IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 June 2024, accepted 17 June 2024, date of publication 21 June 2024, date of current version 28 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3417822

RESEARCH ARTICLE

Handling the Class Imbalance Problem With an Improved Sine Cosine Algorithm for Optimal Instance Selection

RAJALAKSHMI SHENBAGA MOORTHY[1], ARIKUMAR K. SELVARAJ[2], SAHAYA BENI PRATHIBA[3], (Member, IEEE), GOKUL YENDURI[4], SACHI NANDAN MOHANTY[4], (Senior Member, IEEE), AND JANJHYAM VENKATA NAGA RAMESH[5]

[1]Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research, Chennai 600116, India
[2]Department of Data Science and Business Systems, College of Engineering and Technology, SRM Institute of Science Technology (SRMIST), Kattankulathur 603203, India
[3]Centre for Cyber Physical Systems, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
[4]School of Computer Science and Engineering (SCOPE), VIT-AP University, Amaravati, Andhra Pradesh 522237, India
[5]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Vijayawada, Andhra Pradesh 522302, India

Corresponding author: Gokul Yenduri (yenduri.gokul@gmail.com)

This work was supported by the VIT-AP University, Amaravathi, India.

ABSTRACT Class imbalance is a significant study problem that is biased, exhibiting excellent performance toward the majority classes in the dataset while showing inferior performance toward minority classes. When dealing with real-world issues, this kind of biased nature affects classification accuracy. The Improved Binary Sine Cosine Algorithm (IBSCA) has been used in this work to identify a subset of the majority class in the best possible way. The proposed IBSCA makes some enhancements over the conventional Binary Sine Cosine Algorithm (BSCA) to address the issue of premature convergence with local optimal solutions. To improve classification accuracy for unbalanced datasets, the proposed IBSCA seeks to identify the optimal collection of instances from the majority class. The advised IBSCA makes use of the alpha agent, beta agent, and random agent's location, which tends to devote considerable time to exploration to find the best possible set of instances. By using the geometric mean (G-mean) and F-score to describe the fitness function, the proposed IBSCA aims to solve the multi-objective optimization issue. On 18 datasets with different imbalance ratios taken from the KEEL repository, experimentation is conducted. Comparisons are made between the suggested IBSCA and the traditional Binary Sine Cosine Algorithm, Binary Particle Swarm Optimization (BPSO), and Binary Grey Wolf Optimization (BGWO). Additionally, the performance of the suggested IBSCA is evaluated against the top outcomes from different research papers. Metrics like sensitivity, F-score, G-mean, and area under curve (AUC) show that the suggested IBSCA outperforms the state-of-the-art algorithms. The statistical findings using the Wilcoxon signed rank test and Friedman test also demonstrate that the suggested IBSCA is more efficient than the other conventional algorithms.

INDEX TERMS Class imbalance, instance selection, K-nearest neighbor, metaheuristic algorithm, sine cosine algorithm.

I. INTRODUCTION

The Class Imbalance problem is just one of the issues that arise when working with various datasets as a result of the rapid growth of machine learning techniques and the demand for them from small businesses to major healthcare organizations [1], [2]. Class imbalance is a concern when there are more training cases in one class and few instances in the other class [3]. The distribution of majority and minority instances is quantified by the Imbalance Ratio (IR), which is

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

provided in Equation (1).

$$IR = \frac{\text{Number of Instances in Majority Class}}{\text{Number of Instances in Minority Class}} \quad (1)$$

The datasets that are imbalanced and have an IR value between 1.5 and 3 are categorized as less imbalanced datasets. IR values between 3 and 9 are considered to be medium-sized imbalanced datasets. The datasets with a value of IR higher than 9 are considered to be highly unbalanced [4]. The most frequent inquiries that come up when working with datasets with class imbalance are:

Q1. Do datasets with actual class imbalances have an impact on the classifier's performance?

Q2. Are there any methods for coping with the issue of class imbalance?

These inquiries prompted the researchers to concentrate on class imbalance issues and to suggest a cutting-edge methodology for mitigating the harms that come along with those types of datasets. The impact of the K-NN classifier is used to discuss the response to the question: 1. The response to Query 2 uses a variety of resampling techniques, including

- Oversampling the instances in the minority class.
- Undersampling the instances in the majority class.

Typically, two categories of approaches–data-level approaches and classifier-level approaches–are used to handle class imbalance issues [5]. Approaches at the data level work with instances that are either oversampled or undersampled. Oversampling is most frequently achieved through random oversampling, in which samples from minority classes are arbitrarily selected for replication. Despite being a useful strategy, random selection has an overfitting issue [6], [7]. The Synthetic Minority Oversampling Method (SMOTE) uses K-NN to resolve the issue caused by random oversampling [8], [9]. Data clustering is another general term for oversampling the instances in the minority class while incorporating intra-cluster and inter-cluster distance into account [10], [11], [12]. The next data-level technique uses undersampling, which eliminates some instances from the majority class. The most popular method of undersampling is random sampling, which involves selecting some examples at random from the majority class and deleting them. Data clustering has also been used for undersampling, where instances from the majority class are grouped into clusters with the number of clusters equal to the number of instances from the minority class, and the nearest cluster is selected, with the remaining clusters being discarded [13]. To make the algorithm suitable for datasets with imbalances, classifier-level techniques modify the algorithm's process. Figure 1 depicts a variety of possible solutions to class imbalance issues.

Additionally, imbalanced datasets are quite prevalent in many real-world applications, including the identification of fraudulent calls [14], [15], healthcare datasets [14], [15], and network intrusion [14], [15], where they have an impact on the success of the classifier. Although there are many ways to deal with unbalanced datasets, overfitting is the most prevalent issue that current methodologies

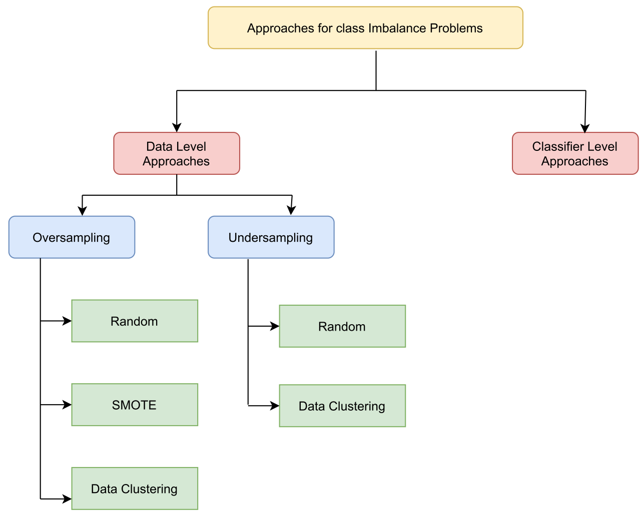

FIGURE 1. Various approaches to class imbalance problem.

encounter, which drops the classifier's accuracy. For NP-Hard challenges including resource allocation, path optimization, hyperparameter tuning, feature selection, etc., metaheuristic algorithms are the go-to answer. In this work, instances from the majority class are determined using metaheuristic algorithms [16], [17], [18]. This research used the Binary Sine Cosine Algorithm(BSCA) to address the issue of class imbalance. The premature convergence and gradual local convergence of BSCA leads to degrade the performance of the algorithm [19]. To overcome the above-mentioned drawback alpha agents, beta agents, and random agents were introduced in the proposed IBSCA. The idea behind the alpha agent, beta agent, and random agent is to promote exploration and exploitation to find global optimal solutions. Alpha agents and beta agents are introduced to promote exploration capability and variety in the search process. The agent with the highest fitness value is known as the alpha agent, and the agent with the second highest fitness value is known as the beta agent. The prime goal of the alpha agent and the beta agent is to improve the global search process and to prevent the agent from getting trapped in the local optimal solution. The random agents are randomly chosen from the population to promote diversification in the solution space. In other words, Random agents allow for a more diversified investigation of the search space, which could improve the algorithm's odds of discovering global optima. There is always space for improvement because no algorithm is effective for all real-world problems. Thus, the Improved Binary Sine Cosine Optimization algorithm (IBSCA) meta-heuristic algorithm is used to create a novel algorithm to address the class imbalance issue. The conventional BSCA is improved for convergence and stagnation by incorporating extra agents.

IBSCA utilizes SCA's vital notions, inspired by the behavior of sine and cosine functions, to balance exploration and exploitation in the multi-objective search space. In the multi-objective optimization problem, two or more objectives must be optimized simultaneously. These objectives can be either minimum or maximum. In the case of a class imbalance problem, the optimization focuses on maximizing G-mean and F-score. Furthermore, the suggested IBSCA seeks to work with a population of agents, each of which evaluates the objective function. IBSCA also incorporates alpha agents, beta agents, and random agents to help with intensification and diversity, preventing premature convergence. For dealing with the class imbalance problem, IBSCA distributes weight to each objective function, namely G-mean, and F-score. The suggested technique combines multiple objectives into a single value that includes the model's capacity to recognize positive cases (sensitivity) while limiting false positives (precision). This aggregated value is more likely to evaluate the model's performance in dealing with class imbalance than either metric alone. Finally, in each iteration, IBSCA tries to choose the solution that has a better solution than others in the search space. The ability of IBSCA to combine objectives, evaluate objectives, and maintain high-quality solutions makes it suitable and efficient for multi-objective optimization problems.

The primary objective and novelty of the research comprises the following:

- The creation of the IBSCA introduces improvements and alterations to the traditional BSCA. These enhancements try to solve the issue of premature convergence to local optima, which is a prevalent problem in optimization methods.
- The selection of a subset of instances from the majority class is crucial in class imbalance issues. Thus, the proposed IBSCA chooses the optimal subset of instances thereby maximizing F-score, G-mean
- The formulation of class imbalance problem as a multi-objective optimization problem with objectives including G-mean and F-score. The multi-objective formulation helps to explore exploit and produce promising solutions.
- The utilization of alpha agent, beta agent, and random agent promotes exploration and exploitation. The exploration prevents the algorithm from trapping in the local optimal solution.
- IBSCA is contrasted with other common metaheuristic algorithms, such as BSCA, BPSO, and BGWO.
- Additionally, IBSCA is contrasted with cutting-edge techniques like SMOTE and Borderline SMOTE (BL SMOTE).
- All algorithms are set up to execute 30 times, with the average result being used as a benchmark.
- Sensitivity, F-Score, Geometric mean (G-mean), and AUC metrics are compared to those of other existing approaches.
- Wilcoxon signed rank and Friedman test has been used to demonstrate the statistical efficacy of the suggested IBSCA.

Details about the issue of the class imbalance are provided in the introduction section. The remaining portions of the article are structured as follows: The various current methods for addressing the issue of class imbalance are described in the literature survey section. The suggested IBSCA is explained in the next section. The full experimental findings and analysis are presented in the experiment results section. The work is concluded with future scope.

II. LITERATURE SURVEY

The different methods for addressing the class imbalance issue in real-world datasets are described in this section. The authors started off the research by handling imbalanced datasets with the conventional method of instance selection before moving on to the most effective method of selecting the instances using metaheuristic algorithms.

A. STATE-OF-THE-ART

The traditional methods of handling the class imbalance are represented in Table 1. Convolutional Neural Networks(CNN) have been used to train the datasets that suffer from class imbalance [5]. The methods used to address class imbalance are random minority oversampling, random majority oversampling, Two-phase training with pre-training on a randomly oversampled dataset, Two-phase training with pre-training on a randomly undersampled dataset, thresholding with prior class probabilities, oversampling with thresholding, sampling with thresholding [5]. SMOTE, Adaptive Synthetic Sampling Approach (ADASYN), BL SMOTE, and Safe level SMOTE have been compared for various performance metrics on the accuracy, sensitivity, specificity, precision, F-measure, G-mean, and Area Under Curve (AUC) on various datasets for different classifiers such as Naïve Bayes, Support Vector Machine and Nearest Neighbor [20]. It was found that safe level SMOTE outperformed other methods in terms of F-measure and G-mean [21]. The class imbalance problem in intrusion detection is done using Siamese Neural Network. The KDD and NSL-KDD datasets had been taken in which Remote to Local (R2L) and user to Root (U2R) are less in number than denial of service and probe attack. Siamese-NN works based on a few-shot learning technique, which was initially used for signature verification. Siamese-NN obtained higher recall for R2L and U2R attacks than Deep Neural Networks (DNN) and CNN [22]. A Class Specific Extreme Learning Machine (CS-ELM) was used to handle class imbalance to overcome the drawbacks of weighted ELM. The algorithm has been evaluated on different datasets taken from the KEEL repository. The Wilcoxon signed-rank test and Friedman mean-rank test were performed to validate the strength of CS-ELM [23]. SMOTE which was functioning using K-Nearest Neighbors was evaluated mathematically using multivariate Gaussian and multivariate Laplacian distribution to find the distribution of instances that were replicated. SMOTE intends to create replicated instances for the minority class to make the size of the minority instances equal to that of the majority instances [24], [25]. The G-Mean was the standard way to measure the performance of the classifier when dealing with binary class imbalance problems. The G-Mean was

TABLE 1. Methods used for handling class imbalance.

Concept	Method	Pros	cons
Over sampling ([30]), ([31])	Random oversampling	Does not remove instances from minority class resulting no information loss. Risk of overfitting is reduced .	Increase in size of the data. Since it duplicates existing data, no new information is added thereby limiting the model's ability.
	SMOTE - generate synthetic instances from instances in minority class	Preserve information. Overfitting is reduced.	Sensitive to noise. Computationally complex.
	Borderline–SMOTE– generate synthetic instances near the decision boundary of minority class	Robust for unseen data. Reduce the chance of overfitting.	Sensitive to noise. Computationally complex.
Under sampling ([30]), ([32])	Random under sampling – reduce the number of instances from majority class	Dataset size is reduced. Reduce the risk of overfitting.	Leads to loss of information. Risk of removing some important instances.
	Tomek links – find pair of instances from different class that are close to each other with different class label	Preserving valuable information. Removes instances that are on the decision boundary.	Sensitive to noise.

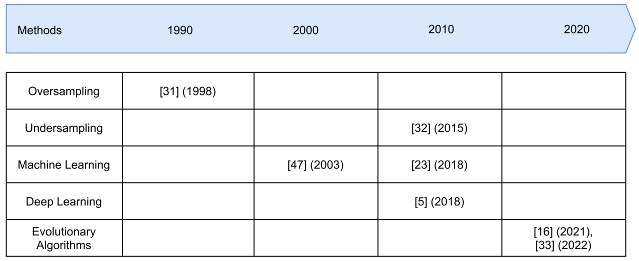

FIGURE 2. Evolution of methods for handling class imbalance.

measured using Random Undersampling (RUS), Ensemble of RUS (ERUS), Boosting of RUS (RUSBOOST), Evolutionary Undersampling (EUS), Particle Swarm Optimization(PSO), Tomek links (TL), OSS (One-sided selection), TL + CNN, Neighbourhood cleaning rule (NCL) on 66 datasets and it was found that G-mean was better when the datasets become balanced rather than on imbalanced dataset [26]. Disjuncts - Random Oversampling was used to generate synthetic samples to alleviate the problem of class imbalance that generates cone structures starting from minority class towards decision boundary of majority class [27]. Privacy-preserving Federated learning framework was designed to address the problem of class imbalance for wind turbine dataset [28]. Personalized Retrogress - resilient Federated learning framework for medical data using progressive Fourier aggregation [29].

B. METAHEURISTIC ALGORITHMS FOR INSTANCE SELECTION

A novel ensemble method had been designed to have the benefit of using ensemble learning with the noble under-sampling method. The undersampling was done through Binary PSO (BPSO) and it got the benefit of ensemble classifiers to find the right number of instances from the majority class [33]. The very most prevalent problem that the real-world dataset suffers from is class imbalance. The imbalanced dataset affects the performance of any classifier and the classifier taken into account was Multi-Layer Perceptron (MLP). The fitness function taken into account was G-Mean. The metaheuristic algorithm aimed to select the optimal set of instances thereby boosting the performance of MLP was Grey-Wolf Optimization (GWO), PSO, and Salp Swarm Algorithm (SSA) [34]. Simulated Annealing had been used for under-sampling the instances from the majority class. The classifiers such as support vector machine, decision tree, K-Nearest Neighbor, and discriminant analysis were evaluated on 51 real-world imbalanced datasets The designed simulated annealing-based undersampling is better than SMOTE in terms of G-Mean and F-Score [35]. The Large-Scale instance selection (LRIS) algorithm was designed to handle class imbalance problems. The idea behind LRIS was the length reduction strategy. The evolution operators such as crossover and mutation were used to generate a child population from the parent population. The probability that the instance is represented as the gene in the individual was based on the importance of the instance [36]. Memetic Algorithms (ME) with variable Neighborhood Search (VNS) were used to do both instance and feature selection. The designed ME with VNS tends to do balanced exploration and exploitation and worked well for noisy IoT Data [37]. Often Genetic Algorithm (GA) has been widely used for instance selection with the aid of solving class imbalance problems. The major drawbacks of GA-based instance selection are that it is computationally complex, and when the size of the dataset grows, performance is badly affected. This was addressed through Fuzzy clustering by dividing the dataset into regions, and instance selection based on GA was done in each cluster, and the final result was obtained through ensemble voting [38]. The sampling approach called Clustering Based Instance Selection (CBIS) was designed to overcome the problems of class-imbalanced datasets, which tend to affect the classifier in classifying the minority class from the majority class. The CBIS was evaluated on datasets taken from the KEEL repository. The performance of MLP together with bagging and boosting was better than other traditional algorithms [39]. The Fig.2 represents the evolution of various methods for handling class imbalance [40].

According to the study, different researchers created unique approaches to the class imbalance issue. Each approach offers a unique solution to the issue, though they all have some possible drawbacks. Although the algorithms for the conventional method of handling imbalance problems select instances from the majority class to balance with instances in the minority class, the algorithms suffer greatly from an issue of overfitting [41], [42]. When the imbalance ratio is high, the algorithms tend to hinder the classifier's performance [43], [44]. The current metaheuristic algorithms for choosing the best possible set of examples to handle class imbalance have the issue of stagnation, blocked in the local optimal solution, which leads to premature convergence [45]. According to the information gathered through conducting a literature review, only a small number of research studies

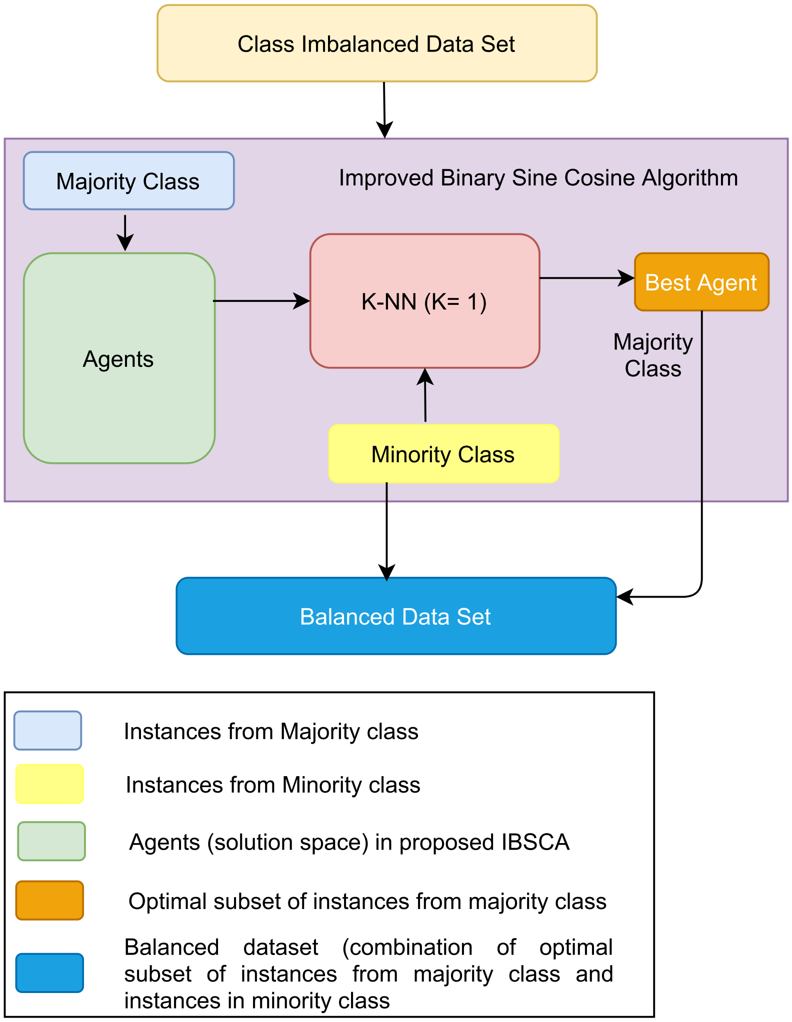

FIGURE 3. Work flow of proposed improved binary sine cosine algorithm.

used a metaheuristic method, for instance, selection. This is the first study that, as far as the author is aware, used IBSCA for instance selection. The beta agent and random agent have been added to IBSCA to address the class imbalance and the issues of overfitting and early convergence.

III. IMPROVED BINARY SINE COSINE ALGORITHM

The proposed IBSCA tends to solve the class imbalance issue in the best possible manner while maximizing accuracy. To stop the classifier from becoming biased, the IBSCA algorithm tends to undersample the instances in the majority class. The suggested IBSCA algorithm employs sine and cosine functions as a population-based optimization strategy. Most real-world data sets have a peculiar issue with class imbalance. The problem arises when instances in one class, such as the majority class M_c are higher than instances in another class, such as the minority class m_c. Such disparity seriously impairs the classifier's performance and raises the possibility that the decisions it makes will not be helpful.

The analysis of the literature shows how metaheuristic algorithms are frequently used to address the issue of class imbalance. IBSCA has been suggested in this article to produce an ideal subset of instances from the majority class by undersampling the instances of the majority class. The BSCA, which uses the sine and cosine mathematical functions to handle real-world problems, has difficulties with premature convergence and stagnation. In addition to the alpha agent α, the agent with the greatest fitness, the shortcomings of BSCA have been addressed in IBSCA by using two additional agents, the beta agent β, and the random agent $rand$. The issue of early convergence and stagnation is resolved by the inclusion of the Beta agent and random agent. Additionally, the discrete class imbalance issue is handled by converting the continuous search space of the IBSCA algorithm to a binary search space using a V-shaped transfer function [46]. Figure 3 depicts the operation of IBSCA.

A. INITIALIZATION OF POPULATION

The proposed IBSCA algorithm works on a collection of agents called population. P_{IBSCA} represents the number of agents in the population. The dimensions of the agent $A_i \in P_{IBSCA}$ depend on the instances in the majority class

I_1	I_2	I_3	I_4	I_5	I_6			I_{M-2}	I_{M-1}	I_M
1	0	1	0	0	0	......	...	1	1	0

FIGURE 4. **Representation of agent.**

M. Depending on whether or not the instance is chosen, the value of A_{ij} represents the j^{th} dimension of the i^{th} instance and can be either 1 or 0. If A_{ij} is set to 0, the instance is not chosen; otherwise, the instance is chosen. Figure 4 depicts the depiction of an agent.

B. ASSESSING THE AGENT USING THE FITNESS FUNCTION

The goal of the fitness function is to select the best solution from a collection of possible ones in state space. The goals of maximizing the G-Mean and F-score are used to evaluate each agent in the population. Each objective is assigned a weight, w_i, and the fitness function is the linear combination of the weight and objective function. The agents are evaluated using the fitness function. When evaluating the fitness function, the classifier K-NN with K=1 is used. If there are a lot of instances in the majority class, K-NN, a non-parametric classifier that identifies K-Nearest Neighbors of an instance, may be biased towards that class [47]. To optimize the performance of K-NN, various methodologies, including Fuzzy K-NN [48] and SCA K-NN [49], have been used for real-world problems. This paper aims to use the proposed IBSCA to solve the issue of class imbalance while maximizing the performance of K-NN. Finding the ideal subset of instances from the majority class is the goal of IBSCA to maximize the G-mean and F-score, which are displayed in Equation (2).

$$Fit_{A_i} \leftarrow w_1 * GMean\left(OS_{M_c}\right) + w_2 * Fscore\left(OS_{M_c}\right) \tag{2}$$

where $GMean\left(OS_{M_c}\right)$ represents the G-mean obtained for the optimal subset of the majority class and $Fscore\left(OS_{M_c}\right)$ represents the F-score for the optimal subset of the majority class. w_1 and w_2 assigns the weight to the G-mean and F-score of the classifier, where $w_1 \in 0, 1$ and $w_2 = 1 - w_1$ such that $w_1 + w_2 = 1$.

$GMean\left(OS_{M_c}\right)$ is described as the root of the product of the True Positive Rate (TPR) and the True Negative Rate (TNR), as shown Equation (3).

$$GMean \leftarrow \sqrt{TPR * TNR} \tag{3}$$

$Fscore\left(OS_{M_c}\right)$ represents the F-score calculated as the harmonic mean of precision and recall as shown in Equation (4).

$$Fscore\left(OS_{M_c}\right) \leftarrow \frac{2 * r * p}{r + p} \tag{4}$$

where r represents recall and p represents precision which are represented in Equation (5) and Equation (6), respectively.

$$r \leftarrow \frac{n_{tp}}{n_{tp} + n_{tn}} \tag{5}$$

$$p \leftarrow \frac{n_{tp}}{n_{tp} + n_{fp}} \tag{6}$$

where n_{tp} represents the number of instances that are correctly classified as positive, n_{fp} represents the number of positive instances that are classified as negative, and n_{fn} represents the number of negative instances that are classified as positive. Since the F-score takes into account the harmonic mean of recall and precision, it is regarded as one of the objective functions. When dealing with heavily biased datasets, precision and recall aim to measure an algorithm's effectiveness to a great extent. Additionally, false positives and false negatives have an impact on the F-score, which evaluates the particle and provides improved performance.

C. UPDATION OF AGENT

In every iteration, each agent A_i updates its location based on the alpha, beta, and random agent. The computation of the agent's position is shown in Equation (7).

$$A_{ij}^{t} \leftarrow \begin{cases} A_{ij}^{t-1} + r_1 * sinr_2 + \left|r_3 * A_{\alpha j}^{t-1} - A_{ij}^{t-1}\right| \\ + \left|(1 - r_3) * A_{\beta j}^{t-1} - A_{ij}^{t-1}\right| \quad if\ r_4 < 0.5 \\ A_{ij}^{t-1} + r_1 * cosr_2 + \left|r_3 * A_{\alpha j}^{t-1} - A_{ij}^{t-1}\right| + \\ \left|(1 - r_3) * A_{randj}^{t-1} - A_{ij}^{t-1}\right| \quad else \end{cases} \tag{7}$$

For paving exploration at the initial state and exploitation of the agent towards convergence, the variable r_1 tends to decline linearly over the generations. The computation of r_1 is shown in Equation (8). The value r_{max} is set as 0.9 and r_{min} is set as 0.1. Max_Iter represents the maximum number of generations and t represents the current iteration number. Exploration and exploitation are achieved using the random variable r_2 if the value of the sine and cos functions falls within the range $(-1,1)$ the agent intensifies; otherwise, the agent diversifies. The weight of the alpha agent is determined by the random variable r_3. The agents tend to move towards the alpha agent if the value of r_3 is higher than 1, otherwise, the agents may tend to move towards the beta agent or a random agent with random probability r_4. The value of the agent i at the j^{th} dimension during step t is represented by A_{ij}^{t}. The value of agent "alpha" at the j^{th} dimension during step $t-1$ is represented by $A_{\alpha j}^{t-1}$. The beta agent and random agent at the j^{th} dimension during iteration $t - 1$ are represented by the variables $A_{\beta j}^{t-1}$ and A_{randj}^{t-1}.

$$r_1 \leftarrow (r_{max} - r_{min}) * \frac{(Max\_Iter - t)}{Max\_Iter} + r_{min} \tag{8}$$

D. HANDLING DISCRETE CLASS IMBALANCE PROBLEM

The position of the agent that was calculated is a continuous measure. Transfer functions can be applied to the position of the agent in discrete class Imbalance issues where the agents must be expressed as either 0 or 1, indicating whether the instance is selected or not. To confine the agent in a discrete area, a transfer function is used. When the position of the

Algorithm 1 Improved Binary Sine Cosine Algorithm.

Input: Instances in Majority class $\{I_1, \ I_2, \ \ldots \ I_{|M_c|}\}$,
Instances in minority class $\{I_1, \ I_2, \ \ldots \ I_{|m_c|}\}$,
Number of generations: Max_Iter,
Number of agents: $|P_{IBSCA}|$
Output: Optimal subset of instances from the majority class M_c, OS_{M_c} where $|OS_{M_c}| < |M_c|$

/* Generation of Agents */
1: $P_{IBSCA} \leftarrow \{\}$
2: **for** each agent A_i where $i \leq |P_{IBSCA}|$ **do**
3: **for** each dimension j where $j \leq M_c$ **do**
4: $A_{ij} \leftarrow Rand\_Initialize(0, 1)$
5: **end for**
6: **end for**
7: $t = 1$
8: **while** $t \leq Max\_Iter$ **do**
9: **for** each agent $A_i \in P_{IBSCA}$ **do**
/*Computing Fitness */
10: Compute Fitness Fit_{A_i} using 2
11: **end for**
12: $A_{\alpha}^{t} \leftarrow First\_Max\_Agent\left\{Sort\left(A_i|Max\left(Fit_{A_i}\right)\right)\right\}$
13: $A_{\beta}^{t} \leftarrow Next\_Max\_Agent\left\{Sort\left(A_i|Max\left(Fit_{A_i}\right)\right)\right\}$
14: $A_{rand}^{t} \leftarrow Choose\_Random\_Agent\ (P_{IBSCA})$
15: Compute r_1 using 8
16: $r_2 \leftarrow rand()$
17: $r_3 \leftarrow rand()$
18: **for** each agent A_i where $i \leq |P_{IBSCA}|$ **do**
19: **for** each dimension j where $j \leq M_c$ **do**
20: $r_4 \leftarrow rand(0, 1)$
21: Compute position using 7
22: **end for**
23: **end for**
24: **for** each agent A_i where $i \leq |P_{IBSCA}|$ **do**
25: **for** each dimension j where $j \leq M_c$ **do**
26: Generate rand
27: Compute $V_sTF\left(A_{ij}^{t}\right)$ using 9
28: **if** $rand < V_sTF\left(A_{ij}^{t}\right)$ **then**
29: $A_{ij}^{t} \leftarrow 1$
30: **else**
31: $A_{ij}^{t} \leftarrow 0$
32: **end if**
33: **end for**
34: **end for**
35: **end while**
36: return $OS_{M_c} \leftarrow A_{\alpha}^{t}$

agent is large, the transfer function aims to accelerate the agent with the greatest probability. When the position of an agent is significant, it indicates that the agent is not the best. As a result, for the following iteration, the agent's value will be altered. However, when the position of the agent is small, the transfer function only slightly accelerates the agent. The V-shaped transfer function V_sTF is applied over the agent as represented in Equation (9).

$$V_sTF\left(A_{ij}^{t}\right) \leftarrow \left|tanh\left(A_{ij}^{t}\right)\right| \tag{9}$$

The i^{th} agent's corresponding dimension is set to 1, which denotes the selection of the instance if the V-shaped transfer function V_sTF of the i^{th} agent A_{ij}^{t} for the j^{th} dimension is larger than a random value. If not, the associated j^{th} dimension of the i^{th} agent is set to 0, which symbolizes the deselection of the instance. The choosing and deselection of instances are displayed in Equation (10).

$$A_{ij}^{t} \leftarrow \begin{cases} 1, & rand < V_sTF(A_{ij}^{t}) \\ 0, & else \end{cases} \tag{10}$$

The working of the proposed IBSCA is shown in Algorithm 1. The working of Algorithm 1 is given as follows: Lines 1 to 6 represent the random initialization of the position of each agent as 0 or 1. Line 7 represents the initialization of the iteration variable. Lines 8 to 35 will be repeated for the maximum number of iterations. Lines 9 to 11 represent the computation of the fitness function using K-NN for each agent. Line 12 represents finding the alpha agent which is the agent with maximum fitness. Line 13 represents finding the beta agent which is the agent with the next maximum fitness. Line 14 represents finding the random agent. Line 15 represents the computation of variable r_1. Line 16 and Line 17 represent the computation of random variables r_2 and r_3. Line 18 to Line 19 represents the computation of values for each dimension of the agent based on the random variable r_4. Since the value computed using Equation (7) is continuous, with the aid of converting the value to discrete, a V-shaped transfer function is used. Line 24 to Line 34 represent the conversion of continuous value to discrete value based on the V-shaped transfer function and a random value. Line 36 represents the position of the alpha agent.

E. ALPHA AGENT, BETA AGENT, AND RANDOM AGENT

IBSCA's main goal is to keep the agents from settling into the local optimal solution and causing an early convergence. With the help of the best agent, the agents must conduct exploitation to figure out the global value. Alpha agent and beta agent are so introduced. The agent with the maximum fitness value is called the alpha agent, and the agent with the next maximum fitness value is called the beta agent. The alpha agent is manipulated with weight r_3 and the beta agent is manipulated with weight $(1 - r_3)$ when the agent's position is updated, increasing the likelihood that the best solution will be exploited because both the second-best solution and the best agent will be taken into account. Additionally, when the value of r_4 is larger than or equal to 0.5, the position of the random agent is taken into account together with the position of the alpha agent to ensure that the agent has a reasonable amount of exploration. This causes the algorithm to flip between exploration and exploitation with a probability of 0.5, which forces it to look for a globally optimal solution.

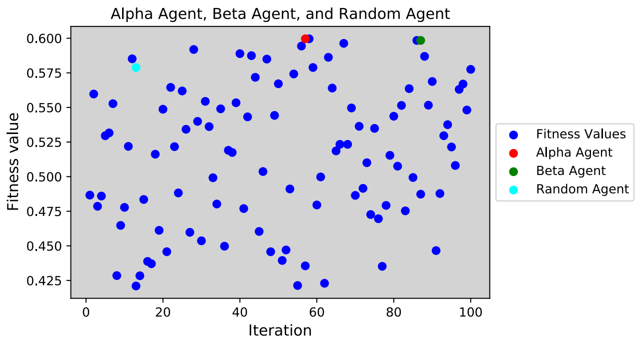

FIGURE 5. Alpha agent, beta agent, and random agent.

Figure 5 represents the alpha agent, beta agent, and random agent at the start of the iteration.

IV. EXPERIMENT RESULTS

The test is run on an Intel® CoreTM i7-8565U CPU running at 1.80 GHz with 16 GB of Memory and an X64-based processor. This part describes the dataset that was used to evaluate the efficiency of the proposed IBSCA. Tables 3 and 4 list the algorithms used to compare the effectiveness of the suggested algorithm along with the parameter definitions. This section elaborates on the metrics used for performance evaluation as well as the outcomes attained.

A. DESCRIPTION OF THE DATASETS

The proposed algorithm is evaluated with 18 imbalanced datasets taken from the KEEL repository [50]. The datasets taken into account are Glass1, Wisconsin, Pima, Iris0, Vehicle2, Ecoli1, Ecoli2, Ecoli3, Glass 2, Yeast-1_vs_7, Glass4, Ecoli4, page-blocks-1-2_vs_4, Abalone9-18, Glass5, Yeast4, Yeast5, Ecoli-0-1-3-7 vs 2-6. The full description of the imbalanced datasets taken into account is provided in Table 2. Small-scale datasets with fewer than 19 characteristics are included in every dataset taken into account. The datasets were selected with varying imbalance ratios because the paper does not address feature selection and instead focuses on class imbalance. The IR is computed using Equation (1). The datasets chosen for experimentation are categorized into three groups viz. Low-Level imbalanced dataset, Medium-Level imbalanced dataset, and High-level imbalanced dataset. The experiment is repeated 30 times, and the mean, min, max, and standard deviation are chosen for further comparison. Additionally, the datasets are evaluated using 5-fold cross-validation

B. ALGORITHMS TAKEN FOR COMPARISON

The proposed IBSCA has been compared with the conventional BSCA, BPSO, and Binary GWO (BGWO) algorithm. Various base classifiers such as K-NN (K = 1), Support Vector Machine (SVM), Random Forest (RF), and Neural Network (NN) were taken for experimentation. Each algorithm is designed to execute 30 times, with the average value being taken into account. The population size is

TABLE 2. Description of dataset.

Type of Dataset	Dataset	Number of Features	Number of Instances Majority Class	Minority Class	Imbalance Ratio
Low Level Imbalanced datasets	Glass1	9	139	77	1.81
	Wisconsin	9	445	240	1.85
	Pima	8	501	269	1.86
	Iris0	4	101	51	1.98
	Vehicle2	18	629	219	2.87
Medium Level Imbalanced datasets	Ecoli1	7	260	78	3.33
	Ecoli2	7	285	53	5.38
	Ecoli3	7	302	36	8.39
High Level Imbalanced datasets	Glass2	9	198	18	11.00
	Yeast-1_vs_7	7	429	30	14.3
	Glass4	9	201	13	15.47
	Ecoli4	7	316	20	15.8
	Page-blocks-1-3_vs_2	10	444	28	15.86
	Abalone9-18	8	689	42	16.4
	Glass5	9	205	9	22.78
	Yeast4	8	1433	51	28.1
	Ecoli-0-1-3-7_vs_2-6	7	274	7	39.14
	Yeast6	8	1449	35	41.4

TABLE 3. Parameters of metaheuristic algorithms taken for instance selection.

Algorithms	Size of Population	Dimension	Values for Parameters
BSCA	20	Number of instances in majority class	Number of Iterations $Max\_Iter = 100$, constant $a = 2$, $r_2 \in (0, 2\pi)$, $r_3 \in (0, 1)$, $r_4 \in (0, 1)$
BPSO	20	Number of instances in majority class	Cognitive component $c_1 = 2$, Social Component $c_2 = 2$, Number of Iterations $Max\_Iter = 100$
BGWO	20	Number of instances in majority class	Number of Iterations $Max\_Iter = 100$, Control Parameter $a = 2$, $r_1 \in (0, 1)$, $r_2 \in (0, 1)$
IBSCA (Proposed Work)	20	Number of instances in majority class	Number of Iterations $Max\_Iter = 100$, constant $a = 2$, $r_2 \in (0, 2\pi)$, $r_3 \in (0, 1)$, $r_4 \in (0, 1)$, $r_{max} = 0.9$, $r_{min} = 0.1$

fixed at 20, and the number of iterations is set at 100. Table 3 contains a summary of the metaheuristic algorithm parameters used for the experimentation. Additionally, the conventional SMOTE, BL SMOTE, and the suggested IBSCA have all been evaluated. The proposed algorithm has also been contrasted with other approaches currently in use by researchers, such as GWO with Multi-Layer Perceptron (GWO-MLP) [34], Simulated Annealing with Discriminant Analysis (SA-DA) [35], Neighborhood Search (NB-Basic) [51], Modified Tomek Link Search (NB-Tomek) [51], Common Nearest Neighbors search (NB-Comm) [51], and Recursive Search (NB-Rec). Table 4 lists the characteristics of several other current methods that are used for instance selection.

C. METRICS TAKEN FOR COMPARISON

Sensitivity (recall), F-Score, G-Mean, and AUC are some of the metrics used to assess how well the suggested algorithm performs. The metrics are all calculated based on the number of true positives, true negatives, false positives, and false negatives, abbreviated as n_{tp}, n_{tn}, n_{fp} and n_{fn}, respectively. Equation (3), (4), (5), and (6) show how G-Mean, F-score,

TABLE 4. Parameters for other instance selection algorithms.

Methodology	Algorithms	Values for Parameters
Over Sampling	SMOTE	$NumOfNeighbors = 5$
	BL SMOTE	$NumOfNeighbors = 5$, $NeighborsForConsideringInstance = 3$
	Deep Learning and Fuzzy Support Vector Machine [52]	Penalty Cost c and width of the gaussian kernel σ are selected using Grid Search Method, margin m and number of neurons is determined using Grid Search Method
Under Sampling	SA – DA [35]	Initial Temperature $Temp = 80000$, Cooldown factor $x = 0.8$
	NB Basic [51]	$K = \sqrt{size\ of\ dataset} + \sqrt{imbalance\ ratio}$
	NB Tomek [51]	$K = \sqrt{size\ of\ dataset} + \sqrt{imbalance\ ratio}$
	NB Comm [51]	$K = \sqrt{size\ of\ dataset} + \sqrt{imbalance\ ratio}$
	NB Rec [51]	$K = \sqrt{size\ of\ dataset} + \sqrt{imbalance\ ratio}$
	ISND [53]	$NumOfNeighbors = 5, \sigma = \sigma_0, \sigma = 2\sigma_0/3$, $\sigma = \sigma_0/3$ where σ_0 is the standard deviation of original minority normalized data
	GWO MLP [34]	$sizeofpopulation = 20, Max\_Iter = 50$

Recall, and Precision are calculated. AUC, which is described in Equation (11), is a quantitative representation of the Receiver Operating Characteristic curve that is built on a TPR and a False Positive Rate(FPR).

$$AUC = (1 + TPR - FPR)/2 \tag{11}$$

D. RESULT ANALYSIS

In this research, the IBSCA for class imbalance is evaluated and contrasted against various other algorithms such as BSCA, BPSO, and BGWO. The evaluation is done using various metrics such as sensitivity, F-score, G-mean, and AUC. The research also investigates the fitness values of the proposed IBSCA in terms of mean, best, worst, and standard deviation. Additionally, the G-Mean and F-Score of IBSCA are compared to those of conventional SMOTE approaches. Furthermore, using F-score and G-mean as assessment metrics, the analysis investigates how well IBSCA performs on datasets with various degrees of class imbalance, including low, medium, and large imbalanced datasets.

Statistical tests, such as the Wilcoxon signed-rank test and the Friedman test, are used to support the comparisons between IBSCA and other algorithms. Additionally, the study evaluates the effectiveness of different base classifiers without taking class imbalance into account, concentrating on sensitivity, F-score, G-mean, and AUC. Finally, using the same assessment measures (sensitivity, F-score, G-mean, and AUC), the performance of IBSCA is assessed when combined with different base classifiers such as Support Vector Machine (SVM), Random Forest (RF) and Neural Network (NN). The effectiveness of IBSCA in managing class imbalance and its possible impact on classification tasks are crucial insights provided by these thorough comparisons and analyses. Furthermore, the effectiveness of IBSCA in locating optimal or nearly optimal solutions for various functions can be evaluated only by applying it to certain benchmark functions. This analysis offers useful information regarding the efficiency and efficacy of IBSCA in handling various optimization landscapes as well as insights into its strengths and weaknesses in doing so.

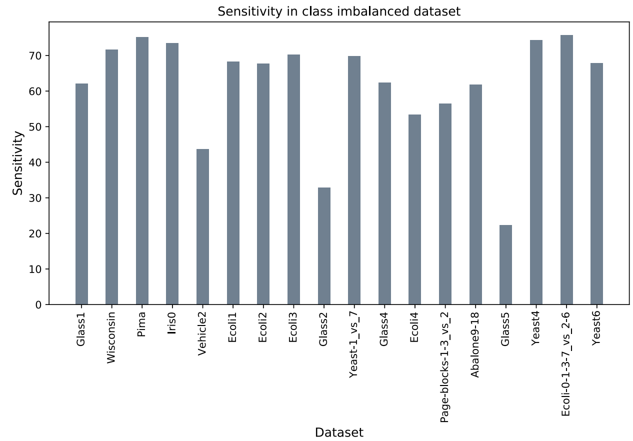

FIGURE 6. Sensitivity of different datasets using 1-NN.

1) COMPARISON OF IBSCA WITH BSCA, BPSO, AND BGWO

To address the issue of class imbalance and prevent the classifier from being biased towards the majority class, instance selection algorithms pick instances from the majority class. The first degree of experimentation is carried out to determine sensitivity. Sensitivity is a metric used to assess the precision of the minority class. It is regarded as the most crucial measure in datasets with class imbalances. The sensitivity of various datasets when K-NN is used as a predictor with K=1 is shown in Figure 6.

The comparison of the sensitivity of different instance selection algorithms is shown in Table 5. Table 5 clearly shows that IBSCA achieves a high level of sensitivity compared to other algorithms used as comparisons. It's because IBSCA employs two agents–Alpha, which has the best fitness value, and Beta, which has the next-best fitness value–which promotes the exploitation of the best solution. The suggested IBSCA offers exploration by taking the random agent's position into account with a probability of 0.5. IBSCA performs better than BSCA, BPSO, and BGWO for the low imbalanced Vehicle2 dataset, where the imbalance ratio is 2.87, by 0.63%, 2.37%, and 6.91%, correspondingly. Similarly, IBSCA achieves 7.68%, 8.9%, and 10.59% higher sensitivity than BSCA, BPSO, and BGWO, respectively, for the medium imbalanced Ecoli3 dataset, where the imbalance ratio is 8.39. Additionally, 10 extremely unbalanced datasets are tested, and IBSCA outperforms all other conventional metaheuristic algorithms in terms of sensitivity. For instance, BSCA, BPSO, and BGWO have reduced minimal sensitivity than IBSCA for the Yeast6 dataset, where the imbalance ratio is 41.4.

The next degree of comparison is used to calculate the F-Score, a crucial metric for unbalanced datasets. When the dataset is skewed, the F-Score is regarded as a crucial measure because it balances precision and recall. The F-score values derived for various datasets using various metaheuristic algorithms are shown in Table 6. IBSCA performs 4.41% better than BSCA, 5.63% and 8.53% better than BPSO and BGWO, respectively, for the less imbalanced Pima

TABLE 5. Comparison of sensitivity.

Dataset	IBSCA				BSCA				BPSO				BGWO			
	Mean	Min	Max	Std	Mean	Min	Max	Std	Mean	Min	Max	Std	Mean	Min	Max	Std
Glass1	74.35	74.13	74.50	0.09	72.13	71.92	72.42	0.10	69.19	68.99	69.41	0.09	66.0	65.75	66.17	0.10
Wisconsin	96.34	96.08	96.67	0.11	95.45	95.29	94.68	0.089	94.3	94.10	94.56	0.11	91.2	90.95	91.40	0.11
Pima	92.3	92.14	92.51	0.08	90.1	89.94	90.35	0.09	89.1	88.17	89.29	0.11	84.3	84.17	84.54	0.10
Iris0	99.1	98.89	99.32	0.13	97.32	97.13	97.48	0.088	95.3	94.99	95.41	0.09	94.3	94.11	94.48	0.09
Vehicle2	82.5	82.29	82.65	0.08	81.98	81.72	82.15	0.10	80.54	80.33	80.78	0.09	76.8	76.57	76.94	0.086
Ecoli1	96.12	95.92	96.24	0.08	94.62	94.44	94.79	0.09	93.23	93.01	93.42	0.12	91.14	90.85	91.31	0.09
Ecoli2	93.2	92.95	93.36	0.08	91.3	91.12	91.48	0.09	88.12	87.87	88.32	0.10	72.12	72.01	72.37	0.07
Ecoli3	94.35	94.17	94.53	0.08	87.10	86.94	87.30	0.09	85.89	85.62	86.04	0.11	84.35	84.16	84.53	0.09
Glass2	53.4	53.16	53.58	0.09	54.67	54.37	54.88	0.11	51.87	51.68	52.11	0.08	56.32	56.13	56.49	0.10
Yeast-1_vs_7	84.44	84.22	84.63	0.09	82.05	81.87	82.17	0.08	81.87	81.65	82.10	0.11	80.45	80.27	80.67	0.085
Glass4	93.21	92.99	93.37	0.08	90.69	90.51	90.89	0.09	87.41	87.22	87.55	0.08	86.92	86.67	87.09	0.09
Ecoli 4	95.87	95.73	96.03	0.07	91.23	91.01	91.39	0.08	89.73	91.01	91.39	0.08	87.64	87.46	87.81	0.08
Page-blocks-1-3_vs_2	96.14	96.02	96.35	0.08	92.79	92.63	92.99	0.10	91.71	91.52	91.99	0.11	90.45	90.24	90.66	0.10
Abalone9-18	76.01	75.76	76.19	0.10	74.14	73.94	74.24	0.08	74.30	74.09	74.58	0.10	73.83	73.63	74.04	0.08
Glass5	53.14	52.96	53.35	0.10	51.06	50.82	51.29	0.10	50.89	50.71	51.05	0.10	49.10	48.85	49.27	0.10
Yeast4	89.62	89.37	89.83	0.11	88.21	88.05	88.47	0.09	87.14	86.95	87.39	0.10	86.44	86.18	86.62	0.11
Ecoli-0-1-3-7_vs_2-6	95.14	94.83	95.34	0.11	91.32	91.15	91.48	0.09	90.01	89.80	90.21	0.09	89.91	89.70	90.09	0.09
Yeast6	87.05	86.87	87.24	0.09	85.17	85.02	85.32	0.08	84.83	84.65	84.98	0.08	81.40	81.19	81.68	0.11

TABLE 6. Comparison of F-Score.

Dataset	IBSCA				BSCA				BPSO				BGWO			
	Mean	Min	Max	Std	Mean	Min	Max	Std	Mean	Min	Max	Std	Mean	Min	Max	Std
Glass1	59.45	59.17	59.60	0.10	56.34	56.14	56.55	0.10	58.43	58.17	58.64	0.10	55.24	55.01	55.46	0.09
Wisconsin	98.3	98.09	98.54	0.10	95.24	95.01	95.48	0.11	94.3	94.10	94.44	0.08	93.2	92.93	93.33	0.09
Pima	70.3	70.00	70.51	0.10	67.2	66.93	67.46	0.10	66.34	66.20	66.65	0.09	64.3	64.13	64.43	0.09
Iris0	99.32	99.09	99.48	0.09	95.5	95.29	95.72	0.09	92.1	91.92	92.22	0.07	91.38	91.21	91.56	0.10
Vehicle2	97.35	97.14	97.57	0.10	96.43	96.18	96.60	0.10	95.28	95.07	95.53	0.12	84.34	84.12	84.53	0.10
Ecoli1	88.97	88.81	89.19	0.09	87.34	87.15	87.54	0.10	85.32	85.17	85.57	0.10	82.12	81.91	82.33	0.11
Ecoli2	94.45	94.29	94.67	0.10	87.34	87.13	87.55	0.10	89.65	89.45	89.93	0.12	74.34	74.14	74.51	0.09
Ecoli3	75	74.76	75.17	0.10	72.3	72.06	72.57	0.12	69.12	68.91	69.33	0.11	67.85	67.65	68.00	0.08
Glass2	52.22	52.06	52.40	0.08	51.34	51.18	51.54	0.09	49.34	49.19	49.60	0.10	44.23	44.05	44.43	0.10
Yeast-1_vs_7	69.21	69.00	69.47	0.11	55.65	55.48	55.81	0.08	54.86	54.65	55.02	0.09	51.72	51.59	51.86	0.07
Glass4	81.07	80.88	81.28	0.11	80.67	80.55	80.82	0.07	75.33	75.20	75.45	0.07	74.86	74.63	75.06	0.09
Ecoli 4	76.89	76.67	77.04	0.09	73.85	73.65	74.09	0.11	71.74	71.56	71.96	0.09	70.82	70.57	71.02	0.11
Page-blocks-1-3_vs_2	98.15	97.91	98.35	0.10	97.14	96.98	97.35	0.08	96.7	96.41	96.83	0.10	91.71	91.42	91.93	0.11
Abalone9-18	61.30	61.14	61.57	0.10	59.54	59.33	59.77	0.12	55.71	55.56	55.88	0.08	52.25	52.10	52.51	0.09
Glass5	64.25	64.00	64.46	0.09	61.33	61.09	61.54	0.08	60.15	59.93	60.36	0.08	58.48	58.30	58.74	0.09
Yeast4	44.76	44.58	45.02	0.10	41.12	40.99	41.27	0.08	40.52	40.34	40.69	0.11	39.18	39.02	39.44	0.10
Ecoli-0-1-3-7_vs_2-6	69.43	69.31	69.67	0.09	68.77	68.57	68.97	0.10	65.45	65.23	65.61	0.11	64.78	64.60	65.03	0.11
Yeast6	72.58	72.34	72.74	0.10	69.52	69.37	69.81	0.11	67.28	67.16	67.48	0.09	64.32	64.08	64.49	0.11

dataset with an imbalance ratio of 1.86. IBSCA outperforms BSCA, BPSO, and BGWO, respectively, by 7.52%, 5.08%, and 21.29% for the moderately imbalanced Ecoli2 dataset. IBSCA has a higher F-score than BSCA, BPSO, and BGWO at 3.95%, 6.69%, and 7.89%, respectively, for the extremely unbalanced Ecoli4 dataset. In other terms, IBSCA performs 1.71 times better than BSCA does. This demonstrates that, compared to other current methods, the proposed method improves the trade-off between sensitivity and specificity by reducing the likelihood of false positive and false negative predictions.

Additional testing is done to evaluate G-mean of the proposed IBSCA-based selection of instances to conventional algorithms. Due to its dependence on both true positive and true negative rates, G-mean is another crucial measure for class-imbalanced datasets [51]. Table 7 shows that, except for the Ecoli3 dataset, the proposed IBSCA algorithm has a greater G-mean than all other algorithms. In the Ecoli3 dataset, where the unbalanced ratio is 8.39, BPSO performs 1% better than IBSCA. However, if we take into account the extremely unbalanced Glass2 dataset, IBSCA performs 0.79% better than BPSO. Additionally, IBSCA performs 1.42%, 2.88%, and 4.77% better than BSCA, BPSO, and BGWO, correspondingly, for the Yeast6 dataset, where the imbalance ratio is 41.4. This demonstrates that IBSCA performs above typical regardless of whether the imbalance ratio is large.

Table 8 compares the suggested IBSCA's AUC to those of other algorithms. It is clear from Table 8 that IBSCA outperforms BSCA, BPSO, and BGWO regarding outcomes. IBSCA performs 0.43%, 0.71%, and 2.11% better than BSCA, BPSO, and BGWO, correspondingly, for the low-imbalanced Iris0 dataset. IBSCA performs 0.46%, 0.25%, and 3.37% better than BSCA, BPSO, and BGWO separately for the medium imbalanced Ecoli3 dataset. IBSCA gets the second-highest AUC of 94.92 for page blocks 1-3 vs 4, while BSCA achieves the highest AUC of 95.23. IBSCA gets the best AUC value among all other algorithms, except for page blocks 1-3 vs 4. IBSCA gets an AUC of 98.93 even for the highly unbalanced Yeast6 dataset, which is 2.70, 3.18, and 5.92% better than BSCA, BPSO, and BGWO, respectively. This is so that the optimal subset of the majority class can be

TABLE 7. Comparison of G-Mean.

Dataset	IBSCA				BSCA				BPSO				BGWO			
	Mean	Min	Max	Std	Mean	Min	Max	Std	Mean	Min	Max	Std	Mean	Min	Max	Std
Glass1	59.91	59.67	60.10	0.09	57.20	57.09	57.48	0.10	57.93	57.70	58.21	0.10	54.6	54.41	54.85	0.11
Wisconsin	97.67	97.52	97.92	0.09	96.12	95.90	96.31	0.10	94.35	94.11	94.49	0.09	94.8	94.55	95.11	0.11
Pima	75.17	75.01	75.38	0.11	74.20	73.98	74.37	0.10	74.18	73.93	74.35	0.09	72.11	71.87	72.30	0.11
Iris0	99.67	99.54	99.81	0.07	99.12	98.85	99.28	0.10	99.15	98.99	99.33	0.09	97.13	97.01	97.38	0.10
Vehicle2	83.45	83.22	83.63	0.09	82.13	81.79	82.30	0.10	81.12	80.95	81.37	0.09	79.91	79.75	80.14	0.09
Ecoli1	94.3	94.16	94.48	0.09	89.43	89.28	89.65	0.10	87.69	87.42	87.92	0.10	88.34	88.17	88.57	0.09
Ecoli2	97.1	96.93	97.24	0.09	95.43	95.25	95.59	0.09	91.94	91.59	92.13	0.11	94.12	93.89	94.33	0.09
Ecoli3	95.67	95.46	95.81	0.08	94.3	94.17	94.57	0.08	96.63	96.43	96.85	0.10	91.43	91.22	91.67	0.11
Glass2	79.13	78.91	79.40	0.11	74.3	74.16	74.51	0.09	78.50	78.36	78.67	0.08	74.24	73.97	74.53	0.11
Yeast-1_vs_7	79.84	79.62	80.12	0.10	77.24	77.03	77.41	0.10	75.03	74.75	75.19	0.10	73.26	73.12	73.46	0.09
Glass4	84.32	84.09	84.44	0.08	82.14	81.95	82.45	0.12	81.92	81.70	82.17	0.10	80.73	80.53	80.99	0.09
Ecoli 4	99.51	99.37	99.69	0.07	97.34	97.04	97.52	0.10	96.12	95.92	96.35	0.09	93.47	93.20	93.76	0.12
Page-blocks-1-3_vs_2	99.62	99.42	99.81	0.09	98.01	97.77	98.24	0.11	95.39	95.18	95.57	0.10	93.16	92.87	93.45	0.13
Abalone9-18	70.13	69.92	70.30	0.09	68.45	68.32	68.75	0.10	67.14	66.92	67.40	0.11	66.32	66.09	66.51	0.10
Glass5	97.37	97.14	97.52	0.09	94.68	94.44	94.94	0.11	91.14	90.99	91.31	0.09	89.43	89.25	89.67	0.09
Yeast4	52.61	52.39	52.84	0.09	50.53	50.29	50.63	0.09	49.42	49.24	49.62	0.09	47.13	46.97	47.32	0.10
Ecoli-0-1-3-7_vs_2-6	98.71	98.46	98.89	0.10	95.16	94.99	95.37	0.08	94.71	94.53	94.88	0.08	91.18	90.93	91.43	0.11
Yeast6	93.15	92.98	93.30	0.07	91.82	91.64	92.01	0.09	90.46	90.26	90.72	0.09	88.71	88.49	88.91	0.10

found. The suggested IBSCA algorithm uses random agents in addition to alpha and beta agents, and all of the agents exhibit high levels of exploration and exploitation.

2) COMPARISON OF FITNESS

The fitness value of the suggested IBSCA is contrasted with those of other algorithms for experimentation in Table 9. Comparing the suggested IBSCA's Mean, Best, and Worst fitness values to those of other algorithms, it can be seen that IBSCA outperforms them. The conclusions drawn from Table 9 are as follows:

- For the Glass dataset, the mean, best, and worst fitness is higher than the other algorithms which validates the introduction of alpha, beta, and random agents. The BGWO has the highest standard deviation indicating that it has greater variability in fitness. For the Wisconsin dataset, solutions in BGWO are more sensitive which results in high variability in fitness as indicated by its higher standard deviation. Similarly, BPSO and BSCA have moderate performance while considering fitness, but they have less variability when compared to BGWO. For the PIMA dataset, the mean fitness of IBSCA (0.67) is higher than BPSO (0.50), BGWO (0.43), and BSCA (0.63). Also, IBSCA has the smallest standard deviation (0.01). For the Iris 0 and Vehicle 2 datasets, IBSCA has superior performance in mean and best fitness showing its ability to find good solutions. But the highest worst fitness indicates the chance of getting trapped in local optima occasionally which needs further refinement
- For all the 4 medium imbalanced datasets, IBSCA achieves good value for mean fitness and minimum value for standard deviation validating the performance is superior to other algorithms. This is because the algorithm not only considers the alpha agent, and beta agent which represent the first and next best solution, but also considers the random agent thereby making exploration equal to exploitation.
- Except for page blocks 1-3_vs_2, for all of the highly unbalanced datasets, the proposed IBSCA has a superior mean, best, and worst fitness than other algorithms used in experimentation. The mean fitness of IBSCA is 0.67889 while the mean fitness of BSCA is 0.81941, resulting in the best AUC of 95.23 for the page-blocks 1-3_vs_2 dataset.

3) COMPARISON OF IBSCA WITH TRADITIONAL SMOTE TECHNIQUES

In terms of F-score and G-mean, the suggested IBSCA is contrasted with conventional SMOTE and BL SMOTE techniques. Table 10 compares the F-Scores of the IBSCA, SMOTE, and BL SMOTE. As opposed to SMOTE and its variations, the proposed IBSCA is seen to have a higher degree of F-score. For the Glass1 dataset, the conventional SMOTE has a 6.76% lower F-Score than IBSCA. IBSCA raises the F-Score for the Ecoli3 dataset by 14.46% and 30.25%, respectively. Additionally, IBSCA increases the F-Score for the highly unbalanced Glass2 dataset by 4.20% and 7.69% compared to SMOTE and BL SMOTE. The proposed IBSCA also uses a random agent to investigate and exploit the solution space, producing an Optimal Subset of the Majority class.

The findings of comparing the G-mean of the IBSCA with SMOTE and BL SMOTE are tabulated in Table 10. Table 10 makes it clear that IBSCA has a higher G-mean score than SMOTE and BL SMOTE. The efficiency of IBSCA is superior to SMOTE and BL SMOTE for the low-level imbalanced Pima dataset by 11.30% and 11.97%, respectively. IBSCA also gets a 9.39% higher G-mean than SMOTE and an 8.21% higher value than BL-SMOTE for the medium imbalanced Ecoli2 dataset. IBSCA's G-mean for the large unbalanced Glass2 dataset is 79.13, which is significantly higher than SMOTE's G-mean result of 58.19, which is 26.46% lower than IBSCA.

4) ANALYSIS ON LOW AND MEDIUM IMBALANCED DATASETS

Researchers use a variety of algorithms to address the class imbalance issue, and the performance of the proposed IBSCA

TABLE 8. Comparison of area under curve (AUC).

Dataset	IBSCA				BSCA				BPSO				BGWO			
	Mean	Min	Max	Std	Mean	Min	Max	Std	Mean	Min	Max	Std	Mean	Min	Max	Std
Glass1	59.94	59.73	60.13	0.11	57.45	57.24	57.69	0.09	57.94	57.73	58.12	0.09	56.78	56.57	56.92	0.09
Wisconsin	98.01	97.82	98.21	0.09	97.23	97.04	97.47	0.09	95.01	94.70	95.19	0.11	95.27	95.09	95.45	0.10
Pima	76.88	76.59	77.10	0.11	76.16	76.05	76.37	0.08	76.73	76.58	76.95	0.09	73.27	73.10	73.43	0.09
Iris0	99.91	99.69	100.09	0.10	99.48	99.36	99.64	0.07	99.20	98.99	99.48	0.09	97.80	97.58	97.99	0.10
Vehicle2	87.65	87.43	87.90	0.12	87.17	86.94	87.36	0.10	81.43	81.26	81.63	0.09	80.58	80.38	80.73	0.09
Ecoli1	97.89	97.72	98.10	0.09	97.55	97.29	97.74	0.11	87.69	87.49	87.86	0.10	89.16	88.94	89.48	0.11
Ecoli2	97.95	97.58	98.14	0.11	97.83	97.62	98.06	0.09	92.61	92.41	92.68	0.07	94.70	94.49	94.93	0.10
Ecoli3	95.71	95.49	95.97	0.12	95.27	95.06	95.55	0.10	95.47	95.16	95.71	0.10	92.48	92.25	92.62	0.09
Glass2	82.34	82.09	82.51	0.10	81.37	81.15	81.61	0.11	79.77	79.61	79.96	0.09	76.32	76.11	76.51	0.09
Yeast-1_vs_7	82.56	82.36	82.86	0.11	81.55	81.38	81.69	0.09	80.92	80.80	81.28	0.11	78.79	78.59	79.02	0.09
Glass4	78.10	78.01	78.34	0.08	73.56	73.33	73.86	0.11	72.81	72.67	73.02	0.09	72.06	71.91	72.30	0.09
Ecoli 4	94.55	94.34	94.78	0.10	93.92	93.76	94.13	0.09	91.15	90.88	91.27	0.08	90.05	89.79	90.25	0.10
Page-blocks-1-3_vs_2	94.92	94.76	95.14	0.09	95.23	95.00	95.50	0.11	90.18	89.92	90.42	0.11	88.27	88.08	88.50	0.10
Abalone9-18	94.54	94.33	94.69	0.09	92.13	92.00	92.37	0.10	90.45	90.24	90.65	0.13	89.46	89.36	89.67	0.08
Glass5	89.86	89.68	90.09	0.10	88.15	87.95	88.41	0.10	87.79	87.58	88.11	0.12	86.04	85.83	86.28	0.11
Yeast4	82.16	82.00	82.26	0.07	80.86	80.64	81.07	0.11	79.95	79.78	80.21	0.09	77.56	77.37	77.75	0.09
Ecoli-0-1-3-7_vs_2-6	94.87	94.66	95.08	0.09	93.70	93.55	93.83	0.07	91.79	91.66	91.96	0.08	90.24	90.06	90.36	0.08
Yeast6	98.93	98.65	99.10	0.10	96.26	96.10	96.50	0.09	95.79	95.56	96.10	0.12	93.07	92.96	93.22	0.07

algorithm is compared with those algorithms. Figure 7 clearly shows that the suggested IBSCA's F-score value is higher than that of other algorithms. IBSCA's F-score for the glass1 dataset is 7.65% higher than GWO- MLP [34]. However, the suggested IBSCA receives a 5.48% lower F-score than DFSVM. Because DFSVM gets the highest F-Score of 69.2 for the Glass 1 dataset, the proposed IBSCA comes in second. In a similar vein, IBSCA's success in Wisconsin is 3.40 percent better than that of NB-Basic [51], NB-Tomek [51], NB-Comm [51], NB-Rec [51], and SA-DA [35]. However, ISND outperforms IBSCA in terms of Wisconsin's F-Score by 0.48%. IBSCA outperforms NB-Basic [51], NB-Tomek [51], NB-Comm [51], NB-Rec [51], ISND [53], and SA-DA [35] by 18.72%, 17.084%, 17.511%, 26.799%, 14.21%, and 4.125%, respectively. Iris0's F-score for the proposed IBSCA is 99.32, placing it second overall behind GWO-MLP [34], whose F-Score is 99.7. IBSCA performs 22%, 18.84%, 15.147%, 6.66%, 15.14%, and 14.82% better than GWO-MLP [34], NB-Basic [51], NB-Tomek [51], NB-Comm [51], NB-Rec [51], and SA-DA [35] for the medium-level imbalanced Ecoli3 dataset.

The comparison between G-mean and the other algorithms is described in [34], [35], [51], and [53] and is shown in Figure 8. The suggested IBSCA comes in second place for the low-level imbalanced Glass1 dataset with a G-Mean of 59.91, while GWO-MLP [34] comes in first place with a G-Mean of 62. For the Pima Dataset, IBSCA outperforms NB-Basic [51], NB-Tomek [51], NB-Comm [51], ISND [51], and SA-DA [35] by 36.81%, 29.719%, 26.54%, 10.77%, and 0.61%, respectively. In comparison to GWO-MLP [34], the G-mean of the iris0 dataset is 99.67, which is 0.23% lower. However, for the vehicle2 dataset, where the unbalanced ratio is 2.87, GWO-MLP [34] performs 10.48% worse than the suggested IBSCA. The G-Mean of the suggested IBSCA is improved for the Medium Unbalanced Ecoli3 dataset by 8.74%, 3.62%, 0.83%, 2.69%, and 8.43% for GWO-MLP [34], NB-Basic [51], NB-Tomek [51], NB-Comm [51], NB-Rec [51], and SA-DA [35].

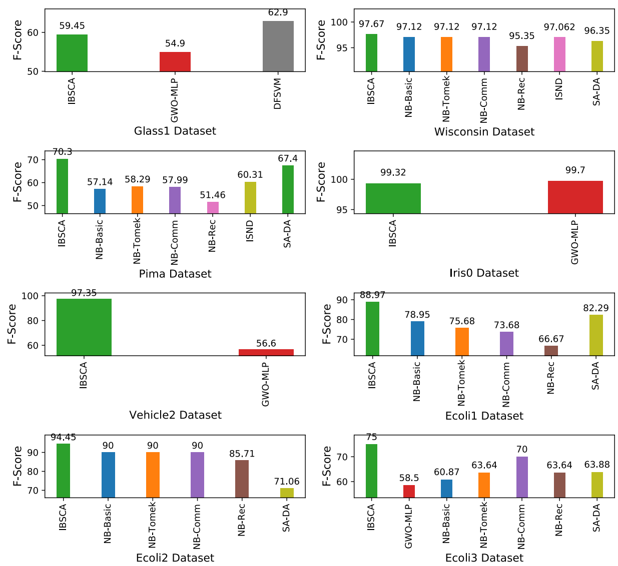

FIGURE 7. Comparison of F-Score on low and medium imbalanced dataset.

When it comes to maximizing F-Score for the Wisconsin and Iris0 datasets, Improved SMOTE based on the normal distribution and GWO-MLP ranks first, followed by the proposed IBSCA. Out of 5 less-imbalanced datasets, Glass2 and Iris0 are ranked first by GWO MLP, and suggested IBSCA is ranked second. The proposed IBSCA outperforms other current mechanisms in terms of G-mean for the three mediumly imbalanced datasets.

5) ANALYSIS ON LARGE IMBALANCED DATASETS

For the mainly unbalanced Glass2 dataset, the F-Score of IBSCA is higher than those of NB-Basic [51], NB-Tomek [51], NB-Comm [51], NB-Rec [51], SA-DA [35], and SA-DA by 36.17%, 23.40%, 36.17%, 45.28%, and 36.17%, respectively. For the Glass2 dataset, the G-Mean of IBSCA is 6.53% higher than NB-Basic [51], 3.65% higher than NB-Tomek [51], 6.53% higher than NB-Comm [51], and 9.50% higher than NB-Rec [51]. However, for the Glass2 dataset,

TABLE 9. Comparison of fitness.

Dataset	Method	Characteristics of Fitness			
		Mean	Best	Worst	Std
Glass1	IBSCA	0.75795	0.99506	0.58451	0.00928
	BSCA	0.68613	0.79604	0.56781	0.02979
	BPSO	0.58289	0.5997	0.55154	0.06431
	BGWO	0.55257	0.59735	0.5026	0.11697
Wisconsin	IBSCA	0.83138	0.97139	0.10955	0.06819
	BSCA	0.74482	0.95057	0.32497	0.11574
	BPSO	0.64403	0.94218	0.56362	0.19504
	BGWO	0.53949	0.94047	0.71272	0.2607
Pima	IBSCA	0.69664	0.71149	0.68218	0.008
	BSCA	0.63438	0.70185	0.56363	0.04008
	BPSO	0.50098	0.69391	0.32151	0.10904
	BGWO	0.4383	0.7263	0.10285	0.17896
Iris0	IBSCA	0.82654	0.99459	0.71295	0.07178
	BSCA	0.7516	0.95371	0.56441	0.1142
	BPSO	0.62702	0.94686	0.32811	0.18054
	BGWO	0.57199	0.9393	0.11654	0.2504
Vehicle2	IBSCA	0.7794	0.8928	0.71204	0.24394
	BSCA	0.71848	0.89115	0.56223	0.32179
	BPSO	0.62788	0.881	0.32179	0.56223
	BGWO	0.53585	0.82001	0.10211	0.71204
Ecoli1	IBSCA	0.78586	0.90961	0.71288	0.03984
	BSCA	0.71863	0.88095	0.56667	0.09192
	BPSO	0.61983	0.86368	0.32297	0.18292
	BGWO	0.53465	0.85222	0.10596	0.22244
Ecoli2	IBSCA	0.77961	0.95196	0.5638	0.03668
	BSCA	0.74029	0.90593	0.71236	0.10378
	BPSO	0.53335	0.90588	0.3214	0.16849
	BGWO	0.60304	0.84154	0.10258	0.25383
Ecoli3	IBSCA	0.7571	0.83092	0.71239	0.02466
	BSCA	0.61249	0.83061	0.32036	0.13905
	BPSO	0.52575	0.8229	0.26577	0.16446
	BGWO	0.46991	0.79606	0.10437	0.23042
Glass2	IBSCA	0.6424	0.70771	0.59332	0.03343
	BSCA	0.46676	0.6451	0.32174	0.08711
	BPSO	0.44402	0.63847	0.26615	0.11564
	BGWO	0.36885	0.62727	0.10428	0.16249
Yeast-1_vs_7	IBSCA	0.66686	0.73949	0.62498	0.09389
	BSCA	0.47972	0.71158	0.32002	0.11227
	BPSO	0.44966	0.65862	0.26691	0.19731
	BGWO	0.41285	0.64918	0.10822	0.02758
Glass4	IBSCA	0.74741	0.81151	0.71215	0.01708
	BSCA	0.55874	0.79786	0.32355	0.14789
	BPSO	0.51665	0.78108	0.27177	0.15288
	BGWO	0.44677	0.77457	0.10121	0.20143
Ecoli4	IBSCA	0.76895	0.87921	0.71587	0.0305
	BSCA	0.59295	0.83793	0.32354	0.14956
	BPSO	0.53162	0.83447	0.26889	0.16769
	BGWO	0.51604	0.8175	0.7137	0.22935
Page-blocks-1-3_vs_2	IBSCA	0.67889	0.98346	0.32151	0.06436
	BSCA	0.81941	0.97079	0.27307	0.19734
	BPSO	0.63116	0.955	0.10805	0.20225
	BGWO	0.5817	0.92156	0.1021	0.26426
Abalone9-18	IBSCA	0.65212	0.71147	0.59456	0.03462
	BSCA	0.48877	0.65205	0.32286	0.09836
	BPSO	0.42787	0.63461	0.26621	0.10669
	BGWO	0.38811	0.6017	0.10998	0.16165
Glass5	IBSCA	0.72505	0.80287	0.71202	0.00757
	BSCA	0.53855	0.77109	0.32324	0.12491
	BPSO	0.49534	0.75242	0.2703	0.13403
	BGWO	0.43726	0.73953	0.10282	0.19857
Yeast4	IBSCA	0.57603	0.71163	0.4342	0.03981
	BSCA	0.39381	0.4847	0.32022	0.0503
	BPSO	0.36267	0.45625	0.27134	0.08283
	BGWO	0.28769	0.44951	0.10385	0.11638
Ecoli-0-1-3-7_vs_2-6	IBSCA	0.74714	0.83293	0.71218	0.02057
	BSCA	0.57103	0.81801	0.32447	0.15356
	BPSO	0.53059	0.79053	0.26743	0.1562
	BGWO	0.46466	0.77961	0.10262	0.21181
Yeast6	IBSCA	0.73617	0.81184	0.71202	0.01553
	BSCA	0.57908	0.80267	0.32549	0.1296
	BPSO	0.51739	0.787	0.26637	0.14638
	BGWO	0.48108	0.7468	0.12554	0.19742

the G-Mean of SA-DA is 1.98% higher than the proposed IBSCA. For maximizing G-Mean and F-Score for the yeast-1 vs 7, the proposed IBSCA comes first. The suggested IBSCA outperforms the NB-Basic [51], NB-Tomek [51], NB-Comm [51], NB-Rec [51], and SA-DA [35] in terms of F-Score by 76.56%, 87.43%, 76.88%, 74.66%, and 31.31%,

TABLE 10. Comparison of IBSCA with SMOTE techniques.

Dataset	IBSCA		SMOTE		BL SMOTE	
	F-Score	G-Mean	F-Score	G-Mean	F-Score	G-Mean
Glass1	59.45	59.91	55.43	55.14	55.89	54.03
Wisconsin	98.3	97.67	94.67	96.61	95.14	96.63
Pima	70.3	75.17	66.17	66.67	66.18	66.17
Iris0	99.32	99.67	91.32	91.36	91.32	91.67
Vehicle2	97.35	83.45	89.12	81.12	90.19	81.89
Ecoli1	88.97	94.3	81.1	81.2	77.32	83.2
Ecoli2	94.45	97.1	87.13	87.98	87.89	89.12
Ecoli3	75	95.67	64.15	62.78	52.31	46.09
Glass2	52.22	79.13	49.18	58.19	47.39	58.23
Yeast-1$_v s_7$	69.21	79.84	0	0	0	0
Glass4	81.07	84.32	66.67	70.71	66.67	70.71
Ecoli4	76.89	99.51	100	100	85.71	86.6
Page-blocks-1-3$_v s_2$	98.15	99.62	100	100	88.89	89.44
Abalone9-18	61.30	70.13	18.18	35.1	0	0
Glass5	64.25	97.37	0	0	0	0
Yeast4	44.76	52.61	33.33	54.29	33.33	54.29
Ecoli-0-1-3-7$_v s_2 - 6$	69.43	98.71	100	100	100	100
Yeast6	72.58	93.15	66.67	75.46	20	37.67

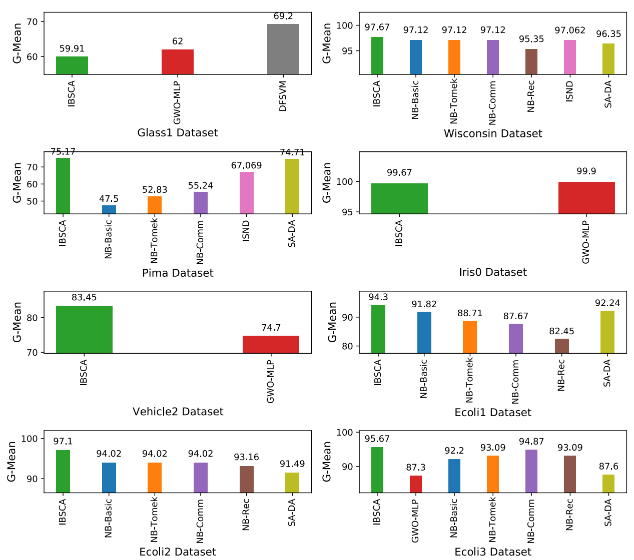

FIGURE 8. Comparison of G-Mean on low and medium imbalanced dataset.

respectively. Similarly, IBSCA outperforms NB-Basic [51], NB-Tomek [51], NB-Comm [51], NB-Rec [51], and SA-DA [35] in terms of G-Means, achieving 27.48%, 53.93%, 35.32%, 22.56%, and 1.74% higher G-Means. The suggested IBSCA comes in second place for the Glass 4 dataset, where the imbalance ratio is 15.47, while SA-DA comes in first place and boosts G-Means by 14.15%. IBSCA, however, raises the F-Score by 21.20% more than SA-DA. When it comes to optimizing F-Score for the Ecoli4 dataset, NB-Comm [51] performs 30.05% better than IBSCA. By 0.49% less than NB-Comm, the suggested IBSCA comes in second place for maximizing G-Mean [51]. The proposed IBSCA places first with an F-Score of 61.3 and a G-Mean of 70.13 for the Abalone 9-18 dataset, where the imbalance ratio is 16.4, while SA-DA [35] gets an F-Score of 59.44 and a G-Mean of 68.02. The F-Score for the Ecoli 4 dataset for NB-Comm [51], is 100%, which is 62.71% lower than the F-Score for the suggested IBSCA. For the Glass 5 datasets, where the vast majority of techniques result in F-score and G-Mean values of 0. This demonstrates yet again how well the suggested

TABLE 11. Performance comparison of IBSCA with existing works for Glass 2 Dataset.

Dataset	Glass 2	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	**52.22**	79.13
NB-Basic [51]	33.33	73.96
NB-Tomek [51]	40	76.24
NB-Comm [51]	33.33	73.96
NB-Rec [51]	28.57	71.61
SA-DA [35]	39.22	**80.7**

TABLE 12. Performance comparison of IBSCA with existing works for Yeast-1_vs_7 Dataset.

Dataset	Yeast-1_vs_7	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	**69.21**	**79.84**
NB-Basic [51]	16.22	57.9
NB-Tomek [51]	8.7	36.78
NB-Comm [51]	16	51.64
NB-Rec [51]	17.54	61.83
SA-DA [35]	47.54	78.45

TABLE 13. Performance comparison of IBSCA with existing works for GLASS 4 Dataset.

Dataset	Glass 4	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	**81.07**	84.32
NB-Basic [51]	66.67	70.71
NB-Tomek [51]	66.67	70.71
NB-Comm [51]	66.67	70.71
NB-Rec [51]	23.53	82.16
SA-DA [35]	63.89	**96.25**

IBSCA improves the classification of instances belonging to the majority and minority classes. Additionally, the proposed IBSCA outperforms other existing methodologies for the extremely unbalanced Yeast6 dataset, where the imbalance ratio is 41.4. The proposed IBSCA ranks first for seven of the ten largely unbalanced datasets in terms of maximizing F-score and for five of the ten in terms of maximizing G-means.

The conclusions drawn from Table 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 are as follows:

- The proposed IBSCA outperforms other existing mechanisms in terms of F-score, which quantifies the trade-off between sensitivity and specificity. In other words, the proposed technique reduces the number of false positives and negatives, improving predictions of true positives and true negatives.
- Even in circumstances where the current mechanisms are unable to generate G-mean, the proposed IBSCA has a high G-mean. This demonstrates that the suggested mechanism effectively balances the classification of classes into majorities and minorities.

6) STATISTICAL TESTS

To compare the effectiveness of the suggested IBSCA with other mechanisms, the Wilcoxon Signed-rank test was used. IBSCA is superior to other algorithms for choosing the best subset of instances from the majority class, according

TABLE 14. Performance comparison of IBSCA with existing works for Ecoli4 Dataset.

Dataset	Ecoli4	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	76.89	99.51
NB-Basic [51]	88.89	99.2
NB-Tomek [51]	88.89	99.2
NB-Comm [51]	**100**	**100**
NB-Rec [51]	80	98.4
SA-DA [35]	66.67	76.97

TABLE 15. Performance comparison of IBSCA with existing works for Page-blocks-1-3_vs_2 Dataset.

Dataset	Page-blocks-1-3_vs_2	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	**98.15**	**99.62**
NB-Basic [51]	71.43	97.7
NB-Tomek [51]	71.43	97.7
NB-Comm [51]	71.43	97.7
NB-Rec [51]	35.71	89.19
SA-DA [35]	53.26	81.42

TABLE 16. Performance comparison of IBSCA with existing works for Abalone9-18 Dataset.

Dataset	Abalone9-18	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	**61.30**	**70.13**
NB-Basic [51]	12.9	56.35
NB-Tomek [51]	13.04	52.84
NB-Comm [51]	22.86	64.5
NB-Rec [51]	16	64.5
SA-DA [35]	59.44	68.02

TABLE 17. Performance comparison of IBSCA with existing works for Glass5 Dataset.

Dataset	Glass5	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	**64.25**	**97.37**
NB-Basic [51]	0	0
NB-Tomek [51]	0	0
NB-Comm [51]	0	0
NB-Rec [51]	0	0
SA-DA [35]	62.12	96.29

TABLE 18. Performance comparison of IBSCA with existing works for Yeast4 Dataset.

Dataset	Yeast4	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	**44.76**	52.61
NB-Basic [51]	23.88	81.42
NB-Tomek [51]	30.19	83.79
NB-Comm [51]	32	**84.29**
NB-Rec [51]	17.58	77.19
SA-DA [35]	41.93	76.52

to the research hypothesis, with a p-value of 0.05 and a 95% confidence interval. The comparison of p values of the suggested IBSCA with those of other mechanisms for sensitivity, F-Score, G-Mean, and AUC are shown in Table 21. Table 21 shows that the p-values are less than 0.05, indicating that the research hypothesis is accepted. In addition to the Wilcoxon Signed rank test, Friedman test [54] had been conducted to validate the performance of the proposed algorithm. Table 22 represents the result obtained from the

TABLE 19. Performance comparison of IBSCA with existing works for Ecoli-0-1-3-7_vs_2-6 Dataset.

Dataset	Ecoli-0-1-3-7_vs_2-6	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	69.43	98.71
NB-Basic [51]	50	98.13
NB-Tomek [51]	50	98.13
NB-Comm [51]	**100**	**100**
NB-Rec [51]	66.67	99.07
SA-DA [35]	66.67	70.71

TABLE 20. Performance comparison of IBSCA with existing works for Yeast6 Dataset.

Dataset	Yeast6	
Techniques	F-Score	G-Mean
IBSCA (Proposed)	**72.58**	**93.15**
NB-Basic [51]	52.17	90.97
NB-Tomek [51]	54.55	91.93
NB-Comm [51]	42.11	74.54
NB-Rec [51]	52.17	90.97
SA-DA [35]	31.03	59.41

TABLE 21. p values of Wilcoxon signed rank test.

Techniques	BSCA	BPSO	BGWO
Metric	**Sensitivity**		
IBSCA (Proposed)	0.000016	0.00032	0.00041
Metric	**F-Score**		
IBSCA (Proposed)	0.000124	0.00021	0.00076
Metric	**G-Mean**		
IBSCA (Proposed)	0.00012	0.00049	0.00091
Metric	**AUC**		
IBSCA (Proposed)	0.00034	0.00069	0.00018

TABLE 22. p-value of friedman test.

Metric	P-value
Sensitivity	9.87E-10
F-Score	3.19E-11
G-Mean	1.36E-09
AUC	5.49E-10

Friedman test. The Null and research hypotheses are given as: Null Hypothesis: No significant difference among the instances chosen by different algorithms. Research Hypothesis: Significant difference among the instances chosen by algorithms. The Friedman test is a non-parametric test, where the test statistic does not have explicit distributions. The distribution is computed using the chi-square test and the value of the statistic is found to be 44.87 for comparing the sensitivity of all Algorithms. The p-value of the Friedman test was found as 9.87e-10. The degrees of freedom are kept as 3. From the table 22 it is depicted that all p-values are much smaller than the significance level of 0.05. Thus, it is strongly evident the rejection of the null hypothesis and acceptance of the research hypothesis.

7) PERFORMANCE OF VARIOUS BASE CLASSIFIERS WITHOUT CLASS IMBALANCE

The proposed IBSCA K-NN has been compared with the other base classifiers such as K-NN, SVM, RF, and NN as shown in Table 23. For the less imbalanced Glass dataset, IBSCA K-NN improves sensitivity by 19.73% than K-NN,

TABLE 23. Comparison of proposed IBSCA K-NN with other classifiers without class imbalance.

Metrics	Sensitivity					F-Score					G-Mean					AUC				
Dataset	IBSCA K-NN	K-NN	SVM	RF	NN	IBSCA K-NN	K-NN	SVM	RF	NN	IBSCA K-NN	K-NN	SVM	RF	NN	IBSCA K-NN	K-NN	SVM	RF	NN
Glass1	**74.35**	62.1	51.9	52.29	55.79	**59.45**	43.69	37.39	38.4	40.61	**59.91**	44.02	39.61	40.28	41.74	**59.94**	44.05	40.15	40.85	43.52
Wisconsin	**96.34**	71.6	68.39	74.84	72.91	**98.3**	72.35	66.14	67.45	69.36	**97.67**	71.88	68.64	72.08	69.18	**98.01**	72.13	65.79	70.69	70.33
Pima	**92.3**	75.19	64.41	63.25	69.34	**70.3**	50.9	46.26	47.8	49.72	**75.17**	54.43	50.97	53.16	55.15	**76.88**	55.66	51.26	51.88	55.22
Iris0	**99.1**	73.52	65.41	69.12	74.15	**99.32**	71.7	64.13	67.02	70.32	**99.67**	71.95	62.64	62.75	68.05	**99.91**	72.13	64.9	67.77	71.27
Vehicle2	**82.5**	43.71	54.76	58.11	59.64	**97.35**	69.03	64.66	68.55	69.85	**83.45**	59.18	57.57	59.75	60.95	**87.65**	62.15	58.99	60.79	61.96
Ecoli1	**96.12**	68.3	71.61	71.41	71.87	**88.97**	65.1	59.35	59.58	60.23	**94.3**	69	66.39	65.21	67.61	**97.89**	71.63	67.74	68.79	69.88
Ecoli2	**93.2**	67.74	63.94	65.78	66.26	**94.45**	69.53	62.37	62.54	63.4	**97.1**	71.48	62.18	64.03	67.02	**97.95**	72.1	69.07	67.74	68.62
Ecoli3	**94.35**	70.18	66.64	63.03	67.98	**75**	53.6	52.8	51.59	54.03	**95.67**	68.37	62.42	61.4	62.63	**95.71**	68.4	66.93	65.91	68.5
Glass2	**53.4**	32.89	36.13	36.42	36.66	**52.22**	37.68	34.57	35.16	35.12	**79.13**	57.1	48	48.39	54.62	**82.34**	59.41	58.13	58.81	58.52
Yeast-$1_v s_7$	**84.44**	69.74	63.24	64.53	62.54	**69.21**	49.92	42.06	39	39.42	**79.84**	57.59	57.96	57.11	55.42	**82.56**	59.55	58.5	57.63	58.98
Glass4	**93.21**	62.41	62.85	65.25	66.97	**81.07**	59.28	55	54.89	56.82	**84.32**	61.65	57.97	58.24	61.48	**78.1**	57.11	53.72	55.7	56.38
Ecoli4	**95.87**	53.3	66.21	69.81	75.41	**76.89**	57.15	50.74	52.62	56.68	**99.51**	73.97	67.61	69.73	74.04	**94.55**	70.28	65	65.34	69.19
Page-blocks-1-$3_v s_2$	**96.14**	56.45	68.3	64.26	74.5	**98.15**	71.25	7.2	64.42	72.43	**99.62**	72.32	59.72	65.48	69.92	**94.92**	68.91	66.7	65.98	69.52
Abalone9-18	**76.01**	61.83	53.79	53.93	54.72	**61.3**	44.76	39.54	40.13	40.15	**70.13**	51.21	49.96	46.13	48.03	**94.54**	69.03	66.46	65.61	67.07
Glass5	**53.14**	22.3	36.43	37.9	37.06	**64.25**	46.31	43.4	43.67	43.71	**97.37**	70.19	45.76	53.14	60.39	**89.86**	64.78	62.54	63.61	62.33
Yeast4	**89.62**	74.3	63.96	63.47	68.78	**44.76**	32.83	28.48	29.9	31.38	**52.61**	38.59	37.74	36.01	43.53	**82.16**	60.26	57.01	58.02	58.93
Ecoli-0-1-3-$7_v s_2 - 6$	**95.14**	75.67	64.73	71.59	68.37	**69.43**	51.53	45.03	47.3	47.89	**98.71**	73.25	61.97	64.72	68.6	**94.87**	70.4	64.91	68.49	66.37
Yeast6	**87.05**	67.82	64.07	60.95	68.43	**72.58**	51.51	47.39	47.72	51.91	**93.15**	66.11	61.79	58.48	68.03	**98.93**	70.21	67.35	65.94	69.69

TABLE 24. Comparison of IBSCA with classifiers K-NN, SVM, RF, NN.

Metrics	Sensitivity				F-Score				G-Mean				AUC			
Optimization Algorithm	IBSCA				IBSCA				IBSCA				IBSCA			
Dataset	K-NN	SVM	RF	NN	K-NN	SVM	RF	NN	K-NN	SVM	RF	NN	K-NN	SVM	RF	NN
Glass1	**74.35**	71.23	72.35	73.88	**59.45**	52.34	53.98	55.81	**59.91**	55.45	56.62	57.37	**59.94**	56.21	57.43	59.81
Wisconsin	**96.34**	92.13	93.78	94.87	**98.3**	91.71	90.05	94.47	**97.67**	95.18	96.23	94.22	**98.01**	91.23	94.37	95.78
Pima	**92.3**	87.13	86.14	89.71	**70.3**	64.29	66.65	67.51	**75.17**	70.83	74.12	74.89	**76.88**	71.23	72.34	74.98
Iris0	**99.1**	92.65	93.43	95.82	**99.32**	91.23	93.09	95.43	**99.67**	89.11	87.16	92.34	**99.91**	92.32	94.14	96.71
Vehicle2	**82.5**	76.31	77.15	79.18	**97.35**	91.23	94.35	96.13	**83.45**	81.22	82.23	83.89	**87.65**	83.23	83.67	85.28
Ecoli1	**96.12**	91.23	92.45	94.43	**88.97**	79.94	80.92	82.45	**94.3**	89.42	88.57	92.54	**97.89**	91.23	93.44	95.65
Ecoli2	**93.2**	87.32	89.35	91.67	**94.45**	87.09	87.09	89.12	**97.1**	86.83	89.17	94.21	**97.95**	96.45	94.34	96.45
Ecoli3	**94.35**	87.33	88.87	91.45	**75**	72.17	73.22	74.86	**95.67**	85.32	87.14	86.78	**95.71**	91.49	93.54	94.91
Glass2	**53.4**	48.21	49.05	50.14	**52.22**	47.70	48.74	49.08	**79.13**	66.24	67.09	76.32	**82.34**	80.21	81.54	81.78
Yeast-$1_{v}s_{7}$	**84.44**	80.12	81.23	83.37	**69.21**	56.48	52.19	54.36	**79.84**	77.83	76.43	76.43	**82.56**	78.56	77.13	81.34
Glass4	**93.21**	88.32	89.17	90.32	**81.07**	77.93	76.67	78.84	**84.32**	82.14	81.35	85.3	**78.10**	76.12	77.81	78.23
Ecoli4	**95.87**	91.32	93.29	94.43	**76.89**	71.21	72.66	75.65	**99.51**	94.89	96.29	98.81	**94.55**	91.23	90.23	92.34
Page-blocks-1-$3_{v}s_{2}$	**96.14**	89.71	90.24	95.19	**98.15**	9.85	91.24	97.71	**99.62**	81.74	92.74	94.32	**94.92**	91.29	93.45	93.78
Abalone9-18	**76.01**	71.29	73.45	74.92	**61.30**	54.37	55.96	56.14	**70.13**	68.7	64.32	67.16	**94.54**	91.38	91.49	93.78
Glass5	**53.14**	47.43	48.12	49.35	**64.25**	59.12	58.71	60.24	**97.37**	62.34	71.45	83.23	**89.86**	85.19	85.53	85.91
Yeast4	**89.62**	84.32	83.24	89.62	**44.76**	39.05	40.89	42.77	**52.61**	51.75	49.25	59.32	**82.16**	78.17	79.34	80.32
Ecoli-0-1-3-$7_{v}s_{2}-6$	**95.14**	91.38	92.21	94.38	**69.43**	63.95	64.08	67.24	**98.71**	88.01	87.67	96.32	**94.87**	92.18	92.78	93.19
Yeast6	**87.05**	84.61	85.89	86.78	**72.58**	65.04	67.71	69.74	**93.15**	84.81	82.98	91.41	**98.93**	92.43	93.56	93.63

43.26% than SVM, 42.19% than RF, and 33.27% than NN. Similarly, for the medium imbalanced dataset Ecoli 3 dataset, the proposed IBSCA K-NN improves sensitivity by 34.44%, 41.58%, 49.69%, 38.79%than K-NN, SVM, RF, and NN respectively. Also, for the large imbalanced Yeast6 dataset, the proposed IBSCA K-NN is 28.35% 35.87%, 42.82%, and 27.21% respectively for K-NN, SVM, RF, and NN. IBSCA K-NN achieves F-Score of Ecoli-0-1-3-7_vs_2-6 is 69.43 which is 34.74% 54.19%, 46.79%, 44.98% greater than K-NN SVM, RF, NN respectively. For the Yeast4 dataset where the imbalance ratio is 28.1, the proposed IBSCA K-NN improves G-mean by 36.33%. than IBSCA K-NN. The IBSCA K-NN improves AUC by 36.95%, 42.25%, 44.09%, and 40.95% than K-NN, SVM, RF, and NN respectively for Abalone 9-18 dataset. The reason behind the improved performance is that IBSCA K-NN does a great level of exploration and exploitation by utilizing the alpha agent, beta agent, and random agent thereby finding the global solution.

8) PERFORMANCE OF IBSCA WITH VARIOUS BASE CLASSIFIERS

The proposed IBSCA with K-NN as a classifier has been validated with other classifiers like Support Vector Machines, Random Forests, and Neural Networks. It has been found from table 24 that IBSCA improves the performance of K-NN (K=1) more than other classifiers when working with an imbalanced dataset. Since the performance of 1-NN is high in most cases, it has been chosen as a classifier for validating the subset of instances chosen by IBSCA. For the low-level imbalanced Glass dataset, the performance of IBSCA is improved by 4.38% than IBSCA SVM, 2.76% than IBSCA RF, and 0.64% than IBSCA NN. For the medium imbalanced Ecoli2 dataset, IBSCA K-NN achieves 8.45%, 8.45%, and 5.98% greater F-score than SVM, RF, and NN respectively. This validates that IBSCA chooses an optimal subset of instances from the majority class and thereby improves the performance of K-NN. Also, K-NN is a non-parametric algorithm that does not require any underlying distribution of data making it well suitable for imbalanced datasets [55].

9) ANALYSIS OF IBSCA ON VARIOUS BENCHMARK FUNCTIONS

The proposed IBSCA has been evaluated by the IEEE Congress on Evolutionary Computation (CEC) benchmark functions such as the sphere function, Rosen Brock's Banana function, and Ackley Function. It has been observed from Figure 9 that the proposed IBSCA has superior performance than conventional BSCA, BPSO, and BGWO. IBSCA utilizes alpha agents, beta agents, and random agents for performing exploration and exploitation to find the global optimal solution. The addition of random agents opens the door for exploitation to conduct a fruitful search without becoming snared in a regionally optimal solution. A well-liked benchmark function to verify the optimization algorithm is the sphere function. The suggested IBSCA uses random agents, alpha agents, and beta agents to try to discover the minimal

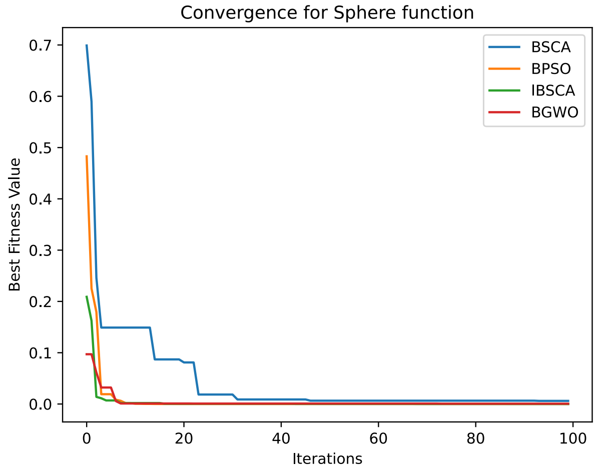

FIGURE 9. Convergence for sphere function by IBSCA, BSCA, BPSO and BGWO.

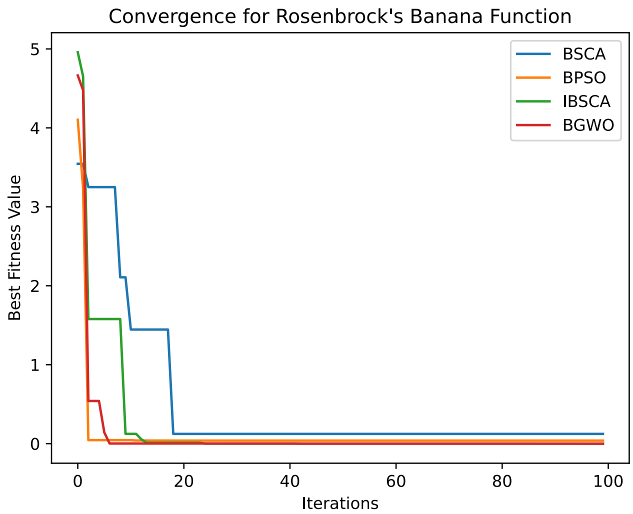

FIGURE 10. Convergence for Rosenbrock's banana function by IBSCA, BSCA, BPSO and BGWO.

value. Figure 9 illustrates how several optimization strategies used in experiments converged for the sphere function. Inferred from the figure 9 is that the minimum value of fitness for IBSCA is 9.288660e-07, but for BPSO, BSCA, and BGWO, it is 9.388671e-07, 0.006077, and 0.000641, respectively. At iteration 88, IBSCA convergence begins. For BPSO, BSCA, and BGWO, it stands at 91, 94, and 91, respectively.

Rosenbrock's banana function, a non-convex optimization procedure, is the subsequent benchmark function taken into consideration. The convergence of various optimization algorithms for Rosenbrock's banana function is depicted in Figure 10 The suggested IBSCA obtains a mean fitness of 0.212514, compared to 0.111672, 0.524871, and 0.109802 for BPSO, BSCA, and BGWO, respectively. Additionally, the BPSO has a standard deviation of 0.514481, which is lower than the IBSCA's of 0.773346, indicating that some solutions are unstable. IBSCA still outperforms BPSO in terms of minimum mean fitness, with a minimum fitness of 0.000003 compared to 0.038413.

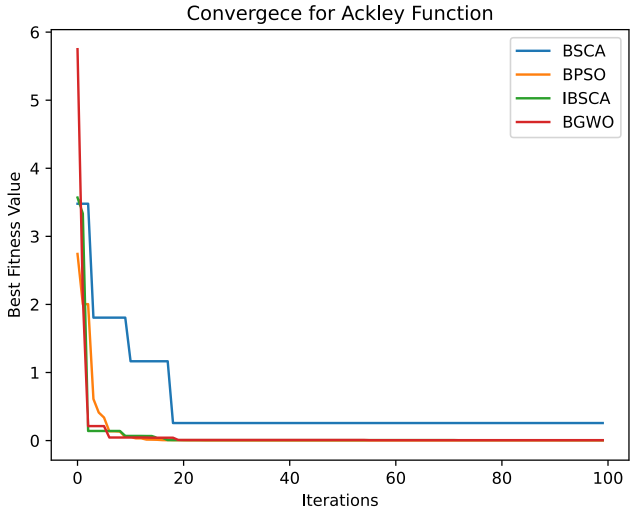

FIGURE 11. **Convergence for Ackley function by IBSCA, BSCA, BPSO and BGWO.**

Ackley function is another standard benchmark function used to evaluate the performance of the proposed IBSCA. The comparison of convergence for the Ackley function is shown in Figure 11. From Figure 11 it is evident that the proposed IBSCA method outperforms the other algorithms BSCA, BPSO, and BGWO by having the lowest mean value of 0.087480. Also, the proposed IBSCA achieves a minimum standard deviation (0.395721) which implies that the solutions are more consistent and stable than BPSO (0.484575), BSCA (0.687513), and BGWO (0.609256). The high mean value of BSCA (0.535216) suggests that on average, it achieves higher fitness than other algorithms. Thus, improvements have been made to the conventional BSCA for finding global optimal solutions.

V. CONCLUSION

It is quite typical for the real-world dataset to contain a larger proportion of instances from one class and a smaller proportion from another. Such unbalanced datasets have a significant negative impact on the classifier's performance by favoring the dominant class. Researchers have occasionally developed a variety of methods to address the issue of class imbalance. In this paper, we proposed an innovative metaheuristic algorithm called Improved Binary Sine Cosine Algorithm using 1-NN as a classifier for dealing with the problems of imbalanced datasets by selecting the instances from the majority class as optimally as possible to tackle the classifier's bias towards one class. In addition to the alpha agent, the introduction of beta and random agents tends to encourage appropriate levels of exploration and exploitation. The suggested algorithm's effectiveness is compared to other traditional metaheuristic algorithms, cutting-edge techniques, and various algorithms for dealing with imbalanced datasets. 18 datasets were used to assess the suggested algorithm. Our suggested algorithms beat the other algorithms in terms of sensitivity for 17 out of the 18 datasets. The main revelations in this article are:

- Beta agent and random agent are used to find the global optimal solution, preventing the agents from settling for the local optimal solution and accelerating convergence.
- On 18 benchmarking datasets, the proposed IBSCA is compared to other conventional algorithms, and the findings show that IBSCA is superior to other algorithms for dealing with class-imbalanced datasets.
- Accelerating the agent so that exploration occurs at the beginning and exploitation occurs at the point of convergence.

In terms of metrics like sensitivity, F-score, G-mean, and AUC, the experimental findings showed that the proposed IBSCA works better than other algorithms for handling imbalanced datasets. Additionally, the outcomes demonstrate that IBSCA performs significantly in datasets with low, middle, and high levels of imbalance. IBSCA is superior to other algorithms, according to the study hypothesis, which has a 95% confidence interval. Although the proposed IBSCA outperforms various metrics like sensitivity, F-score, G-mean, and AUC for real-world datasets, it also has some limitations that result in future enhancement. The convergence speed of the algorithm increases when dealing with complex and relatively large dimensional problems. Though the inclusion of alpha agent, beta agent, and random agent balances between exploration and exploitation, it is quite common for any metaheuristic algorithm to trap in local optima. The scalability becomes an issue when working with large datasets and a greater number of agents thereby increasing the computational complexity. To address these challenges, dimensionality reduction can be done beforehand. Also, multiple optimization algorithms can be combined to deal with complex optimization problems. To speed up the search process, the agents can be processed in parallel to find a global optimal solution.

REFERENCES

[1] G. Zhou, Z. Wang, and Q. Li, "Spatial negative co-location pattern directional mining algorithm with join-based prevalence," *Remote Sens.*, vol. 14, no. 9, p. 2103, Apr. 2022.

[2] Q. Meng, Q. Ma, and Y. Shi, "Adaptive fixed-time stabilization for a class of uncertain nonlinear systems," *IEEE Trans. Autom. Control*, vol. 68, no. 11, pp. 6929–6936, Nov. 2023.

[3] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proc. Int. Conf. Artif. Intell.*, vol. 56, 2000, pp. 111–117.

[4] A. Mahani and A. R. B. Ali, "Classification problem in imbalanced datasets," in *Recent Trends in Computational Intelligence*. London, U.K.: IntechOpen, 2019, pp. 1–23.

[5] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2018.

[6] A. Ghazikhani, H. S. Yazdi, and R. Monsefi, "Class imbalance handling using wrapper-based random oversampling," in *Proc. 20th Iranian Conf. Electr. Eng. (ICEE)*, May 2012, pp. 611–616.

[7] R. S. Moorthy, K. Arikumar, and B. S. B. Prathiba, "An improved whale optimization algorithm for optimal placement of edge server," in *Advances in Distributed Computing and Machine Learning*. Cham, Switzerland: Springer, 2023, pp. 89–100.

[8] B. S. Raghuwanshi and S. Shukla, "SMOTE based class-specific extreme learning machine for imbalanced learning," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104814.

[9] P. J. Huang, *Classification of Imbalanced Data Using Synthetic Oversampling Techniques*. Los Angeles, CA, USA: Univ. California, 2015.

[10] L. Gong, S. Jiang, and L. Jiang, ''Tackling class imbalance problem in software defect prediction through cluster-based over-sampling with filtering,'' *IEEE Access*, vol. 7, pp. 145725–145737, 2019.
[11] K. S. Arikumar, S. B. Prathiba, M. Alazab, T. R. Gadekallu, S. Pandya, J. M. Khan, and R. S. Moorthy, ''FL-PMI: Federated learning-based person movement identification through wearable devices in smart healthcare systems,'' *Sensors*, vol. 22, no. 4, p. 1377, Feb. 2022.
[12] K. S. Arikumar, A. D. Kumar, T. R. Gadekallu, S. B. Prathiba, and K. Tamilarasi, ''Real-time 3D object detection and classification in autonomous driving environment using 3D LiDAR and camera sensors,'' *Electronics*, vol. 11, no. 24, p. 4203, Dec. 2022.
[13] S.-J. Yen and Y.-S. Lee, ''Cluster-based under-sampling approaches for imbalanced data distributions,'' *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5718–5727, Apr. 2009.
[14] K. S. Babu and Y. N. Rao, ''A study on imbalanced data classification for various applications,'' *Revue d'Intell. Artificielle*, vol. 37, no. 2, pp. 517–524, Apr. 2023.
[15] M. M. Rahman and D. N. Davis, ''Addressing the class imbalance problem in medical datasets,'' *Int. J. Mach. Learn. Comput.*, vol. 3, no. 2, pp. 224–228, 2013.
[16] S. S. Shaw, S. Ahmed, S. Malakar, L. Garcia-Hernandez, A. Abraham, and R. Sarkar, ''Hybridization of ring theory-based evolutionary algorithm and particle swarm optimization to solve class imbalance problem,'' *Complex Intell. Syst.*, vol. 7, no. 4, pp. 2069–2091, Aug. 2021.
[17] O. M. Alyasiri, Y.-N. Cheah, and A. K. Abasi, ''Hybrid filter-wrapper text feature selection technique for text classification,'' in *Proc. Int. Conf. Commun. Inf. Technol. (ICICT)*, Jun. 2021, pp. 80–86.
[18] O. M. Alyasiri, Y.-N. Cheah, A. K. Abasi, and O. M. Al-Janabi, ''Wrapper and hybrid feature selection methods using Metaheuristic algorithms for English text classification: A systematic review,'' *IEEE Access*, vol. 10, pp. 39833–39852, 2022.
[19] Y. Luo, W. Dai, and Y.-W. Ti, ''Improved sine algorithm for global optimization,'' *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 118831.
[20] K. S. Arikumar, S. B. Prathiba, R. S. Moorthy, K. Tamilarasi, M. M. V. Chalapathi, and A. D. Kumar, ''The role of machine learning in IoT: A survey,'' in *Proc. 3rd Int. Conf. Smart Electron. Commun. (ICOSEC)*, Oct. 2022, pp. 451–457.
[21] A. Gosain and S. Sardana, ''Handling class imbalance problem using oversampling techniques: A review,'' in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 79–85.
[22] P. Bedi, N. Gupta, and V. Jindal, ''Siam-IDS: Handling class imbalance problem in intrusion detection systems using Siamese neural network,'' *Proc. Comput. Sci.*, vol. 171, pp. 780–789, Jan. 2020.
[23] B. S. Raghuwanshi and S. Shukla, ''Class-specific extreme learning machine for handling binary class imbalance problem,'' *Neural Netw.*, vol. 105, pp. 206–217, Sep. 2018.
[24] D. Elreedy and A. F. Atiya, ''A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance,'' *Inf. Sci.*, vol. 505, pp. 32–64, Dec. 2019.
[25] A. Estabrooks, T. Jo, and N. Japkowicz, ''A multiple resampling method for learning from imbalanced data sets,'' *Comput. Intell.*, vol. 20, no. 1, pp. 18–36, Feb. 2004.
[26] L. I. Kuncheva, Á. Arnaiz-González, J.-F. Díez-Pastor, and I. A. D. Gunn, ''Instance selection improves geometric mean accuracy: A study on imbalanced data classification,'' *Prog. Artif. Intell.*, vol. 8, no. 2, pp. 215–228, Jun. 2019.
[27] Y. Sun, L. Cai, B. Liao, W. Zhu, and J. Xu, ''A robust oversampling approach for class imbalance problem with small disjuncts,'' *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5550–5562, Jun. 2023.
[28] S. Lu, Z. Gao, Q. Xu, C. Jiang, A. Zhang, and X. Wang, ''Class-imbalance privacy-preserving federated learning for decentralized fault diagnosis with biometric authentication,'' *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 9101–9111, Dec. 2022.
[29] Z. Chen, C. Yang, M. Zhu, Z. Peng, and Y. Yuan, ''Personalized retrogress-resilient federated learning toward imbalanced medical data,'' *IEEE Trans. Med. Imag.*, vol. 41, no. 12, pp. 3663–3674, Dec. 2022.
[30] S. García and F. Herrera, ''Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy,'' *Evol. Comput.*, vol. 17, no. 3, pp. 275–306, Sep. 2009.
[31] C. X. Ling and C. Li, ''Data mining for direct marketing: Problems and solutions,'' in *Proc. KDD*, vol. 98, 1998, pp. 73–79.
[32] J. Prusa, T. M. Khoshgoftaar, D. J. Dittman, and A. Napolitano, ''Using random undersampling to alleviate class imbalance on tweet sentiment data,'' in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Aug. 2015, pp. 197–202.
[33] J. Li, Y. Wu, S. Fong, A. J. Tallón-Ballesteros, X.-S. Yang, S. Mohammed, and F. Wu, ''A binary PSO-based ensemble under-sampling model for rebalancing imbalanced training data,'' *J. Supercomput.*, vol. 78, no. 5, pp. 7428–7463, Apr. 2022.
[34] I. Al-Badarneh, M. Habib, I. Aljarah, and H. Faris, ''Neuro-evolutionary models for imbalanced classification problems,'' *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2787–2797, Jun. 2022.
[35] A. S. Desuky and S. Hussain, ''An improved hybrid approach for handling class imbalance problem,'' *Arabian J. Eng.*, vol. 46, no. 4, pp. 3853–3864, Apr. 2021.
[36] F. Cheng, F. Chu, and L. Zhang, ''A multi-objective evolutionary algorithm based on length reduction for large-scale instance selection,'' *Inf. Sci.*, vol. 576, pp. 105–121, Oct. 2021.
[37] C.-C. Lin, J.-R. Kang, Y.-L. Liang, and C.-C. Kuo, ''Simultaneous feature and instance selection in big noisy data using memetic variable neighborhood search,'' *Appl. Soft Comput.*, vol. 112, Nov. 2021, Art. no. 107855.
[38] M. Kordos, M. Blachnik, and R. Scherer, ''Fuzzy clustering decomposition of genetic algorithm-based instance selection for regression problems,'' *Inf. Sci.*, vol. 587, pp. 23–40, Mar. 2022.
[39] C.-F. Tsai, W.-C. Lin, Y.-H. Hu, and G.-T. Yao, ''Under-sampling class imbalanced datasets by combining clustering analysis and instance selection,'' *Inf. Sci.*, vol. 477, pp. 47–54, 2019.
[40] A. Lorenc and A. Burinskiene, ''Improve the orders picking in e-commerce by using WMS data and BigData analysis,'' *FME Trans.*, vol. 49, no. 1, pp. 233–243, 2021.
[41] P. Mouawad, T. Dubnov, and S. Dubnov, ''Robust detection of COVID-19 in cough sounds: Using recurrence dynamics and variable Markov model,'' *Social Netw. Comput. Sci.*, vol. 2, no. 1, p. 34, Feb. 2021.
[42] H. Patel, D. S. Rajput, G. T. Reddy, C. Iwendi, A. Kashif Bashir, and O. Jo, ''A review on classification of imbalanced data for wireless sensor networks,'' *Int. J. Distrib. Sensor Netw.*, vol. 16, no. 4, Apr. 2020, Art. no. 155014772091640.
[43] S. Bhattacharya, P. K. R. Maddikunta, S. Hakak, W. Z. Khan, A. K. Bashir, A. Jolfaei, and U. Tariq, ''Antlion re-sampling based deep neural network model for classification of imbalanced multimodal stroke dataset,'' *Multimedia Tools Appl.*, vol. 81, no. 29, pp. 41429–41453, Dec. 2022.
[44] N. K. AlShammari, E. U. H. Qazi, A. M. Gabr, A. A. Alzamil, A. S. Alshammari, S. M. Albadran, and G. T. Reddy, ''Post-stroke rehabilitation optimization & recommendation framework using tele-robotic ecosystem: Industry 4.0 readiness approach,'' *J. Intell. Fuzzy Syst.*, vol. 44, no. 5, pp. 8773–8783, Jan. 2023.
[45] W. Zheng and L. Yin, ''Characterization inference based on joint-optimization of multi-layer semantics and deep fusion matching network,'' *PeerJ Comput. Sci.*, vol. 8, p. e908, Apr. 2022.
[46] S. Mirjalili and A. Lewis, ''S-shaped versus V-shaped transfer functions for binary particle swarm optimization,'' *Swarm Evol. Comput.*, vol. 9, pp. 1–14, Apr. 2013.
[47] I. Mani and I. Zhang, ''kNN approach to unbalanced data distributions: A case study involving information extraction,'' in *Proc. Workshop Learn. Imbalanced Datasets*, vol. 126, 2003, pp. 1–7.
[48] W. Shang, H. Huang, H. Zhu, Y. Lin, Z. Wang, and Y. Qu, ''An improved kNN algorithm–fuzzy kNN,'' in *Proc. Int. Conf. Comput. Inf. Sci.*, Xi'an, China. Cham, Switzerland: Springer, Dec. 2005, pp. 741–746.
[49] R. S. Moorthy and P. Pabitha, ''Optimal detection of phising attack using SCA based K-NN,'' *Proc. Comput. Sci.*, vol. 171, pp. 1716–1725, Jan. 2020.
[50] J. Derrac, S. Garcia, L. Sanchez, and F. Herrera, ''Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework,'' *J. Mult. Valued Log. Soft Comput*, vol. 17, pp. 255–287, Jun. 2015.
[51] P. Vuttipittayamongkol and E. Elyan, ''Neighbourhood-based undersampling approach for handling imbalanced and overlapped data,'' *Inf. Sci.*, vol. 509, pp. 47–70, Jan. 2020.
[52] K.-F. Wang, J. An, Z. Wei, C. Cui, X.-H. Ma, C. Ma, and H.-Q. Bao, ''Deep learning-based imbalanced classification with fuzzy support vector machine,'' *Frontiers Bioeng. Biotechnol.*, vol. 9, p. 1448, Jan. 2022.
[53] S. Wang, Y. Dai, J. Shen, and J. Xuan, ''Research on expansion and classification of imbalanced data based on SMOTE algorithm,'' *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, Dec. 2021.

[54] D. W. Zimmerman and B. D. Zumbo, "Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks," *J. Experim. Educ.*, vol. 62, no. 1, pp. 75–86, Jul. 1993.
[55] Z. Shi, "Improving k-Nearest neighbors algorithm for imbalanced data classification," *IOP Conf. Series: Mater. Sci. Eng.*, vol. 719, no. 1, Jan. 2020, Art. no. 012072.

RAJALAKSHMI SHENBAGA MOORTHY received the B.Tech. degree (Hons.) in information technology, the M.E. degree (Hons.) in computer science and engineering, and the Ph.D. degree from Anna University, in 2010, 2013, and 2021, respectively. She is currently an Assistant Professor with the Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research. She has published several book chapters and a book on programming and data structures. She has published various research articles indexed in IEEE, Elsevier, and Springer. Her research interests include analysis of algorithms, machine learning, deep learning, and metaheuristic algorithms. She was awarded with the Chancellor Shri. V. R. Venkataachalam Award for the Best Research Publication.

ARIKUMAR K. SELVARAJ received the bachelor's, master's, and Ph.D. degrees in computer science and engineering from Anna University, Chennai. He is currently an Assistant Professor with the Department of Data Science and Business Systems, SRM Institute of Science and Technology (SRMIST), Kattankulathur. He has 12 years of experience. He has published 25 conference papers and journal articles. His current research interests include wireless sensor networks, machine learning, software-defined networking, autonomous vehicular networks, and the IoT.

SAHAYA BENI PRATHIBA (Member, IEEE) received the bachelor's and master's degrees in computer science and engineering and the Ph.D. degree from Anna University, Chennai, in 2022. She has secured 23rd rank among 2581 candidates in Master of Engineering. She is currently an Assistant Professor with the Centre for Cyber Physical Systems, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai. Her research interests include 5G/6G, vehicle-to-everything, software-defined networking, autonomous vehicular networks, Industry 5.0, and metaverse. She was a recipient of the Anna Centenary Research Fellowship. She is serving as a reviewer for IEEE TRANSACTION and reputed Elsevier journals.

GOKUL YENDURI received the M.Tech. and Ph.D. degrees in IT from Vellore Institute of Technology, in 2013. He is currently an Assistant Professor with VIT-AP University. He was a Senior Research Fellow with the DIVERSASIA Project, co-funded by the Erasmus+ Program of European Union in the past. He has attended several national and international conferences, workshops, and guest lectures. He has published articles in peer-reviewed international journals. His research interests include machine learning and predictive analysis, software engineering, assistive technologies, and metaverse. He is acting as a reviewer of many prestigious peer-reviewed international journals.

SACHI NANDAN MOHANTY (Senior Member, IEEE) received the Ph.D. degree from Indian Institute of Technology Kharagpur, India, in 2015, with an MHRD Scholarship from the Government of India, and the Ph.D. degree from Indian Institute of Technology Kharagpur, in 2019. He has guided nine Ph.D. Scholar and 23 postgraduate students. He has published 120 international journals of international repute. He has authored/edited 42 books, published by IEEE-Wiley, Springer, Wiley, CRC Press, NOVA, and DeGruyter. His research interests include data mining, big data analysis, cognitive science, fuzzy decision-making, brain–computer interface, cognition, and computational intelligence. He was elected as a fellow of the Institute of Engineers, Ambassador to European Alliance Innovation (EAI) Springer, and a Senior Member of the IEEE Computer Society Hyderabad Chapter. He is a member of professional bodies, such as IETE, CSI, and IE. He has received four best paper awards during the Ph.D. studies from the International Conference, Beijing, China, and the other International Conference on Soft Computing Applications organized by IIT Rookee, in 2013. He was a recipient of the Best Thesis Award from the Computer Society of India, in 2015, and many international and national awards from the Government of India, Department of Science and Technology, SERB, funding for international travel for presenting research papers. He has been awarded many awards and fellowships during his career and to name, including the Prof. Ganesh Mishra Memorial Award at the 61st Annual Technical Session and the 19th Prof. Bhubaneswar Behera Lecturer, in June 2020, from The Institution of Engineers (India) Odisha State Center, Bhubaneswar. He was a Reviewer of *Robotics and Autonomous Systems* (Elsevier), *Computational and Structural Biotechnology Journal* (Elsevier), *Artificial Intelligence Review* (Springer), and *Spatial Information Research* (Springer). He is leading as the General Chair of two international conferences, such as ICISML and AIHC, and the Editor-in-Chief of the *International Journal Intelligent Systems* and *Machine Learning with Applications*. He has gone on academic assignments to USA, Paris, Slovakia, Singapore, Abu Dhabi, Dubai, Malaysia, and Germany, to deliver keynote talks and chair sessions in international conferences with travel support from the Department of Science and Technology, Government of India, New Delhi, India.

JANJHYAM VENKATA NAGA RAMESH is currently an Assistant Professor with the Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India. He has 20 years of experience in teaching for UG and PG engineering students. He has published more than 25 papers in IEEE/SCI/Scopus/WoS journals and conferences. He has authored six textbooks and ten book chapters. His research interests include wireless sensor networks, computer networks, deep learning, machine learning, and artificial intelligence. He is a reviewer of various leading journals.