IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 March 2023, accepted 20 March 2023, date of publication 27 March 2023, date of current version 4 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3262460

RESEARCH ARTICLE

Imbalanced Data Classification Method Based on LSSASMOTE

ZHI WANG AND QICHENG LIU
School of Computer and Control Engineering, Yantai University, Yantai, Shandong 264000, China
Corresponding author: Qicheng Liu (ytliuqc@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 62272405.

ABSTRACT Imbalanced data exist extensively in the real world, and the classification of imbalanced data is a hot topic in machine learning. In order to classify imbalanced data more effectively, an oversampling method named LSSASMOTE is proposed in this paper. First, the kernel function parameters and penalty parameters of the support vector machine (SVM) were optimized using levy sparrow search algorithm (LSSA), and a fitness function was correspondingly designed. Then, during the optimization process, SMOTE sampling rate was combined, and LSSA iteration was used to select the best combination of SVM parameters and SMOTE sampling rate. In addition, the oversampled samples were noise processed by Tomek Link. In this case, the LSSASMOTE+SVM classification model was constructed to classify the imbalanced data. Eight of the datasets used in the experiments were obtained on UCI and KEEL, and the other three datasets were created manually. The experimental results confirm that the model can effectively improve the classification accuracy of imbalanced data and can be used as a new imbalanced data classification method.

INDEX TERMS Imbalanced data, machine learning, sparrow search algorithm, support vector machine, oversampling.

I. INTRODUCTION

Imbalanced datasets refer to two different sets of instances with significant imbalances and asymmetries. The class with a larger amount of data in the dataset is called the majority class, while the one with a smaller amount of data is called the minority class [1]. Imbalanced data exist in various applications, such as medical diagnosis, garbage detection, credit risk identification, etc. [2]. When classifying data, most classification algorithms learn models by minimizing the overall misclassification rate without considering the differences in sample sizes between categories. In imbalanced data, where a few categories have a small number of samples, the classifier may prefer to classify the samples as majority classes, leading to biased decision boundaries towards the majority classes. This bias can have negative consequences in practical applications. For example, in identifying bank credit risk, the number of customers with bad credit is much lower than that of customers with good credit. In addition, if wrongly classified as good credit, they can cause financial losses to the bank loan business [3]. Therefore, in some practical situations, it is very important to classify minority classes accurately.

There are various approaches to address the imbalanced data problem, such as resampling, algorithm tuning, integrated learning, and deep learning. The difference between these methods lies in their technical means and implementation, and some new techniques have further innovated and improved in these aspects. For instance, the SMOTE algorithm balances data by synthesizing samples of minority classes, the Focal Loss method [4] improves classification accuracy by reducing the weight of easily classified samples, and the EasyEnsemble algorithm [5] splits data into multiple subsets and trains a classifier on each subset to enhance overall classification performance.

Moreover, solving class imbalance often gives rise to the class overlap problem [6], where the boundaries between different classes become blurred, making it challenging to differentiate them. The methods to address this problem include feature selection, feature extraction, integrated learning, and anomaly detection. These methods offer significant help in the application of imbalanced data and in the classification

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano.

process. Addressing class imbalance and class overlap in a rational manner can be beneficial for the classification process and the application of imbalanced data.

II. RELATED WORK

Existing imbalanced data classification solutions are broadly classified into deep and shallow models. Among them, the deep model is a neural network-based model that automatically learns feature representations suitable for imbalanced data and can take into account the weight differences between different categories during the learning process. The main focus of this paper is to investigate the use of shallow models for handling imbalanced data, which typically refer to traditional machine learning models that rely on algorithm tuning and data preprocessing to handle imbalanced data [7].

From the data level, the data set is mainly processed using the oversampling methods and the undersampling methods, among which the synthetic minority oversampling technique [8] (SMOTE) is a classic oversampling method. It randomly replicates minority samples based on the principle of linear interpolation, increasing thus the number of minority samples and improving the classification accuracy for the minority class. However, since the SMOTE technique fails to consider the distribution of adjacent samples and is blind in the selection of sampling magnification, it is exposed to the risk of introducing noise instances and overfitting problems. Various attempts have been made to improve the SMOTE algorithm based on these two problems. For example, Meng and Li [9] proposed a method of combining the center offset factor and SMOTE, which first removes noise using Tomek Link technology, then calculates the center offset factor to select the sparsely distributed minority class samples, and combines these samples with SMOTE to generate better minority classes, thus improving classification performance on imbalanced datasets. Krawczyk et al. proposed an undersampling method using support vector machine optimization to improve the computing time and classification accuracy [10]. Huo et al. [11] proposed the GASMOTE oversampling method, which selects the SMOTE sampling rate using the genetic algorithm for oversampling, providing the sampling rate with a certain flexibility. However, the classification accuracy of this method still needs to be improved.

From the algorithm level, the representative algorithms are cost-sensitive random forest, support vector machine, etc. [12]. Support vector machine (SVM) [13], proposed by Vapnik, is characterized by a solid theoretical foundation, simple implementation, generalization, and excellent classification performance. However, when SVM deals with unbalanced classification problems, the classification accuracy is not optimistic because of misclassification and inseparability. The reason for the misclassification is that the sample distribution of different classes is imbalanced and that the minority class is much smaller than the majority class, which makes the classification hyperplane tilt toward the majority class. In addition, the value of the related parameters in the support vector machine is also significant in the classification process [14]. For example, the penalty parameter C will take a higher value for samples of minority class in the imbalanced data classification process, which may deviate from the probability distribution of the initial data. Therefore, different solutions have been proposed to improve the classification performance of SVM. Luo and Wang [15] proposed the FTL-SMOTE algorithm and introduced the hybrid kernel function to improving SVM, which effectively improves the classification effect of the imbalanced data. Ma and Zhu [16] proposed the IGWOSMOTE algorithm, which combines the gray wolf algorithm with the SMOTE method to improve the blindness of the SMOTE sampling rate, and improves SVM through the gray wolf algorithm, which significantly improves the overall classification accuracy. However, the gray wolf algorithm can easily get stuck in a local optimum, and thus affects the oversampling result. The SMOTE and SVM algorithms have been improved in the above studies, but still, need to be optimized in terms of algorithm parameter selection and classification accuracy.

TABLE 1. Number of iterations of the optimization algorithm.

Data sets	GA	ACO	DE	SSA
10-Dim Sphere	230	280	260	**190**
10-Dim Rastrigin	2986	3125	3452	**2430**
10-Dim Ackley	500	400	600	**300**
20-Dim Ackley	5350	5975	5547	**4755**
20-Dim Rastrigin	3700	3500	4700	**2900**
A 256x256 RGB image	420	480	460	**300**

In this paper, the problems encountered in the imbalanced data classification process are combined, and the swarm intelligence optimization algorithm [17] is used as an inspiration for research. Therefore, an oversampling method based on the levy sparrow search algorithm (LSSA) [18] is proposed. The sparrow search algorithm is a depth-first search-based algorithm that performs better than genetic algorithms (GA) [19], ant colony algorithms (ACO) [20], and differential evolution algorithms (DE) [21] compared to function optimization problems, as shown in Table (1). And LSSA is a sparrow search algorithm based on Lévy flight, which has better search efficiency and results through techniques such as random wandering and pruning.

Based on the excellent performance of LSSA, firstly, this paper introduces LSSA to optimize the kernel function and penalty parameters of SVM, and designs an adaptation function. Second, the SMOTE sampling rate was involved in the optimization process, and the LSSA was used to select the best combination of SVM parameters and the SMOTE sampling rate iteratively. Finally, Tomek Link [22] is used to denoise the oversampled samples and solve the class overlap problem to build the LSSASMOTE+SVM classification model.

III. RELATED THEORIES

A. TOMEK LINK

Tomek Link is a method used to solve the class overlap problem by effectively identifying and removing noisy points

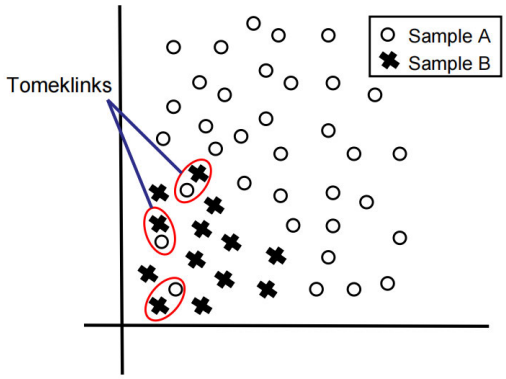

FIGURE 1. Before deleting Tomek links.

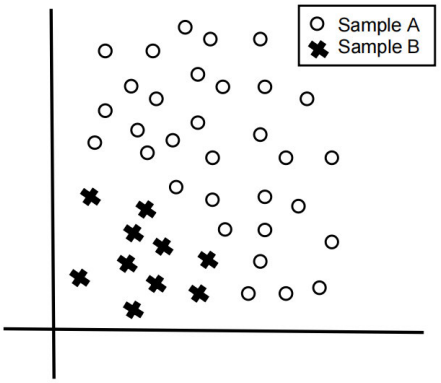

FIGURE 2. After deleting Tomek links.

between adjacent classes, thus improving the performance of the classifier. The core idea of the Tomek Link algorithm is to find those sample points between adjacent classes that are close to each other and labeled with the same Tomek Link, i.e., noisy points that need to be removed.

In Figure (1), samples A and B are samples of different categories. The nearest neighbor of A is B, while the nearest neighbor of B is A, and then A and B are Tomek links. The entire Tomek link is deleted. As shown in Figure (2), the boundaries between the samples become more apparent, the noise samples are removed, the classification difficulty is reduced, and the classification accuracy is improved.

B. SUPPORT VECTOR MACHINES

The Support Vector Machine (SVM), a model for small and medium-sized data samples, maps the feature vectors of the samples to some points in the space and uses SVM to draw a line to distinguish the two types of points. These two points are then used to divide the plane, as shown in (1).

$$\begin{cases} min\dfrac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\varepsilon_i \\ s.t.y_i(\omega^T x_i + b) \geq 1 - \varepsilon_i \end{cases} \quad i = 1, 2, \cdots, m \tag{1}$$

In formula (1), ω represents the normal vector of the hyperplane; x_i represents the training sample; y_i represents the sample category; b represents the threshold of sample training; C represents the penalty parameter; ε_i represents the relaxation variable.

For nonlinear cases, the kernel function $k(x_i, x_j)$ is introduced to map the samples from the low-dimensional space to the high-dimensional space, so that the samples can be separable in the high-dimensional space [23]. As shown in (2).

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i k(x_i, x_j) + b \tag{2}$$

The radial basis kernel function (RBF) is generally adopted by the kernel function $k(x_i, x_j)$. As shown in (3).

$$k(x_i, x_j) = exp(-g\left\|x_i - x_j\right\|^2) \tag{3}$$

According to formulas (1) to (3), the penalty parameter C and kernel parameter g should be optimized by SVM. Therefore, the swarm intelligence algorithm can be used to select the optimal parameters of SVM, to improve the classification performance of SVM.

C. SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

The basic idea of the synthetic minority oversampling technique (SMOTE) is to add a new sample to the dataset from the interval of several types of samples, so that the number of positive and negative samples can be balanced [24]. The basic flow of the algorithm is as follows:

(1) Each sample is set as z and the distance S_{min} between all samples is calculated according to the Euclidean distance to obtain the k nearest neighbor.

(2) The sampling magnification is N. Some samples zn are randomly selected from the k neighbors of a small number of samples z [25].

(3) A new sample is constructed with samples zn and z, as shown in (4).

$$z_{new} = z + rand(0, 1) \cdot |z - zn| \tag{4}$$

D. SPARROW SEARCH ALGORITHM

The sparrow search algorithm (SSA) [26] is a swarm intelligence optimization algorithm that mimics sparrow foraging behavior. Each sparrow has a location attribute indicating the place where it finds food. At the same time, every sparrow can be a finder and follower, and all sparrows can detect and warn. The position of each sparrow in the d-dimensional space is $X = (x_1, x_2, \ldots, x_D)$, with the fitness value of $f_i = f(x_1, x_2, \ldots, x_D)$.

The formula for the position update of the discoverer is shown in (5).

$$x_{i,d}^{t+1} = \begin{cases} x_{i,d}^{t} exp(\dfrac{-i}{\alpha \cdot iter_{max}}), & R_2 < ST \\ x_{i,d}^{t} + Q \cdot D, & R_2 > ST \end{cases} \tag{5}$$

In formula (5), $x_{i,d}^{t+1}$ represents the position of the i individual in the t generation of the population in the d dimension;

α represents a random number in (0,1); Q represents a positively distributed random number; R_2 represents the uniform random number in (0,1); ST represents the alarm threshold and safety value, with a value range of [0.5,1.0]. $R_2 < ST$ indicates that the current environment is safe and that sparrows can find food. $R_2 > ST$ indicates that a predator is approaching, and the sparrow issues an alert. At this time, all the sparrows fly to a safe place to feed.

The primary function of the follower is to follow the discoverer. The formula for a simplified position update is shown in (6).

$$x_{i,d}^{t+1} = \begin{cases} Q \cdot exp(\frac{xw_{i,d}^t - x_{i,d}^t}{i^2}), & i > \frac{n}{2} \\ xb_{i,d}^t + \frac{1}{D}\sum_{d=1}^{D}(rand\{-1,1\} \cdot (|xb_{i,d}^t - x_{i,d}^t|)), & i \le \frac{n}{2} \end{cases} \tag{6}$$

xw represents the worst position of sparrows in the population. xb represents the best position of the sparrows in the population. $i > \frac{n}{2}$ indicates that the i follower is hungry and flies to other places for food. $i \le \frac{n}{2}$ indicates that the follower moves to the sweet spot and stays near the sweet spot, and the variance from the optimal position becomes smaller. Some followers also act as scouts to help discoverers find food. When the scouts find danger, they will immediately abandon their existing food and move to a new place. As shown in (7).

$$x_{i,d}^{t+1} = \begin{cases} x_{i,d}^t + \beta|x_{i,d}^t - xb_{i,d}^t|, & f_i \neq f_g \\ x_{i,d}^t + K\left[\frac{|x_{i,d}^t - xw_{i,d}^t|}{(f_i - f_w) + \varepsilon}\right], & f_i = f_g \end{cases} \tag{7}$$

β represents a random number in a normal distribution; K represents the range between [-1,1]; ε represents a very small and non-zero number; f_g and f_w represents the best and worst fitness values, respectively; f_i represents the individual fitness value of a sparrow; When $f_i = f_g$, they need to change their position quickly and fly to another sparrow to prevent danger.

IV. LSSASMOTE+SVM CLASSIFICATION MODEL

A. LSSASMOTE ALGORITHM

The sparrow search algorithm has the shortcomings of lack of diversity and local optimization ability in the late iteration. Ma and Zhu [27] introduced the mechanism of flight disturbance levy to enhance the optimization performance of SSA, which solves the optimization problem of high-dimensional space to some extent.

During the calculation of the search path $L(\lambda)$ of the levy flight, the calculation formula of the simulated levy flight [28] path was generally used, as shown in (8):

$$s = \frac{u}{|v|^{1/\beta}} \tag{8}$$

In formula (8), s refers to the flight path $L(\lambda)$; the value range of the parameter β is $0 < \beta < 2$, generally taking $\beta = 1.5$; the parameters u and v are typically distributed random numbers, obeying formula (8) shown in the normal distribution; the values of standard deviations σ_u and σ_v of the normal distribution corresponding to formula (9) are in line with the calculation shown in (10):

$$\begin{cases} u\ N(0, \sigma_u^2) \\ v\ N(0, \sigma_v^2) \end{cases} \tag{9}$$

$$\begin{cases} \sigma_u = \left\{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}\right\}^{1/\beta} \\ \sigma_v = 1 \end{cases} \tag{10}$$

Firstly, in the individual selection, the inertia weight factor is adopted in this paper, and the roulette method is used to select sparrow individuals for the Levy flight variation, as shown in (11):

$$f = 1 - iter/Max\_iter \tag{11}$$

In formula (11), f is the inertia weight factor, $iter \in \{1, 2, \ldots, Max\_iter\}$, and Max_iter denotes the number of iterations of sparrow search. If $rand > f$, use roulette wheel selection and take a random number $rand$ to perform levy flight mutation on the selected sparrow individuals.

Secondly, the improved sparrow search algorithm is used to select the parameters of SVM, assign different weights to different categories of samples, and reduce the dimensionality of samples. At the same time, the value of the SMOTE sampling rate is also included in the optimization process. Through the LSSA algorithm, the problem of obtaining the optimal parameters can be transformed into the problem of solving the maximum value of the function. The algorithm is defined as:

$$maximize : y = f(X), \quad X = (x_1, x_2, \ldots, x_D) \tag{12}$$

In formula (12), $f(X)$ is the fitness function, that is, the prediction accuracy of samples of the minority class, and X refers to the position of different sparrows in the D dimension of the initial population of sparrows.

The SMOTE sampling rate in this algorithm is defined as:

$$Z_{min} < round(Z_i) < Z_{max}, \quad i = 1, 2, \ldots, M \tag{13}$$

In formula (13), Z_{min} and Z_{max} are the minimum and maximum values of the sampling ratio Z_i of the minority class samples, respectively, which are determined by the number of minority class samples; Z_i takes a value within this interval, and its value is rounded off by the $round()$ function; M refers to the dimension of decision space, that is, the number of samples from the minority class.

B. DESIGN OF THE FITNESS FUNCTION

In the improved sparrow search algorithm, the individual position of a sparrow is related to its fitness value, and the complexity of its fitness function also directly affects the

efficiency of the algorithm. In this case, the split data set was predicted and the accuracy of the prediction result was used as the fitness value.

The function is constructed as shown in (14):

$$fitness = acc(validation(X)) + acc(train(X)) \tag{14}$$

Combined with formula (12), *validation* in formula (14) represents the classification label of the validation set; *train*, the classification label of the training set; X, the position of different sparrows in the D dimension of the initial population of sparrows; *acc*, the accuracy of the prediction result; and *fitness*, the fitness value, with a higher corresponding position that indicates a better individual position of the sparrow.

The pseudo-code of the fitness value solving process is shown in Algorithm (1):

Algorithm 1 Fitness Solution Algorithm

Input: population array X, validation set *validation*, train set *Train*

Output: *fitness*

Classifier ← *SVM.SVC*(C ← $X[0]$, *kernel*, *gamma* ← $X[1]$)
{The initial fitness value was calculated and the SVM classifier was trained with X }
Train ← *Classifier.Predict*(*Train*)
validation ← *Classifier.Predict*(*validation*)
{Calculate the training set and validation set prediction labels}
Fun(0) ← *acc*(*validation*) + *acc*(*Train*)
{Calculate the initial fitness value Fun (0) }
pop ← *X.shape*(0)
{Construct zero matrix}
for i ← 0 *to pop by* 1 **do**
    fitness[i] ← *Fun*($X[i, :]$)
    { Calculate the fitness value based on the pop size}
end for
fitness[i] ← *Sort*(*fitness*)
{ Sort the fitness values and select the best fitness value}
return *fitness*

C. LSSASMOTE+SVM CLASSIFICATION MODEL

Based on the characteristics of the imbalanced data, the LSSASMOTE+SVM classification model selected the best combination of parameters of the SMOTE sampling rate, the SVM penalty parameter and the kernel parameter. The influence of noise among different samples after sampling is optimized. This classification model not only obtains more ideal balanced data, but also improves the classification accuracy of imbalanced data. The steps to build the model are as follows:

Step 1: Initialize the sparrow population and set the parameters of the LSSA algorithm, which consist of population size *pop*, dimension *dim*, maximum iteration number *MaxIter*, lower boundary *lb* and upper boundary *ub*.

Step 2: Take the penalty parameter C and the kernel parameter g of the SVM as the individual position of the sparrow to learn the training set and construct the fitness function *fitness* by taking the classification accuracy as the fitness value of the individual position of the sparrow.

Step 3: Calculate and sort the fitness value using the LSSA algorithm, iterate according to the number of iterations *MaxIter*, select individuals to mutate using the roulette selection method, and choose the best combination of parameters C and g.

Step 4: Create a new sparrow population based on the number of minority classes in different datasets.

Step 5: Take the sampling rate in the SMOTE algorithm as the individual position of the sparrow, combine it with the optimized SVM algorithm, and select the best sampling rate according to the fitness function.

Step 6: Perform oversampling with the selected sampling ratio and use the Tomek link for noise processing to obtain a data set with balanced sample categories.

Step 7: Balance the data set and combine the selected parameter combinations C and g to establish the LSSASMOTE+SVM classification model.

In the LSSASMOTE+SVM classification model, the SMOTE sampling rate and the parameters of the SVM are the individual positions of the sparrows. Therefore, the optimal parameters were selected to solve the optimal individual positions of the sparrows. The pseudo-code of the solution process is shown in Algorithm (2).

V. EXPERIMENTAL

A. DATA SET PREPARATION

To verify the validity of the LSSASMOTE+SVM model, experimental analyses based on eight imbalanced datasets from UCI and KEEL are conducted in this paper, and the structural characteristics of the datasets are listed. In addition, this paper uses three manually created datasets which are generated by make_classification in sklearn. The performance of the model is further demonstrated by controlling the degree of imbalance and the proportion of noisy data. The proportion of noise in simulated data 1(Sim1) is 10%, the proportion of noise in simulated data 2(Sim2) is 20%, and the proportion of noise in simulated data 3(Sim3) is 30%. See Table (2) for an example.

To ensure the consistency of the sample imbalance rate between the validation and training sets, the data set was divided into 70% of the training set, 15% of the test set and 15% of the validation set. In order to fully validate the classification effect of the algorithm and reduce randomness, the following metric results are the average values obtained after 5 times of hierarchical cross-validation.

B. EVALUATION INDICATORS

The classification accuracy rate is usually taken as the evaluation indicator by the traditional SVM model. However, the accuracy rate is suitable for evaluating the balanced

Algorithm 2 Optimal sparrow individual position

Input: population size *pop*, dimension *dim*, maximum iteration number *MaxIter*, population array X, *fitness*
Output: Optimal individual position *GbestPosition*

$ST \leftarrow 0.6$
{Warning value}
$PD \leftarrow 0.7$
{Ratio of discoverers}
$SD \leftarrow 0.2$
{Be aware of the dangerous proportion of sparrows}
$GbestPosition \leftarrow X$
for $i \leftarrow 1$ *to MaxIter by* 1 **do**
  $X \leftarrow PDUpdate(X, ST, dim)$
  $X \leftarrow JDUpdate(X, PD, dim)$
  $X \leftarrow SDUpdate(X, SD, dim, fitness)$
  $GbestPosition \leftarrow X$
  { Update sparrow position by formula (5) to (7)}
  $factor \leftarrow 1 - i/MaxIter$
  { Inertia factor}
  for $j \leftarrow 1$ *to pop by* 1 **do**
    if $random() > factor$ **then**
      $L \leftarrow Levy(dim)$
      {Levy flight mutation on selected individuals}
      $ds \leftarrow L \cdot (X[j, :] - GbestPosition[0, :])$
      $Temp \leftarrow X[j, :] + ds$
      $fitnew \leftarrow Fun[Temp[0, :]]$
      {Calculate the new fitness value}
    end if
  end for
end for
$fitness \leftarrow Sort(fitness)$
{Sort the new fitness values}
$X \leftarrow SortPosition(X)$
{ Population sorting}
$GbestPositon[0, :] \leftarrow copy.copy(X[0, :])$
{Update the optimal individual position}
return *GbestPosition*

TABLE 2. Data from ten experiments created by UCI and KEEL and manually.

UCI data sets	The total number of	A few class	Most of the class	Imbalance rate
Yeast3	1484	163	1321	8.1
Ecoli2	336	52	284	5.46
Blood	748	178	570	3.2
Glass0	214	70	144	2.06
Seeds	210	70	140	2.0
Pima	768	268	500	1.86
Ionosphere	351	126	225	1.78
Breast	699	241	458	1.90
Sim1	1000	226	774	3.42
Sim2	1000	262	738	2.81
Sim3	1000	282	718	2.54

dataset. In the classification results of imbalanced datasets, the accuracy of a few samples may be too low, so the classification accuracy cannot represent the classification results. Therefore, *F_measure* and *G_mean* were used as evaluation indicators, and both were calculated based on a confusion matrix [29]. See Table (3) for an example.

TABLE 3. Confusion matrix.

	Predicted more class	Predicted less class
More than the actual class	TP	FN
Less than the actual class	FP	TN

According to Table (3), the following evaluation indicators are easy to calculate:

The recall rate for most samples is shown in (15).

$$rr_{TP} = \frac{TP}{TP + FN} \times 100\% \tag{15}$$

The recall rate for minority samples is shown in (16).

$$rr_{TN} = \frac{TN}{FN + TN} \times 100\% \tag{16}$$

The accuracy of the minority samples is shown in (17).

$$pr_{TN} = \frac{TN}{FP + TN} \times 100\% \tag{17}$$

The *G_mean* value is shown in (18).

$$G\_mean = \sqrt{rr_{TN} \times rr_{TP}} \times 100\% \tag{18}$$

The *F_measure* value is shown in (19).

$$F\_measure = \frac{2rr_{TN} \times pr_{TN}}{rr_{TN} + pr_{TN}} \times 100\% \tag{19}$$

The recall rate of the two types of samples is considered by the *G_mean* value, which becomes larger in the case of a larger recall rate of the two types of samples, with a larger *G_mean* value indicating a stronger classification ability of the model for different types of samples. Hence, *G_mean* can perfectly reveal the performance of the model. The accuracy of the classification and the recall rate of minority samples are considered comprehensively by the *F_measure* value, which excellently reveals the accuracy of minority samples, with a larger *F_measure* value indicating a more accurate classification of the model for minority samples. Therefore, the larger *G_mean* and *F_measure* indicate that the model is more effective in classifying imbalanced data. In this case, *G_mean* value and *F_measure* value are mainly used in the experimental part of the paper to evaluate the classification performance of the model.

C. EXPERIMENTAL RESULTS

When the data sets are the same, the sampling multiplier of each sampling method is set to three. The LSSASMOTE+SVM model was hereby compared with the SMOTE+SVM model [8], the SSMOTE+SVM model [30], the LD-SMOTE+SVM model [31], the L-SMOTE+SVM model [32] and the FTL-SMOTE+Mixed Kernel SVM models [15]. The SVM parameter C in the above algorithm uses the default regularization parameter value of 1.0, and the parameter g uses the default value of auto.

The *G_mean* and *F_measure* values of each classification model in different datasets are shown in Table (4). According to the experimental results shown in Table (4), the *F_measure*

TABLE 4. Comparison of the LSSASMOTE+SVM model with other models.

Data set	Classification model of	$G\_mean/\%$	$F\_measure/\%$
Yeast3	SMOTE+SVM	75.43	73.09
	SSMOTE+SVM	76.16	50.81
	LD-SMOTE+SVM	74.47	49.17
	L-SMOTE+SVM	70.06	42.27
	FTL-SMOTE+SVM	81.10	62.77
	LSSASMOTE+SVM	**84.25**	**83.22**
Ecoli2	SMOTE+SVM	94.93	95.30
	SSMOTE+SVM	92.25	84.86
	LD-SMOTE+SVM	90.92	80.79
	L-SMOTE+SVM	87.86	75.95
	FTL-SMOTE+SVM	95.38	93.25
	LSSASMOTE+SVM	**98.80**	**97.78**
Blood	SMOTE+SVM	65.77	60.29
	SSMOTE+SVM	64.79	46.55
	LD-SMOTE+SVM	67.38	50.70
	L-SMOTE+SVM	67.59	49.56
	FTL-SMOTE+SVM	70.25	54.21
	LSSASMOTE+SVM	**82.78**	**74.83**
Glass0	SMOTE+SVM	80.79	73.41
	SSMOTE+SVM	78.92	71.15
	LD-SMOTE+SVM	68.81	62.41
	L-SMOTE+SVM	76.35	69.33
	FTL-SMOTE+SVM	85.94	79.23
	LSSASMOTE+SVM	**94.36**	**88.81**
Seeds	SMOTE+SVM	92.89	89.35
	SSMOTE+SVM	93.60	90.19
	LD-SMOTE+SVM	94.56	92.28
	L-SMOTE+SVM	94.66	91.60
	FTL-SMOTE+SVM	98.08	**97.43**
	LSSASMOTE+SVM	**98.64**	97.13
Pima	SMOTE+SVM	74.85	72.43
	SSMOTE+SVM	75.86	65.87
	LD-SMOTE+SVM	80.25	75.67
	L-SMOTE+SVM	73.42	69.73
	FTL-SMOTE+SVM	87.64	80.35
	LSSASMOTE+SVM	**89.73**	**82.65**
Ionosphere	SMOTE+SVM	90.26	89.12
	SSMOTE+SVM	94.73	93.42
	LD-SMOTE+SVM	92.41	93.83
	L-SMOTE+SVM	92.67	94.69
	FTL-SMOTE+SVM	94.74	**96.16**
	LSSASMOTE+SVM	**97.64**	95.53
Breast	SMOTE+SVM	95.86	94.07
	SSMOTE+SVM	95.93	94.02
	LD-SMOTE+SVM	96.51	94.90
	L-SMOTE+SVM	95.57	94.19
	FTL-SMOTE+SVM	**97.94**	**97.18**
	LSSASMOTE+SVM	97.51	95.86
Sim1	SMOTE+SVM	75.91	74.81
	SSMOTE+SVM	86.73	87.78
	LD-SMOTE+SVM	82.21	81.14
	L-SMOTE+SVM	81.54	84.59
	FTL-SMOTE+SVM	87.64	89.58
	LSSASMOTE+SVM	**94.81**	**93.35**
Sim2	SMOTE+SVM	73.56	74.37
	SSMOTE+SVM	77.57	78.72
	LD-SMOTE+SVM	79.61	74.95
	L-SMOTE+SVM	82.37	84.45
	FTL-SMOTE+SVM	84.14	83.35
	LSSASMOTE+SVM	**89.62**	**90.34**
Sim3	SMOTE+SVM	68.43	70.43
	SSMOTE+SVM	70.33	71.85
	LD-SMOTE+SVM	76.75	74.30
	L-SMOTE+SVM	72.94	70.39
	FTL-SMOTE+SVM	82.31	83.42
	LSSASMOTE+SVM	**86.13**	**85.28**

TABLE 5. Comparison of LSSASMOTE optimized SVM classifier with other classifiers.

Data set	Classification model of	$G\_mean/\%$	$F\_measure/\%$
Yeast3	SVM	76.73	77.32
	Bayes	55.65	54.56
	C4.5	93.72	83.15
	Random Forest	82.82	80.33
	AdaBoost	81.33	85.24
	LSSA-SVM	**84.25**	**83.22**
Ecoli2	SVM	84.22	83.76
	Bayes	77.48	79.23
	C4.5	96.62	93.32
	Random Forest	98.22	96.92
	AdaBoost	69.52	68.26
	LSSA-SVM	**98.80**	**97.78**
Blood	SVM	**82.93**	**76.77**
	Bayes	60.06	62.14
	C4.5	82.44	74.45
	Random Forest	83.13	76.76
	AdaBoost	72.24	68.43
	LSSA-SVM	82.78	74.83
Glass0	SVM	83.76	81.15
	Bayes	71.24	67.74
	C4.5	93.04	86.62
	Random Forest	93.87	90.16
	AdaBoost	53.42	51.13
	LSSA-SVM	**94.36**	**88.81**
Seeds	SVM	96.54	94.58
	Bayes	94.76	92.75
	C4.5	98.17	96.33
	Random Forest	97.25	94.52
	AdaBoost	97.36	94.46
	LSSA-SVM	**98.64**	**97.13**
Pima	SVM	78.55	74.77
	Bayes	75.86	75.72
	C4.5	88.26	77.88
	Random Forest	89.15	80.33
	AdaBoost	80.95	85.24
	LSSA-SVM	**89.75**	**82.13**
Ionosphere	SVM	97.37	**96.96**
	Bayes	84.92	87.72
	C4.5	91.25	83.32
	Random Forest	95.74	92.25
	AdaBoost	94.24	88.87
	LSSA-SVM	**97.63**	95.54
Breast	SVM	96.91	95.62
	Bayes	96.85	97.21
	C4.5	95.24	90.71
	Random Forest	**98.34**	**96.72**
	AdaBoost	98.06	96.17
	LSSA-SVM	97.78	95.64
Sim1	SVM	82.97	84.24
	Bayes	79.16	80.61
	C4.5	81.68	90.56
	Random Forest	94.24	90.87
	AdaBoost	86.03	84.47
	LSSA-SVM	**94.81**	**93.32**
Sim2	SVM	85.75	85.66
	Bayes	80.43	80.13
	C4.5	82.73	86.18
	Random Forest	88.46	89.13
	AdaBoost	88.65	86.46
	LSSA-SVM	**89.62**	**90.34**
Sim3	SVM	80.14	80.24
	Bayes	73.75	71.73
	C4.5	83.75	78.74
	Random Forest	85.34	84.13
	AdaBoost	83.13	81.15
	LSSA-SVM	**86.13**	**85.28**

values of the LSSASMOTE+SVM classification model in the Yeast3, Ecoli2, Blood, Glass0, and Pima datasets are better than those of other models. Moreover, the *G_mean* value has also achieved favorable results in six datasets, namely Yeast3, Ecoli2, Blood, Seeds, Pima, and Ionosphere, and the average *G_mean* across all eight datasets is 9.11% higher than that of the basic SMOTE+SVM model. The

TABLE 6. The running time of each algorithm in a partial data set.

Data set	Classification model of	Running time/s
Yeast3	SMOTE+SVM	2.68
	SSMOTE+SVM	2.29
	LD-SMOTE+SVM	3.63
	L-SMOTE+SVM	2.94
	FTL-SMOTE+SVM	57.75
	LSSASMOTE+SVM	**148.61**
Ecoli2	SMOTE+SVM	1.34
	SSMOTE+SVM	1.26
	LD-SMOTE+SVM	2.74
	L-SMOTE+SVM	1.42
	FTL-SMOTE+SVM	10.68
	LSSASMOTE+SVM	**30.52**
Blood	SMOTE+SVM	1.47
	SSMOTE+SVM	1.93
	LD-SMOTE+SVM	2.68
	L-SMOTE+SVM	2.43
	FTL-SMOTE+SVM	6.78
	LSSASMOTE+SVM	**19.94**

LSSASMOTE+SVM model removes the effects of noise during its process, resulting in the model having better classification results on three artificially created simulated datasets. However, due to the increasing percentage of noise, the classification performance of the model in this paper shows a decreasing trend.

To further assess the classification effectiveness of the LSSASMOTE+SVM model and its superiority over other classifiers, the model was compared experimentally with unoptimized SVM, Bayes [33], C4.5 [34], Random Forest [35], and AdaBoost [36] using eleven datasets presented in Table (2). Both the LSSASMOTE+SVM and other classifiers used the dataset after LSSA-SMOTE optimization. The experimental results are presented in Table (5).

Based on the experimental results, it can be concluded that the LSSA-SVM model outperforms most of the other classifiers for the majority of datasets. Although, in some cases, the random forest and C4.5 algorithms also exhibit good performance. Therefore, the proposed LSSASMOTE+SVM model not only optimizes imbalanced datasets at the data level but also selects a suitable combination of parameters for the SVM classifier at the algorithm level, resulting in more accurate classification of imbalanced data.

During the experiments, it is found that the LSSASMOTE+SVM model has a longer running time than other classification models. Some of the dataset runtimes are shown in Table (6). Table (6) displays three datasets with different imbalance rates, and it is evident that the LSSASMOTE+SVM algorithm requires more computational time. This is because the LSSA algorithm employs multiple sparrow individuals for the search process, along with a significant number of random perturbations and local search strategies, which necessitate more computational resources and time. Furthermore, the LSSA algorithm in this paper optimizes parameter search for both the sampling and classification algorithms, which also contributes to the higher running time.

VI. CONCLUSION

The LSSASMOTE+SVM classification model was proposed by combining the improved sparrow search algorithm with the SMOTE and SVM algorithms. The experiments confirm the feasibility of the proposed model. LSSA selects the best combination of parameters for this model, improves the blindness of the selection of sampling rate of the SMOTE algorithm, and obtains a better balanced and stable data set. The choice of this parameter combination is conducive to improving the classification accuracy of the SVM classifier while dealing with imbalanced data. Overall, the LSSASMOTE+SVM classification model proposed in this paper has a good classification effect on imbalanced data. However, more efforts are still required to reduce the run time and improve the classification accuracy of the LSSASMOTE algorithm.

In addition, there is still room for improvement in the multi-classification problem, and further research is needed on the time complexity and boundary partition.

REFERENCES

[1] J. Wang and J. Yan, "Classification algorithm based on undersampling and cost-sensitiveness for unbalanced data," *Comput. Appl.*, vol. 41, no. 1, pp. 48–52, 2021.

[2] C. Tian and L. Zhou, "Credit assessment method based on majority weight minority oversampling technique and random forest," *Comput. Appl.*, vol. 39, no. 6, pp. 1707–1712, 2019.

[3] B. Yang, L. Shi, G. Chi, and Y. Dong, "Design and application of credit rating model based on BPNN-LDAMCE based on unbalanced data," *J. Quant. Tech. Econ.*, vol. 39, no. 3, pp. 152–169, 2022.

[4] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[5] P. Iba and W. Langley, "Experiments with easyensemble," in *Proc. Int. Conf. Artif. Intell. (ICAI)*, 2000, pp. 9–16.

[6] M. S. Santos, P. H. Abreu, N. Japkowicz, A. Fernández, C. Soares, S. Wilk, and J. Santos, "On the joint-effect of class imbalance and overlap: A critical review," *Artif. Intell. Rev.*, vol. 55, no. 8, pp. 6207–6275, Dec. 2022.

[7] J. Chen and Z. Zheng, "Over-sampling method on imbalanced data based on WKmeans and smote," *Comput. Eng. Appl.*, vol. 57, no. 23, pp. 106–112, 2021.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[9] D. Meng and Y. Li, "An imbalanced learning method by combining SMOTE with center offset factor," *Appl. Soft Comput.*, vol. 120, May 2022, Art. no. 108618.

[10] B. Krawczyk, C. Bellinger, R. Corizzo, and N. Japkowicz, "Undersampling with support vectors for multi-class imbalanced data classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–7.

[11] H. Yudan, G. Qing, C. Zhihua, and Y. Lei, "Classification method for imbalance dataset based on genetic algorithm improved synthetic minority over-sampling technique," *J. Comput. Appl.*, vol. 35, no. 1, pp. 121–124, 2015.

[12] L. Wang, M. Han, X. Li, N. Zhang, and H. Cheng, "Review of classification methods for unbalanced data sets," *Comput. Eng. Appl.*, vol. 57, no. 22, pp. 42–52, 2021.

[13] V. Vapnik, "Statistical learning theory," *Ann. Inst. Stat. Math.*, vol. 55, no. 2, pp. 371–389, Aug. 2003.

[14] Z. Sun, G. Wang, M. Gao, L. Gao, and A. Jiang, "Research on excess location method of sealed electronic equipment based on parameter optimization support vector machine," *Electron. Meas. Instrum.*, vol. 35, no. 8, pp. 162–174, 2021.

[15] K. Luo and G. Wang, "Research on imbalanced data classification based on L-SMOTE and SVM," *Comput. Eng. Appl.*, vol. 55, no. 17, pp. 55–61, 2019.

[16] H. Ma and M. Zhu, "IgwoSMOTE: An over sampling method based on improved gray wolf algorithm for SVM imbalanced data classification," *Comput. Eng. Sci.*, vol. 44, no. 6, pp. 1133–1140, 2022.
[17] R. Xiao, Z. Feng, and J. Wang, "Concept discrimination, research progress and application analysis of swarm intelligence," *J. Nanchang. Inst. Technol.*, vol. 41, no. 1, pp. 1–21, 2022.
[18] L. Zhang, Y. Zhang, and G. Song, "LSSA-based feature extraction and classification of hyperspectral images with small training samples," *Remote Sens. Lett.*, vol. 8, no. 7, pp. 625–634, 2017.
[19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
[20] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Aug. 1997.
[21] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
[22] I. Tomek, "Two modifications of CNN," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 11, pp. 769–772, Nov. 1976.
[23] X. Fan and L. Cui, "Antitumor drug target prediction method based on network attribute and its application," *Data Anal. Knowl. Discovery*, vol. 2, no. 12, pp. 98–108, 2018.
[24] T.-B. Du, G.-H. Shen, Z.-Q. Huang, Y.-S. Yu, and D.-X. Wu, "Automatic traceability link recovery via active learning," *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, no. 8, pp. 1217–1225, Aug. 2020.
[25] L. Cai, G. Li, J. Fang, and L. Yu, "Research on imbalanced data clustering mining for urban hot spots," *Comput. Sci.*, vol. 46, no. 8, pp. 16–22, 2018.
[26] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020.
[27] W. Ma and X. Zhu, "Sparrow search algorithm based on Levy flight disturbance strategy," *J. Appl. Sci.*, vol. 40, no. 1, pp. 116–130, 2022.
[28] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 49, no. 5, pp. 4677–4683, May 1994.
[29] A. Arshad, S. Riaz, and L. Jiao, "Semi-supervised deep fuzzy C-mean clustering for imbalanced multi-class classification," *IEEE Access*, vol. 7, pp. 28100–28112, 2019.
[30] C. Wang, Z. Pan, L. Dong, and C. Ma, "Research on classification for imbalanced dataset based on improved smote," *Comput. Eng. Appl.*, vol. 49, no. 2, pp. 184–187, 2013.
[31] X. Wen, J. Chen, W. Jing, and K. Xu, "Research on optimization of classification model for imbalanced data set," *Comput. Eng.*, vol. 44, no. 4, pp. 268–273, 2018.
[32] B. Yi, J. Zhu, and J. Li, "Imbalanced data classification on micro-credit company customer credit risk assessment using improved smote support vector machine," *Chin. J. Manag. Sci.*, vol. 24, no. 3, pp. 24–30, 2016.
[33] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
[34] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.
[35] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[36] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, pp. 119–139, Aug. 1995.

ZHI WANG was born in 1998. He is currently pursuing the degree with the School of Computer and Control Engineering, Yantai University, Yantai, Shandong, China. His main research interest includes imbalanced data.

QICHENG LIU was born in 1970. He received the Ph.D. degree. He is currently a Professor with the School of Computer and Control Engineering, Yantai University, Yantai, Shandong, China. His main research interests include big data, intelligent information processing, and data mining.