment is relatively safe, allowing for extensive foraging exploration. Conversely, if R is larger than S, it indicates that some individuals have detected predators and issued an alarm to move towards a safe zone, ensuring the safety of the population.

The other individuals in the population are followers, who come to forage based on the information provided by the discoverers. The update of their positions is expressed as Equation (4):

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp(\frac{x_w - x_{i,j}^t}{\alpha \cdot iter_{\max}}), i \geq \frac{n}{2} \\ X_b^{t+1} + \left| X_{i,j}^t - X_b^{t+1} \right| \cdot A^T (AA^T)^{-1} \cdot L, otherwise \end{cases} \tag{4}$$

where X_P represents the current optimal position of the discoverer; A is a $1 \times d$ matrix with elements randomly assigned as 1 or -1. The variable i in $i \geq \frac{n}{2}$ represents the quantity of followers within the population, with n indicating the population's size. This condition is commonly referred to as the "hunger judgment" and is employed to assess whether an individual necessitates foraging.

In addition, to effectively avoid predator attacks, the algorithm also introduces an early warning mechanism that selects a certain proportion of individuals as scouts, who are responsible for detecting and warning of potential threats. The positions of the scouts are updated using Equation (5):

$$X_{i,j}^{t+1} = \begin{cases} X_b^t + \beta \left| X_{i,j}^t - X_b^t \right|, f_i > f_g \\ X_{i,j}^t + k\left(\frac{\left|X_{i,j}^t - X_w^t\right|}{(f_i - f_w) + \gamma}\right), f_i = f_g \end{cases} \quad (5)$$

where β denotes the learning rate, which is a normally distributed random number that controls the speed at which the model updates parameters during each iteration. The random variable k is a value within the interval $[-1, 1]$ used to control the direction of sparrow movement and is a small constant employed to prevent division by zero. Finally, f_i represents the value of the objective function at the current position.

4. Model Construction

4.1. Framework

The overall framework structure of the SALGAN-CatBoost-SSAGA framework is shown in Figure 1, and mainly consists of four parts: data cleaning, data augmentation, risk-warning prediction, and model evaluation.

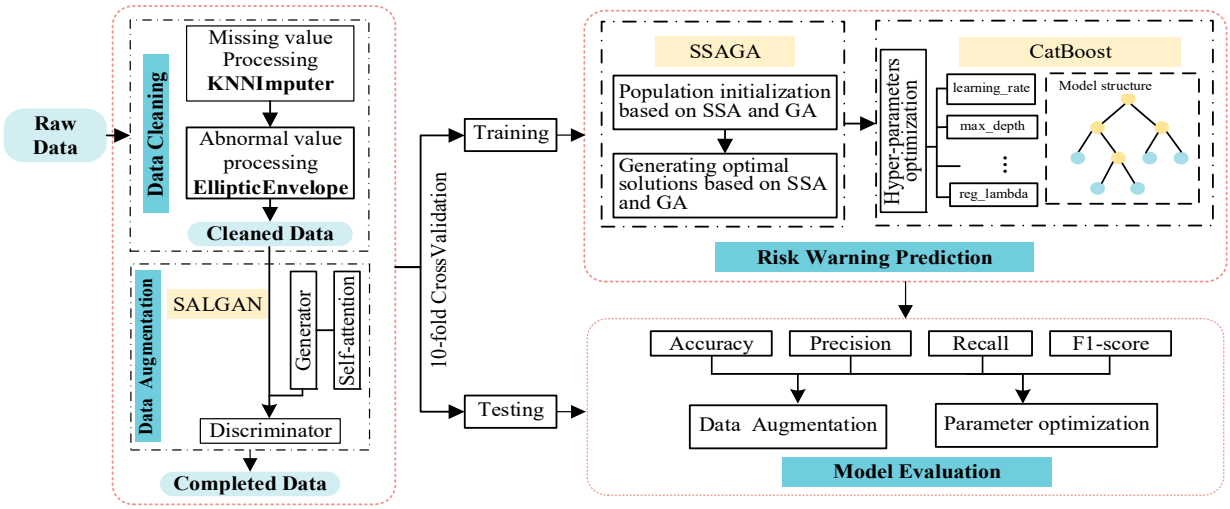

Figure 1. Framework of SALGAN-CatBoost-SSAGA.

The workflow of the SALGAN-CatBoost-SSAGA model is shown as follows:

(1) Data cleaning

For feature incompleteness, the KNNImputer algorithm is utilized to interpolate missing data from the original datasets, while the EllipticEnvelope algorithm is applied to remove outliers, thus completing the datasets.

(2) Data augmentation

In order to address the challenges associated with small-sample learning, this study implements a generative adversarial network that is utilized with a label-aware self-attention mechanism (SALGAN), aiming to generate high-quality synthetic sample data through this methodology.

(3) Risk-warning prediction

We improve the SSA using the GA and apply the improved SSAGA to optimize the hyperparameters of CatBoost, including n_estimators, learning_rate, max_depth, reg_lambda, and subsample. We then train the classifier.

(4) Model evaluation

We propose to test the performance of the evaluation indexes of the new framework, using this as the basis for the evaluation of the SALGAN-CatBoost-SSAGA risk-warning model.

SALGAN-CatBoost-SSAGA consists of two main parts: the SALGAN data augmentation method and the CatBoost-SSAGA prediction model. It is primarily used to address prediction problems in small-sized datasets. Firstly, for data augmentation, the SALGAN combines self-attention with a GAN, and, more importantly, introduces label awareness. The SALGAN not only autonomously adapts to various data features and relationships but also enhances the generator's ability to effectively grasp and replicate data distributions across different labels, thereby generating more realistic and contextually relevant synthetic data samples. By using the SALGAN to generate virtual samples, the diversity of the training data for model training is increased, thus avoiding insufficient learning in the subsequent models. Secondly, to enhance the prediction accuracy of CatBoost, we opted to optimize it using the SSA. However, considering that the SSA may suffer from issues such as a poor quality of randomly generated initial populations and being prone to local optima, we chose to use the GA to further optimize the SSA. By incorporating the GA into the process, a hybrid optimization approach called the SSAGA was formed. The SSAGA not only enhances the global search capability and accelerates the convergence speed, but also increases the possibility of finding the global optimal solution.

4.2. Data Cleaning

Two key steps for data cleaning are employed in this study, aimed at enhancing the completeness and accuracy of the data, thereby ensuring high reliability in subsequent analyses.

(1) Missing value interpolation

We utilize the KNNImputer algorithm, which has demonstrated strong performance in multiple studies, to estimate the values of missing data. The core of this algorithm is to extract the k-closest samples in the dataset, and then use the distribution of these samples to fill in the missing data values. If the missing values are discrete, the plurality of the k-nearest neighbors will be voted to fill them; otherwise, the average of the nearest neighbors will be used to fill them.

(2) Outlier removal

The EllipticEnvelope algorithm is utilized to identify outliers. This algorithm assumes that the normal sample data conform to a multivariate Gaussian distribution, while the abnormal sample data do not follow this distribution. Its objective is to find the smallest ellipse that can cover the majority of the samples and consider the points outside of the ellipse as outliers.

4.3. Data Augmentation Based on the SALGAN

GANs are capable of capturing and learning the complex distribution characteristics of data, including various implicit correlations and dependencies, which aids in generating highly realistic data instances. However, achieving a balance in the learning process between the generator and discriminator can be challenging, leading to model instability and convergence difficulties.

Self-attention mechanisms enhance GANs by focusing on different segments of the data, aiding GANs to better grasp global structures. This results in capturing finer details and patterns in data generation, reducing model collapse issues. The model adapts autonomously based on varying features and relationships within the data.

Considering the distinct characteristics of differently labeled samples, modeling the data distribution for various labels allows the generator to better capture and reflect each label's unique features. This label-aware synthetic data generation approach facilitates the generator's more effective learning and mimicking of data distributions under different labels, yielding more realistic and contextually accurate synthetic samples.

We introduce the SALGAN for data augmentation. Compared to a traditional GAN, the SALGAN not only adapts autonomously to diverse data features and relationships but also enables the generator to more effectively learn and mimic data distributions across different labels, creating more realistic and context-relevant synthetic data samples. The experimental process of the SALGAN is illustrated in Figure 2, wherein the input is an N × M matrix, and the output is an (N + T) × M matrix; N is the number of original data items; M is the number of data items; and T is the number of generated virtual sample data items.

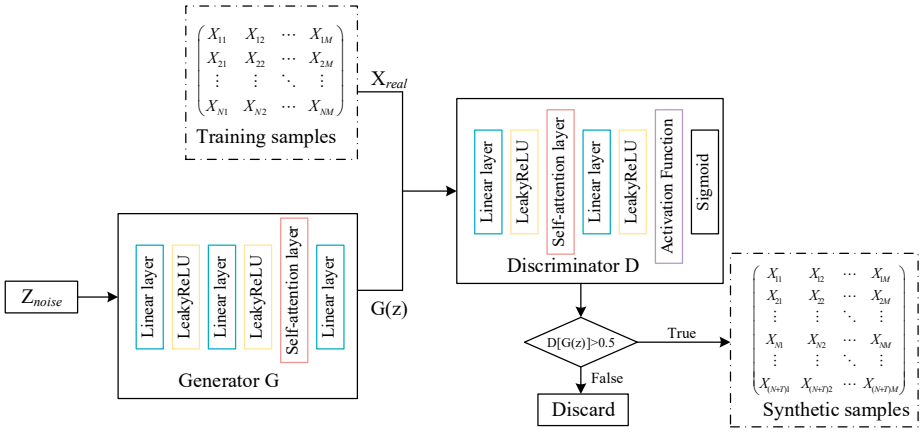

Figure 2. Flowchart of the SALGAN's process.

Data are classified by label types and generated in batches. Random noise Z_{noise}, which is a vector randomly generated from a normal distribution, serves as the input for the generator network. The generator uses Z_{noise} to generate a set of synthetic data. The discriminator receives the real data and fake data generated by the generator, and its task is to distinguish the two sets of data and output a probability value indicating the possibility that the data are real. The results of the loss function calculation are used to correct the back-propagation error and refine the parameters of the two networks. This iterative process continues until the generator and the discriminator reach a balanced state. The generator can create enough convincing data to copy the discriminator, and the discriminator is good at accurately distinguishing real data from false data.

1. Generator G

(1) Fully Connected Layer: We receive the input data and map them to a higher-dimensional hidden layer space, providing the basis for subsequent processing. The first linear layer maps the input size to the hidden layer size, and the second linear layer maps the hidden layer size back onto itself. Each linear layer is followed by the LeakyReLU Activation Function. This configuration introduces nonlinearity, enabling the model to capture more complex data patterns.

(2) Self-Attention Layer: The self-attention layer captures relationships among the input data, as illustrated in Figure 3.

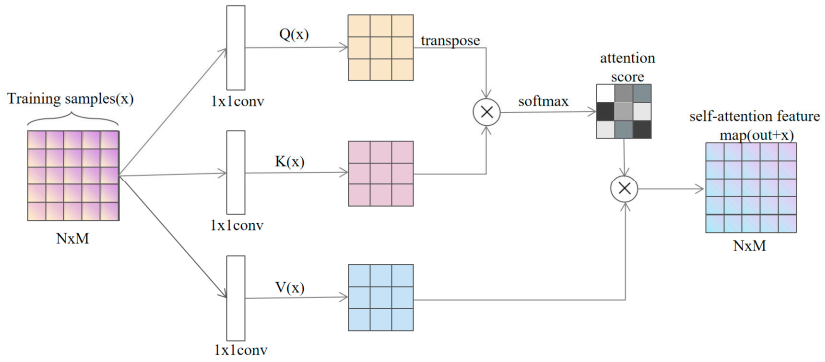

Figure 3. Flowchart of the self-attention layer's process.

First, for the input feature matrix $x \in R^{N*M}$, we perform linear transformation to generate query (Q), key (K), and value (V) matrices:

$$\begin{aligned} Q &= XW^Q \\ K &= XW^K \\ V &= XW^V \end{aligned} \tag{6}$$

where $W^Q \in R^{M*d_k}$, $W^K \in R^{M*d_k}$, and $W^V \in R^{M*d_k}$ are the learnable weight matrices, and d_k and d_v represent the dimensions of the keys and values, respectively.

Second, attention scores are calculated, as shown in Equation (7), by performing matrix multiplication between the transpose of $Q(x)$ and $K(x)$ to calculate attention scores.

$$\text{Attention Score} = QK^T \tag{7}$$

Third, weighted value vectors are calculated. Using the softmax function, the attention scores are normalized, as shown in Equation (8), and then the softmaxed attention scores are multiplied by $V(x)$ according to Equation (9) to obtain the weighted values, which represent the comprehensive information of all input features weighted by their corresponding attention scores.

$$\text{Softmax Score} = \text{softmax}\left(\frac{\text{Attention Score}}{\sqrt{d_k}}\right) \tag{8}$$

$$\text{Weighted Value} = \text{Softmax Score} \times V \tag{9}$$

(3) Output Layer: We map the high-dimensional representation of the hidden layer to the output layer. This consists of a linear layer and the Tanh Activation Function for generating the final generated data.

The generated data samples are compared with real data samples and adjust their own weights according to the output of the discriminator. During training, the constant confrontation between the generator and the discriminator enables the generator to produce progressively more realistic and higher-quality data samples.

2. Discriminator D

The discriminator analyzes the input data and effectively distinguishes between the real data X_{real} and generated data G(Z). The output of the discriminator is used for self-adjustment and is fed back to the generator to guide improvements in the generation process. Binary cross-entropy loss (BCELoss) is used as the main loss function of the discriminator, as shown in Equation (10). This loss function makes the discriminator judgement more accurate and generates data closer to the actual data by minimizing the binary cross-entropy loss.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \bullet \log(p(y_i)) + (1 - y_i) \bullet \log(1 - p(y_i)) \tag{10}$$

where y represents a binary label, either 0 or 1, and $p(y)$ denotes the probability of the output belonging to a given label. N signifies the number of groups of objects for which the model makes predictions.

4.4. Risk-Warning Model Based on CatBoost-SSAGA

CatBoost is an algorithm that supports various categorical variables, effectively reduces the prediction bias in GBDT, thus reducing the risk of overfitting, and has a high accuracy. However, the processing of categorical features takes a long time and CatBoost has many hyperparameters that need to be tuned. Therefore, this study combines the SSA and GA to construct a hybrid algorithm, named the SSAGA, to optimize the parameters of CatBoost.

4.4.1. SSAGA

The initial quality of the population randomly generated by the SSA is poor. It is easy for the SSA to fall into local optima in large or complex optimization spaces, sometimes even jumping out of the optimization space.

The GA introduces new solutions through crossover and mutation operations, which is conducive to increasing the diversity of the population. It aids in carrying out local searches and evolutionary optimization, which can further refine the solution in the found good region. In addition, the GA is more stable and able to optimize in a complex optimization space.

By combining the SSA and GA, this paper proposes the hybrid SSAGA, which could enhance global search capabilities, accelerate convergence speed, and increase the likelihood of finding global optima. This methodology aims to effectively obtain the optimal hyperparameters for the CatBoost model. The specific workflow of the model is depicted in Figure 4.

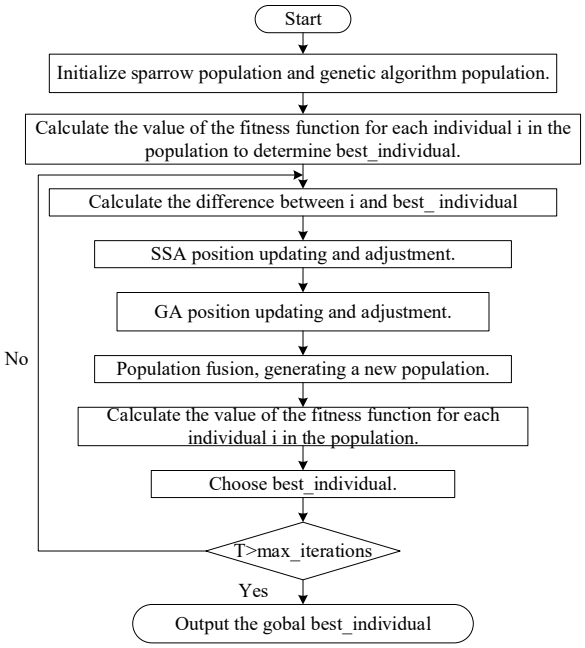

Figure 4. Flowchart of the SSAGA.

The optimization process of the SSAGA is as follows:

Step 1 Initialization: The population size (pop_size), maximum iteration number (max_iterations), and parameter dimension (dimension) are set. The Sparrow Search Algorithm population and Genetic Algorithm population are initialized, and random parameter vectors are generated.

Step 2 Iterative optimization: Within the specified maximum iteration number, the algorithm alternately executes the following steps:

a. Sparrow Search Algorithm Phase: Initially, each sparrow's performance in the population is evaluated using a fitness function, identifying the current optimal- and least-fit individuals. Subsequently, the location of the sparrow population is updated. In each iteration, sparrows adjust their positions based on the current best (best_sparrow) and worst (worst_sparrow) locations. This updating mechanism mimics sparrows' foraging behavior, where some sparrows converge towards the best solution (leader sparrows and followers), while others explore in the opposite direction of the worst solution (scouts). The updated parameter values are constrained within their valid range.
b. Genetic Algorithm Phase: Parental selection is conducted using the select_parents() function, employing a roulette-wheel selection method based on the fitness function, with the selection probability being proportional to the expected fitness. A crossover operation on the selected parental individuals is performed using the crossover() function, where a crossover point is randomly chosen to mix the genes of two parental individuals in a certain proportion. Mutation operation on post-crossover individuals is executed using the mutate() function, introducing random perturbations to certain genes of the individuals. The new individuals obtained from the crossover and mutation are merged with the original population to form a new Genetic Algorithm population.
c. Optimal Individual Update: The Sparrow Search Algorithm and Genetic Algorithm populations are merged, and the optimal individual is selected based on the fitness function, specifically the individual with the lowest fitness function value.
d. Termination Condition Assessment: The iteration process concludes when either the maximum number of iterations is reached or specific stopping criteria are satisfied (e.g., the fitness function value falls below a certain threshold).

Step 3 Output Results: The optimal parameter combination and the fitness function value are returned as the optimization outcomes.

By integrating the two algorithms, the SSAGA continually updates its position and individual evolution during the hybrid process to effectively enhance the optimization, leading to superior hyperparameter configurations.

4.4.2. CatBoost-SSAGA

According to the proposed optimization strategies, the mechanism of CatBoost-SSAGA is shown in Algorithm 1. First, we initialize the parameters of the SSAGA. Secondly, we generate populations for both the SSA and GA, separately, then merge them into a new population. Third, we update the position of the sparrows and the global fitness. Finally, Algorithm 1 returns the best position and its corresponding fitness value, which represent the optimal hyperparameters for CatBoost.

CatBoost-SSAGA involves multiple steps such as parameter optimization, model training, and validation, each of which affects the overall complexity. The following presents a complexity analysis of these steps:

(1) Data Preparation and Preprocessing: The complexity of the data loading and preprocessing is typically O(n $\times$ m), where n is the number of samples and m is the number of features.
(2) Parameter Optimization and Model Training

Population Initialization: The complexity of the SSA and GA population initialization is O(pop_size $\times$ dimension), where pop_size is the population size, and dimension is the parameter dimension.

Iterative Optimization:

In each iteration, the complexity of updating the sparrow positions and performing the GA operations is also O(pop_size $\times$ dimension).

The training complexity of the CatBoost model depends on n_estimators (the number of trees), max_depth (the tree depth), the number of samples, and the number of features. The training complexity of CatBoost can be roughly represented as O(n_estimators $\times$ n $\times$ m).

Due to the use of ten-fold cross-validation, the computational complexity is further increased as the model needs to be trained and evaluated on each fold, making the complexity approximately O(10 × n_estimators × n × m).

(3) Optimal Solution Selection: The complexity of selecting the optimal solution from the merged population is O(pop_size), because it needs to iterate in the population to find the individual with the highest fitness.

Algorithm 1. CatBoost-SSAGA

Input: Population size P, Dimension D, Upper bound ub, Lower bound lb, Maximum iterations ε, Strategy parameter S.

Output: Best fitness value f_{Gb} and Best position X_{Gb}.

1. Initialize empty lists: X = [], F = []
2. Generate initial population for SSA and GA:
   a. For SSA (Sparrow Search Algorithm):
      - Use initialize_sparrows function with inputs pop_size = P, dimension = D, lb, and ub to create sparrows population
   b. For GA (Genetic Algorithm):
      - Use initialize_ga_population function with the same inputs to create GA population
   c. Combine both populations: X = SSA population + GA population
3. For each iteration t from 1 to ε:
   a. Calculate decay rate ϵ = 1–(t/ε)
   b. For each individual I in X:
      - Evaluate fitness using evaluate_fitness function
      - F = CatBoost.fit(X)
   c. Get X_b, f_b, X_w, f_w
   d. Update positions of first pdNum individuals in X using SSA strategy:
      - Apply update_sparrow_positions_enhanced influenced by X_b and X_w
   e. Update positions of remaining individuals in X using GA strategy:
      - Select parents from X
      - Perform crossover and mutation to generate new offspring
      - Replace corresponding individuals in X with new offspring
   f. Re-evaluate fitness of entire population X
   g. If a better fitness is found (indicating higher accuracy from CatBoost), update f_{Gb} and X_{Gb}
   h. End
4. Return f_{Gb} and X_{Gb}

4.5. Indicators of Model Evaluation

The proposed risk-warning model was mainly applied to a binary classification problem, wherein the samples were divided into positive and negative classes, and the prediction results were evaluated using a confusion matrix, i.e., Table 1. Accuracy, precision, recall, F1-scores, and the AUC were used as evaluation metrics for the proposed model. Accuracy refers to the percentage of samples correctly predicted by the classifier in the total samples. Recall represents the proportion of correctly predicted positive samples. The F1-score is the harmonic average of precision and recall, which considers both evaluation indicators and reflects the model's robustness. The AUC (area under the curve) measures the area under the ROC (Receiver Operating Characteristic) curve, where a higher AUC indicates a better classification effect of the classifier.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

Table 1. Confusion matrix.

	Positive	**Negative**
True	True Positive (*TP*)	True Negative (*TN*)
False	False Positive (*FP*)	False Negative (*FN*)

5. Experiments

5.1. Datasets and Preprocessing

1. Datasets

For this study, five independent and usable datasets of heart disease are selected, namely cleveland.data, hungarian.data, switzerland.data, VA-long-beach.data, and basel.data, which were obtained from the UCI Machine Learning Repository [29]. After merging, the dataset consists of 1190 data instances and contains three attributes with significant missing values, hence it is classified as a small and incomplete dataset. The merged dataset consists of a complete set of 14 attributes, of which 13 are used for predicting heart disease as feature attributes, and the remaining 1 is used as a labeled sample. The specific attribute descriptions are shown in Table 2.

2. Preprocessing

After the cleaning process, the experimental dataset is left with 981 samples, each of which contains 14 common attributes, of which 13 are used to predict the characteristic attributes of heart disease and the remaining one is used as a labeled sample. For data augmentation, virtual data are generated using the SALGAN, in accordance with the types of labels. As a result, the final dataset comprises 1981 samples.

3. Model training

Cross-validation effectively reflects the robustness of the model. In the experiments, all of the evaluation metrics in the experiments are based on 10-fold cross-validation and calculated as averages and standard deviations.

CatBoost incorporates numerous hyperparameters, including iterations, learning_rate, depth, reg_lambda, subsample, border_count, and so on. This study focuses on fine-tuning the hyperparameters that significantly impact model performance. The primary parameters we optimized are the learning rate, tree depth, maximum number of trees, and regularization coefficient, among five other parameters. Table 3 lists the default values of these parameters as well as the ranges we used for optimization.

Table 2. Dataset attribute descriptions.

Name	**Description**
Age	Continuously variable values
Sex	0 = Female
	1 = Male
Cp	0 = Classic angina pectoris
	1 = Atypical angina pectoris
	2 = Non-angina pectoris
	3 = Asymptomatic
Trestbps	Continuously variable values
Chol	Continuously variable values
Fbs	0 = <120mg/d
	1 = >120mg/d

Table 2. *Cont.*

Name	Description
Restecg	0 = Normal
	1 = Existence of a segment exception
	2 = Possible or definite left ventricular hypertrophy
Thalach	Continuously variable values
Exang	0 = No
	1 = Yes
Oldpeak	Continuously variable values
Slope	0 = Up
	1 = Float
	2 = Down
Ca	0
	1
	2
	3
Thal	1 = Normal
	2 = Irreparable
	3 = Reparable
Target	0 = No
	1 = Yes

Table 3. Optimal parameters of CatBoost-SSAGA.

Name	Optimization Scope	SSAGA-CatBoost
learning_rate	[0.001, 0.2]	0.00298
max_depth	[4, 10]	9
n_estimators	[1100, 1500]	1153
reg_lambda	[0.01, 10]	2.62938
subsample	[0.9, 1]	0.98761

5.2. Comparative Experiments

For this section, we conducted three sets of comparative experiments and ablation studies on the UCI heart disease dataset to validate the performance of the proposed SALGAN-CatBoost-SSAGA model.

1. Performance comparison between different algorithms

To verify the effectiveness of the proposed framework, we conducted comparative experiments between five commonly used tree models and SALGAN-CatBoost-SSAGA, including XGBoost version 2.0.3, LightGBM version 3.3.5, and Scikit-learn version 1.0.2. The experimental results in terms of the accuracy, precision, recall, F1-scores, and AUC are shown in Table 4 and Figure 5. The results demonstrate the superior performance of SALGAN-CatBoost-SSAGA over the individual machine learning models. Table 4 indicates that SALGAN-CatBoost-SSAGA performs the best in all metrics, with an accuracy of 90.56%, a precision of 87.79%, a recall of 87.45%, and an F1-score of 87.54%, which are much higher than those of the other models, indicating that the proposed framework performs well. It also shows smaller standard deviations, indicating greater stability. As shown in Figure 5, the area under the curve for SALGAN-CatBoost-SSAGA is the largest, reaching 0.96, which illustrates that it has high accuracy and the best overall classification performance.

Table 4. The algorithms' performances.

Model	Accuracy	Precision	Recall	F1-Score
RF	79.71 ± 0.03	81.29 ± 0.03	79.14 ± 0.05	80.10 ± 0.03
lightGBM	77.78 ± 0.02	79.52 ± 0.03	77.18 ± 0.06	78.16 ± 0.03
xgBoost	77.37 ± 0.02	78.85 ± 0.03	77.17 ± 0.04	77.90 ± 0.03
AdaBoost	71.66 ± 0.03	72.66 ± 0.02	72.65 ± 0.06	72.55 ± 0.03
Decision Tree	71.15 ± 0.03	72.76 ± 0.03	70.87 ± 0.04	71.76 ± 0.03
SALGAN-CatBoost-SSAGA	90.56 ± 0.01	87.79 ± 0.02	87.45 ± 0.03	87.54 ± 0.02

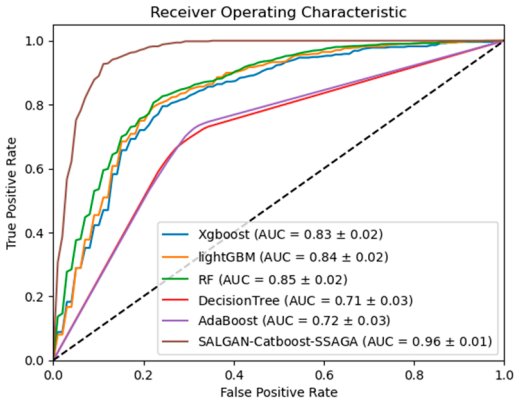

Figure 5. ROC for SALGAN-CatBoost-SSAGA and other models.

2. Impact of data augmentation on prediction results

In order to verify the effectiveness of the data enhancement method, we used the standard CatBoost model to compare the risk predictions of the data before and after data enhancement. The experimental results are shown in Table 5. These experimental results indicate that augmenting the dataset through data enhancement techniques led to a notable improvement in the performance of the CatBoost model across all evaluation metrics. Compared to the performance prior to data augmentation, there was an increase of over 10% in all metrics.

Table 5. Comparison of prediction results based on data augmentation.

Datasets	Accuracy	Precision	Recall	F1-Score
Before	79.57 ± 0.04	79.94 ± 0.05	79.58 ± 0.07	79.50 ± 0.04
After	87.83 ± 0.01	87.87 ± 0.02	87.81 ± 0.02	87.81 ± 0.01

3. Impact of parameter optimization on prediction results

To evaluate the effectiveness of the SSAGA hybrid algorithm, we utilized a dataset augmented through SALGAN data enhancement to compare its performance with that achieved by optimizing the CatBoost model parameters using solely the SSA or GA. The experimental results are presented in Table 6. These results demonstrate that, compared to the individual use of the SSA or GA, the SSAGA showed the best performance across all evaluation metrics, particularly in terms of accuracy.

Table 6. Comparison of prediction results based on the SSAGA.

Algorithm	Accuracy	Precision	Recall	F1-Score
SSA	89.45 ± 0.02	86.57 ± 0.07	85.63 ± 0.09	86.63 ± 0.07
GA	87.58 ± 0.02	84.03 ± 0.08	83.27 ± 0.1	83.87 ± 0.09
SSAGA	90.56 ± 0.01	87.79 ± 0.02	87.45 ± 0.03	87.54 ± 0.02

4. Ablation experiments

In order to evaluate the overall performance of SALGAN-CatBoost-SSAGA and verify the necessity of each module, we conducted ablation experiments on the SALGAN and SSAGA. As shown in Figure 6, SALGAN-CatBoost-SSAGA outperformed the other two models in key performance metrics such as accuracy, precision, recall, and F1-scores. This finding not only highlights the excellent predictive ability of SALGAN-CatBoost-SSAGA but also clearly proves the rationality of our choice of this combined model. It shows the obvious advantages of the combination model in improving the prediction performance compared with a single model, thus verifying the effectiveness of our combination strategy.

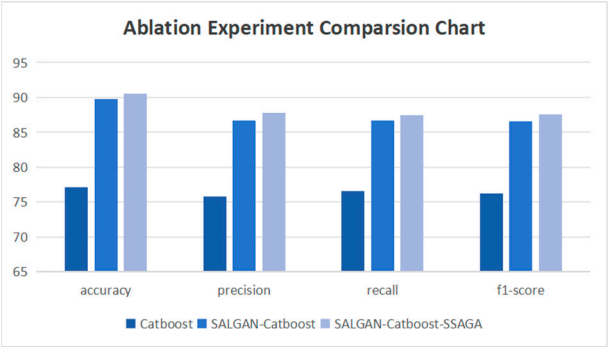

Figure 6. Ablation experiment comparison chart.

6. Conclusions

This paper introduces the SALGAN-CatBoost-SSAGA predictive model, which is designed for small-sample and incomplete datasets. The SALGAN is capable of learning various hidden correlations and dependencies within different types of labeled data, thereby facilitating the generation of highly realistic data instances. This enhancement aids in the model's learning process with sample data. Additionally, in order to find the global optimal parameters of the CatBoost algorithm, we propose the SSAGA, an algorithm that combines the SSA and GA, which helps CatBoost find global optimal parameters more effectively, avoiding local optima and improving the accuracy and stability of the prediction model. The experimental results show that the framework achieves the expected effect in data augmentation and improving prediction accuracy. The performance of the framework is better than other comparison methods in all selected indicators, which proves its feasibility and effectiveness. Therefore, it is very suitable for prediction tasks with small-sample datasets. Future works will focus more on directly incorporating small-sample incomplete datasets from different fields to validate further the generalizability of the model proposed in this study.

Author Contributions: Conceptualization, Y.P. and Y.L.; methodology, Y.P.; software, Y.L.; validation, Y.P., Y.L. and X.L.; resources, X.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.P., J.W. and X.L.; visualization, Y.L.; supervision, J.W.; project administration, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, Grants number 62172287 and number 62102273.

Data Availability Statement: The original data presented in the study are openly available at http://archive.ics.uci.edu/dataset/45/heart+disease (accessed on 9 April 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, K.; Jin, X.; Wang, Y. Survey on few-shot learning. *J. Softw.* **2021**, *32*, 349–369.
2. Ansarullah, S.I.; Kumar, P. A systematic literature review on cardiovascular disorder identification using knowledge mining and machine learning method. *Int. J. Recent Technol. Eng.* **2019**, *7*, 1009–1015.

3. Yekkala, I.; Dixit, S.; Jabbar, M.A. Prediction of heart disease using ensemble learning and Particle Swarm Optimization. In Proceedings of the 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon), Bengaluru, India, 17–19 August 2017; pp. 691–698.
4. Li, F.F.; Fergus, R.; Perona, P. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 594–611.
5. Liu, Y.; Long, M.; Cao, Z.; Wang, J. Few-Shot Object Recognition from Machine-Labeled Web Images. *IEEE Trans. Image Process.* **2020**, *29*, 594–611.
6. Wang, X.; Huang, T.E.; Darrell, T.; Gonzalez, J.E.; Yu, F. Frustratingly Simple Few-Shot Object Detection. *arXiv* **2020**. [CrossRef]
7. Wang, J.; Chen, Y. *Introduction to Transfer Learning*; Electronic Industry Press: Beijing, China, 2021.
8. Hu, X.; Chen, S. A survey of few-shot learning based on machine learning. *Intell. Comput. Appl.* **2021**, *11*, 191–195+201.
9. Zhu, X.; Ghahramani, Z.; Lafferty, J.D. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003), Washington, DC, USA, 21–24 August 2003; pp. 912–919.
10. Gliozzo, J.; Mesiti, M.; Notaro, M.; Petrini, A.; Patak, A.; Puertas-Gallardo, A.; Paccanaro, A.; Valentini, G.; Casiraghi, E. Heterogeneous data integration methods for patient similarity networks. *Brief. Bioinform.* **2022**, *23*, bbac207. [CrossRef]
11. Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Kumar, A.; Feris, R.; Giryes, R.; Bronstein, A. Delta-encoder: An effective sample synthesis method for few-shot object recognition. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 2850–2860.
12. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
13. Kataoka, Y.; Matsubara, T.; Uehara, K. Image generation using generative adversarial networks and attention mechanism. In Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, 26–29 June 2016; pp. 1–6.
14. Park, N.; Mohammadi, M.; Gorde, K.; Jajodia, S.; Park, H. Data Synthesis based on Generative Adversarial Networks. *Proc. VLDB Endow.* **2018**, *11*, 1071–1083. [CrossRef]
15. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: Gradient boosting with categorical features support. *arXiv* **2018**, arXiv:1810.11363.
16. Diao, L.; Niu, D.; Zang, Z.; Chen, C. Short-term weather forecast based on wavelet denoising and catboost. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 3760–3764.
17. Kumar, P.S.; Kumari, A.; Mohapatra, S.; Naik, B.; Nayak, J.; Mishra, M. CatBoost ensemble approach for diabetes risk prediction at early stages. In Proceedings of the 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON), Bhubaneswar, India, 8–9 January 2021; pp. 1–6.
18. Wang, B.; Feng, H.; Wang, F.; Qin, X.; Huang, P.; Dang, D.; Zhao, J.; Yi, J. Application of CatBoost model based on machine learning in predicting severe hand-foot-mouth disease. *Chin. J. Infect. Control* **2019**, *18*, 12–16.
19. Chen, D.; Chen, Y.; Feng, X.; Wu, S. Retrieving suspended matter concentration in rivers based on hyperparameter optimized CatBoost algorithm. *J. Geo-Inf. Sci.* **2022**, *24*, 780–791.
20. Jin, C.; Yu, J.; Wang, Q.; Chen, L.J. Prediction of blasting Fragment large block percentage ratio based on ensemble learning CatBoost model. *J. Northeast. Univ. (Nat. Sci.)* **2023**, *44*, 1743–1750.
21. Xu, L.; Guo, C. Predicting Survival rates for gastric cancer based on ensemble learning. *Data Anal. Knowl. Discov.* **2021**, *5*, 86–99.
22. Yang, C.; Liu, L.; Zhang, Y.; Zhu, W.; Zhang, S. Machine learning based on landslide susceptibility assessment with Bayesian optimized the hyper parameters. *Bull. Geol. Sci. Technol.* **2022**, *41*, 228–238.
23. Tikhamarine, Y.; Souag-Gamane, D.; Kisi, O. A new intelligent method for monthly streamflow prediction: Hybrid wavelet support vector regression based on grey wolf optimizer (WSVR-GWO). *Arab. J. Geosci.* **2019**, *12*, 540. [CrossRef]
24. Feng, T.; Peng, Y.; Wang, J. ISGS: A combinatorial model for hysteresis effects. *Acta Electron. Sin.* **2023**, *51*, 2504–2509.
25. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
26. Meng, K.; Chen, C.; Xin, B. MSSSA: A multi-strategy enhanced sparrow search algorithm for global optimization. *Front. Inf. Technol. Electron. Eng.* **2022**, *23*, 1828–1847. [CrossRef]
27. Ou, Y.; Yu, L.; Yan, A. An Improved Sparrow Search Algorithm for Location Optimization of Logistics Distribution Centers. *J. Circuits Syst. Comput.* **2023**, *32*, 2350150. [CrossRef]
28. Wang, J.; Wang, Z.; Li, J.; Peng, Y. An Interpretable Depression Prediction Model for the Elderly Based on ISSA Optimized LightGBM. *J. Beijing Inst. Technol.* **2023**, *32*, 168–180.
29. Janosi, A.; Steinbrunn, W.; Pfisterer, M.; Detrano, R. Heart Disease. UCI Machine Learning Repository. 1988. Available online: https://archive.ics.uci.edu/dataset/45/heart+disease (accessed on 9 April 2024).