Journal of Computer and Knowledge Engineering, Vol. , No.. 2023.

DOI:

A Difficulty-aware Approach to Fair Classification on Imbalanced Datasets*

Niloufar Kashefi[1], Javad Hamidzadeh[2], and Mona Moradi[3]

Abstract**-- Class imbalance in real-world datasets often biases standard classifiers toward the majority class, degrading performance on the minority class. While existing methods like sample re-weighting can mitigate this, they may increase overall misclassification errors or fail to consider the difficulty of training instances. To address these shortcomings, we introduce a difficulty-aware classification framework based on multi-objective evolutionary optimization. Our approach uses a specialized fitness function to simultaneously optimize for minority-class recall and overall accuracy, guiding the selection of the most informative training samples. We quantify sample difficulty using a fuzzy approach, which then modulate class-specific weights to refine the classifier's decision boundary. Furthermore, we incorporate chaotic dynamic maps into the evolutionary operators to accelerate convergence and maintain population diversity. Evaluated on various UCI benchmark datasets with 10-fold cross-validation, our method improves minority-class performance on imbalanced data without compromising accuracy on balanced data. Comparative analysis using AUC, G-mean, and F-measure confirms our approach achieves a superior trade-off between minority-class detection and overall accuracy compared to state-of-the-art methods.**

Index Terms**-- Instance reduction; Fuzzy weighted average distance-based decision surface; Chaotic imperialist competitive algorithm; Reduction rate.**

INTRODUCTION

Class imbalance is a pervasive challenge in machine learning, significantly impairing model performance across diverse domains such as medical diagnosis [1], streaming data mining [2], fraud detection [3], natural language processing [4], and image analysis [5]. In these applications, the training data contains a skewed class distribution, which biases classifiers toward the majority class. As a result, instances from the minority class—which often carry the most critical information—are frequently misclassified. Existing solutions to mitigate class imbalance can be broadly categorized into three paradigms:

1. Data-level approaches: These techniques modify the training data distribution to create a more balanced class representation. The primary methods are oversampling and undersampling. Oversampling methods increase the number of minority class instances, ranging from simple duplication to synthetic generation of new samples with techniques like SMOTE. Conversely, undersampling methods reduce the size of the majority class by discarding redundant or noisy samples. However, these risks losing valuable information.

The core challenge in these approaches lies in sample selection. Effective methods must distinguish informative instances from noisy or redundant ones to avoid overfitting while preserving data diversity. To achieve this, neighborhood-based algorithms like *k*-Nearest Neighbors (*k*-NN) can identify redundant samples based on feature space proximity. In contrast, clustering-based methods (e.g., *k*-means) can sample representative centroids. More advanced techniques, such as the adaptive clustering proposed in [6], reduce computational overhead by dynamically tuning the cluster count and using alternative distance metrics for sample selection.

2. Algorithm-level approaches: These approaches modify the learning algorithm without changing the distribution of the training data. This is often achieved through cost-sensitive learning [7, 8], where higher misclassification costs are assigned to minority class instances, directly embedding class-specific penalties into the objective function. While effective at improving minority recall, this can increase model complexity. Other algorithmic strategies include one-class classification and modifications to specific algorithms like Support Vector Machines (SVMs) [12–14], though these can require specialized tuning.

3. Hybrid Approaches: This category combines data-level and algorithm-level techniques to leverage the strengths of both. The goal is to create a more robust solution where a modified data distribution is fed into a specially adapted learning algorithm. These methods are often highly effective because they address the problem from two angles simultaneously. Popular examples include ensemble methods combined with sampling, such as SMOTEBoost [15], which integrates the SMOTE oversampling technique into the AdaBoost framework, and RUSBoost [11], which combines random undersampling with Boosting. By synergistically applying both strategies, hybrid methods can often achieve superior performance, particularly in cases of severe imbalance where a single approach is insufficient.

Over the past decade, research on class imbalance has advanced significantly. The focus has broadened from binary classification to multi-class long-tail (skewed) distributions, spurring the development of hybrid augmentation methods like GAN-based synthetic sample generation [15]. In parallel, uncertainty-aware frameworks, such as Bayesian inference [16–18] and fuzzy logic [19], have been integrated to refine

* Manuscript received Revised, , accepted .

[1] Faculty of computer engineering and information technology, Sadjad University, Mashhad, Iran, Email: N.Kashefi@sadjad.ac.ir

[2] Corresponding author. Faculty of computer engineering and information technology, Sadjad University, Mashhad, Iran, Email: J_Hamidzadeh@sadjad.ac.ir

[3] Faculty of computer engineering and information technology, Sadjad University, Mashhad, Iran, Email: Mona_Moradi@sadjad.ac.ir

both data augmentation and hyperparameter optimization [20, 21]. These advancements culminate in state-of-the-art strategies like evolutionary bagging with data augmentation [22], which aim to enhance model fairness and robustness.

Despite this progress, a pervasive limitation persists in many rebalancing techniques: they either indiscriminately remove majority instances or apply static, higher weights to minority classes. Such strategies often overlook the crucial concept of *instance difficulty*, which affects instances in all classes, including the majority. Instance difficulty refers to the inherent difficulty of a sample that can negatively affect a model's learning process. This difficulty can arise from several factors:

Noise: Irrelevant data within the sample.

Ambiguity: The sample's features make it hard to clearly assign to one class.

Proximity to a decision boundary: The sample is located very close to the line or surface that separates different classes, making it a challenging case for the classifier.

Neglecting these difficult instances can compromise classifier performance and discard information critical for robust decision-making.

To bridge this gap, we propose a difficulty-aware decision surface derived via a chaotic-evolutionary optimization process. This decision surface integrates the designed instance difficulty metric to simultaneously perform instance selection and adaptive weighting, enabling nuanced rebalancing sensitive to sample difficulty. The main contributions of the paper are:

- A hybrid rebalancing strategy that combines subset selection and fuzzy-based weight modulation, providing a more granular and context-sensitive mechanism than traditional methods.

- Explicit consideration of sample-level difficulty—including ambiguous, noisy, or boundary samples—within the decision-making framework.

The remainder of this paper is organized as follows: Section 2 reviews relevant literature; Section 3 outlines theoretical backgrounds; Section 4 details the proposed method; Section 5 presents the experimental evaluation; and Section 6 concludes with insights and potential future directions.

RELATED WORK

This section reviews key literature related to the proposed methodology, structured into two subsections: Imbalance Ratio and Overlapping as problem, and Kernel-based Density Estimation Strategy as solution.

A. *Problems: Imbalance Ratio and Overlapping*

Imbalanced datasets are characterized by a significantly higher count of majority-class instances relative to minority-class instances. The Imbalance Ratio (IR), defined in Equation (1), is widely used to quantify this skew.

$$IR = \frac{\text{size of the majority class}}{\text{size of the minority class}} \tag{1}$$

In multi-label settings, IR must be computed per label since individual labels may exhibit distinct positive-negative class distributions; moreover, instances may simultaneously belong to both majority and minority labels, complicating imbalance assessment [23].

TABLE I summarizes different types of problems that may arise when encountering class imbalance.

TABLE I
Different Types of Problems in Class Imbalance

Problem	**Description**	**Consequence**	**Ref.**
Class Overlap	Regions in feature space where instances of multiple classes coexist.	Degraded classifier performance, even in balanced datasets; misclassification at class boundaries.	[24-27]
Label Noise	Incorrectly labeled instances in the dataset.	Complicates training; can lead to poor generalization.	[26, 5]
Aggressive Reduction	Indiscriminate removal of majority instances by rebalancing methods.	Undermining of model fairness and stability; creation of new imbalances.	Proposed method

B. *A good solution: Kernel-based Density estimation strategy*

Comprehending the local sample density is pivotal in addressing both class imbalance and overlap. By modeling separate probability density functions (PDFs) for majority and minority classes, the density ratio indicates how far a sample lies from the overlapping zone: values near unity suggest equal class density, whereas higher ratios denote proximity to minority high-density regions. Kernel Density Estimation (KDE) has been adopted to estimate these PDFs and guide density-aware sampling. KDE-based sampling can outperform conventional oversampling methods by generating synthetic instances in dense minority-class regions, reducing overfitting and improving metrics such as F-score and G-mean, even under severe IR conditions. While density-based under-sampling methods offer effective mitigation of class imbalance, they frequently falter in scenarios involving nonlinearly separable class distributions. To address this limitation, one can employ nonlinear mappings—such as kernel transformations or feature-space embeddings—that project the original input data into a high-dimensional feature space, thereby converting inherently nonlinear decision boundaries into linearly separable representations suitable for density estimation and sampling [25, 28]. Several recent KDE-related methodologies are introduced in [29-38]. To facilitate a comparative understanding of existing KDE-based approaches and their limitations, Table II provides a concise summary of representative methods recently proposed in the literature. This comparison helps to clarify the conceptual differences and performance gaps that our proposed difficulty-aware framework aims to address.

TABLE II
Overview of Recent KDE-Based Methods for Imbalanced Data Classification

Methodology	**Key Contribution**	**Limitation**

Adaptive KDE [29]	Adjusts bandwidths based on local sparsity, preserving informative instances.	Lacks sensitivity to sample hardness and local ambiguity.
Localized Density Ratio Sampling (LDRS) [30]	Identifies overlap-prone regions and applies selective resampling.	Uniformly treats boundary instances, disregarding instance-level difficulty.
Density-Aware SMOTE (DA-SMOTE) [31]	Restricts synthetic sample generation to high-density areas.	Lacks adaptive mechanisms for instance difficulty or class overlap beyond density.
KDE with Graph Embedding [32]	Uses graph-based embeddings for nonlinear-aware sampling.	Does not incorporate instance-level selection or adaptive weighting based on difficulty.
KMM-HR [33]	Aligns inter-class distributions and reweights boundary/noisy instances.	Its nature limits fine-grained control over local structures.

In addition, several complementary strategies—including rebalancing [39], ensemble learning [40], transfer learning [41], and architecture decoupling [42]—have been introduced to tackle the previously outlined challenges from different methodological perspectives.

Another challenge arises where the number of samples in different categories follows a long-tailed distribution. Recent research can be broadly categorized into the following four groups, as shown in TABLE III:

TABLE III
Representative Studies Addressing Long-Tailed Distributions

Category	Strategy	Advantages	Disadvantages
Resampling & Reweighting [43-45]	Adjusts class distributions or assigns weights to instances.	Foundational; can highlight challenging samples.	Risks overfitting minority classes and underfitting majority classes.
Transfer Learning [46]	Leverages knowledge from data-rich (head) classes.	Improves performance on data-scarce (tail) classes.	Challenging to construct effective transfer modules and manage training dynamics.
Two-stage Training [47]	Decouples the feature extractor from the classifier.	Learns robust feature representations.	Can interfere with feature learning.
Grouping & Ensemble Methods [48-50]	Divides data into subsets for independent model training and aggregates outputs.	Preserves head-class performance; reduces prediction variance and bias.	Can be complex to implement and coordinate multiple models.

Background: Imperialist Competitive Algorithm

Imperialist Competitive Algorithm (ICA) [51] is a population-based metaheuristic within the evolutionary optimization paradigm, where candidate solutions are modeled as *countries*. The algorithm is structured into the following stages:

(A) Initialization and role assignment: Initially, a population of countries is generated randomly in the solution space. Each country's quality (or power) is evaluated using a cost (fitness) function. Based on these cost values, countries are divided into two groups: imperialists (the stronger countries with lower cost values) and colonies (the weaker countries). The role assignment of colonies to imperialists are based on their relative powers: the better countries become imperialists and own a number of colonies proportional to their strength. The cost value (fitness) of each country is computed by **Error! Reference source not found.**.

$$C_n = f_{\text{cost}}^{(imp,n)} - \max_i \left(f_{\text{cost}}^{(imp,i)} \right) \tag{2}$$

where $f_{\cos t}^{(imp,n)}$ is the cost of n^{th} imperialist and C_n is its normalized cost. The normalized power of each imperialist is evaluated by **Error! Reference source not found.**.

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \tag{3}$$

where P_n is the normalized power value of n^{th} imperialist. The imperialists are the more powerful countries (have higher values for P_i variables). To evaluate the number of colonies of an empire, **Error! Reference source not found.** must be used.

$$NC_n = Round\left\{P_n . N_{col}\right\} \tag{4}$$

where NC_n is the number of initial colonies of n^{th} imperialist which is chosen from empiricists randomly. Besides, *Round* $\{.\}$ assigns an integer value to NC_n.

(B) Assimilation mechanism and position update: In ICA, a colony moves toward the imperialist by a random value that is between 0 and $\beta \times d$. The new location of colonies is evaluated by **Error! Reference source not found.**.

$$\{z\}_{new} = \{z\}_{old} + U\left(0, \beta \times d\right) \times \{V_1\} \tag{5}$$

where β is an adjustable parameter and d is the distance between colony and imperialist. $\{V_1\}$ is a vector which its start point is $\{z\}_{old}$ and its direction is toward $\{z\}_{\text{new}}$. The length of this vector is set to unity [42]. One way to increase the searching around the imperialist is by adding a random amount of deviation θ to the direction of movement. The moving model is shown in Fig. 1.

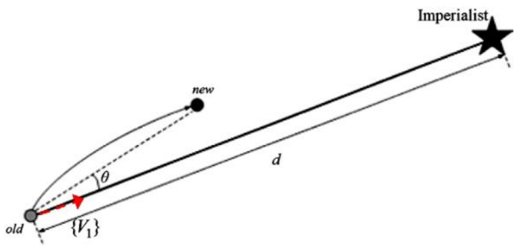

Fig. 1. Movement of colonies to its new location in the original ICA.

(C) Leadership transition based on fitness superiority: In the course of the optimization process, a colony may discover a solution with a lower cost (i.e., higher fitness) than that of its associated imperialist. When such an improvement occurs, a position exchange mechanism is triggered: the colony and the imperialist swap roles, and the colony is promoted to the status of imperialist by occupying the superior position in the search space. This dynamic update strategy ensures that the search process is continuously guided by the most promising candidates and facilitates convergence toward optimal or near-optimal solutions.

(D) Imperialistic competition and empire collapse mechanism: During the evolution of the ICA, less dominant imperialists gradually lose control over their colonies due to their relatively inferior fitness. In contrast, more powerful imperialists competitively absorb these orphaned colonies, thereby expanding their influence. When an empire loses all of its colonies, it is considered collapsed and is eliminated from the search process. A probabilistic mechanism governs the acquisition of the collapsed empire's remaining resources, allowing competing empires to assimilate its remnants. The total cost of the *n*th empire is evaluated by **Error! Reference source not found.**.

$$TC_n = f_{\cos t}^{(imp,n)} + \xi \frac{\sum_{i=1}^{NC_n} f_{\cos t}^{(col,i)}}{NC_n} \tag{6}$$

where ξ is an adjustable positive number [52].

(E) Convergence Criterion and Termination Condition: The iterative process of the Imperialist Competitive Algorithm continues until a single empire successfully dominates all others by assimilating every remaining colony within the population. At this convergence point, the prevailing empire is identified as the optimal solution or instance selection pattern. If such a global domination is not achieved, the algorithm returns to step (B), and the competition among empires resumes, ensuring ongoing exploration and exploitation within the solution space.

Talatahari, et al. [47] presented an improved ICA that uses a point out of vector which its start point is $\{z\}_{old}$ and its direction is toward $\{z\}_{new}$, as indicated in Fig. 2. This improved algorithm is generated by changing **Error! Reference source not found.** into **Error! Reference source not found.**.

$$\begin{aligned} \{Z\}_{new} = \{Z\}_{old} &+ \beta \times d \times \{CM\} \otimes \{V_1\} \\ &+ CM \times \tan(\theta) \times d \times \{V_2\}, \\ &\{V_1\}.\{V_2\} = 0, \\ &\|\{V_2\}\| = 1 \end{aligned} \tag{7}$$

where CM and θ are the random parameters and $\{V_2\}$ is perpendicular to $\{V_1\}$.

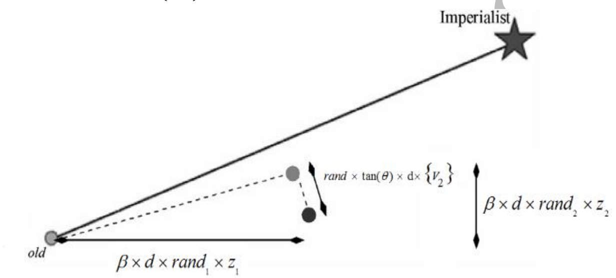
Imperialist
$rand \times \tan(\theta) \times d \times \{V_2\}$
$\beta \times d \times rand_2 \times z_2$
old
$\beta \times d \times rand_1 \times z_1$

Fig. 2. Movement of colonies to their new location in the improved ICA.

PROPOSED METHOD

To address the class imbalance problem in binary classification tasks, the proposed method comprises two phases, as illustrated in Fig. 3:

1. Generation of candidate sample subsets: Candidate subsets are produced via a global search strategy based on the Chaotic Imperialist Competitive Algorithm (CICA). By leveraging chaos theory within the ICA, this approach enhances both the diversity of the candidate population and the algorithm's convergence behavior, promoting exploration and reducing the risk of premature convergence.
2. Assessment via a fuzzy weighted average distance-based decision surface: The evaluation phase employs a fuzzy weighted average distance-based decision surface. This method is designed to deliver robust class boundary modeling. By integrating uncertainty handling and instance-level difficulty metric, it ensures that the constructed decision boundaries are sensitive to the distinctive characteristics of imbalanced datasets and can better distinguish between minority and majority classes.

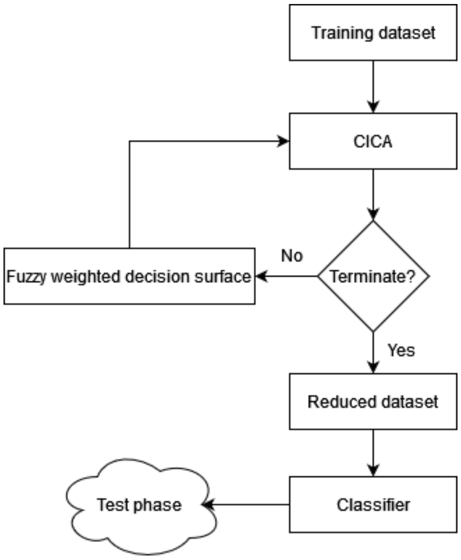

Fig. 3. Flowchart of the proposed method.

A. Exploring the Solution Space

Instance selection can be formulated as a multi-objective optimization problem, where multiple candidate solutions—each representing a reduced dataset—are explored within a defined solution space. In the proposed approach, the conflicting objectives include maximizing the F-measure, G-mean, and accuracy, while simultaneously minimizing the reduction rate of the minority class. Equation (8) formalizes this unconstrained optimization problem aimed at selecting instances from an imbalanced dataset.

$$\begin{aligned} f_{\text{cost}} &= w_1.Fmeasure + w_2.Gmean \\ &+ w_3.Accuracy + w_4.\left(p_1.Red_{\min} + p_2.Red_{\text{maj}}\right) \\ &\sum_{i=1}^{4} w_i = 1, \quad p_1 > p_2 \end{aligned} \tag{8}$$

where

$$Fmeasure = \frac{(1+\alpha^2).precision.recall}{\alpha^2.precision.recall} \tag{9}$$

$$Gmean = \sqrt{TP \times TN} \tag{10}$$

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

$$Reduction = \frac{\text{Total instances} - \text{Retained instances}}{\text{Total instances}} \tag{12}$$

where TP, FP, FN, and TN denote true positives, false positives, false negatives, and true negatives, respectively. A TP corresponds to a target instance correctly classified as target. A FP occurs when a non-target instance is erroneously classified as target. Conversely, a FN arises when a target instance is incorrectly identified as non-target. Finally, a TN indicates the correct classification of a non-target instance as non-target.

In **Error! Reference source not found.**, w_1, w_2, w_3 and w_4 are adaptive weighting parameters of each metric and p_1 , p_2 are the penalty parameters. Regarding the minimization problem, p_1 and p_2, i.e., reduction rate of the minority class should be lower than the reduction rate of the majority class.

The Chaotic Imperialist Competitive Algorithm (CICA), facilitates exploration across both minority and majority classes. It initializes with a random population represented as a binary array, where each element corresponds to a country. The presence or absence of a country is encoded as 1 or 0, respectively, with varying configurations representing different candidate solutions. The solution exhibiting the highest fitness value is selected, corresponding to the optimal instance selection state. By leveraging CICA, a solution pool is generated, each optimized for one of four distinct objectives, from which an appropriate reduced dataset can be chosen. Here, 10 percent of the countries are considered as empires ($N_{imp} = 10\%$) and 90 percent of them are used as colonies ($N_{col} = 90\%$). Besides, the cost function represented in **Error! Reference source not found.** evaluates the corresponding cost value of each country presented in **Error! Reference source not found.**. Since utilizing random values for θ presented in **Error! Reference source not found.** may get stuck in a local solution and speed down the convergence, the proposed method uses some chaotic maps listed in Table IV for finding globally optimal solutions.

TABLE IV
Chaotic Functions

Name	Function
Logistic map	$z_{k+1} = az_k\left(1-z_k\right)$
Tent map	$z_{k+1} = \begin{cases} 2z_k, & z_k < 0.5 \\ 2\left(1-z_k\right), & z_k \geq 0.5 \end{cases}$
Sinusoidal map	$z_{k+1} = az^2_k \sin\left(\pi z_k\right)$
Gauss map	$z_{k+1} = \begin{cases} 0 & z = 0 \\ \dfrac{1}{z_k} mod\left(1\right) & otherwise \end{cases}$

B. Difficulty-aware Decision Surface

Now, a classification method is introduced that integrates difficulty awareness into the decision-making process by employing a fuzzy membership function. This function assigns weights to training data points (candidate solutions) according to their assessed difficulty level. Initially, each training sample is evaluated using a difficulty metric that quantifies the ambiguity or uncertainty associated with that sample. The fuzzy membership function then converts these difficulty scores into corresponding weights, effectively emphasizing the influence of more challenging instances.

When classifying a new data point, the proposed decision surface assigns its label based on the proximity to the nearest class boundary. Importantly, the distance calculation incorporates the fuzzy membership weights, allowing the classification to factor in not only the feature similarity but also the difficulty of comparable training samples. This approach enhances robustness by recognizing that harder-to-classify instances contribute differently to shaping the decision boundary.

Consider *N* points $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ where $\boldsymbol{x}_i = \{x_{i1}, ..., x_{id}\}$ distributed within a *d*-dimensional feature space. A label y_i assigns each point to one of l distinct classes. The distance between two points z_i and z_j in a high dimensional feature space given by **Error! Reference source not found.** [53]:

$$\begin{aligned} D(\phi(z_i), \phi(z_j)) &= \left\| \phi(z_i) - \phi(z_j) \right\|_2 \\ &= \sqrt{(\phi(z_i) - \phi(z_j)).(\phi(z_i) - \phi(z_j))} \\ &= \sqrt{(\phi(z_i).\phi(z_i)) - 2(\phi(z_i).\phi(z_j)) + (\phi(z_j).\phi(z_j))} \\ &= \sqrt{k(z_i, z_i) - 2k(z_i, z_j) + k(z_j, z_j)} \end{aligned} \tag{13}$$

where ϕ is a mapping function transferred training instances into high-dimensional feature space, and $k(z_i, z_j)$ is a kernel function. Regarding a binary classification task, the class centers of positive and negative classes in a high-dimensional feature space are given by Eq. **Error! Reference source not found.**.

$$\begin{aligned} C^+ &= \frac{1}{l^+} \sum_{z_j \in \{+1\}} \phi(z_j) \\ C^- &= \frac{1}{l^-} \sum_{z_j \in \{-1\}} \phi(z_j) \end{aligned} \tag{14}$$

where l^+ and l^- are the size of positive and negative class, respectively.

The normalized distance between point z_i and the center of its class is determined by **Error! Reference source not found.** :

$$\Delta(z_i) = \begin{cases} \dfrac{\left\| \phi(z_i) - C^+ \right\|}{\max \left\| \phi(z_i) - C^+ \right\|}, & \text{if } z_i \in \{+1\} \\ \dfrac{\left\| \phi(z_i) - C^- \right\|}{\max \left\| \phi(z_i) - C^- \right\|}, & \text{if } z_i \in \{-1\} \end{cases} \tag{14}$$

By taking the median of all *k*-NN around instance z_i, the normalized density for z_i is computed as follows.

$$\rho(z_i) = \frac{median\left\{ \dfrac{1}{\left\| \phi(z_i) - \phi(z_1) \right\|}, ..., \dfrac{k}{\left\| \phi(z_i) - \phi(z_k) \right\|} \right\}}{\max\left\{ \dfrac{1}{\left\| \phi(z_i) - \phi(z_1) \right\|}, ..., \dfrac{k}{\left\| \phi(z_i) - \phi(z_k) \right\|} \right\}} \tag{15}$$

Finally, the membership degree of instance z_i is given by:

$$\mu(z_i) = \frac{\delta.\rho(z_i)}{\Delta(z_i)} \tag{16}$$

where $\delta, \delta \in (0,1]$ is a positive parameter which determines the importance of density. Clearly, the membership degree of each sample follows $\mu(z_i) \in [0,1]$.

A notable challenge emerges when data points exhibit comparable membership degrees across multiple classes. As illustrated in Fig. 4, instances located near the decision boundary (denoted by "+") may possess identical membership values, yet their contributions to each class can differ significantly. To enhance the modeling of such ambiguous and difficult instances, the framework is extended by incorporating non-membership and hesitation degrees, enabling a more nuanced representation of uncertainty.

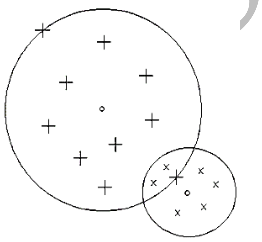

Fig. 4. Illustration two instances (+) with equal distance to the center of the class.

Consider *k*-NN for instance z_i . Let z_m be the m^{th} nearest neighbor of instance z_i. The hesitation degree for instance z_i is defined by **Error! Reference source not found.**.

$$h(z_i) = \begin{cases} \dfrac{\left| \left\{ z_j \left| \left\| \phi(z_m) - C^- \right\| \le \left\| \phi(z_m) - C^+ \right\| \right. \right\}_{\substack{j=1 \\ m=1}}^{\substack{j=k \\ m=k}} \right|}{k}, & y_i = +1 \\ \dfrac{\left| \left\{ z_j \left| \left\| \phi(z_m) - C^+ \right\| \le \left\| \phi(z_m) - C^- \right\| \right. \right\}_{\substack{j=1 \\ m=1}}^{\substack{j=k \\ m=k}} \right|}{k}, & y_i = -1 \end{cases} \tag{17}$$

where $h(z_i) \in [0,1]$. Fig. 5 shows an example for computing the hesitation degree. The non-membership degree of instance z_i is defined by **Error! Reference source not found.**.

$$\eta(z_i) = \frac{h(z_i)}{\mu(z_i)} \tag{18}$$

According to **Error! Reference source not found.**, if $\mu(z_i) = 0$ then $\eta(z_i) \to \infty$. Also, the non-membership degree becomes zero if $h(z_i) = 0$; So, $\eta(z_i) \in [0, \infty)$.

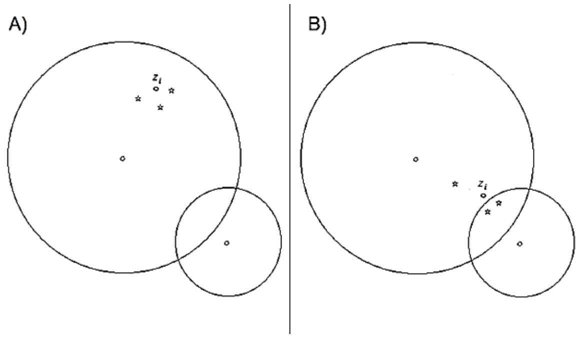

Fig. 5. Computing the hesitation degree for instance z_i with different position. For k=3, the hesitation degree of z_i in (A) is zero whilst in (B) is 2/3.

According to **Error! Reference source not found.**, if $\mu(z_i)=0$ then $\eta(z_i)\to\infty$. Also, the non-membership degree becomes zero if $h(z_i)=0$; So, $\eta(z_i)\in[0,\infty)$

Now, the weight of each instance is computed by the degree of membership and the degree of non-membership in **Error! Reference source not found.**:

$$\omega_i=\begin{cases}\mu_i & h_i\le 0.3\\ \dfrac{\mu_i}{\mu_i+\eta_i} & otherwise\end{cases} \tag{19}$$

Thus, the weighted average of each class is computed by **Error! Reference source not found.**.

$$\begin{aligned}C_\omega^{+}&=\frac{1}{l^{+}}\sum_{y_i=+1}\omega_i z_i\\ C_\omega^{-}&=\frac{1}{l^{-}}\sum_{y_i=-1}\omega_i z_i\end{aligned} \tag{20}$$

where l^{+} and l^{-} are the size of the positive and negative classes, respectively. Then, instance z_i is classified by **Error! Reference source not found.**.

$$y_i=\operatorname{sgn}\left(D\left(\phi(z_i),C_\omega^{-}\right)-D\left(\phi(z_i),C_\omega^{+}\right)\right) \tag{21}$$

EXPERIMENTS

The experiments were conducted in a Jupyter Notebook environment using an Intel Core i5 processor and 16 GB of RAM. To assess the performance of the proposed method, classifiers including SVM, Convolutional Neural Network (CNN), and k-Nearest Neighbors with k=10 were employed. The experimental evaluation involved multiple kernel functions, such as Radial Basis Function (RBF) and polynomial kernels, alongside various chaotic maps including logistic, tent, sinusoidal, and Gaussian maps. Table V summarizes the optimal parameter settings utilized throughout the experiments.

TABLE V
Values of the Parameters

Parameter	Val.	Parameter	Val.	Parameter	Val.
δ	0.6	ww_1	0.2	Imperialist size	10
α	0.8	w_2	0.2	p_1	0.9
ξ	0.3	w_3	0.2	p_2	0.1
Population size	100	w_4	0.3		

A. Experiments on benchmark datasets

In this section, experimental evaluations are performed on datasets of moderate scale. The experiments are carried out on a diverse set of UCI datasets. The detailed statistics of these datasets are summarized in Table VI. As shown in the last column of the table, which reports the IR, some datasets are relatively balanced (IR < 5), while most exhibit varying degrees of class imbalance. This distribution enables a thorough evaluation of the proposed method across both balanced and imbalanced settings. These datasets provide a representative testbed for validating the effectiveness and generalizability of the proposed approach.

The initial phase of the experiments involves examining various kernel functions and chaotic maps within the proposed framework to identify the optimal configuration. Performance metrics—including Accuracy, Reduction Rate, Fmeasure, and Effectiveness—are reported using 10-fold cross-validation across both balanced and imbalanced datasets, employing SVM, k-NN, and CNN classifiers. Figures 6 through 8 illustrate these evaluation metrics corresponding to the SVM, k-NN, and CNN classifiers, respectively. Notably, to jointly assess reduction rate and Fmeasure, the Effectiveness metric is defined as the *Reduction rate* × *Fmeasure*.

TABLE VI
Dataset Statistics

Dataset	#samples	#features	Class1 (%)	Class2 (%)	IR (%)
Yeast-6	1484	8	2.36	97.64	41.38
Yeast-5	1484	8	2.70	97.30	36.03
Yeast-1289vs7	947	8	3.17	96.83	30.55
Yeast-4	1484	8	3.44	96.56	28.07
Yeast-3	1484	8	10.98	89.02	8.11
Yeast-2vs8	483	8	4.14	95.86	23.15
Yeast-1	1484	8	28.90	71.10	2.46
Yeast-1458vs7	693	8	4.33	95.67	22.09
Yeast-1vs7	459	8	6.54	93.46	14.29
Ecoli-4	336	7	5.95	94.05	15.80
Ecoli-3	336	7	10.42	89.58	8.60
Ecoli-2	336	7	15.48	84.52	5.46
Ecoli-1	336	7	22.92	77.08	3.36
Ecoli-0vs1	220	7	35	65	1.86
Ecoli-0137vs26	282	7	2.48	97.52	39.32
Glass-6	214	9	6.07	93.93	15.50
Glass-5	214	9	4.20	95.80	22.80
Glass-016vs5	184	9	4.89	95.11	19.45
Glass-5vs12	159	9	8.18	91.82	11.26
Glass-2	214	9	13.55	86.45	6.38
Glass-123vs567	214	9	23.83	76.17	3.20
Glass-1	214	9	35.51	64.49	1.82
Glass-0	214	9	32.71	67.29	2.09

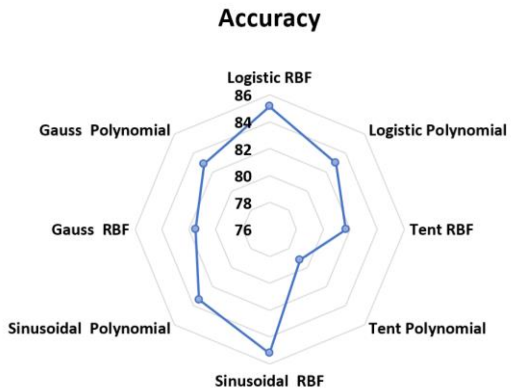

Fig 6. A. Accuracy of the proposed method on different kinds of kernel functions and chaotic maps using SVM classifier.

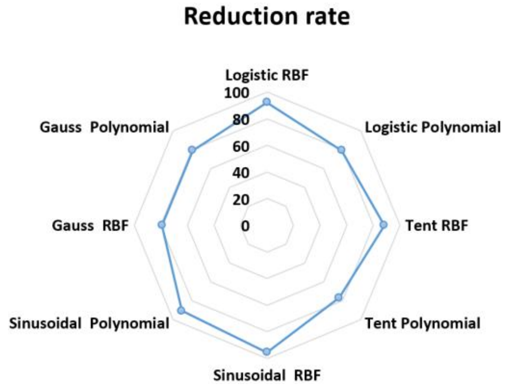

Fig 6. B. Reduction rate of the proposed method on different kinds of kernel functions and chaotic maps using SVM classifier.

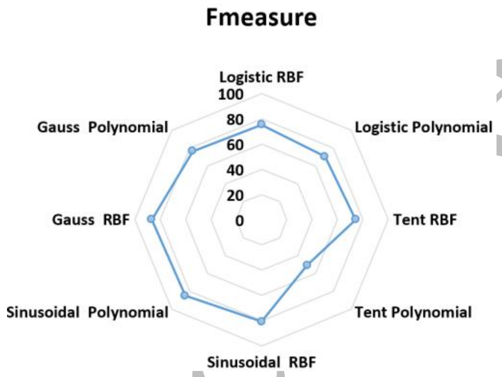

Fig 6. C. Fmeasure of the proposed method on different kinds of kernel functions and chaotic maps using SVM classifier.

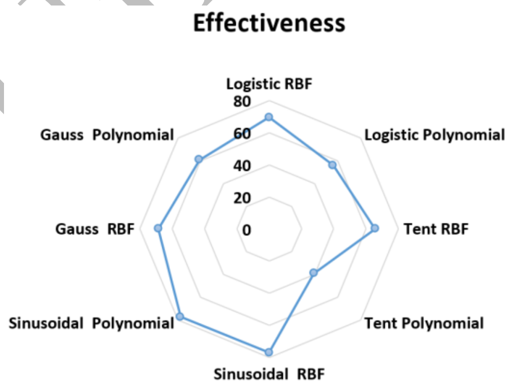

Fig 6. D. Effectiveness of the proposed method on different kinds of kernel functions and chaotic maps using SVM classifier.

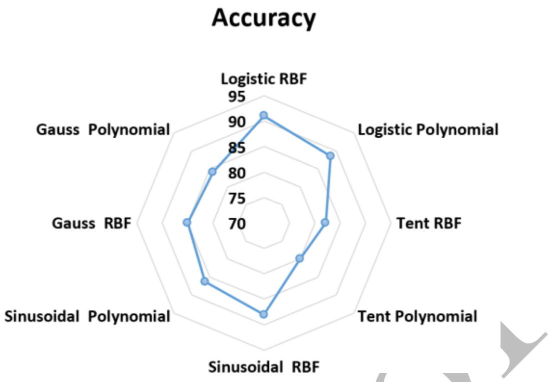

Fig 7. A. Accuracy of the proposed method on different kinds of kernel functions and chaotic maps using *k*-NN classifier.

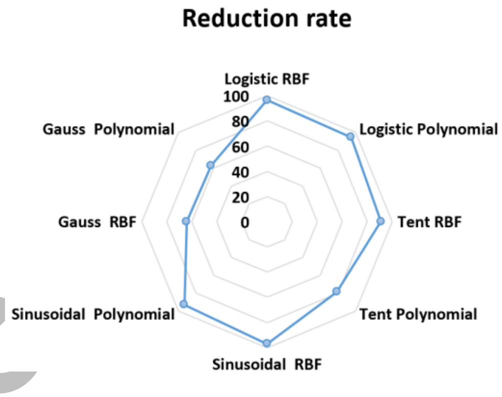

Fig 7. B. Reduction rate of the proposed method on different kinds of kernel functions and chaotic maps using *k*-NN classifier.

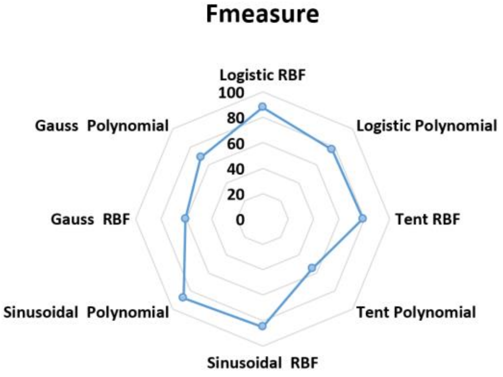

Fig 7. C. Fmeasure of the proposed method on different kinds of kernel functions and chaotic maps using *k*-NN classifier.

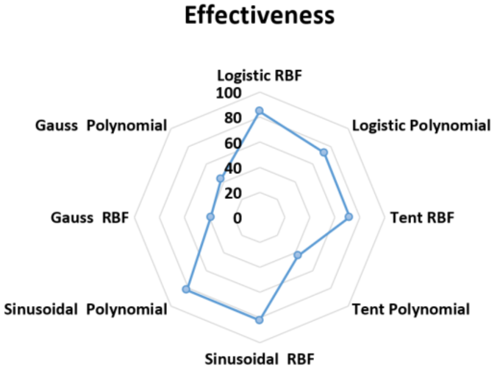

Fig 7. D. Effectiveness of the proposed method on different kinds of kernel functions and chaotic maps using *k*-NN classifier.

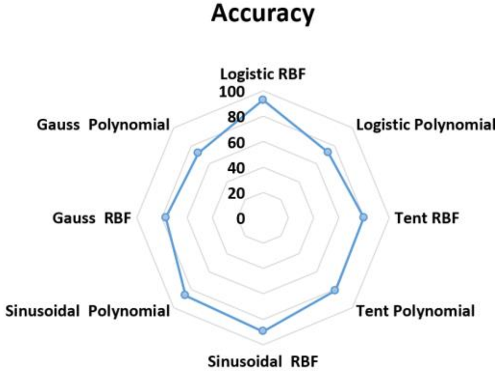

Fig 8. A. Accuracy of the proposed method on different kinds of kernel functions and chaotic maps using CNN classifier.

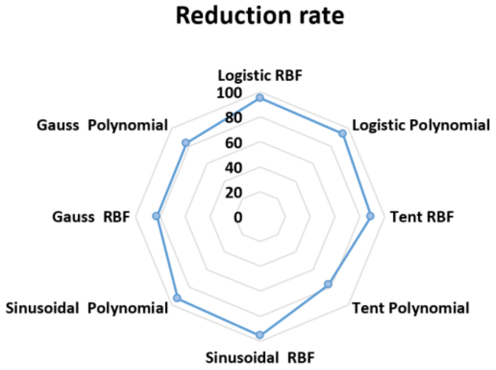

Fig 8. B. Reduction rate of the proposed method on different kinds of kernel functions and chaotic maps using CNN classifier.

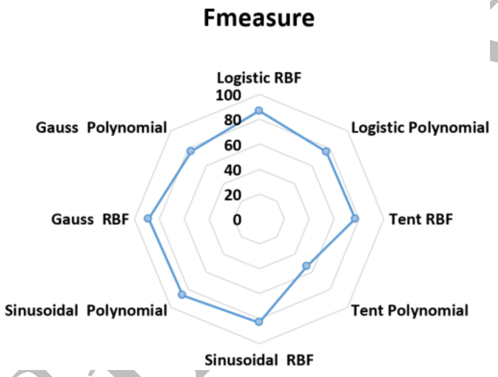

Fig 8. C. Fmeasure of the proposed method on different kinds of kernel functions and chaotic maps using CNN classifier.

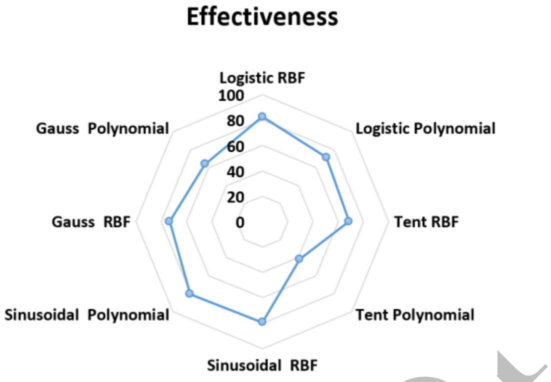

Fig 8. D. Effectiveness of the proposed method on different kinds of kernel functions and chaotic maps using CNN classifier.

According to Figures 6–8, the experimental results demonstrate that the RBF kernel consistently outperforms the polynomial kernel across all evaluation metrics. Although the polynomial kernel occasionally achieves competitive performance, it lacks stability and fails to deliver consistently acceptable results across all criteria. In contrast, the RBF kernel exhibits greater robustness and reliability, making it the preferred choice within the proposed framework.

Additionally, the figures highlight that the sinusoidal and logistic chaotic maps outperform other chaotic functions. The choice of chaotic map has a significant impact on both the reduction rate and classification performance metrics, such as accuracy and F-measure. These results emphasize the sensitivity of the method to the underlying chaotic dynamics. Based on the observations, the logistic map is recommended when minimizing classification error is the primary objective, while the sinusoidal map is more suitable when prioritizing dimensionality reduction. Both functions perform consistently well across most metrics; however, selecting the appropriate chaotic function based on the application's goals can further enhance overall effectiveness.

The class distribution before and after applying the proposed method is illustrated in Figure 9. As discussed earlier, two major challenges in instance selection are:

- Preserving inter-class balance in originally balanced datasets, avoiding the introduction of skew during reduction;
- Retaining sufficient minority-class instances in imbalanced datasets to prevent information loss.

To evaluate the impact of the proposed method on class distribution, Figure 9 presents a comparative analysis of the IR, as well as the proportions of minority and majority instances, before and after instance selection. The results clearly show an increase in the proportion of minority-class instances and a decrease in majority-class instances. This indicates that the method selectively removes more majority instances, effectively emphasizing the preservation of minority-class information.

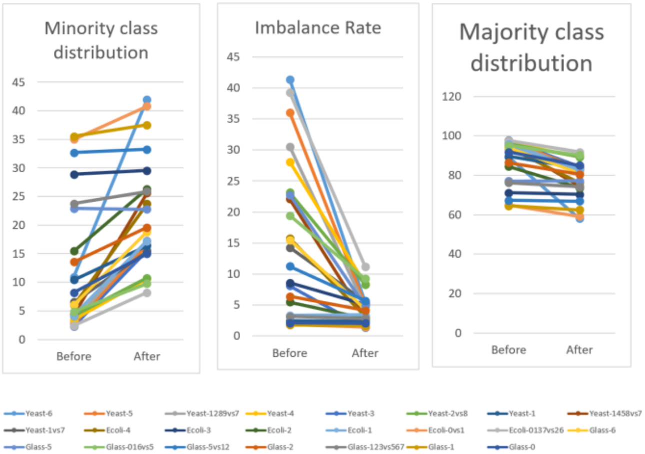

Fig 9. Class distribution (%) before and after using the proposed method.

Moreover, the results confirm that the method maintains inter-class balance in balanced datasets and improves class representation in imbalanced settings. This section has presented strong empirical evidence supporting the framework's validity and effectiveness. The next section builds on these findings with further experiments.

A. Experimental results in comparison with other methods on benchmark datasets

This section presents a comparative analysis between the proposed method and several competing methods. The evaluation uses the logistic chaotic map and the RBF kernel within our framework. To ensure a fair comparison, all methods employ a common SVM classifier.

The competing methods are selected based on their conceptual alignment with our approach, incorporating at least one of the following principles: evolutionary optimization, fuzzy modeling, difficulty estimation, or density-aware evaluation. This deliberate selection ensures a meaningful benchmark by attributing performance differences to the innovations of our design rather than fundamental methodological disparities.

The results are presented in Tables VII, VIII, and IX, on the Yeast, Ecoli, and Glass datasets. Each table reports the average values of three key metrics—F-measure, G-mean, and AUC—along with their corresponding ranks. The Average Rank (AR) is provided in the final row of each table. Metric abbreviations include F1 (F-measure), Gm (Geometric Mean), AUC, and AR (Average Rank), consistently applied throughout.

According to the reported ranks, the proposed method consistently outperforms all competing approaches. Figures 10–12 complement the tables by graphically depicting the outcomes for F-measure, G-mean, and AUC, offering an intuitive visual comparison.

TABLE VII
Evaluation Metrics for the Proposed Method and the State-of-the-art Methods on Yeast Dataset (Average)

Metric	SGO	GE-IFRVFL-CIL	KDEBO	KDENDS_SMOTE	SVDDSMOTE	Proposed method
F1	49.12	73.03	74.12	69.34	70.66	78.16
	(6)	(3)	(2)	(5)	(4)	**(1)**
Gm	51.20	79.32	79.19	70.82	80.08	85.65
	(6)	(3)	(4)	(5)	(2)	**(1)**
AUC	57.05	82.64	86.98	71.41	81.68	85.98
	(6)	(3)	**(1)**	(5)	(4)	(2)
AR	6	3	2.33	5	3.33	1.33
	6	3	2	5	4	**(1)**

TABLE VIII
Evaluation Metrics for the Proposed Method and the State-of-the-art Methods on Echoli Dataset (Average)

Metric	SGO	GE-IFRVFL-CIL	KDEBO	KDENDS_SMOTE	SVDDSMOTE	Proposed method
F1	79.78	80.64	82.34	84.84	80.83	83.51
	(6)	(5)	(3)	**(1)**	(4)	(2)
Gm	87.18	85.34	87.91	88.87	89.63	92.45
	(5)	(6)	(4)	(3)	(2)	**(1)**
AUC	87.71	89.12	91.03	88.90	91.66	93.25
	(6)	(4)	(3)	(5)	(2)	**(1)**
AR	5.6	5	3.33	3	3.33	1.33
	(5)	(4)	(3)	(2)	(3)	**(1)**

TABLE IX
Evaluation Metrics for the Proposed Method and the State-of-the-art Methods on Glass Dataset (Average)

Metric	SGO	GE-IFRVFL-CIL	KDEBO	KDENDS_SMOTE	SVDDSMOTE	Proposed method
F1	75.22	82.05	76.02	78.37	80.8	94.64
	(6)	(2)	(5)	(4)	(3)	**(1)**
Gm	75.50	88.33	81.85	85.54	87.5	96.42
	(6)	(2)	(5)	(4)	(3)	**(1)**
AUC	87.47	89.12	86.81	87.43	89.25	96.8
	(4)	(3)	(6)	(5)	(2)	**(1)**
AR	5.33	2.33	5.33	4.33	2.66	1
	(5)	(2)	(5)	(4)	(3)	**(1)**

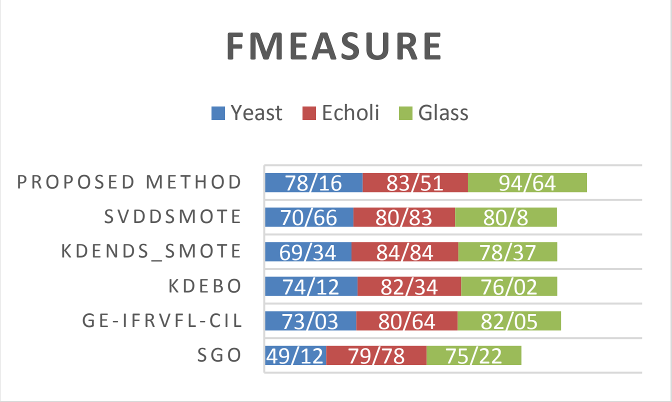

Fig 10. Fmeasure metric for the proposed method and other methods - all datasets.

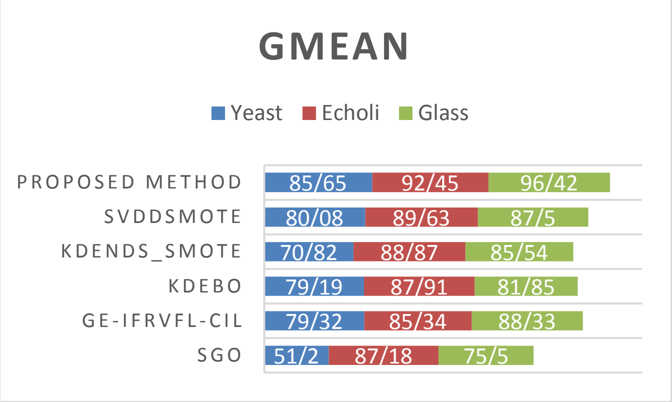

Fig 11. Gmean metric for the proposed method and other methods - all datasets.

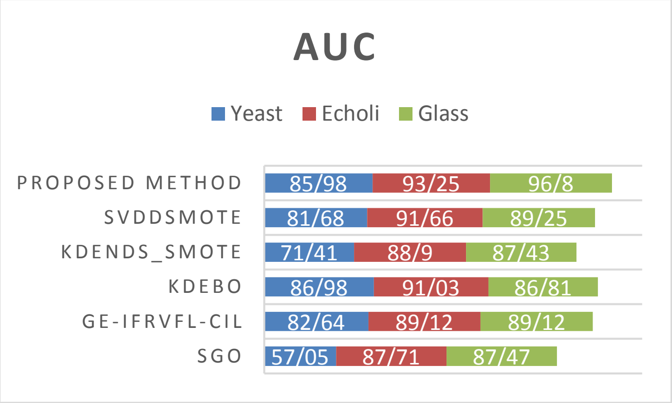

Fig 12. AUC metric for the proposed method and other methods - all datasets.

A. Discussion

The comparative results in Tables VII–IX and Figures 10–12 confirm the superiority of the proposed method across all key evaluation metrics. Several factors contribute to this performance advantage over techniques such as SGO, GE-IFRVFL-CIL, KDEBO, KDENDS_SMOTE, and SVDDSMOTE:

- Instance Selection over Generation: Unlike most methods that focus on generating synthetic data, our approach emphasizes informed instance selection. This preserves data integrity and improves classifier generalization by avoiding synthetic noise and overlap.
- Fuzzy Awareness and Adaptive Penalties: Unlike SVDD-based oversampling methods with fixed parameters, our method leverages fuzzy membership weighting to dynamically adapt penalties, allowing better boundary modeling in sparse regions.
- Fine-Grained Fitness Evaluation: Our integration of fuzzy difficulty-awareness into the multi-objective evolutionary process allows instance contribution to be evaluated more precisely, outperforming approaches like GE-IFRVFL-CIL, which use fuzzy logic in a more coarse-grained fashion.
- Chaotic Evolutionary Dynamics: The use of chaotic maps in the evolutionary algorithm (unlike conventional mechanisms used in SGO or KDEBO) improves search-space exploration and convergence stability, particularly in noisy or high-dimensional scenarios.
- Preservation of Class Distribution: Our method maintains balance in originally balanced datasets and improves minority representation in imbalanced datasets without generating artificial data—a capability evidenced by Figure 9 and the superior ranks across evaluation tables.

B. Real-world datasets with high class imbalance across diverse domains

To evaluate the scalability and robustness of the proposed method under realistic, data-intensive conditions, we conduct extensive experiments on the large-scale dataset regarding CLIMB benchmark suite [58]. This benchmark is specifically designed to reflect the core challenges of industrial-scale imbalanced learning, including extreme class imbalance, high-dimensional feature spaces, noisy distributions, and structural irregularities[4]. The CLIMB includes six datasets with IR>50. Table X summarizes the key characteristics of these datasets. The proposed method is compared against the methods summarized in Table XI.

TABLE X
Summary of Real-World Datasets with Severe Class Imbalance

Dataset Name	#samples	#features	IR	Domain
Dis	3772	29	64.03	Biology
Satellite	5100	36	67.0	Remote sensing
Employee-Turnover-at-TECHCO	34452	2	68.74	Human resources
Page-blocks	5473	10	175.46	Document processing
allbp	3772	29	257.79	Biology
Credit Card Fraud Detection	284807	30	577.88	Finance

[4] https://imbalanced-ensemble.readthedocs.io/en/latest/api/datasets

To guarantee methodological alignment with CLIMB and ensure reproducibility, we replicate their entire evaluation pipeline:

- **Preprocessing**: Numerical features are standardized to zero mean and unit variance; categorical features are one-hot encoded; no imputation is required due to dataset completeness.
- **Validation Protocol**: A stratified 5-fold cross-validation scheme is adopted to preserve class ratios in each fold. Within each training fold, we perform a nested 3-fold tuning of hyperparameters using Optuna [59], with 100 trials per method and all random seeds fixed to 42. Optuna is an open-source hyperparameter optimization framework designed to automate the process of searching for the best hyperparameters for machine learning models. It provides an efficient, flexible way to handle optimization tasks by using algorithms such as Tree-structured Parzen Estimator (TPE) and integrates seamlessly with existing machine learning pipelines. Optuna is particularly noted for its ability to optimize hyperparameters dynamically, using a "define-by-run" approach, where users specify the optimization logic in a flexible and modular way.
- **Base Classifier**: SVM classifiers were used to evaluate the effect of instance reduction and sampling strategies, allowing a fair and interpretable comparison across all baselines.
- **Implementation Details**: All algorithms are implemented in a unified Python codebase, parallelized across folds, and executed on a server with 64 GB RAM. Large-scale encoding leverages sparse matrix representations to accommodate large scale datasets.

Table XI
Baseline Methods Selected for Evaluation

Selected Method	Category	Description	Ref.
Edited Nearest Neighbors (ENN)	Cleaning	Removes mislabeled/noisy majority samples by local disagreement filtering	[60]
Instance Hardness Threshold (IHT)	Simple Undersampling	Eliminates hard-to-classify majority examples via local hardness estimation	[61]
Easy Ensemble	Ensemble Undersampling	Trains multiple classifiers on balanced subsets with aggregated prediction	[62]
ADASYN	Oversampling	Adaptive oversampling focused on sparse and borderline minority regions	[63]
SMBA	Ensemble Oversampling	Applies bagging with local density-aware synthetic sampling per ensemble member	[64]
Cost-sensitive SVM	Cost-sensitive Learning	Adjusts misclassification costs to penalize majority-class errors less	[65]
KDEBO	Kernel density-based methods	Density-guided differential evolution oversampling in high-density minority regions to avoid noise	[35]

Table XII reports the experimental results in terms of AUC, G-Mean, F-measure, and accuracy. The proposed method consistently outperforms all baseline approaches across these metrics. While ENN and IHT help clarify decision boundaries, they often fail to retain the most informative minority instances that our method preserves. EasyEnsemble and KDEBO partially address class imbalance but introduce high variance or synthetic noise. Similarly, AsymBoost improves minority recall but struggles to maintain overall accuracy, unlike the proposed method's multi-objective selection strategy. In contrast to synthetic methods such as ADASYN and SMBA, which expand the dataset, the proposed method selectively retains high-impact minority instances based on their difficulty and discriminative value. Several key factors contribute to the superior performance of the proposed method:

- Overlap Reduction: By integrating a fuzzy hardness metric with a chaotic multi-objective evolutionary process, the method avoids selecting noisy or overlapping samples—an issue that commonly degrades oversampling methods.
- Scalability and Stability: Experimental results demonstrate that the method scales well with data size, maintaining stable convergence and computational efficiency even on datasets exceeding 500,000 instances.
- Granular Difficulty Modeling: Fuzzy hardness enables more precise estimation of instance importance compared to binary selection rules or density-based heuristics.
- Chaotic Multi-objective Optimization: The use of the Imperialist Competitive Algorithm enhanced with chaotic maps promotes both rapid convergence and diverse solutions within a large search space.

These findings confirm that strategically guided instance selection—grounded in difficulty-awareness and optimized through chaotic evolutionary dynamics—provides a robust and scalable solution to extreme class imbalance in large-scale tabular learning.

Table XII
Performance Comparison on Real-World Datasets with Severe Class Imbalance

Method	AUC	G-Mean	Fmeasure	Balanced Acc.
ENN [60]	45.0	75.4	75.1	75.8
IHT [61]	33.0	74.6	68.0	**79.9**
EasyEnsemble [62]	35.5	83.0	89.8	87.3
ADASYN [63]	34.3	76.2	70.3	76.3
SMBA [64]	56.0	74.9	76.6	74.4

Cost-sensitive SVM [65]	41.7	73.9	74.9	74.8
KEDBO[35]	53.4	76.08	71.92	79.17
Proposed method	**58.33**	**77.24**	**91.65**	78.32

CONCLUSION AND FUTURE WORK

This paper introduced a difficulty-aware instance reduction method that combines chaotic evolutionary optimization with fuzzy-informed instance selection to enhance classification. The method uses a chaotic Imperialist Competitive Algorithm (ICA) guided by a multi-objective fitness function balancing accuracy and fairness. A distance-weighted decision surface further ensures structural class separation. Comprehensive experiments on datasets with different intensities of imbalance demonstrated the method's superiority over competing methods. We also investigated the effect of kernel functions and chaotic maps. The RBF kernel with the logistic map yielded the best performance when accuracy was prioritized, while the sinusoidal map was more effective for dimensionality reduction. The chaotic components accelerated convergence and enhanced diversity in the solution space.

Key strengths of the proposed method include:

- Robust handling of class imbalance without synthetic data
- Flexibility for different optimization goals (e.g., accuracy vs. reduction rate)
- Consistent generalization across small and large datasets

Future work will explore the following directions: extending the method to multi-class scenarios with overlapping distributions, enhancing scalability through dimensionality-aware optimization and sparse instance modeling, integrating instance selection with graph-based learning or self-supervised representations, and applying the method to real-time or streaming environments with incremental instance selection and dynamic class balancing.

REFERENCES

[1] T. Xia, T. Dang, J. Han, L. Qendro, and C. Mascolo, "Uncertainty-aware health diagnostics via class-balanced evidential deep learning," *IEEE J. Biomed. Health Inform.*, 2024.

[2] Z. Nouri, V. Kiani, and H. Fadishei, "Rarity updated ensemble with oversampling: An ensemble approach to classification of imbalanced data streams," *Stat. Anal. Data Min.: ASA Data Sci. J.*, vol. 17, no. 1, p. e11662, 2024.

[3] L. Ren *et al.*, "Improving fraud detection via imbalanced graph structure learning," *Mach. Learn.*, vol. 113, no. 3, pp. 1069–1090, 2024.

[4] S. Raza, M. Garg, D. J. Reji, S. R. Bashir, and C. Ding, "Nbias: A natural language processing framework for BIAS identification in text," *Expert Syst. Appl.*, vol. 237, p. 121542, 2024.

[5] C.-Q. Zhang, Y. Deng, M.-Z. Chong, Z.-W. Zhang, and Y.-H. Tan, "Entropy-based re-sampling method on SAR class imbalance target detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 209, pp. 432–447, 2024.

[6] Q. Zhou and B. Sun, "Adaptive K-means clustering based under-sampling methods to solve the class imbalance problem," *Data Inf. Manag.*, p. 100064, 2023.

[7] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in *Proc. IEEE*, 2010, pp. 1–8.

[8] M. Moradi and J. Hamidzadeh, "Handling class imbalance and overlap with a hesitation-based instance selection method," *Knowl.-Based Syst.*, vol. 294, p. 111745, 2024.

[9] J. Bi and C. Zhang, "An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme," *Knowl.-Based Syst.*, vol. 158, pp. 81–93, 2018.

[10] I. Czarnowski, "Weighted ensemble with one-class classification and over-sampling and instance selection (WECOI): An approach for learning from imbalanced data streams," *J. Comput. Sci.*, vol. 61, p. 101614, 2022.

[11] M. S. E. Shahabadi, H. Tabrizchi, M. K. Rafsanjani, B. B. Gupta, and F. Palmieri, "A combination of clustering-based under-sampling with ensemble methods for solving imbalanced class problem in intelligent systems," *Technol. Forecast. Soc. Change*, vol. 169, p. 120796, 2021.

[12] B. B. Hazarika and D. Gupta, "Density-weighted support vector machines for binary class imbalance learning," *Neural Comput. Appl.*, vol. 33, no. 9, pp. 4243–4261, 2021.

[13] B. B. Hazarika and D. Gupta, "Affinity based fuzzy kernel ridge regression classifier for binary class imbalance learning," *Eng. Appl. Artif. Intell.*, vol. 117, p. 105544, 2023.

[14] B. B. Hazarika, D. Gupta, and P. Borah, "Fuzzy twin support vector machine based on affinity and class probability for class imbalance learning," *Knowl. Inf. Syst.*, vol. 65, no. 12, pp. 5259–5288, 2023.

[15] S. Guan, X. Zhao, Y. Xue, and H. Pan, "AWGAN: An adaptive weighting GAN approach for oversampling imbalanced datasets," *Inf. Sci.*, p. 120311, 2024.

[16] Z. Wang and H. Wang, "Variational imbalanced regression: Fair uncertainty quantification via probabilistic smoothing," *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

[17] S. Tao, P. Peng, Y. Li, H. Sun, Q. Li, and H. Wang, "Supervised contrastive representation learning with tree-structured parzen estimator Bayesian optimization for imbalanced tabular data," *Expert Syst. Appl.*, vol. 237, p. 121294, 2024.

[18] C. Ye, R. Tsuchida, L. Petersson, and N. Barnes, "Label shift estimation for class-imbalance problem: A Bayesian approach," 2024, pp. 1073–1082.

[19] A. Kumari, M. Tanveer, and C. T. Lin, "Class probability and generalized Bell fuzzy twin SVM for imbalanced data," *IEEE Trans. Fuzzy Syst.*, 2024.

[20] Z. Deng, P. Cui, and J. Zhu, "Towards accelerated model training via Bayesian data selection," *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

[21] Y. Pu, W. Yao, and X. Li, "EM-IFCM: Fuzzy c-means clustering algorithm based on edge modification for imbalanced data," *Inf. Sci.*, vol. 659, p. 120029, 2024.

[22] S. Bano *et al.*, "Self-paced ensemble and big data identification: A classification of substantial imbalance computational analysis," *J. Supercomput.*, vol. 80, no. 7, pp. 9848–9869, 2024.

[23] N. García-Pedrajas, J. M. Cuevas-Muñoz, and A. de Haro-García, "Evolutionary simultaneous under and oversampling of instances for dealing with class-imbalance datasets in multilabel problems," *Appl. Soft Comput.*, p. 111618, 2024.

[24] X. Lu, X. Ye, and Y. Cheng, "An overlapping minimization-based over-sampling algorithm for binary imbalanced classification," *Eng. Appl. Artif. Intell.*, vol. 133, p. 108107, 2024.

[25] Y. Liu, L. Zhu, L. Ding, H. Sui, and W. Shang, "A hybrid sampling method for highly imbalanced and overlapped data classification with complex distribution," *Inf. Sci.*, vol. 661, p. 120117, 2024.

[26] T. Ma, S. Lu, and C. Jiang, "A membership-based resampling and cleaning algorithm for multi-class imbalanced overlapping data," *Expert Syst. Appl.*, vol. 240, p. 122565, 2024.

[27] A. Kumar, D. Singh, and R. S. Yadav, "Class overlap handling methods in imbalanced domain: A comprehensive survey," *Multimedia Tools Appl.*, pp. 1–48, 2024.

[28] Z. Sun, W. Ying, W. Zhang, and S. Gong, "Undersampling method based on minority class density for imbalanced data," *Expert Syst. Appl.*, vol. 249, p. 123328, 2024.

[29] H. Yoon, J. Kim, and S. Lee, "Adaptive bandwidth kernel density estimation for imbalanced classification," *Pattern Recognit. Lett.*, vol. 139, pp. 12–20, 2020.

[30] J. Kang, Y. Li, and X. Liu, "Localized density ratio-based sampling for class-imbalance learning," *Neurocomputing*, vol. 438, pp. 280–292, 2021.

[31] X. Gong, L. Wang, and Y. He, "DA-SMOTE: Density-aware synthetic oversampling for imbalanced classification," *Knowl.-Based Syst.*, vol. 239, p. 107985, 2022.

[32] J. Zhang, H. Zheng, and F. Liu, "Graph-based embedding and density estimation for imbalanced learning," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023.

[33] Z. Shen, T. Wu, and C. Xu, “Kernel mean matching with hardness-aware reweighting for imbalanced classification,” *Inf. Sci.*, vol. 610, pp. 76–94, 2022.
[34] C. Chen, *An empirical study of adaptive kernel density estimation in detecting distributional overlap*. Ph.D. dissertation, Delft Univ. Technol., 2023.
[35] B. Turkoglu, “KDEBO: A kernel density estimation-guided differential evolution-based oversampling technique for imbalanced learning,” *Electronics*, vol. 11, no. 17, p. 2703, 2022.
[36] F. Kamalov, S. Moussa, and J. A. Reyes, “KDE-based ensemble learning for imbalanced data,” *Electronics*, vol. 11, no. 17, p. 2703, 2022.
[37] F. Kamalov and A. Elnagar, “Kernel density estimation-based sampling for neural network classification,” in *Proc. 2021 Int. Symp. Networks, Comput. Commun. (ISNCC)*, 2021, pp. 1–6.
[38] F. Kamalov, “Kernel density estimation based sampling for imbalanced class distribution,” *Inf. Sci.*, vol. 512, pp. 1192–1201, 2020.
[39] Y.-C. Wang and C.-H. Cheng, “A multiple combined method for rebalancing medical data with class imbalances,” *Comput. Biol. Med.*, vol. 134, p. 104527, 2021.
[40] L. Yu, R. Zhou, L. Tang, and R. Chen, “A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data,” *Appl. Soft Comput.*, vol. 69, pp. 192–202, 2018.
[41] H. Cui, Q. Wang, Y. Ye, Y. Tang, and Z. Lin, “A combinational transfer learning framework for online transient stability prediction,” *Sustain. Energy, Grids Networks*, vol. 30, p. 100674, 2022.
[42] B. Hu, Y. Liu, N. Chen, L. Wang, N. Liu, and X. Cao, “SEGCN-DCR: A syntax-enhanced event detection framework with decoupled classification rebalance,” *Neurocomputing*, vol. 481, pp. 55–66, 2022.
[43] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[44] J. Liu, Y. Sun, C. Han, Z. Dou, and W. Li, “Deep representation learning on long-tailed data: A learnable embedding augmentation perspective,” 2020, pp. 2970–2979.
[45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2017, pp. 2980–2988.
[46] J. Liu, F. Guo, H. Gao, Z. Huang, Y. Zhang, and H. Zhou, “Image classification method on class imbalance datasets using multi-scale CNN and two-stage transfer learning,” *Neural Comput. Appl.*, vol. 33, pp. 14179–14197, 2021.
[47] Z. Kang *et al.*, “Periodic variable star classification with deep learning: Handling data imbalance in an ensemble augmentation way,” *Publ. Astron. Soc. Pac.*, vol. 135, no. 1051, p. 094501, 2023.
[48] L. Xiang, G. Ding, and J. Han, “Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification,” in *Proc. Springer*, 2020, pp. 247–263.
[49] X. Wang, L. Lian, Z. Miao, Z. Liu, and S. X. Yu, “Long-tailed recognition by routing diverse distribution-aware experts,” *arXiv preprint arXiv:2010.01809*, 2020.
[50] Y. Chen, X. Yang, and H.-L. Dai, “Cost-sensitive continuous ensemble kernel learning for imbalanced data streams with concept drift,” *Knowl.-Based Syst.*, vol. 284, p. 111272, 2024.
[51] E. Atashpaz-Gargari and C. Lucas, “Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition,” in *Proc. 2007 IEEE Congr. Evol. Comput.*, 2007, pp. 4661–4667.
[52] S. Talatahari, B. F. Azar, R. Sheikholeslami, and A. Gandomi, “Imperialist competitive algorithm combined with chaos for global optimization,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 3, pp. 1312–1319, 2012.
[53] V. H. Moghaddam and J. Hamidzadeh, “New Hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier,” *Pattern Recognit.*, vol. 60, pp. 921–935, 2016.
[54] W. Pei *et al.*, “EvoSampling: A granular ball-based evolutionary hybrid sampling with knowledge transfer for imbalanced learning,” *arXiv preprint arXiv:2412.10461*, 2024.
[55] M. A. Ganaie *et al.*, “Graph embedded intuitionistic fuzzy random vector functional link neural network for class imbalance learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 11671–11680, 2024.
[56] R. Zhang *et al.*, “A density-based oversampling approach for class imbalance and data overlap,” *Comput. Ind. Eng.*, vol. 186, p. 109747, 2023.
[57] X. Tao *et al.*, “SVDD-based weighted oversampling technique for imbalanced and overlapped dataset learning,” *Inf. Sci.*, vol. 588, pp. 13–51, 2022.
[58] Z. Liu, Z. Li, Z. Yang, *et al.*, “CLIMB: Class-imbalanced learning benchmark on tabular data,” *arXiv:2505.17451*, 2025.
[59] T. Akiba *et al.*, “Optuna: A next-generation hyperparameter optimization framework,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019.
[60] D. Wilson, “Asymptotic properties of nearest neighbor rules using edited data,” *IEEE Trans. Syst., Man, Cybern.*, 1972.
[61] T. Smith and T. Martinez, “Instance hardness: A measure for hard-to-classify instances,” *IJCAI*, 2011.
[62] X.-Y. Liu, J. Wu, and Z.-H. Zhou, “EasyEnsemble: A bagging ensemble for imbalanced data,” in *Proc. IEEE Int. Conf. Data Min.*, 2008.
[63] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *Proc. IEEE Int. Joint Conf. Neural Networks*, 2008, pp. 1322–1328.
[64] R. Blagus and L. Lusa, “SMBA: Ensemble learning with synthetic sampling for imbalanced data,” *Expert Syst. Appl.*, vol. 42, no. 24, pp. 9044–9055, 2015.
[65] P. Viola and M. Jones, “Fast and robust classification using asymmetric AdaBoost,” in *Adv. Neural Inf. Process. Syst.*, 2001.