#### **ORIGINAL ARTICLE**



# Hybridization of ring theory-based evolutionary algorithm and particle swarm optimization to solve class imbalance problem

Sayan Surya Shaw<sup>1</sup> · Shameem Ahmed<sup>2</sup> · Samir Malakar<sup>3</sup> · Laura Garcia-Hernandez<sup>4</sup> · Ajith Abraham<sup>5</sup> · Ram Sarkar<sup>2</sup>

Received: 27 July 2020 / Accepted: 22 February 2021 © The Author(s) 2021

#### Abstract

Many real-life datasets are imbalanced in nature, which implies that the number of samples present in one class (minority class) is exceptionally less compared to the number of samples found in the other class (majority class). Hence, if we directly fit these datasets to a standard classifier for training, then it often overlooks the minority class samples while estimating class separating hyperplane(s) and as a result of that it missclassifies the minority class samples. To solve this problem, over the years, many researchers have followed different approaches. However the selection of the true representative samples from the majority class is still considered as an open research problem. A better solution for this problem would be helpful in many applications like fraud detection, disease prediction and text classification. Also, the recent studies show that it needs not only analyzing disproportion between classes, but also other difficulties rooted in the nature of different data and thereby it needs more flexible, self-adaptable, computationally efficient and real-time method for selection of majority class samples without loosing much of important data from it. Keeping this fact in mind, we have proposed a hybrid model constituting Particle Swarm Optimization (PSO), a popular swarm intelligence-based meta-heuristic algorithm, and Ring Theory (RT)-based Evolutionary Algorithm (RTEA), a recently proposed physics-based meta-heuristic algorithm. We have named the algorithm as RT-based PSO or in short RTPSO. RTPSO can select the most representative samples from the majority class as it takes advantage of the efficient exploration and the exploitation phases of its parent algorithms for strengthening the search process. We have used AdaBoost classifier to observe the final classification results of our model. The effectiveness of our proposed method has been evaluated on 15 standard real-life datasets having low to extreme imbalance ratio. The performance of the RTPSO has been compared with PSO, RTEA and other standard undersampling methods. The obtained results demonstrate the superiority of RTPSO over state-of-the-art class imbalance problem-solvers considered here for comparison. The source code of this work is available in https://github.com/Sayansurya/RTPSO\_Class\_imbalance.

Keywords Class Imbalance Problem · PSO · Ring Theory based Evolutionary Algorithm · RTPSO · Evolutionary Algorithm

Laura Garcia-Hernandez ir1gahel@uco.es

Sayan Surya Shaw sayansurya1999@gmail.com

Shameem Ahmed shameemahmed20apr2000@gmail.com

Samir Malakar malakarsamir@gmail.com

Ajith Abraham ajith.abraham@ieee.org

Ram Sarkar ram.sarkar@jadavpuruniversity.in

Published online: 09 March 2021

- <sup>1</sup> Department of Computer Science and Engineering, University of Calcutta, Kolkata, India
- <sup>2</sup> Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
- <sup>3</sup> Department of Computer Science, Asutosh College, Kolkata, India
- <sup>4</sup> Area of Project Engineering, University of Cordoba, Cordoba, Spain
- <sup>5</sup> Machine Intelligence Research Labs: Auburn, Auburn, Washington, US



## Introduction

In classical machine learning, the direct application of standard classifiers is logical only when the number of samples found in each class of the considered classification problem is balanced which is an ideal case. But in some real-life datasets like disease prediction datasets the number of samples in each class is often unequal i.e., the problem of class imbalance is present therein. Standard classifiers are not enough to predict the results precisely on these datasets. Imbalance class means there is a disproportionate ratio of observations in each class. Training the classifiers directly on such datasets may affect the model performance [69]. In many cases, the imbalanced ratio is so extreme that the standard classifiers in use are often become biased towards the majority class (sometimes called as "negative" class) and overlook the minority class (sometimes called as "positive" class) examples during training for estimating class separating hyperplane(s) and as a result, these classifiers tend to predict samples of majority class as minority class.

This class imbalance problem is very common in many applications like medical diagnosis or monitoring, detection of oil spills in satellite radar images, fraud detection [33], text classification [41], information retrieval and filtering [16, 36], twitter spam detection [42], detection of adverse drug reaction [53], 5G future network [2] and many more. In this scenario, standard classifiers become almost biased towards the majority class (class having more samples) instances and try to predict them correctly, whereas treating the samples from minority class as noise or sometimes ignore them [27]. As a result, the minority class samples are often misclassified as the members of the majority class. To be specific, here the challenge is how precisely these minority class samples can be predicted, also to preserve the accuracy in the prediction of the majority class samples.

Throughout the years, many researchers have used different approaches to deal with the class imbalance problem. Two categories of techniques are mainly followed to cope up with the class imbalanced datasets that are cost sensitive techniques and sampling techniques [65]. A cost sensitive technique is a sub-field of machine learning and it is used to minimize the cost of training by taking the costs of prediction errors and potentially other costs into account. These techniques can be divided into three groups: data resampling, algorithm modification, and ensemble method. A cost-sensitive technique does not improve the data distribution [13]. Rather the goal of the cost sensitive learning is to minimize the cost of a model on a training dataset. The other category of techniques i.e., sampling techniques encompass two different methods: oversampling and undersampling. In oversampling, the minority class instances are increased by adding more synthesized training data to balance the ratio of the two classes. It can be random oversampling or syn-



thetic minority oversampling technique (SMOTE) [14]. But SMOTE is not beneficial for high-dimensional data [9]. In undersampling techniques, the majority class instances are merged or removed to make a good balance between the number of samples of the two classes. Some of the undersampling techniques are random undersampling (RU) [47], edited nearest neighbors (ENN) rule [70], nearmiss undersampling (NMU) [7], Condensed nearest neighbors (CNN) [55] etc. However, the limitation of using only one of these undersampling methods is that it may not be able to select the most important data samples from the majority class and thereby it might remove some of the crucial and important data from the majority class samples.

Moreover, the above techniques are largely data-dependent and may fail if the same algorithm is applied to other datasets. Hence, a more flexible and self-adaptable algorithm that considers the essence of the majority class data in the underlying classification problem prior to removal is required [65]. To fulfill the requirement, in the recent past, some researchers have applied different optimization algorithms because optimization algorithms are more self-adaptable and can upgrade their fitness value in each iteration. The mostly used optimization algorithms for this problem are Genetic Algorithm (GA) [37], Particle Swarm Optimization (PSO) [39], Ant Colony Optimization [22] [66] etc.

Considering the success of these methods over the traditional ones, we have proposed a hybrid method, termed RTPSO, where two meta-heuristic algorithms Ring Theory (RT)-based Evolutionary Algorithm (RTEA) [34] and PSO are used. RTEA has been used for feature selection (FS) by Ahmed et al. [1] with Harmony Search (HS) algorithm [30] and produced promising results, which has inspired us to apply it with PSO for solving the problem in hand. From the best of our knowledge, the proposed method is completely original and it has been used for the first time to solve said problem. We have used Area Under Curve (AUC) of the ROC (ROC-AUC) score, precision-recall (PR) of AUC (PR-AUC) and F1 score of the AdaBoost classifier [35] for evaluation of our proposed model because typical recognition accuracy score does not reflect the misclassification rate of the minority class samples. The proposed RTPSO algorithm has been experimented on 15 standard and publicly available class imbalance datasets. We have also compared our method with some classic as well as recently proposed methods related to the class imbalance problem.

The entire process of our work is organized as follows: we have discussed some past works in this domain in section "Literature survey" and we have discussed some preliminary techniques used for this research work in section "Preliminaries". In section "Proposed method" the proposed method has been described in detail. We have discussed the datasets and analysed the experimental results elaborately in section "Experimental results and discussion". In section "Experimental results and discussion" we have also compared performance of our method with other state-of-the-art methods and in section "conclusion" we have concluded our work.

#### Literature survey

Over the years many researchers have developed different methods to deal with this class imbalance problem. In this section, we have discussed some popular algorithms that have been used to solve the class imbalance problem.

Chawla et al. [14] introduced the combination of undersampling on majority class samples and oversampling on minority class samples. They oversampled the minority class samples by creating "synthetic" examples. The authors claimed to get a better classification performance on receiver operating characteristic (ROC) curve than normal undersampling technique on Pima Indian diabetes, Phoneme datasets etc. Selecting the nearest neighbors with a focus on examples improved the misclassification rate. But, this approach was not able to handle datasets with all nominal features. Yang et al. [65] proposed a method using PSO with multiple classifiers and evaluation metrics. They experimented on class imbalanced datasets like breast cancer, diabetes etc. But the method did not consider highly imbalanced datasets, hence the performance of the proposed method on such datasets can not be ensured. Liu et al. [43] used RU on SEER breast cancer datasets to balance it. They used the Bagging algorithm to construct an integration of the decision tree model. However, the use of only RU in data preprocessing stage might not be enough because the samples which were removed from the majority class might hold important data, and as a result, the model might fail to produce correct results on the unknown samples.

Anand et al. [3] used the undersampling technique on highly imbalanced datasets with support vector machine (SVM) as classifier to improve the sensitivity. The authors selected the "boundary samples" from the majority classes i.e., the samples of two classes lying close to each other. They used their proposed method on four datasets: micropred, xwchen, active-site and cysteine. In another work, Thomas [57] proposed prototype generation using K-means clustering algorithm and claimed that it can be used in high dimensional datasets also. But, if the variances of the clusters are not so different, the proposed method's performance is guite similar to normal K-means approach and the border region development could be more improved. Gao et al. [28] proposed a method in their paper, where a combination of SMOTE and PSO with radial basis function (RBF) classifier used on Pima Indian diabetes, ADI and Haberman survival datasets. In this work, the authors had considered only a few datasets to evaluate their proposed method and hence the method may be dataset dependant. Cao et al. [12] proposed a wrapper approach along with cost-sensitive neural network model, where the optimization was based on the PSO. They claimed that the experimental results on datasets like hepatitis, abalone and segment are effective than normal sampling methods. Samma et al. [52] introduced a model using PSO and Fuzzy SVM (FSVM), named as PSO-FSVM model for tackling the class imbalance problem. The experiment was performed only on MIAS mammogram dataset which indicates that the algorithm might be dataset dependant.

Prusa et al. [47] used RU method and claimed to have a significant improvement in classification performance. Zhu et al. [70] implemented ENN undersampling method and adaptive synthetic oversampling approach to solve the class imbalance problem and they also used the two-step FS technique to optimize the feature set. The technique mentioned by Bunkhumpornpat and Sinapiromsaran [11], used the density-based majority undersampling technique (DBMUTE) that has the ability to adapt directly density reachable graph. They showed improved results on UCI health monitoring datasets: Haberman's survival and diabetes.

Bao et al. [7] proposed a new method called: Boosted Near-miss Under-sampling on SVM ensembles (BNU-SVMs) and they also used a kernel-distance pre-computation technique to improve the model performance in high dimensional features. Shekarforoush et al. [55] performed a case study in the resampling techniques like CNN, Cluster Centroids (CC) and SMOTE. Vu et al. [60] introduced an application of the deep learning-based approach, called Auxiliary Classifier Generative Adversarial Network, to address the class imbalance problem by generating synthesized data samples in network traffic data classification. But, in general, deep learning models need a huge amount of data to get trained properly which may not be available for many real-world datasets. To handle the class imbalance problem, Rayhan et al. [48] introduced a new clustering-based undersampling approach with boosting (AdaBoost) algorithm i.e., CUSBoost. The authors claimed that CUSBoost algorithm is capable to handle highly imbalanced datasets effectively. They evaluated the proposed method on 13 imbalance datasets.

In the paper, Aydogan et al. [5] reported a new costsensitive classification method (CBR-PSO) using PSO and rough set theory. The algorithm tested on datasets like a brain tumour, leukaemia and lung cancer. The proposed method could be combined with other multi-objective heuristic algorithms or extended with rule pruning approaches to produce a better result. Zhang and Chen [68] proposed a method by incorporating random oversampling, K-means and SVM, dubbed as RK-SVM model. The authors worked on the imbalanced WDBC, Pima and Iris datasets. But this approach is computationally expensive and the algorithm is not tested on highly imbalanced datasets.

In the recent past, many researchers used evolutionary algorithms to improve prediction accuracy. Yu et al. [67]



worked with accelerating evolutionary computation, Cheng et al. [17] introduced various model-based evolutionary algorithms (MBEAs) and He et al. [34] proposed evolutionary multi-objective optimization and used it on real-world applications. Gautheron et al. [29] introduced Mahalanobis metric learning (IML) algorithm. The authors have used the datasets like Pima, Balance, Splice and Heart to evaluate their model's performance. The proposed algorithm could be adapted to learn non-linear metrics. Unal et al. [59] created diversity in using Multi-Objective PSO (MOPSO) by using random immigrants approach. The application of the proposed solution is tested in four different sets using Generational Distance, Spacing, Error Ratio, and Run Time performance measures. Wang et al. worked with Multiple-Strategy Learning PSO (MSL-PSO) algorithm [61] to solve the problem efficiently with large scale variables.

Li et al. [40] proposed adaptive cost-sensitive learning, by developing the model on a sparse cost matrix with a diagonal form. They also used evolutionary extreme learning machine with multi-objective function to optimize the solution. They experimented with their method on real-world datasets like yeast, hayes-roth, ecoli and page blocks. In another work, Wang et al. [62] used a sampling approach for an imbalanced dataset via self-placed learning (ISPL). ISPL is so designed that it can select high-quality samples from the majority class to balance between majority and minority classes. They executed the proposed method on four publicly available breast cancer datasets. But the imbalance ratio was low in all the cases. Therefore, the proposed method did not guarantee to work the same on highly imbalanced datasets. Ghosh et al. [31] proposed a meta-heuristic algorithm, namely adaptive  $\beta$ -hill climbing (A $\beta$ HC) with BSF optimizer (A $\beta$ BSF) in solving FS problem. Ahmed et al. [1] also introduced a new hybrid meta-heuristic FS model based on a well-known metaheuristic HS algorithm and a recently proposed RT-based Evolutionary Algorithm (RTEA), which was named as RTbased HS (RTHS). Both the methods achieved decent results in the FS domain. Therefore, we have implemented these two algorithms on class imbalance problem and have compared them with our proposed method.

From the above discussion, it can be observed that if we can apply a generalized and self-adaptable method in the data preprocessing stage before classifying data in the imbalanced datasets, the classification result would be much better. With this line of thought, we have proposed a hybrid optimization method, called RTPSO, to balance the imbalanced dataset more intelligently, and evaluated our model performance by AdaBoost classifier. Highlights of this work are as follows:

1. Well-known swarm intelligence based optimization algorithm PSO is hybridized with RTEA, a recently proposed optimization algorithm, to solve the class imbalance problem for the first time to the best of our knowledge.



- The proposed method, called RTPSO, has been assessed in terms of ROC-AUC, PR-AUC or F1 score using AdaBoost classifier and evaluated on 15 standard and publicly available datasets where imbalance ratio varies from moderate to extremely high.
- 3. The performance of the proposed method has been compared with some conventional as well as recently published methods.

## **Preliminaries**

#### Particle swarm optimization

PSO, proposed by Kennedy and Eberhart [39], is a swarm intelligence-based meta-heuristic algorithm that can solve complex optimization problems. It is inspired by the social behavior of a flock of birds, school of fishes etc. [18]. PSO algorithm uses a bunch of particles that are called swarm. Each particle, denoted by a point in a D dimensional space, is initialized with random velocity, which can move around and explore the search space. Here, D represents the dimension of the search space. In every iteration, each particle keeps track of their individual best fitness value and the best fitness value acquired by the whole population. By simultaneous updating of the best position (position with the best fitness value), it moves towards the global optimum position [6].

This algorithm also encompasses some tuning parameters that make a great impact on the performance of this algorithm, often expressed as exploration-exploitation trade-off [58]. Exploration implies of probing various regions in the search space with the hope of finding a better solution, maybe the global one. Exploitation means searching only on promising candidates to find the local optimum solution accurately. The mathematical illustration of the PSO algorithm is as follows.

Let,  $X_i$  is the *i*<sup>th</sup> particle in the *D* dimensional space *S* and it is denoted as below.

$$X_i = (X_{i_1} X_{i_2} X_{i_3} X_{i_4} \dots X_{i_D})$$
(1)

Let, there are N particles in S. Now, the whole population can be presented as:

$$X = \{X_1, X_2, X_3, X_4, \dots, X_N\}$$
(2)

The velocity  $(v_i)$  and position  $(X_i)$  of  $i^{th}$  particle at time k+1 are estimated using Eqs. 3 and 4 respectively [25].

$$v_i^{k+1} = wv_i^k + c_1r_1(P_i^k - X_i^k) + c_2r_2(P_g^k - X_i^k)$$
(3)

$$X_i^{k+1} = X_i^k + v_i^{k+1} (4)$$

In Eq. 3,  $w \in [0.8, 1.4]$  is called as inertia factor that decides the contribution rate of the velocity of the particle from previous to current time [23].  $v_i^k$  represents the velocity of the  $i^{th}$  particle at time  $k. c_1 \in [1.5, 2]$  and  $c_2 \in [2, 2.5]$  are the cognitive coefficient and social coefficient respectively.  $P_i^k$  and  $P_g^k$  represent the personal and global best solution at time k respectively.  $r_1$  and  $r_2$  are two diagonal matrices of dimension D with uniform random numbers between 0 and 1. In Eq. 4,  $X_i^k$  represents the position of the  $i^{th}$  particle at time k.

Inertia weight (i.e., w) plays an important role while PSO searches for the global best solution. With the larger value of w, the searching ability of PSO is improved while considering the whole search space. On the contrary, the searching ability of it is improved with a smaller value of w while considering partial space. According to the work mentioned in [6], this algorithm can quickly converge to the near optimal solution from a bigger search space when the value of w decreases linearly from 0.9 to 0.4. Altogether the iteration of the process is controlled by Eqs. 3 and 4 and continues until it reaches either the predefined fitness value (i.e., global optimum) or exceeds the maximum number of iterations [46].

#### **Ring theory-based evolutionary algorithm**

RTEA, proposed by He et al. [34], is a physics-based metaheuristic algorithm. It is inspired by the RT in mathematics. It is based on the algebraic theory and is mainly used in the combinatorial optimization problem. Two evolution parameters global exploration operator (R-GEO) and local development operator (R-LDO) are used for generating a new population following a greedy strategy.

## Ring

Definition: A nonempty set R (i.e.,  $R \neq \emptyset$ ) equipped with two binary operations, addition (+) and multiplication (.), is called a ring (mathematically represented as (R, +, .)) if it follows the ring axioms [26] [50] defined below.

Ring axioms:

- $\forall c, d \in R, c + d = d + c$
- $\forall c, d, e \in R, (c+d) + e = c + (d+e)$
- $-\exists 0 \in R$  such that  $\forall c \in R, c + 0 = 0 + c = c$ , here this 0 is called additive identity.
- $\forall c \in R, \exists d \in R \text{ such that } c + d = d + c = 0, \text{ here this } d \text{ is called additive inverse of } c, \text{ can be written as } -c.$
- ∃1 ∈ R such that  $\forall c \in R, c.1 = 1.c = c$ , here 1 is called multiplicative identity
- $\forall c, d, e \in R, (c.d).e = c.(d.e)$
- $\forall c, d, e \in R, c.(d+e) = c.d+c.e; (d+e).c = d.c+e.c$

Let,  $W_q = \{[0], [1], \dots, [q-1]\}$  be a collection of remainder classes of modulo q, where  $[f] = \{u \in W | u \cong f \pmod{q}\}$ , where  $0 \le u \le q-1, q > 1$ , and W is the set of integers. Here additive and multiplicative, two binary operations can be defined as following:

$$\label{eq:generalized_states} \begin{split} [\mathbf{i}] \oplus [\mathbf{j}] = [(i+j)mod \; q], [\mathbf{i}] \odot [\mathbf{j}] = [(ij)mod \; q], \forall i, \; j \in W_q \end{split}$$

Now, we can easily show that  $W_q$  along with the binary operations  $\oplus$  and  $\odot$  satisfies all the ring axioms and hence we can say  $(W_q, \oplus, \odot)$  is a ring.

#### **Direct product of rings**

The direct product of rings can be used as a method to construct a new ring with the help of two or more rings. It is a special type of ring, where every element is an ordered m-tuple (i.e., m number of rings are used to generate new ring). If  $R_j$  ( $j \in \tau = \{1, 2, 3, ..., q\}$  and  $q \ge 2$ ) are rings, then  $\prod_{j \in \tau} R_j = R_1 \times R_2 \times \cdots \times R_q$  is a ring consists of two binary operations ( $\odot$  and  $\oplus$ ) that can be defined by  $\langle c_1, c_2, ..., c_q \rangle \odot \langle d_1, d_2, ..., d_q \rangle = \langle c_1 d_1, c_2 d_2, ..., c_q d_q \rangle$  and  $\langle c_1, c_2, ..., c_q \rangle \oplus \langle d_1, d_2, ..., d_q \rangle = \langle c_1 + d_1, c_2 + d_2, ..., c_q + d_q \rangle$ . Here  $\prod_{j \in \tau}$  is the direct product of  $R_j$ ,  $j \in \tau$  [1].

#### **Overview of RTEA**

There exists a bijection  $V : \prod_{j=1}^{q} \mathbb{A}_{s_j} \longrightarrow \mathbb{A}[s_1, s_2, \dots, s_q]$ . So  $\mathbb{A}[s_1, s_2, \dots, s_q] = \{0, 1, \dots, s_1 - 1\} \times \{0, 1, \dots, s_2 - 1\} \times \dots \times \{0, 1, \dots, s_q - 1\}$ . Then RTEA has been proposed after drawing supports from addition, multiplication and inverse operations on  $\prod_{j=1}^{q}$ .

Let us have four randomly selected q-dimensional integer vectors:  $L_1, L_2, L_3$  and  $L_4$  from  $\mathbb{A}[s_1, s_2, \dots, s_q]$ , where  $L_1 = \langle l_{11}, l_{12}, \dots, l_{1q} \rangle, L_2 = \langle l_{21}, l_{22}, \dots, l_{2q} \rangle, L_3 = \langle l_{31}, l_{32}, \dots, l_{3q} \rangle, L_4 = \langle l_{41}, l_{42}, \dots, l_{4q} \rangle$ . These four vectors can be used to create a new q-dimensional integer vector  $L = \langle l_1, l_2, \dots, l_q \rangle \in \mathbb{A}[s_1, s_2, \dots, s_q]$  using Eq. 5

$$L_{j} = \begin{cases} \{L_{1j} + L_{4j} \times [L_{2j} + (s_{j} - L_{3j})]\} (mod \ s_{j}) \\ \text{if } rndm(k) \le 0.5; \\ \{L_{1j} + [L_{2j} + (s_{j} - L_{3j})]\} (mod \ s_{j}) \text{ else} \end{cases}$$
(5)

In Eq. 5, the procedure that generates a new q-dimensional integer vector, which has the ability of global learning, is called R-GEO. But the local exploration ability is also required with the global exploration to maintain the balance between local and global search abilities of the algorithm [4]. The new operator used to implement local search is called R-LDO in Algorithm 1.



#### Algorithm 1 Pseudocode of R-LDO

**Input**  $X = (x_1, x_2, ..., x_q) \in \mathbb{Z}[s_1, s_2, ..., s_q]$ local search probability  $Prb_m \in (0, 0.5]$ **Output**  $X = (x_1, x_2, ..., x_a)$ 1: 2: for i = 1 ... q do 3: if  $rndm1 < Prb_m$  then 4: if  $rndm^2 < 0.5$  and  $x_i \neq 0$  then 5:  $x_i \leftarrow s_i - x_i;$ 6٠ else 7:  $x_i \leftarrow rndm(\{0, 1, \dots, s_i - 1\} - \{x_i\});$ 8: end if Q٠ end if 10: end for

## Proposed method

#### Proposed model and solution representation

As the optimization algorithms have been proving their capabilities to solve problems efficiently of different domains for a long time, researchers are trying to come up with some new ideas to contribute more to the field of optimization. Because of such huge interest of the researchers, several new optimization algorithms have been proposed in the past decade. It may seem like there is no need for any such algorithm anymore. But, as per No Free Lunch (NFL) theorem [64], there is no such algorithm that is capable enough to solve every type of optimization problem efficiently. This conclusion of NFL theorem keeps the research area as active as earlier, and also keeps us motivated to come up with a new idea to solve a specific optimization problem, we have considered here i.e., class imbalance problem. In the present work, we have hybridized one of the most popular meta-heuristic algorithms PSO with a recently proposed meta-heuristic RTEA. Applicability of these two algorithms have already been shown in various optimization related problems.

The main reason for proposing a hybridization of two algorithms that work well in isolation is to fix some issues such as premature convergence and stagnation at local optima. Meta-heuristic algorithms suffer mainly from these two problems. Whereas, hybrid algorithms try to converge the solution and find the global optima with the help of exploration and exploitation operators of both the algorithms. Since the hybrid algorithm can be viewed as a union of the underlying algorithms so we expect it to perform better. In PSO algorithm, the particles update their position based on the past best positions and present the global best candidate in the solution. This strategy is used to explore as well as exploit the search space properly. RTEA uses the R-GEO operator for exploration and R-LDO operator for exploitation. However, the core searching strategies of PSO and RTEA may not be efficient. Here, the RTPSO is superior as it takes advantage of the exploration and the exploitation phases of its parent algo-



rithms for strengthening the search process. Not only does it come up with strong exploration and exploitation phases, but also successfully balances these two important phases. Since RTEA updates the solution using four randomly chosen solutions so it may mislead the search process without exploiting the neighbour. On the contrary, PSO has extensive exploitation capabilities but lacks of proper exploration ability and hence may lead to immature convergence. The union of these two helps us to overcome the disadvantages of the individual algorithms.

Normally, there are two ways of hybridizing metaheuristic algorithms: high-level approach and low-level approach [56]. In the high-level approach, we use the output of one algorithm as the input to the other to form a pipeline model. In this approach, methods are executed one after another until the termination condition is reached. The lowlevel style addresses the functional configuration of a single optimization algorithm. In this approach, one meta-heuristic is embedded into the other in such a way that a function in a meta-heuristic is replaced by another meta-heuristic. The proposed RTPSO follows the high-level approach of hybridization between PSO and RTEA. RTPSO is created in the anticipation of finding better solutions and a better convergence rate than PSO and RTEA. The flowchart of the proposed method is depicted in Fig. 1.

#### Working procedure

Initially, we divide our datasets into train and test sets. Then, we further divide the train set into temporary train and validation sets. The test set is utilized at the end to evaluate the model performance. First, the samples belonging to the majority class and minority class in the train set are identified and then from the samples of these majority class, we try to find out the samples (same in the number of minority class samples) which can represent the characteristics of whole majority class samples in the best way possible. In this context, we would like to mention that we have kept minority class samples intact. To guide this process, we take the help of the validation set for evaluating performance in each iteration of RTPSO. In short, in each iteration, we train the classifier on the selected train set and evaluate the performance of the learned module on the validation set. This performance score is used to calculate the fitness value of RTPSO algorithm.

At the initial phase, randomly generated population represents potential solutions which are updated in every iteration using the operators of PSO and RTEA. These solutions represent the combination of some samples that belong to the majority class of the train set.

To check the quality of a solution we take the help of ROC-AUC, PR-AUC score or F1 score. To calculate the fitness value in each iteration of RTPSO we follow the following steps.

## **Fig. 1** Flowchart of the proposed RTPSO used for solving class imbalance problem





- Initially, selected majority class samples and existing minority class samples of train set are combined to form a temporary train dataset.
- Temporary train dataset is used to train AdaBoost classifier and then evaluated on samples of the validation set to calculate the ROC-AUC, PR-AUC and F1 score.

After finding the near optimal solution i.e., the best combination of majority class samples from the train set, we create the final dataset by combining the selected majority class samples and the existing minority class samples of the train set. Next, we train our model based on this dataset and evaluate it on the test set which we have created at the beginning.

#### **Fitness function**

Fitness function is the guide of any optimization algorithm. The goal of the fitness function is to find a test datum that fits a given test criterion [8]. A well-constructed fitness function can increase the chance of finding a better solution by less number of iterations. In our case, it governs the update of the personal best value and global best value in each iteration. From [10] we can observe that ROC-AUC score exhibits some preferable properties than the typical accuracy score. It gives the correct indication of classification result as it is scale-invariant. ROC-AUC score is also decision threshold independent. F1 score is the weighted average of precision and recall. It takes both false positive and false negative into account for evaluation. For these aforementioned reasons, we have used both the AUC scores and F1 score as fitness function alternatively. We have used each metric individually as fitness function.

*ROC-AUC score*: ROC-AUC is the measurement of the model's classification performance. It indicates to which extent the model can classify the positive and negative classes accurately, thus it ranks them correctly [49]. ROC-AUC curve is given by Eq. 6:

$$ROC - AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{6}$$

where,  $T P_{rate}$  represents the true positive rate and  $F P_{rate}$  is the false positive rate. Equation 7 depicts the fitness function which uses ROC-AUC score:

$$fitness_1 = ROC - AUCScore(particle) \tag{7}$$

*F1 score*: Harmonic mean of precision and recall. It can be defined by:

$$F1\,score = \frac{2 \times recall \times precision}{recall + precision} \tag{8}$$

In Eq. 8 precision =  $\frac{TP}{TP+FP}$  and recall =  $\frac{TP}{TP+FN}$  where TP, FP and FN represent true positive, false positive and false negatives respectively [32,45]. Equation 9 represents the fitness function which uses F1 score as fitness measurement:

$$fitness_2 = F1Score(particle) \tag{9}$$

However, when dealing with highly skewed datasets, PR curves give a more informative picture of an algorithms performance. So, we have also calculated the PR-AUC score to evaluate the performance of the proposed method more precisely.

*PR-AUC score*: It is the score that combines precision and recall in single variable. The PR curve shows the trade-off between precision and recall for different threshold values [20,51]. We can compute PR-AUC [38] by parameterizing the PR curve by Eq. 10.

$$PR - AUC = \int_0^1 p(r)dr \tag{10}$$

where p and r denote precision and recall respectively. ROC-AUC looks at true positive and false positive cases while PR-AUC looks at positive predictive value and true positive score. Equation 11 depicts the fitness function which uses PR-AUC score:

$$fitness_3 = PR - AUCScore(particle)$$
(11)

## **Transfer function**

Since class imbalance is a binary optimization problem [65] so to convert the continuous optimization to a binary variant, a transfer function is used. We use '0' and '1' for non-selection and selection of a sample to prepare the final training dataset. To perform this action, the sigmoid function has been used in the present work. A sigmoid curve is an S-shaped curve (see Fig. 2) whose output range  $\in [0, 1]$ . This transfer function is defined as:

$$T(x) = \frac{1}{1 + e^{-x}}$$
(12)

Now, based on the output of the sigmoid function, we update the positions of the particle as:

$$X_{i,j} = \begin{cases} 1 & \text{if } T(X_{ij}) > 0.5\\ 0 & \text{if } T(X_{ij}) \le 0.5 \end{cases}$$
(13)

where,  $X_{ii}$  is the position of  $i^{th}$  particle in  $j^{th}$  dimension.

مدينة الملك عبدالعزيز KACST للعلوم والتقنية Springer



Fig. 2 Graphical representation of the sigmoid transfer function used in the present work

## **Experimental results and discussion**

This section presents the results of the proposed method. A set of experiments are conducted to evaluate the performance of the proposed method. Firstly, we have shown the effect of different population sizes on the performance, and then convergence curves are plotted to show the process of converging towards the near optimal solution. Secondly, results of the RTPSO are compared with the parent optimization techniques i.e., PSO and RTEA. The results are compared based on ROC-AUC and PR-AUC as they represent the best decision boundary between values of true positive (TP) and false positive (FP) [48]. Finally, we have compared performances with the state-of-the-art techniques and made statistical test to justify the worth of our work.

#### **Dataset description**

We have considered 15 real-world class imbalance datasets to evaluate the performance of our proposed method. The datasets are taken from three different popularly used repositories *namely*, namely, UCI<sup>1</sup>, Keel<sup>2</sup> and LIBSVM<sup>3</sup>. These datasets are selected from different domains like disease detection, predicting the cellular localization sites of proteins, and prediction of the age of abalone. We have included the datasets with low (e.g., like WDBC and heart datasets) to moderate (e.g., Hayes-roth and Page-blocks0 datasets) imbalance ratio as well as with very high imbalance ratio (e.g., Abalone19 and Yeast5 datasets) to establish the robustness of our model i.e., how well RTPSO behaves on the datasets having low to very high imbalance ratio. Hayesroth, New-thyroid, SPECTF and Heart are small datasets having the number of samples 160, 215, 267 and 270 respectively. There are also some large datasets like Page-blocks0, Abalone, Abalone19 and Segment0 having the number of samples 5472, 4177, 4174 and 2308 respectively. Most of the datasets are used for binary classification problems. For simplicity, we have divided these datasets into two groups, one with imbalance ratio  $\leq 6$  and the other with imbalance ratio > 6.

We have redefined the multi-class problems as binary class problems since class imbalance problem is mainly designed for binary classification problem [44]. For this, we have converted a certain combination of classes into minority class and the rest of the classes into the majority class following the similar convention as described in the work [29]. Related information of these two categories of datasets is provided in Tables 1 and 2.

For experimental need, we have initially divided each of the datasets into a training set and a test set having 80% and 20% of samples of the corresponding dataset respectively. The division is made by maintaining the original imbalance ratio in both sets. However, later during optimization of majority class samples using RTPSO, we have considered 20% of training samples as validation samples i.e., for calculating the fitness value of an optimization algorithm. Finally, the optimal sample set generated from RTPSO is used to train AdaBoost classifier and the model is evaluated on the test set samples.

The code is written in Python 3 and the graphs are plotted using matplotlib.

#### Parameter tuning

In meta-heuristic, the parameters play an important role in determining the end result. So, it is very important to find the proper parameter values. Since with the increase of population size and iteration number, the computational time also increases so we have performed some experiments to test the effect of population sizes on the performance and the rate of convergence concerning the number of iterations. Experiments are also performed to find out the proper values of  $c_1$ ,  $c_2$  and  $P_m$ .

During the experiment, we have varied one parameter and kept others constant. The effect of different population sizes on the performance of the model is plotted in Fig. 3 using ROC-AUC vs population size graph. To show the convergence of solutions we have plotted graphs for fitness values vs iteration number that are shown in Fig. 4. We have varied the value of  $c_1$  from 0 to 2 with step size 0.25. The obtained ROC-AUC scores are exhibited in Fig. 5.  $c_2$  is also varied from 0 to 2 with a step size of 0.25. The findings of the experiments are shown in Fig. 6. The value of  $P_m$  is varied



<sup>&</sup>lt;sup>1</sup> https://archive.ics.uci.edu/ml/index.php

<sup>&</sup>lt;sup>2</sup> http://sci2s.ugr.es/keel/datasets.php

<sup>&</sup>lt;sup>3</sup> https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Table 1Description of thedatasets having imbalance ratio $\leq 6$  used to evaluate theproposed method

Sl. No.	Dataset	Imbalance ratio	# samples	# classes	Classe(s) considered as minority class
01	Hayes-roth	4.16	160	3	3
02	SPECTF	3.85	267	2	0
03	New-thyroid	2.31	215	3	2,3
04	WDBC	1.68	569	2	М
05	BUPA	1.38	345	2	1
06	heart	1.25	270	2	2

Table 2 Description of the datasets having imbalance ratio > 6 used to evaluate the proposed method. In all the datasets class 1 is taken as a minority class

Sl. No.	Dataset	Imbalance ratio	# samples	#classes	Classe(s) considered as minority class
01	Abalone19	129.44	4174	2	1
02	Yeast5	32.73	1484	2	1
03	Kddcup-guess_passwd_vs_satan	29.98	1642	2	1
04	poker-9_vs_7	29.50	244	2	1
05	Abalone9-18	16.40	731	2	1
06	Led7digit-0-2-4-5-6-7-8-9_vs_1	10.97	443	2	1
07	Page-blocks0	8.79	5472	2	1
08	Abalone	6.35	4177	28	8
09	Segment0	6.02	2308	2	1

from 0 to 0.9 with step a size of 0.1. The ROC-AUC values obtained are exhibited in Fig. 7.

The graphs in Fig. 3 confirm that RTPSO attains peak accuracy with a population size around 20 while the graphs in Fig. 4 depict that RTPSO converges around  $50^{th}$  iteration in most of the cases. Based on this observation and keeping the computational time in mind, we have chosen population size as 20 and iteration number as 50 for further experiments. From Figs. 5 and 6 we can observe that  $c_1 = 2$  and  $c_2 = 2$  produce relatively better results. Similarly, from Fig. 7 we can conclude that  $P_m = 0.2$  produces relatively better results than other values. So for further experiments, we have set  $c_1 = 2$ ,  $c_2 = 2$  and  $P_m = 0.2$ .

## **Analysis of results**

This section reports the results obtained by the proposed RTPSO algorithm on the datasets mentioned in Tables 1 and 2. We run PSO, RTEA and RTPSO algorithms for 15 times on the present datasets and recorded the performance scores. Table 3 reveals the results of the RTPSO algorithm and comparison with the original PSO and RTEA algorithms in terms of ROC-AUC score. From Table 3, we can clearly observe that the RTPSO achieves the best ROC-AUC score for most of the datasets. For example, on SPECTF dataset using RTPSO, the best result has been obtained i.e., 0.8974 which is better



than the original (0.8351), PSO (0.8731) and RTEA (0.8583) results. On Hayes-roth and New-thyroid, all the methods have obtained 1 as ROC-AUC score. The proposed method is also performing well for Abalone dataset compared to the rests. On Segment0 dataset, only RTPSO has achieved 1 as ROC-AUC score. We have achieved 0.9917 score using PSO, 0.9957 by RTEA and 0.9987 by RTPSO while evaluating on WDBC dataset. Our proposed RTPSO technique has acquired 0.9451 as ROC-AUC score in the Heart dataset. Abalone19 is the most imbalanced dataset with an imbalance ratio 129.44 and on that also our proposed method obtains 0.9295 as ROC-AUC score, and outperforms the other methods. The result of Poker-9 vs 7 dataset is 1 by RTPSO while 0.8404 by original, 0.9574 by PSO and 0.9308 by RTEA. For Kddcup-guess\_passwd\_vs\_satan dataset, RTPSO and PSO have obtained 1 as ROC-AUC score. In case of Page-blocks0, we have obtained 0.9954 as ROC-AUC score using RTPSO, 0.9850 by original, 0.9895 by PSO and 0.9927 by RTEA.

The proposed method has obtained 1 as ROC-AUC score for 5 datasets (33.33% of all datasets): Hayes-roth, New-thyroid, Kddc-up-guess\_passwd\_vs\_satan and Poker-9\_vs\_7. However, in case of BUPA, Abalone9-18 and Led7digit-0-2-4-5-6-7-8-9\_vs\_1, RTPSO can not outperform PSO and RTEA in terms of ROC-AUC score. Out of 15 datasets, RTPSO achieves the highest score for 12 datasets (80% of all the datasets). Also, for 12 datasets (80%) the



Fig. 3 Graphs for achieved ROC-AUC scores using different population size for 15 datasets using PSO, RTEA and RTPSO





Fig. 4 Graphs showing the convergence of the solutions over numbers of iterations for 15 datasets using PSO, RTEA and RTPSO



Table 3 Comparison of RTPSO with PSO and RTEA in terms of ROC-AUC score. Best result achieved for each dataset is highlighted in bold font

SL No.	Dataset	ROC-AUC	score using	5			
		Original	Undersar	Undersampled data prepared using			
			PSO	RTEA	RTPSO		
01	Hayes-roth	1.0000	1.0000	1.0000	1.0000		
02	SPECTF	0.8351	0.8731	0.8583	0.8974		
03	New-thyroid	1.0000	1.0000	1.0000	1.0000		
04	WDBC	0.9903	0.9917	0.9957	0.9987		
05	BUPA	0.8189	0.8724	0.8474	0.8527		
06	Heart	0.9149	0.9308	0.8984	0.9451		
07	Abalone19	0.8753	0.9043	0.8675	0.9295		
08	Yeast5	0.9877	0.9897	0.9852	0.9957		
09	Kddcup-guess_passwd_vs_satan	0.9995	1.0000	0.9996	1.0000		
10	Poker-9_vs_7	0.8404	0.9574	0.9308	1.0000		
11	Abalone9-18	0.9032	0.9281	0.9383	0.9067		
12	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.9895	0.9910	0.9965	0.9952		
13	Page-blocks0	0.9850	0.9895	0.9927	0.9954		
14	Abalone	0.7679	0.7766	0.7757	0.8087		
15	Segment0	0.9947	0.9966	0.9998	1.0000		

Table 4 Comparison of RTPSO with PSO and RTEA in terms of average and standard deviation of ROC-AUC score. Best result achieved for each dataset is highlighted in bold font

Dataset	PSO		RTEA		RTPSO	
	Avg.	SD	Avg.	SD	Avg.	SD
Hayes-roth	0.9912	0.0182	0.9936	0.0245	0.9946	0.0015
SPECTF	0.7966	0.0653	0.7908	0.0799	0.8466	0.0632
New-thyroid	0.9670	0.0487	0.9717	0.0244	0.9670	0.0223
WDBC	0.9910	0.0062	0.9887	0.0052	0.9887	0.0072
BUPA	0.7358	0.0689	0.7585	0.0560	0.8485	0.0234
Heart	0.8576	0.0452	0.8531	0.0686	0.8791	0.0329
abalone19	0.7184	0.1032	0.7196	0.1261	0.8096	0.0281
Yeast5	0.9798	0.0170	0.9667	0.0286	0.9798	0.0096
kddcup-guess_passwd_vs_satan	0.9992	0.0014	0.9935	0.0124	0.9992	0.0209
poker-9_vs_7	0.6929	0.2001	0.6861	0.1928	0.8129	0.0620
Abalone9-18	0.7782	0.0996	0.8074	0.0595	0.7787	0.0985
led7digit-0-2-4-5-6-7-8-9_vs_1	0.9457	0.0425	0.9243	0.0534	0.9243	0.0271
Page-blocks0	0.9830	0.0058	0.9743	0.0107	0.9949	0.0056
Abalone	0.7554	0.0184	0.7462	0.0199	0.8052	0.0058
Segment0	0.9936	0.0004	0.9983	0.0025	0.9997	0.0005

proposed method has acquired greater than 0.9 ROC-AUC score. We have also shown the comparisons of RTPSO with PSO and RTEA in terms of average and standard deviation (SD) of ROC-AUC score in the Table 4. From the table, we can clearly observe that RTPSO is achieving the best results most of the time. Similarly, Tables 5 and 6 represent the comparisons of results in terms of F1 score. From Table 5 RTPSO has acquired 1 as F1 score in Hayes-roth, Kddcupguess\_passwd\_vs\_satan and Poker-9\_vs\_7 datasets. We have also achieved the best F1 score in most of the datasets. In Table 6 also the proposed technique is executing well in terms of average and SD.

From the Table 7 we can observe that, the proposed method is performing really well in terms of PR-AUC score as compared to PSO and RTEA. RTPSO has acquired 1 as PR-AUC score in Hayes-roth, Kddcup-guess\_passwd\_vs\_satan datasets. RTPSO has achieved the highest PR-AUC score except for New-thyroid, BUPA, Yeast5 and Page-blocks0 datasets. We have also compared the results of RTPSO with PSO and RTEA in terms of average and standard deviation of PR-AUC score in Table 8. Here also the proposed method



Table 5 Comparison of RTPSO with PSO and RTEA in terms of F1 score. Best result achieved for each dataset is highlighted in bold font

SL No.	Dataset	F1 score using						
		Original	Undersar	npled data pr	epared using			
			PSO	RTEA	RTPSO			
01	Hayes-roth	1.0000	1.0000	1.0000	1.0000			
02	SPECTF	0.7280	0.8480	0.7777	0.7658			
03	New-thyroid	0.9276	1.0000	0.94704	0.9730			
04	WDBC	0.9589	0.9804	0.9776	0.9809			
05	BUPA	0.7590	0.8230	0.7918	0.8441			
06	Heart	0.8541	0.8653	0.9000	0.8999			
07	Abalone19	0.5590	0.6781	0.6957	0.8490			
08	Yeast5	0.9250	1.0000	0.9427	0.9759			
09	Kddcup-guess_passwd_vs_satan	0.8867	0.9934	0.9989	1.0000			
10	Poker-9_vs_7	0.6771	0.8280	0.8269	1.0000			
11	Abalone9-18	0.6999	0.8224	0.8081	0.8576			
12	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.8038	0.9543	0.8374	0.9635			
13	Page-blocks0	0.8130	0.9355	0.8852	0.9265			
14	Abalone	0.5087	0.6974	0.6883	0.7556			
15	Segment0	0.9903	0.9936	0.9912	0.9956			

Table 6 Comparison of RTPSO with PSO and RTEA in terms of average and standard deviation of F1 score. Best result achieved for each dataset is highlighted in bold font

Dataset	PSO		RTEA		RTPSO	
	Avg	SD	Avg	SD	Avg	SD
Hayes-roth	0.9957	0.0022	0.9918	0.01632	0.9985	0.0011
SPECTF	0.6840	0.0841	0.6523	0.0734	0.7566	0.0027
New-thyroid	0.8722	0.0601	0.8947	0.0503	0.9270	0.0348
WDBC	0.9630	0.0117	0.9486	0.0155	0.9617	0.0212
BUPA	0.6722	0.0674	0.7014	0.0489	0.7985	0.0302
Heart	0.7712	0.0582	0.8305	0.0198	0.8291	0.0568
abalone19	0.6595	0.0430	0.6695	0.0397	0.7996	0.0286
Yeast5	0.8302	0.0635	0.7206	0.1090	0.8998	0.0464
kddcup-guess_passwd_vs_satan	0.9899	0.0193	0.9395	0.0856	0.9910	0.0259
poker-9_vs_7	0.6525	0.1609	0.6031	0.1103	0.8558	0.0890
Abalone9-18	0.7512	0.0894	0.7945	0.0190	0.8182	0.0264
led7digit-0-2-4-5-6-7-8-9_vs_1	0.8306	0.0689	0.7228	0.0758	0.8643	0.0757
Page-blocks0	0.9133	0.0149	0.8658	0.0131	0.9012	0.0111
Abalone	0.6887	0.0295	0.6469	0.0253	0.6892	0.0147
Segment0	0.9844	0.0105	0.9778	0.0136	0.9901	0.0050

is achieving the best results in most of the cases. Although, in general, it is quite difficult for any particular algorithm to handle the datasets with low to extremely high imbalance ratio, but the results in Tables 3, 5 and 7 confirm that the current RTPSO performs really well for all these said datasets. Hence we can safely comment that RTPSO is more effective than the individual algorithms (i.e., PSO and RETA) to solve the class imbalance problem.

Factorial analysis of variance (ANOVA) test [19] is performed as statistical test to ensure that the obtained results are statistically significant. The null hypothesis is that the two sets of results have same group means. A factorial ANOVA works with more than one independent variable [21,24]. It has two or more independent variables that split the samples in four or more groups. The simplest case of a factorial ANOVA uses two binary variables as independent variables, thus creating four groups within the samples. If the obtained p-values are < 0.05, then we can conclude that there are significant differences among the treatments at 5% significance level. Now, from Table 9, we can see that the obtained p-values produced by factorial ANOVA test considering ROC-AUC,



Table 7 Comparison of RTPSO with PSO and RTEA in terms of PR-AUC score. Best result achieved for each dataset is highlighted in bold font

SL No.	Dataset	PR-AUC score using					
		Original	Undersampled data prepared using				
			PSO	RTEA	RTPSO		
01	Hayes-roth	1.0000	1.0000	1.0000	1.0000		
02	SPECTF	0.7632	0.8270	0.8088	0.8976		
03	New-thyroid	0.9538	1.0000	0.9347	0.9731		
04	WDBC	0.9811	0.9931	0.9851	0.9974		
05	BUPA	0.8146	0.8709	0.8957	0.8944		
06	Heart	0.8772	0.8946	0.9234	0.9761		
07	Abalone19	0.5800	0.7835	0.7595	0.9035		
08	Yeast5	0.9300	1.0000	0.8900	0.9750		
09	Kddcup-guess_passwd_vs_satan	0.8600	0.9989	0.9978	1.0000		
10	Poker-9_vs_7	0.6600	0.8602	0.8200	1.0000		
11	Abalone9-18	0.7100	0.8518	0.8642	0.9054		
12	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.9200	0.9375	0.8506	0.9575		
13	Page-blocks0	0.7800	0.9530	0.8729	0.9046		
14	Abalone	0.7696	0.8527	0.8415	0.9554		
15	Segment0	0.9900	0.9935	0.9852	0.9935		

Table 8 Comparison of RTPSO with PSO and RTEA in terms of average and standard deviation of PR-AUC score. Best result achieved for each dataset is highlighted in bold font

Dataset	PSO		RTEA		RTPSO	RTPSO		
	Avg	SD	Avg	SD	Avg	SD		
Hayes-roth	0.9978	0.0032	0.9942	0.0076	0.9993	0.0014		
SPECTF	0.7902	0.0120	0.7576	0.0481	0.8563	0.0010		
New-thyroid	0.8652	0.0539	0.8773	0.0550	0.9259	0.0109		
WDBC	0.9879	0.0010	0.9750	0.0071	0.9846	0.0098		
BUPA	0.8258	0.0446	0.8740	0.0037	0.8485	0.0130		
Heart	0.8172	0.0433	0.8326	0.0444	0.8991	0.0085		
abalone19	0.7592	0.0230	0.7339	0.0297	0.8682	0.0086		
Yeast5	0.9502	0.0017	0.8444	0.0147	0.9249	0.0058		
kddcup-guess_passwd_vs_satan	0.9821	0.0339	0.9108	0.1068	0.9891	0.0026		
poker-9_vs_7	0.7936	0.0121	0.7591	0.0317	0.9545	0.0085		
Abalone9-18	0.8038	0.0310	0.7594	0.0176	0.8582	0.0098		
led7digit-0-2-4-5-6-7-8-9_vs_1	0.8579	0.0519	0.7871	0.0076	0.9095	0.0237		
Page-blocks0	0.8631	0.0025	0.8483	0.0159	0.8512	0.0129		
Abalone	0.8052	0.0142	0.8299	0.0119	0.9046	0.0011		
Segment0	0.9769	0.0169	0.9641	0.0211	0.9905	0.0028		

F1 and PR-AUC score separately confirms that the analysis is significant and hence we reject the null hypothesis.

## Comparison with state-of-the-art methods

To validate the effectiveness of our proposed method, in this section we have compared our results with different methods applied on these same datasets according to ROC-AUC score and F1 score (as discussed in the section "Fitness function") which are taken from literature.

For the datasets with imbalance ratio  $\leq 6$ , using F1 score we have compared the results with the other standard methods which are frequently used in class imbalance problem in Table 10. The methods include RU [47], ENN [70], NMU [7], CNN [55], prototype generation using K-means clustering (PK) [57], SMOTE [14], Imbalanced Metric Learning (IML) [29], which follows Mahalanobis metric learning algorithm [63], RTHS [1] and A $\beta$ BSF [31]. Some of these methods are very popular and useful to deal with the class imbalance problem. RTHS and A $\beta$ BSF are recently used evolutionary algorithms. From the Table 10, it is clear that for all the



 Table 9
 p-values produced by the ANOVA test considering ROC-AUC, F1 score and PR-AUC score using RTPSO, PSO and RTEA

Dataset	PSO			RTEA				
	ROC-AUC	F1	PR-AUC	ROC-AUC	F1	PR-AUC		
Hayes-roth	0.0137	0.0002	0.0002	0.0316	0.0004	0.0003		
SPECTF	0.0217	0.0001	0.0001	0.1557	0.4523	0.0001		
New-thyroid	0.0008	0.0001	0.0001	0.3657	0.0013	0.0002		
WDBC	0.0002	0.0001	0.0001	0.3597	0.5216	0.0001		
BUPA	0.0001	0.0001	0.0001	0.0229	0.0001	0.0001		
heart	0.0001	0.0001	0.0001	0.0001	0.0002	0.0001		
Abalone19	0.0001	0.0001	0.0001	0.0727	0.0006	0.0001		
Yeast5	0.0455	0.0001	0.0001	0.7630	0.9743	0.0001		
Kddcup-guess_passwd_vs_satan	0.0290	0.0001	0.0001	0.1667	0.0001	0.0004		
poker-9_vs_7	0.0174	0.0003	0.0001	0.5992	0.1768	0.0082		
Abalone9-18	0.0014	0.0001	0.0001	0.5109	0.0001	0.0002		
Led7digit-0-2-4-5-6-7-8-9_vs-1	0.0110	0.0002	0.0001	0.2626	0.0001	0.0001		
Page-blocks0	0.2840	0.0001	0.0002	0.2752	0.0001	0.0013		
Abalone	0.0031	0.0001	0.0002	0.9255	0.2942	0.0470		
Segment0	0.0005	0.0001	0.0001	0.0001	0.0001	0.0001		

**Table 10** Comparison of RTPSO with the state-of-the-art class imbalance problem-solving methods on the datasets having imbalance ratio  $\leq 6$ . The comparison is made based on F1 score. Best result achieved for each dataset is highlighted in bold font

Sl. No.	Dataset	RU [47]	ENN [70]	NMU [7]	CNN [55]	IML [29]	PK [57]	SMOTE [14]	RTHS[1]	$A\beta BSF[31]$	RTPSO
01	Abalone	0.51	0.46	0.27	0.46	$0.23\pm0.01$	0.53	0.46	0.59	0.60	0.75
02	Hayes-roth	0.70	0.78	0.50	0.64	$0.54\pm0.09$	0.70	0.68	0.91	0.93	1.00
03	SPECTF	0.49	0.46	0.44	0.44	$0.42\pm0.08$	0.42	0.54	0.55	0.57	0.76
04	New-thyroid	0.90	0.73	0.33	0.24	$0.91\pm0.02$	0.79	0.73	0.92	0.95	0.97
05	WDBC	0.91	0.92	0.87	0.83	$0.95\pm0.01$	0.92	0.94	0.95	0.97	0.98
06	BUPA	0.44	0.30	0.41	0.30	$0.52\pm0.05$	0.49	0.70	0.57	0.63	0.84
07	Heart	0.81	0.74	0.83	0.65	$0.77\pm0.03$	0.84	0.83	0.82	0.80	0.89

**Table 11** Comparison of RTPSO with state-of-the-art class imbalance problem-solving methods on the datasets having imbalance ratio > 6. Thecomparison is made based on ROC-AUC score. Best result achieved for each dataset is highlighted in bold font

Sl. No.	Dataset	RU [47]	ENN [70]	NMU [7]	PK [57]	SB [48]	CB [48]	RB [48]	RTHS[1]	A $\beta$ BSF [31]	RTPSO
01	Abalone19	0.74	0.33	0.75	0.87	0.59	0.62	0.65	0.64	0.73	0.93
02	Yeast5	0.98	0.98	0.88	0.99	0.93	0.91	0.95	0.98	0.99	0.99
03	Kddcup-guess_passwd_vs_satan	1.00	1.00	1.00	1.00	0.8602	0.88	0.91	0.99	1.00	1.00
04	Poker-9_vs_7	0.67	0.36	0.40	0.65	0.69	0.96	0.73	0.98	0.99	1.00
05	Abalone9-18	0.79	0.20	0.79	0.86	0.70	0.74	0.80	0.70	0.84	0.90
06	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.97	0.97	0.53	0.96	0.94	0.95	0.88	0.97	0.98	0.99
07	Page-blocks0	0.92	0.94	0.77	0.94	0.85	0.90	0.93	0.94	0.97	0.99
08	Segment0	0.99	0.99	0.89	0.99	0.96	0.95	0.98	0.98	1.00	1.00



> other state	e of the art me	citodo considere	a nere						
	RU	ENN	NMU	CNN	IML	РК	SMOTE	RTHS	AβBSF
RTPSO	0.0149	0.0149	0.0149	0.0149	0.0149	0.0149	0.0149	0.0452	0.0364
PSO	0.0149	0.0149	0.0149	0.0149	0.0149	0.0149	0.0149	0.0345	0.0.082
RTEA	0.0149	0.0149	0.0149	0.0149	0.0149	0.0149	0.0149	0.0149	0.0870
PSO RTEA	0.0149 0.0149	0.0345 0.0149	0.0.0 0.08′						

 Table 12
 p-values produced by the ANOVA test for the F1 scores achieved by the RTPSO algorithm, PSO algorithm and RTEA as compared with 9 other state-of-the art methods considered here

 Table 13 p-values produced by the ANOVA test for the ROC-AUC scores achieved by the RTPSO algorithm, PSO algorithm and RTEA with 9 other state-of-the art methods considered here

	RU	ENN	NMU	РК	SB	СВ	RB	RTHS	AβBSF
RTPSO	0.0149	0.0149	0.0149	0.0149	0.0224	0.0224	0.0420	0.0234	0.0341
PSO	0.0341	0.0341	0.0341	0.0535	0.0149	0.0279	0.0149	0.1090	0.1240
RTEA	0.0279	0.0279	0.0279	0.2450	0.0250	0.0279	0.0149	0.0149	0.0149

datasets, our proposed method has achieved the best F1 score than state-of-the-art methods with which present method is compared. It has also obtained 1 as F1 score in Hayes-roth dataset.

For the datasets which have the imbalance ratio > 6, we have added some more standard methods compared to the previous case. We have compared the results in terms of AUC score with RU [47], ENN [70], NMU [7], PK [57], SMOTE-Boost (SB) [15], CUSBoost (CB) [48], RUSBoost (RB) [54], RTHS [1] and A $\beta$ BSF [31] methods. From the Table 11, it is obvious that our proposed method has acquired the highest ROC-AUC score in all of the datasets. we have obtained above 0.9 as ROC-AUC score for all the datasets. We have also achieved 1 as ROC-AUC score for 3 datasets: Kddcupguess\_passwd\_vs\_satan, Poker-9\_vs\_7 and Segment0.

A statistical test is performed to ensure that our obtained results are statistically significant. The goal is to determine whether there is enough evidence to "reject" a conjecture or hypothesis about the process. The conjecture is called the null hypothesis. For our case, the null hypothesis states that the two sets of results have the same distribution. So, to determine the statistical significance of RTPSO algorithm, ANOVA test has been performed. From the test results provided in Table 12 (in terms of F1 score) and Table 13 (in terms of ROC-AUC score), we can conclude that the results of the proposed RTPSO algorithm are found to be statistically significant.

## Conclusion

In this work, we have proposed a hybrid meta-heuristic method, called RTPSO, to deal with class imbalance problem. RTPSO is based on a popular swarm-intelligence based meta-heuristic algorithm PSO and a recently proposed metaheuristic algorithm RTEA. This hybrid method is proposed to overcome the demerits of PSO and RTEA. From the best of our knowledge, the proposed approach is totally original and it has been used for the first time to solve the class imbalance problem. As RTPSO is self-adaptable to different datasets, so it can be integrated with different classifiers and evaluation parameters easily for any class imbalance datasets. The proposed method has experimented on 15 standard real-life datasets having low to extreme class imbalanced ratio. It has been compared with its parent algorithms PSO and RTEA along with some standard sampling methods using AdaBoost classifier. From the Tables 3, 5, 7, 10 and 11 we can clearly observe that RTPSO achieves better results in most of the cases than the other methods. We have acquired the highest score in 12 datasets (80.00%) out of 15 datasets in Table 3. RTPSO has also obtained 1 as ROC-AUC score for 5 datasets (33.33%). These results verify the advantages and the excellent performance of the proposed approach, which helps us to conclude that it can be used for any class imbalance datasets. As a future scope of the work, we can add more powerful and advanced classifiers to our proposed algorithm to reach better solutions. Additionally, it can be used in more interesting and popular research problems. Finally, RTPSO can be successfully applied on high dimensional datasets also.

**Acknowledgements** We would like to thank the CMATER research laboratory of the Computer Science and Engineering Department, Jadavpur University, India for providing us the infrastructural support.

#### Declarations

Conflict of interest Authors declare that there is no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material



is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copy-right holder. To view a copy of this licence, visit http://creativecomm ons.org/licenses/by/4.0/.

## Appendix

See Figs. 5, 6 and 7



Fig. 5 Graphs for achieved ROC-AUC scores using different values of c1 for 15 datasets using RTPSO

مدينة الملك عبدالعزيز KACST للعلوم والنقنية KACST



Fig. 6 Graphs for achieved ROC-AUC scores using different values of c2 for 15 datasets using RTPSO





Fig. 7 Graphs for achieved ROC-AUC scores using different values of  $P_m$  for 15 datasets using RTPSO



## References

- Ahmed S, Ghosh KK, Singh PK, Geem ZW, Sarkar R (2020) Hybrid of harmony search algorithm and ring theory-based evolutionary algorithm for feature selection. IEEE Access 8:102629– 102645. https://doi.org/10.1109/access.2020.2999093
- Amin A, Anwar S, Adnan A, Nawaz M, Howard N, Qadir J, Hawalah A, Hussain A (2016) Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study. IEEE Access 4:7940–7957. https://doi.org/10.1109/ access.2016.2619719
- Anand A, Pugalenthi G, Fogel GB, Suganthan PN (2010) An approach for classification of highly imbalanced data using weighting and undersampling. Amino Acids 39(5):1385–1391. https:// doi.org/10.1007/s00726-010-0595-2
- 4. Ashlock D (2006) Evolutionary computation for modeling and optimization. Springer Science & Business Media, New York
- Aydogan EK, Ozmen M, Delice Y (2018) CBR-PSO: cost-based rough particle swarm optimization approach for high-dimensional imbalanced problems. Neural Comput Appl 31(10):6345–6363. https://doi.org/10.1007/s00521-018-3469-2
- Bai Q (2010) Analysis of particle swarm optimization algorithm. Comput Inf Sci 3(1):180
- Bao L, Juan C, Li J, Zhang Y (2016) Boosted near-miss undersampling on SVM ensembles for concept detection in large-scale imbalanced datasets. Neurocomputing 172:198–206. https://doi. org/10.1016/j.neucom.2014.05.096
- Baresel A, Sthamer H, Schmidt M (2002) Fitness function design to improve evolutionary structural testing. In: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, GECCO'02, p 1329–1336
- Blagus R, Lusa L (2013) SMOTE for high-dimensional classimbalanced data. BMC Bioinform 14(1), https://doi.org/10.1186/ 1471-2105-14-106
- Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145–1159. https://doi.org/10.1016/s0031-3203(96)00142-2
- Bunkhumpornpat C, Sinapiromsaran K (2016) DBMUTE: densitybased majority under-sampling technique. Knowl Inf Syst 50(3):827–850. https://doi.org/10.1007/s10115-016-0957-5
- Cao P, Zhao D, Zaïane OR (2013) A PSO-based cost-sensitive neural network for imbalanced data classification. In: Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp 452–463, https://doi.org/10.1007/978-3-642-40319-4\_39
- Chang F, Ma L, Qiao Y (2005) Target tracking under occlusion by combining integral-intensity-matching with multi-block-voting. In: Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp 77–86, https://doi.org/10.1007/11538059\_9
- Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: Synthetic minority over-sampling technique. J Artif Intell Res 16:321–357. https://doi.org/10.1613/jair.953
- Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) SMOTE-Boost: Improving prediction of the minority class in boosting. In: Knowledge Discovery in Databases: PKDD 2003, Springer Berlin Heidelberg, pp 107–119, https://doi.org/10.1007/978-3-540-39804-2\_12
- Chawla NV, Japkowicz N, Kotcz A (2004) Editorial. ACM SIGKDD Explorations Newsletter 6(1):1–6. https://doi.org/10. 1145/1007730.1007733
- 17. Cheng R, He C, Jin Y, Yao X (2018a) Model-based evolutionary algorithms: a short survey. Complex Intell Syst 4(4):283–292. https://doi.org/10.1007/s40747-018-0080-1

- Cheng S, Lu H, Lei X, Shi Y (2018b) A quarter century of particle swarm optimization. Complex Intell Syst 4(3):227–239. https:// doi.org/10.1007/s40747-018-0071-2
- Crump M, Navarro D, Suzuki J (2019) Answering questions with data (textbook): Introductory statistics for psychology students https://doi.org/10.17605/OSF.IO/JZE52
- Davis J, Goadrich M (2006) The relationship between precisionrecall and ROC curves. In: Proceedings of the 23rd international conference on Machine learning - ICML '06, ACM Press, https:// doi.org/10.1145/1143844.1143874
- Dinno A (2015) Nonparametric pairwise multiple comparisons in independent groups using dunn's test. J Promot Commun Stat Stata 15(1):292–300. https://doi.org/10.1177/1536867x1501500117
- Dorigo M, Caro GD (1999) Ant colony optimization: a new metaheuristic. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), IEEE, https://doi.org/ 10.1109/cec.1999.782657
- Eberhart RC, Shi Y (1998) Comparison between genetic algorithms and particle swarm optimization. In: Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp 611–616, https://doi.org/ 10.1007/bfb0040812
- Embretson SE (1996) Item response theory models and spurious interaction effects in factorial ANOVA designs. Appl Psychol Meas 20(3):201–212. https://doi.org/10.1177/014662169602000302
- Fourie P, Groenwold A (2002) The particle swarm optimization algorithm in size and shape optimization. Struct Multidiscip Optim 23(4):259–267. https://doi.org/10.1007/s00158-002-0188-0
- 26. Fraleigh JB (2003) A first course in abstract algebra. Pearson Education India
- 27. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2012) A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. IEEE Trans Syst Man Cybern Part C (Applications and Reviews) 42(4):463–484. https://doi.org/10.1109/tsmcc.2011.2161285
- Gao M, Hong X, Chen S, Harris CJ (2011) A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems. Neurocomputing 74(17):3456–3466. https://doi.org/10. 1016/j.neucom.2011.06.010
- Gautheron L, Habrard A, Morvant E, Sebban M (2020) Metric learning from imbalanced data with generalization guarantees. Pattern Recogn Lett 133:298–304. https://doi.org/10.1016/j.patrec. 2020.03.008
- Geem ZW, Kim JH, Loganathan G (2001) A new heuristic optimization algorithm: Harmony search. SIMULATION 76(2):60–68. https://doi.org/10.1177/003754970107600201
- Ghosh KK, Ahmed S, Singh PK, Geem ZW, Sarkar R (2020) Improved binary sailfish optimizer based on adaptive β-hill climbing for feature selection. IEEE Access 8:83548–83560. https://doi. org/10.1109/access.2020.2991543
- Goutte C, Gaussier E (2005) A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp 345– 359, https://doi.org/10.1007/978-3-540-31865-1\_25
- Hassan AKI, Abraham A (2015) Modeling insurance fraud detection using imbalanced data classification. In: Advances in Intelligent Systems and Computing, Springer International Publishing, pp 117–127, https://doi.org/10.1007/978-3-319-27400-3\_11
- He Y, Wang X, Gao S (2019) Ring theory-based evolutionary algorithm and its application to d{0-1} KP. Appl Soft Comput 77:714–722. https://doi.org/10.1016/j.asoc.2019.01.049
- Hu W, Hu W, Maybank S (2008) AdaBoost-based algorithm for network intrusion detection. IEEE Trans Syst Man Cybern Part B (Cybernetics) 38(2):577–583. https://doi.org/10.1109/tsmcb. 2007.914695



- Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study1. Intell Data Anal 6(5):429–449. https://doi.org/ 10.3233/IDA-2002-6504
- Jong KD (1990) GENETIC-ALGORITHM-BASED LEARNING. In: Machine Learning, Elsevier, pp 611–638, https://doi.org/10. 1016/b978-0-08-051055-2.50030-4
- Keilwagen J, Grosse I, Grau J (2014) Area under precision-recall curves for weighted and unweighted data. PLoS One 9(3):e92209. https://doi.org/10.1371/journal.pone.0092209
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN95 - International Conference on Neural Networks, IEEE, https://doi.org/10.1109/icnn.1995.488968
- Li H, Yang X, Li Y, Hao LY, Zhang TL (2020) Evolutionary extreme learning machine with sparse cost matrix for imbalanced learning. ISA Trans 100:198–209. https://doi.org/10.1016/j.isatra.2019.11. 020
- Li Y, Sun G, Zhu Y (2010) Data imbalance problem in text classification. In, (2010) Third International Symposium on Information Processing. IEEE. https://doi.org/10.1109/isip.2010.47
- Liu S, Wang Y, Zhang J, Chen C, Xiang Y (2017) Addressing the class imbalance problem in twitter spam detection using ensemble learning. Comput Secur 69:35–49. https://doi.org/10.1016/j.cose. 2016.12.004
- Liu YQ, Wang C, Zhang L (2009) Decision tree based predictive models for breast cancer survivability on imbalanced data. In: 2009 3rd International Conference on Bioinformatics and Biomedical Engineering, IEEE, https://doi.org/10.1109/icbbe.2009.5162571
- 44. López V, Fernández A, García S, Palade V, Herrera F (2013) An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf Sci 250:113–141. https://doi.org/10.1016/j.ins.2013.07.007
- 45. Malakar S, Sarkar R, Basu S, Kundu M, Nasipuri M (2020) An image database of handwritten bangla words with automatic benchmarking facilities for character segmentation algorithms. NEURAL COMPUTING & APPLICATIONS
- Marini F, Walczak B (2015) Particle swarm optimization (PSO). a tutorial. Chemom Intell Lab Syst 149:153–165. https://doi.org/10. 1016/j.chemolab.2015.08.020
- Prusa J, Khoshgoftaar TM, Dittman DJ, Napolitano A (2015) Using random undersampling to alleviate class imbalance on tweet sentiment data. In: 2015 IEEE International Conference on Information Reuse and Integration, IEEE, https://doi.org/10.1109/iri.2015.39
- Rayhan F, Ahmed S, Mahbub A, Jani R, Shatabda S, Farid DM (2017) CUSBoost: Cluster-based under-sampling with boosting for imbalanced classification. In: 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), IEEE, https://doi.org/10.1109/csitss. 2017.8447534
- Rosset S (2004) Model selection via the AUC. In: Twenty-first international conference on Machine learning - ICML 04, ACM Press, https://doi.org/10.1145/1015330.1015400
- 50. Rotman JJ (2008) An introduction to homological algebra. Springer Science & Business Media
- Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS One 10(3):e0118432. https://doi. org/10.1371/journal.pone.0118432
- 52. Samma H, Lim CP, Ngah UK (2013) A hybrid PSO-FSVM model and its application to imbalanced classification of mammograms. In: Intelligent Information and Database Systems, Springer Berlin Heidelberg, pp 275–284, https://doi.org/10.1007/978-3-642-36546-1\_29

- Santiso S, Casillas A, Pérez A (2018) The class imbalance problem detecting adverse drug reactions in electronic health records. Health Inform J 25(4):1768–1778. https://doi.org/10. 1177/1460458218799470
- Seiffert C, Khoshgoftaar TM, Hulse JV, Napolitano A (2010) RUS-Boost: a hybrid approach to alleviating class imbalance. IEEE Trans Syst Man Cybern Part A Syst Hum 40(1):185–197. https://doi.org/ 10.1109/tsmca.2009.2029559
- Shekarforoush S, Green R, Dyer R (2017) Classifying commit messages: A case study in resampling techniques. In: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, https://doi. org/10.1109/ijcnn.2017.7965999
- 56. Talbi EG (2009) Metaheuristics: from design to implementation, vol 74. John Wiley & Sons, Hoboken
- 57. Thomas JCR (2011) A new clustering algorithm based on kmeans using a line segment as prototype. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Springer Berlin Heidelberg, pp 638–645, https://doi.org/10.1007/ 978-3-642-25085-9\_76
- Trelea IC (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett 85(6):317–325. https://doi.org/10.1016/s0020-0190(02)00447-7
- Ünal AN, Kayakutlu G (2020) Multi-objective particle swarm optimization with random immigrants. Complex Intell Syst. https://doi. org/10.1007/s40747-020-00159-y
- Vu L, Bui CT, Nguyen QU (2017) A deep learning based method for handling imbalanced problem in network traffic classification. In: Proceedings of the Eighth International Symposium on Information and Communication Technology - SoICT 2017, ACM Press, https://doi.org/10.1145/3155133.3155175
- Wang H, Liang M, Sun C, Zhang G, Xie L (2020a) Multiplestrategy learning particle swarm optimization for large-scale optimization problems. Complex Intell Syst. https://doi.org/10.1007/ s40747-020-00148-1
- Wang Q, Zhou Y, Zhang W, Tang Z, Chen X (2020b) Adaptive sampling using self-paced learning for imbalanced cancer data prediagnosis. Expert Syst Appl 152:113334. https://doi.org/10.1016/ j.eswa.2020.113334
- Weinberger KQ, Blitzer J, Saul LK (2006) Distance metric learning for large margin nearest neighbor classification. In: Advances in neural information processing systems, pp 1473–1480
- Wolpert D, Macready W (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82. https://doi.org/10. 1109/4235.585893
- Yang P, Xu L, Zhou BB, Zhang Z, Zomaya AY (2009) A particle swarm based hybrid system for imbalanced medical data sampling. BMC Genomics 10(Suppl 3):S34. https://doi.org/10.1186/1471-2164-10-s3-s34
- 66. Yu H, Ni J, Zhao J (2013) ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. Neurocomputing 101:309–318. https://doi.org/10.1016/j.neucom.2012.08.018
- Yu J, Li Y, Pei Y, Takagi H (2019) Accelerating evolutionary computation using a convergence point estimated by weighted moving vectors. Complex Intell Syst 6(1):55–65. https://doi.org/10.1007/ s40747-019-0111-6
- Zhang J, Chen L (2019) Clustering-based undersampling with random over sampling examples and support vector machine for imbalanced classification of breast cancer diagnosis. Comput Assist Surg 24(sup2):62–72. https://doi.org/10.1080/24699322. 2019.1649074



- Zhou ZH, Liu XY (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans Knowl Data Eng 18(1):63–77. https://doi.org/10.1109/tkde.2006. 17
- Zhu Y, Jia C, Li F, Song J (2020) Inspector: a lysine succinylation predictor based on edited nearest-neighbor undersampling and adaptive synthetic oversampling. Anal Biochem 593:113592. https://doi.org/10.1016/j.ab.2020.113592

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.