IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 November 2023, accepted 29 November 2023, date of publication 1 December 2023,
date of current version 11 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339124

RESEARCH ARTICLE

An Approach for Mining Imbalanced Datasets Combining Specialized Oversampling and Undersampling Methods

JOANNA JEDRZEJOWICZ[1] **AND PIOTR JEDRZEJOWICZ**[2]
[1]Institute of Informatics, Faculty of Mathematics, Physics and Informatics, University of Gdańsk, 80-308 Gdańsk, Poland
[2]Department of Information Systems, Gdynia Maritime University, 81-225 Gdynia, Poland

Corresponding author: Joanna Jedrzejowicz (joanna.jedrzejowicz@ug.edu.pl)

ABSTRACT The paper proposes an approach for mining imbalanced datasets combining specialized oversampling and undersampling methods. The oversampling part produces a set of non-dominated synthetic examples using two, possibly conflicting, criteria including classification potential and the distance from the borderline between minority and majority distances. The undersampling part is used to remove from the majority class examples that are likely to cause mistakes and disturbances in the process of mining. To validate the approach an extensive computational experiment has been carried. Performance of the proposed approach has been compared with that of several leading algorithms proposed for balancing minority and majority datasets. To assure fairness of comparisons a singular learner based on Gene Expression Programming (GEP) has been used in all cases. Experiment results confirmed that the proposed approach outperforms other methods investigated in the experiment.

INDEX TERMS Dominance relation, gene expression programming, imbalanced datasets, oversampling, undersampling.

I. INTRODUCTION

Many real-world problems, such as fraud detection, disease diagnosis, and rare event prediction, involve imbalanced datasets. Therefore, addressing imbalanced datasets is essential to create practical and effective machine learning solutions for these applications. Neglecting imbalanced datasets can lead to biased models, where the minority class is often misclassified or ignored. Addressing imbalances is crucial for ensuring fairness and preventing discrimination in machine learning models. In critical domains like healthcare and finance, where the cost of false positives and false negatives can be high, addressing imbalanced datasets is essential to minimize errors and make reliable predictions.. Models trained on imbalanced data without addressing the imbalance may not perform well, as they tend to be biased toward the majority class. By addressing the imbalance, one can improve the model's performance and its ability to generalize new, unseen data. In scenarios where rare events are of significant interest, like identifying defects in manufacturing or detecting security breaches, mining imbalanced datasets is essential to capture these infrequent occurrences effectively. In conclusion, mining imbalanced datasets is crucial to building effective, unbiased, and practical machine-learning models in various real-world scenarios. In this paper we propose a novel approach for mining two-classes imbalanced datasets, denoted further as DOMR. The approach uses both - oversampling and undersampling algorithms. The oversampling part is used for generating a set of non-dominated synthetic examples extending the minority class data. Oversampling aims at producing synthetic minority examples using two possibly conflicting criteria including real-valued classification potential as defined in [18], which should be maximized and, the distance from the borderline between minority and majority instances, which should be minimized. The undersampling part is used to remove from

The associate editor coordinating the review of this manuscript and approving it for publication was Chih-Yu Hsu.

the majority class data examples that are likely to cause mistakes and disturbances in the process of mining. Besides, undersampling helps in reducing computational complexity of the approach.

The proposed approach has been validated in an extensive computational experiment. In the experiment we have compared our method with several other specialized approaches which, according to our knowledge, are known to assure high quality performance when mining imbalanced datasets. As far as we know no other oversampling/undersampling procedure published in the recent years, based on a single classifier, has outperformed the best method from our set of reference methods. The main contributions of the paper include:

- Selecting and describing in a unified way a set of top-performing balancing techniques based on oversampling and undersampling, later used as reference algorithms.
- Proposing a novel technique for balancing imbalanced datasets based on two criteria – classification potential of the synthetic instance, and its average distance to nearest neighbors from majority class.
- Proposing a method for evolving and selecting synthetic examples based on the criterion of the non-domination level.
- Validating experimentally the proposed approach.

The rest of the paper is organized as follows. Section II offers description of the related work. In this section we discuss approaches to imbalanced data mining which are later used for comparisons in the reported computational experiment. Section III contains a detailed and formal description of the proposed method. Section IV reports on the computational experiment carried out to validate our approach. Finally, Section V contains conclusion and ideas for future research.

II. RELATED WORK

The existing body of knowledge and methods specializing in mining imbalanced data has been reviewed in several papers including for example [7], [10], and [19]. Algorithms designed for mining imbalanced data often use oversampling, undersampling, or a combination of both. Both techniques require changes in the data distribution aiming at balancing available datasets prior to inducing learners, and belong to the data-level methods (see [19]). Oversampling algorithms which work by increasing the number of the minority class examples are one of the most effective methods for addressing the class imbalanced problem. The idea is to produce a set of the, so-called, synthetic minority class examples that subsequently can be used to balance the dataset which is to be mined. Oversampling can be also coupled with undersampling techniques working by reducing the number of majority class instances.

According to [15], main categories of the oversampling techniques cover the following: neighborhood-based, density and probability-based, fuzzy and rough sets-based, and structure, and feature-based approaches. The most popular neighborhood-based oversampling algorithms are SMOTE and its numerous extensions. SMOTE proposed in [4] uses only minority class examples. SMOTE works by choosing instances being close in the feature space, drawing a line between them in the feature space, and locating a new synthetic example at a point along that line. Among the most popular extensions of SMOTE one can mention Borderline-SMOTE (B-SMOTE) [11], Safe-Level SMOTE (SL SMOTE) [3], Synthetic Minority Oversampling Method with Adaptive Qualified Synthesizer Selection (ASN SMOTE) [28], and a Feature-Weighted Oversampling Approach (FW-SMOTE) [21]. Neighborhood-based approaches have a subcategory known as clustering-based oversampling. Examples include [2] and [23].

Density-based algorithms concentrate on identifying regions of similar density for the minority class and locate synthetic examples in areas dense with minority class data. Examples include probability density function estimation [9], and Gaussian-SMOTE [24]. Some disadvantages of the density-based algorithms like relying only on minority class data are alleviated by Oversampling with the Majority (SWIM) proposed in [26]. K-Nearest Neighbor Oversampling approach (KNNOR) proposed in [13] considers the relative density of the entire population for generating synthetic examples.

Methods based on fuzzy and rough sets theory are expected to be suitable to deal with cases of noisy data. Examples of the approach include an oversampling method using the neighborhood rough set theory (RSFSAID) [5], and fuzzy C-means clustering (FCM-EBRB) [8].

Among structure and feature-based oversampling well-performing approaches use kernel functions [20], [22], [25]. Another effective approach belonging to the discussed category is using the concept of radial based functions [17], [18].

As can be seen from the above brief review, the body of knowledge available for dealing with imbalanced data is constantly growing and, over the years, becomes more advanced and mature, assuring better performance when dealing with real problems. In what follows the algorithms used in the subsequent validating experiments are described in a unified way. The first one is SMOTE which is one of the most popular class imbalance learning since it was first introduced by Chawla et al. [4]. The idea is to oversample the data from the minority class by generating synthetic instances using linear interpolation of minority instances and their neighbors, details shown in **Algorithm** 1.

In case of ADASYN [12], which is based on SMOTE, the use of minority instances is differentiated by weights which decide how many times each minority instance is used. More

Algorithm 1 SMOTE

```
Require: data from minority class $minC$, parameter $n$ - number of new instances, parameter $k$ - number of neighbors
Ensure: extended minority $exMin$
  $exMin \leftarrow minC$
  for $i = 1$ to $n$ do
    select random minority instance $x \in minC$
    select random $k$-neighbor $x' \in minC$ of $x$
    generate a new instance $x_{new} = x + \delta(x' - x)$, where random $\delta \in (0, 1)$
    append $x_{new}$ to $exMin$
  end for
  return $exMin$
```

synthetic data is generated for minority class examples that are harder to learn that is have more majority instances in their neighborhood, see **Algorithm** 2.

Algorithm 2 ADASYN

```
Require: data from minority class $minC$, data from majority class $majC$, parameter $n$ - number of new instances, parameter $k$ - number of neighbors
Ensure: extended minority $exMin$
  $exMin \leftarrow minC$
  for $i = 0$ to $|minC|$ do
    $x \leftarrow i$-th instance from $minC$
    $\Delta_x \leftarrow$ number of $k$-neighbors of $x$ from the majority class
    $r_x \leftarrow \Delta_x / k$ {normalize $r_x$ to $r'_x$}
    $g_x \leftarrow r'_x \cdot n$ {$g_x$ decides how many times $x$ will be used}
    for $i = 0$ to $g_x$ do
      $z_x \leftarrow$ random $k$-neighbor of $x$
      $\bar{x} \leftarrow x + (z_x - x) \cdot \delta$, random $\delta \in (0, 1)$
      append $\bar{x}$ to $exMin$
    end for
  end for
  return $exMin$
```

When using LAMO [27] only so-called boundary instances take part in generation of synthetic minority instances. They are sampling seeds identified according to the distribution of majority and minority instances in the neighborhood. The local distribution of each seed is examined according to the distance to the nearest minority and majority instance. Two parameters k_1, k_2, defining respectively, the number of neighbors for minority and majority instances, are used. For $x \in minC$ let $NMin(x)$ stand for the set of k_1-neighbors of x, similarly for $x \in majC$ the k_2 neighbors of x define $NMaj(x)$. **Algorithm** 3 describes the first step of LAMO which allows to select minority instances which are borderline ones. In the second step Gaussian Mixture Model is applied to model the probability density function for sampling seeds from *Border*. Finally, similarly as in SMOTE, applying linear interpolation to *Border*, new minority instances are generated. More details are in [27]. Radial-based approach to oversampling RBO, introduced in [18], makes use of the potential estimation to generate new minority instances. The potential of instance x is defined as

$$\Phi(x, majC, minC, \gamma) = \sum_{y \in majC} e^{-(dist(x,y)/\gamma)^2)} - \sum_{y \in minC} e^{-(dist(x,y)/\gamma)^2)} \quad (1)$$

where γ is a parameter which represents the spread of radial based function. In the experiments **Algorithm** 4 was used. CSMOUTE introduced in [16] is a combined method using

Algorithm 3 LAMO - Generating Borderline Instances

```
Require: data from minority class $minC$, data from majority class $majC$, parameter $k_1$ - number of neighbors for instances from $minC$, parameter $k_2$ - number of neighbors for instances from $majC$,
Ensure: $Border$ - border line instances from minority class
  $S^{bor}_{maj} \leftarrow \{q : q \in majC \;\&\; q \in NMin(x) \;\&\; x \in minC\}$
  $S^{bor}_{min} \leftarrow \{x \in minC : x \in NMax(q) \;\&\; q \in S^{bor}_{maj}\}$
  for $x \in S^{bor}_{min}$ do
    $DIS_{min}(x) = dist(x, z^x_{min})$ where $z^x_{min}$ is the closest minority instance
    $DIS_{maj}(x) = dist(x, z^x_{maj})$ where $z^x_{maj}$ is the closest majority instance
    $DIFF(x) = |DIS_{maj}(x) - DIS_{min}(x)|$
  end for
  $DIFF \leftarrow \{DIFF(x) : x \in S^{bor}_{min}\}$
  $\mu(DIFF) \leftarrow$ mean of $DIFF$
  $\sigma(DIFF) \leftarrow$ standard deviation of $DIFF$
  $Border \leftarrow \{x \in S^{bor}_{min} : DIFF(x) \leq \mu(DIFF) + 3\sigma(DIFF)\}$
  return $Border$
```

Algorithm 4 RAD-Radial-Based Oversampling

```
Require: data from minority class $minC$, data from majority class $majC$, parameter $n$ - number of new instances, parameter $it$ - number of iterations, $p$-probability of interrupting iterations, $step$ - optimization step
Ensure: extended minority $exMin$
  $exMin \leftarrow minC$
  for $i = 1$ to $n$ do
    select random minority instance $x \in minC$
    for $j = 1$ to $it$ do
      $dir \leftarrow$ random basis vector of size equal to number of attributes
      $sign \leftarrow$ random value from $\{-1, 1\}$
      $x_{new} \leftarrow x + dir \cdot sign \cdot step$
      if $|\phi(x_{new}, majC, minC, \gamma)| < |\phi(x, majC, minC, \gamma)|$ then
        $x \leftarrow x_{new}$
      end if
      stop iterations with probability $p$
    end for
    append $x_{new}$ to $exMin$
  end for
  return $exMin$
```

SMOTE for oversampling and SMUTE for undersampling. It is shown in **Algorithm** 5.

```
Algorithm 5 CSMOUTE- Combined Undersampling and Oversampling
Require: data from minority class $minC$, data from majority
    class $majC$, parameter $k$ - number of neighbors, parameter
    $r$-ratio of data balancing
Ensure: balanced data set
    $m \leftarrow |majC| - |minC|$ { $m$ defines the size difference}
    $n \leftarrow m \cdot r$ {$n$ synthetic instances are added to $minC$}
    apply SMOTE to $minC$, $majC$, $n$, $k$ to output $exMin$
    {perform SMUTE to undersample $majC$}
    $majN \leftarrow majC$
    $redMaj \leftarrow \emptyset$
    for $i = 1$ to $n$ do
        select random instance $x_1 \in majN$
        select random $k$-neighbor $x_2 \in majN$ of $x_1$
        $x_{new} \leftarrow x_1 + \delta \cdot (x_2 - x_1)$, random $\delta \in (0, 1)$
        delete $x_1, x_2$ from $majN$
        append $x_{new}$ to $redMaj$
    end for
    return $exMin \cup redMaj$
```

The algorithm FW-SMOTE [21] is another method based on SMOTE. The main assumption is that when applying interpolation the importance of attributes should be varied by the introduction of weights. First, Fisher score of each attribute is calculated as the difference of means of the attributes in each of two classes and normalized by the standard deviation:

$$FS(att) = \frac{|\mu_{att}^{maj} - \mu_{att}^{min}|}{(\sigma_{att}^{maj})^2 + (\sigma_{att}^{min})^2} \tag{2}$$

where μ_{att}^{maj} and μ_{att}^{min} are the mean values of the attribute att in, respectively, majority and minority data and similarly for σ_{att}^{maj}, σ_{att}^{min} being the standard deviations. The attributes are sorted in the descending order of Fisher score FS. Assuming sorted order of attributes, weights are calculated as:

$$weight(att) = (\frac{att}{N})^{\alpha} - (\frac{att-1}{N})^{\alpha} \tag{3}$$

for $att = 1, \ldots N$, where N is the number of attributes, α is an input parameter. The distance between two instances x, y of size N is defined as weighted Minkowski distance, which is

$$dist(x, y) = (\sum_{att=1}^{N} weight(att) \cdot |x_{att} - y_{att}|^p)^{1/p} \tag{4}$$

where p is an input parameter.

Recently introduced SMOTE-RkNN [29] makes use of reverse k-nearest neighbors. As the authors claim, the algorithm identifies noise based on probability density rather than local neighborhood information. The algorithm starts with the application of SMOTE to balance the dataset. Then separately for each of two classes, the number of reverse k-nearest neighbors for each instance is counted. Finally, the probability density of an instance within its class is compared with that in the other class and this allows to estimate whether an instance (both majority and minority) is noise, or not. The instances classified as not noise are appended to the final dataset. It is shown as **Algorithm** 7.

```
Algorithm 6 FW-SMOTE
Require: $N$-number of attributes, data from minority class
    $minC$, data from majority class $majC$, parameter $n$ - number
    of new instances, parameter $k$ - number of neighbors,
Ensure: extended minority $exMin$
    for $att = 1$ to $N$ do
        calculate Fisher score $FS(att)$ using (2)
    end for
    sort attributes in descending order according to Fisher score
    for $att = 1$ to $N$ do
        calculate $weight(att)$ using (3)
    end for
    for $i = 1$ to $n$ do
        select random minority instance $x \in minC$
        set $k$- nearest neighbors of $x$ using Minkowski distance (4)
        select random $k$-neighbor $x_n$ of $x$
        generate a new instance $x_{new} = x + \delta(x_n - x)$, where
        random $\delta \in (0, 1)$
        append $x_{new}$ to $exMin$
    end for
    return $exMin$
```

III. THE PROPOSED APPROACH

The proposed approach, named DOMR, is an extension of the algorithm proposed in the earlier work of the authors [14], in particular a new scheme for undersampling part has been used. The approach uses domination relation between instances and genetic algorithms to oversample minority objects in an iterative process. The relation of domination $\prec$ defined for any two instances (rows) makes use of two criteria. Assume majority objects $majC$, minority objects $minC$ fixed. The first criterion makes use of potential as used in (1). For any two instances x, y we write:

$$x \prec_1 y \Longleftrightarrow \phi(x, majC, minC, \gamma) < \phi(y, majC, minC, \gamma) \tag{5}$$

The second criterion employs the average distance of an instance to 25% of nearest neighbors from the majority class. For a fixed instance x and fixed majority dataset $majC$ let $\{x_1, \ldots, x_n\}$ stand for the 25% of nearest neighbors of x from $majC$. Define:

$$\begin{gathered} distMajority(x, majC) = \sum_{i=1}^{n} dist(x, x_i)/n \\ x \prec_2 y \Longleftrightarrow distMajority(x, majC) < distMajority(y, majC) \end{gathered} \tag{6}$$

Algorithm 7 SMOTE - RkNN

```
Require: $D = majC \cup minC$ - initial dataset, parameter $k$ - number of neighbors, control parameter $\lambda$
Ensure: balanced dataset $BD$
  perform SMOTE to balance dataset to $Dn \supset D$
  for $i = 1$ to $|Dn|$ do
    for $j = 0, 1$ do
      $\xi^j(i) \leftarrow$ number of reverse k-nearest neighbors of $i$-th instance from class $j$
    end for
    for $j = 0, 1$ do
      $\overline{\xi}^j \leftarrow$ normalized $\xi^j$
    end for
    $BD \leftarrow \emptyset$
    for $i = 1$ to $|Dn|$ do
      if $i$-th instance in class 0 then
        if $\overline{\xi}^0(i) < \lambda \cdot \overline{\xi}^1(i)$ then
          append $i$-th instance to $BD$
        else if $\overline{\xi}^1(i) < \lambda \cdot \overline{\xi}^0(i)$ then
          append $i$-th instance to $BD$
        end if
      end if
    end for
  end for
  return $BD$
```

Finally, x dominates y iff

$$x \prec y \iff x \prec_1 y \;\&\; x \prec_2 y \tag{7}$$

Balancing the dataset makes use of a genetic algorithm where the objects of the population are minority synthetic instances and the fitness function is defined with the help of non-dominating levels. For the population P a fast-non-dominated-sort algorithm from [6] is applied to define non-domination level of each member of the population. To do that, for each $p \in P$ three values are calculated: S_p - the set of population members dominated by p, n_p - domination count, which is the number of population members that dominate p, $rank(p)$ - the level of non domination of p, and the value of the fitness function at the same time. The details of calculating values of non-domination level are in **Algorithm** 8.

Genetic algorithm uses mutation and crossover operations. Assume that data contain m attributes. Mutation operation is defined in a standard way, that is for a population member x, attribute number $att \leq m$ is randomly chosen, a random value atV from the domain of att is drawn and a new population member $\bar{x}$ is generated which is equal to x except the value of att which is replaced by atV.

As for the crossover two versions are used. In case of one-point crossover, for two members $x = x_1, \ldots, x_m$ and $y = y_1, \ldots, y_m$ of the population a random cutting place is chosen $1 < l \leq m$ and new members $\bar{x} = x_1, \ldots, x_{l-1}, y_l, \ldots, y_m$ and $\bar{y} = y_1, \ldots, y_{l-1}, x_l, \ldots, x_m$ are generated. In case of two point crossover, two cutting places $1 < l < p \leq m$ are drawn and new members are $\bar{x} = x_1, \ldots, x_{l-1}, y_l, \ldots, y_{p-1}, x_p, \ldots, x_m$ and $\bar{y} = y_1, \ldots, y_{l-1}, x_l, \ldots, x_{p-1}, y_p, \ldots, y_m$ are added to the population. After each application of the genetic algorithm, those objects of the population which are on the first level of non-domination are attached to the new minority dataset. The details of the genetic algorithm to balance the dataset are in **Algorithm** 9.

Algorithm 8 Calculation of Non Domination Levels

```
Require: domination relation for members of the population $P$
Ensure: non domination levels $\{rank(p) : p \in P\}$
  $\mathcal{F}_1 = \emptyset$
  for all $p \in P$ do
    $S_p = \emptyset$
    $n_p = 0$
    for all $q \in P$ do
      if $p \prec q$ then
        $S_p = S_p \cup q$
      else if $q \prec p$ then
        $n_p += 1$
      end if
    end for
    if $n_p = 0$ then
      $rank(p) = 1$
      $\mathcal{F}_1 = \mathcal{F}_1 \cup p$
    end if
  end for
  $i = 1$
  while $\mathcal{F}_i \neq \emptyset$ do
    $Q = \emptyset$
    for all $p \in \mathcal{F}_i$ do
      for all $q \in S_p$ do
        $n_q = n_q - 1$
        if $n_q = 0$ then
          $rank(q) = i + 1$
          $Q = Q \cup q$
        end if
      end for
    end for
    $i = i + 1$
    $\mathcal{F}_i = Q$
  end while
  return values of $rank$ function
```

To reduce the size of the majority class instances the algorithm shown as **Algorithm** 10 was used. It starts with generating the centroid of minority instances which is then used to delete those majority instances which are closest.

The overall architecture of the proposed method for mining imbalanced datasets based on dominance relation is shown as **Algorithm** 11. In Figure 1 the workflow of the proposed approach is shown.

Algorithm 9 DOM-Balancing the Dataset With Domination Relation

```
Require: $majC$ majority dataset, $minC$- minority dataset, $n$ - balancing parameter
Ensure: extended minority dataset $newMin$
  $newMin \leftarrow \emptyset$
  $population \leftarrow$ randomly generated $n$ rows from minority class
  while $|newMin| < n$ do
    {prepare domination matrix}
    for all $x, y \in population$ do
      $domin(x, y) \leftarrow x \prec y$
    end for
    using $domin$ determine $\{rank(x) : x \in population\}$ (Algorithm 8)
    $newMin \leftarrow newMin \cup \{x \in population : rank(x) = 1\}$
    for all $x \in population$ do
      $fitness(x) \leftarrow rank(x)$
    end for
    use $fitness$ and roulette to generate $newPopulation$
    $population \leftarrow$ apply mutation and crossover to $newPopulation$
  end while
  return $newMin$
```

Algorithm 10 Undersampling With Minority Class Centroid

```
Require: data from majority class $majC$, data from minority class $minC$, parameter $s$ - size of reduced majority class.
Ensure: reduced majority class $redMaj \subset majC$ of size $s$.
  calculate centroid $CN$ of $minC$
  define distances of $CN$ to majority instances

      $DIST = \{dist(x, CN) : x \in majC\}$

  sort $DIST$ in ascending order $SORT = \{d_1, \ldots, d_{|majC|}\}$
  keep in reduced majority class instances whose distances are in the initial $s$ segment of $DIST$

      $redMaj = \{x \in majC : dist(x, CN) \leq d_s\}$

  return $redMaj$
```

A. COMPUTATIONAL COMPLEXITY ANALYSIS

To estimate the computational complexity of the proposed approach we use the following observations:

- as follows from [6], establishing fitness of the population members requires $O(N^2)$, where N is population size; since after each application of genetic algorithm at least one instance is added to the new minority set, therefore at most N iterations are performed and the complexity of **Algorithm** 9 is $O(N^3)$,
- in case of **Algorithm** 10, the calculation of distances of majority instances to the centroid requires $O(M)$ steps, where M is the number of majority instances, and sorting requires $O(M \log M)$ steps,
- since N is less than the size of the minority data, both N and M are bounded by the size of the dataset, and the

Algorithm 11 Proposed Approach

```
Require: dataset $D = majC \cup minC$, $rL$- reduction level.
Ensure: performance measures F1, AUC, G- geometric mean.
  $redMaj \leftarrow$ result of Algorithm 10 on $D$, reducing majority class size by $rL$
  $n \leftarrow |redMaj| - |minC|$ {$n$ -number of data to balance majority and minority class}
  $newMin \leftarrow$ result of Algorithm 9 applied to dataset $redMaj \cup minC$, with parameter $n$
  $D \leftarrow redMaj \cup minC \cup newMin$ { $D$ is balanced dataset}
  perform 5-CV scheme of GEP classification on $D$
  return performance measures F1, AUC, G-geometric mean
```

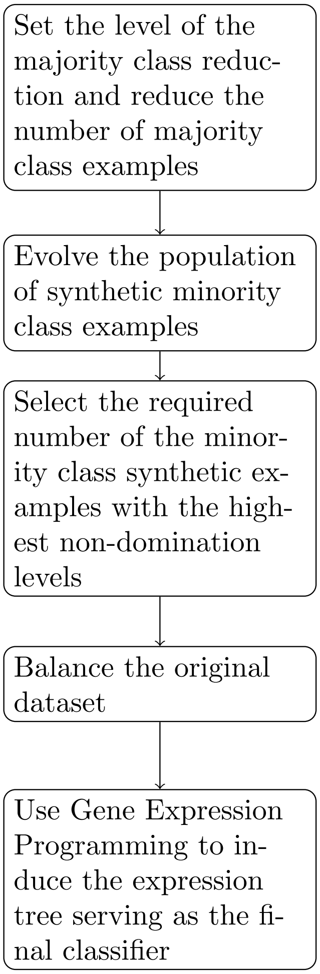
Set the level of the majority class reduction and reduce the number of majority class examples
Evolve the population of synthetic minority class examples
Select the required number of the minority class synthetic examples with the highest non-domination levels
Balance the original dataset
Use Gene Expression Programming to induce the expression tree serving as the final classifier

FIGURE 1. The workflow of the proposed approach.

complexity of **Algorithm** 11 is $O(D^3)$, where D is the size of the considered dataset.

IV. COMPUTATIONAL EXPERIMENT

A. DATASETS

To validate the proposed approach we have carried out an extensive computational experiment. It involved 70 imbalanced datasets from Keel-Dataset Repository [1]. Main characteristics of the discussed datasets are shown in Table 1.

B. EXPERIMENT PLAN

The performance of the proposed approach has been compared with the performance of several state-of-the-art oversampling methods and, additionally, one "classic" technique - ADASYN. Details of the above algorithms are given in the Related Work section. The notation used in further comparisons consists of three parts – acronym of the method, letter R, present only in case the proposed majority class reduction algorithm has been used, and symbol of the performance metric used. Thus, for example, notation "ADAR G" refers to value of the geometric mean produced by the ADASYN algorithm supplemented by the proposed majority class data reduction algorithm.

All subsequently reported experiment results refer to averages calculated over 30 results obtained by running 6 independent repetitions of the 5-CV scheme. In the experiment, for each oversampling method, three performance measures including F1 metric, area under the receiver operating characteristic curve (AUC), and geometric mean (G) have been calculated. Besides, for each oversampling method and each performance measure results have been obtained for two variants – with and without the proposed algorithm for majority class data reduction. When the majority data reduction procedure is applied we use the same level of reduction for all considered methods as in the DOMR. Altogether, results of 42 experiment configurations with 70 cases each, have been produced and, subsequently, reported.

To assure fairness of the performance comparison between the investigated methods we have decided to use a single classifier with the default settings to perform the classification task after the respective oversampling/undersampling method has produced balanced datasets with equal number of examples in the minority and majority classes. As the classifier, we selected the Gene Expression Programming (GEP) technique producing binary classifiers in the form of expression trees. Our motivation for using GEP is two-fold. Firstly, GEP is known to produce good classification results when used as a classifier. Secondly, GEP produced classifiers in the form of expression trees are easily explainable.

C. PARAMETERS

All investigated methods have been implemented by the authors in Java using the Eclipse platform. The software runs

TABLE 1. Datasets used in the reported experiments.

Dataset	#Attr	#Inst	IR
ecoli-0_vs_1	7	220	1.82
ecoli-0-4-6-vs-5	6	203	9.15
wisconsin	9	683	1.86
ecoli-0-2-6-7_vs_3-5	7	224	9.18
pima	8	768	1.87
ecoli-0-3-4-7-vs-5-6	7	257	9.28
glass0	9	214	2.06
ecoli-0-6-7-vs-5	6	220	10.00
haberman	3	306	2.78
ecoli-0-1-4-7-vs-2-3-5-6	7	336	10.59
vehicle3	18	846	2.99
led7digit-0-2-4-5-6-7-8-9-vs-1	7	443	10.97
glass-0-1-2-3_vs_4-5-6	9	214	3.20
ecoli-0-1-vs-5	6	240	11.00
ecoli1	7	336	3.36
glass-0-1-4-6-vs-2	9	205	11.06
new-thyroid1	5	215	5.14
ecoli-0-1-4-7-vs-5-6	6	332	12.28
new-thyroid2	5	215	5.14
cleveland-0-vs-4	13	177	12.62
ecoli2	7	336	5.4
ecoli-0-1-4-6-vs-5	6	280	13.00
glass6	9	214	6.38
dermatology-6	34	358	16.90
yeast3	8	1484	8.10
shuttle-6_vs_2-3	9	230	22.00
ecoli3	7	336	8.60
lymphography-normal-fibr	18	148	23.67
yeast-0-5-6-7-9-vs-4	8	528	9.35
flare-F	11	1066	23.79
vowel0	13	988	9.98
car-good	6	1728	24.04
glass-0-1-6-vs-2	9	192	10.29
car-vgood	6	1728	24.58
glass2	9	214	11.59
kr-vs-k-zero-one_vs_draw	6	2901	26.63
yeast-1-vs-7	7	459	14.3
winequality-red-4	11	1599	29.17
glass4	9	214	15.47
poker-9_vs_7	10	244	29.50
ecoli4	7	336	15.80
winequality-white-9_vs_4	11	168	32.60
page-blocks-1-3-vs-4	10	472	15.86
kr-vs-k-three-vs-eleven	6	2935	35.23
abalone9-18	8	731	16.40
winequality-red-8_vs_6	11	656	35.40
glass-0-1-6_vs_5	9	184	19.44
abalone-17_vs_7-8-9-10	8	2338	39.31
glass5	9	214	22.78
abalone-21_vs_8	8	581	40.50
yeast-2-vs-8	8	482	23.10
winequality-white-3_vs_7	11	900	44.00
yeast4	8	1484	28.10
winequality-red-8_vs_6-7	11	855	46.50
yeast-1-2-8-9-vs-7	8	947	30.57
abalone-19_vs_10-11-12-13	8	1622	49.69
yeast5	8	1484	32.73
kr-vs-k-zero_vs_eight	6	1460	53.07
ecoli-0-1-3-7-vs-2-6	7	281	39.14
winequality-white-3-9_vs_5	11	1482	58.28
yeast6	8	1484	41.40
poker-8-9_vs_6	10	1485	58.40
Abalone19	8	4184	99.44
winequality-red-3_vs_5	11	691	68.10
yeast-0-3-5-9-vs-7-8	8	506	9.12
abalone-20_vs_8-9-10	8	1916	72.69
yeast-0-2-5-7-9-vs-3-6-8	8	1004	9.14
poker-8-9_vs_5	10	2075	82.00
yeast-0-2-5-6_vs_3-7-8-9	8	1004	9.14
poker-8_vs_6	10	1477	85.88

under the open source software license and is available from the authors at request.

In the case of the method for mining imbalanced datasets based on dominance relation the algorithm is parameter-free and returns the required number of synthetic examples to balance majority and minority classes. Parameters for the remaining methods have been set at values used or suggested by their authors in original papers.

Apart from using 7 balancing techniques described in Section III we have decided to consider additionally the possibility of reducing the number of the majority class examples. Such a reduction influences both – the computational time required by a balancing technique and the subsequent performance of the classifier used. From the computational complexity analysis, it is clear that reducing the number of examples in the majority class set of examples reduces the computational time needed to balance minority and majority sets. The influence of reducing the majority class size on subsequent classifier performance is not, however, straightforward. A preliminary study carried out to identify the relation between the reduction level of the majority set has shown that in a majority of cases, a reasonable reduction does not influence negatively the classifier performance and may even improve it. The above claim is also supported by the results of the computational experiment reported in this paper. Based on the preliminary study results we have decided to use the following heuristic rules for controlling the majority set reduction level when using all considered balancing techniques for all datasets.

- 10% reduction for problems with the overall number of instances (#inst.) smaller than 300.
- 20% or 30% reduction for problems with 300 < #inst. < 600.
- 30% or 40% reduction for problems with 601 < #inst. < 1200.
- 50% or 60% reduction for problems with 1201 < #inst. < 1800.
- 70% or 80% reduction for problems with 1801 < #inst. < 2500.
- 80% or 90% reduction for problems with #inst. greater than 2501.

The sequence of actions performed in the process of solving the imbalanced dataset mining problem with the majority set reduction includes the following three steps:

1) Using the proposed heuristics rules and **Algorithm** 10 reduce the majority set size.
2) Perform data balancing using one of the considered techniques.
3) Use Gene Expression Programming to mine the balanced dataset.

Gene Expression Programming classifier used in the experiment requires setting the value of several parameters. In the experiment, for all considered methods and variants, the GEP classifier has been used with the following parameter value settings: population size – 200; number of iterations – 200; probabilities of mutation, RIS transposition, IS transposition, 1-point and 2-point recombination – 0.5, 0.2, 0.2, 0.2, 0.2, respectively. For selection the roulette wheel method has been used.

TABLE 2. Experiment results – F1 performance measure.

Variable	Av. Rank	Mean	St.Dev.
1 DOMR F1	11,4214	0,9621	0,0377
2 DOM F1	10,8143	0,9406	0,0530
3 CSMR F1	10,8786	0,9369	0,0557
4 RAD F1	10,2357	0,9324	0,0631
5 ADA F1	8,8429	0,9183	0,0761
6 CSM F1	8,2786	0,9158	0,0797
7 FWS F1	7,7786	0,9137	0,0805
8 RNN F1	7,6857	0,9007	0,0842
9 RNNR F1	6,8286	0,8981	0,0850
10 LAMR F1	6,5571	0,8982	0,0882
11 FWSR F1	5,0571	0,8843	0,0918
12 RADR F1	4,5357	0,8804	0,0918
13 LAM F1	4,1214	0,8714	0,1010
14 ADAR F1	1,9643	0,8425	0,1040

TABLE 3. Experiment results – AUC performance measure.

Variable	Av. Rank	Mean	St.Dev.
1 DOMR AUC	11,1714	0,9403	0,0541
2 DOM AUC	11,1143	0,9368	0,0552
3 RNNR AUC	10,6786	0,9333	0,0622
4 ADAR AUC	9,1429	0,9207	0,0750
5 RADR AUC	9,0429	0,9158	0,0867
6 CSMR AUC	8,2857	0,9122	0,0833
7 FWSR AUC	8,0286	0,9153	0,0793
8 RAD AUC	7,7786	0,9007	0,0842
9 RNN AUC	7,3071	0,9003	0,0834
10 CSM AUC	6,3286	0,8949	0,0888
11 ADA AUC	5,1357	0,8847	0,0911
12 FWS AUC	4,5071	0,8813	0,0907
13 LAMR AUC	4,3071	0,8742	0,0999
14 LAM AUC	2,1714	0,8439	0,1017

D. EXPERIMENT RESULTS

In Tables 2 – 4 experiment results for each considered performance measure are shown. Results have been sorted according to mean performance measure value from the best to the worst one. Apart from the respective mean value we show average rank and standard deviation of results produced by considered methods. To check whether there are significant differences among results produced by different methods we use the Friedman ANOVA test. The null hypothesis is that there are no such differences. The respective values of the Chi-square statistics (N = 70, df = 13) and p-values are shown in Table 5.

Data from Table 5 allow to observe that for each performance measure null hypotheses do not hold and there are significant differences between results produced by different methods at significance level of 0.05.

Since the Friedman test does not tell which of the investigated methods contributes most to differences among results we have also performed the Nemenyi test to determine exactly which groups of methods produce statistically significant differences of means. The Nemenyi test is a post-hoc

TABLE 4. Experiment results – G performance measure.

Variable	Av. Rank	Mean	St.Dev.
1 DOMR G	11,1714	0,9403	0,0541
2 DOM G	11,1143	0,9368	0,0552
3 RNNR G	10,6786	0,9333	0,0622
4 ADAR G	9,1429	0,9207	0,0750
5 RADR G	9,0429	0,9158	0,0867
6 CSMR G	8,2857	0,9122	0,0833
7 FWSR G	8,0286	0,9153	0,0793
8 RAD G	7,7786	0,9007	0,0842
9 RNN G	7,3071	0,9003	0,0834
10 CSM G	6,3286	0,8949	0,0888
11 ADA G	5,1357	0,8847	0,0911
12 FWS G	4,5071	0,8813	0,0907
13 LAMR G	4,3071	0,8742	0,0999
14 LAM G	2,1714	0,8439	0,1017

TABLE 5. Friedman test results.

Performance measure	Chi-square statistics	p-value
F1	426.0436	0.000000
AUC	391.5576	0.000000
G	413.6753	0.000000

TABLE 6. Nemenyi post hoc test results for F1 performance measure.

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	x	0	0	0	1	1	1	1	1	1	1	1	1	1
2		x	0	0	1	1	1	1	1	1	1	1	1	1
3			x	0	1	1	1	1	1	1	1	1	1	1
4				x	0	1	1	1	1	1	1	1	1	1
5					x	0	0	0	1	1	1	1	1	1
6						x	0	0	0	0	1	1	1	1
7							x	0	0	0	1	1	1	1
8								x	0	0	1	1	1	1
9									x	0	1	1	1	1
10										x	0	1	1	1
11											x	0	0	1
12												x	0	1
13													x	1
14														x

test that compares multiple models after a significant result from Friedman's test. The null hypothesis for Nemenyi is that there is no difference between any two methods, and the alternative hypothesis is that at least one pair of methods performs differently. The test proved that in the investigated case, the null hypothesis has to be rejected and the alternative hypothesis holds true. Results of the Nemenyi test are shown in Tables 6 – 8. In these tables relations between performance of each pair of methods are denoted by 0 or 1. Relation 0 tells that both methods produce statistically similar results, while relation 1 tells that the respective method from column # produces statistically different means. The order of methods in Tables 6 – 8 corresponds strictly to the order of methods in Tables 2 – 4.

E. EXPERIMENT RESULTS ANALYSIS

Data shown in Tables 2 - 4 allow to observe that a clear winner among considered methods and their variants is oversampling for mining imbalanced datasets based on dominance relation (proposed in this paper) and denoted DOMR. The above finding is true for all three performance measures considered.

TABLE 7. Nemenyi post hoc test results for AUC performance measure.

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	x	0	0	1	1	1	1	1	1	1	1	1	1	1
2		x	0	1	1	1	1	1	1	1	1	1	1	1
3			x	0	0	1	1	1	1	1	1	1	1	1
4				x	0	0	0	0	1	1	1	1	1	1
5					x	0	0	0	0	1	1	1	1	1
6						x	0	0	0	1	1	1	1	1
7							x	0	0	0	1	1	1	1
8								x	0	0	1	1	1	1
9									x	0	1	1	1	1
10										x	0	1	1	1
11											x	0	0	1
12												x	0	1
13													x	1
14														x

TABLE 8. Nemenyi post hoc test results for G performance measure.

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	x	0	0	0	1	1	1	1	1	1	1	1	1	1
2		x	0	0	1	1	1	1	1	1	1	1	1	1
3			x	0	0	1	1	1	1	1	1	1	1	1
4				x	0	0	1	1	1	1	1	1	1	1
5					x	0	0	0	1	1	1	1	1	1
6						x	0	0	0	1	1	1	1	1
7							x	0	0	0	1	1	1	1
8								x	0	0	1	1	1	1
9									x	0	0	1	1	1
10										x	0	1	1	1
11											x	0	0	1
12												x	0	1
13													x	1
14														x

In the case of F1 measure, DOMR F1 has obtained mean score over 70 considered datasets of 0.962 with the Standard Deviation of results at 0.037 and the average rank equal to 11.42 out of 14.00 possible. The second best performer has been DOM F1 with scores 0.946, 10.81, and 0.053, respectively. The third best performer has been the CSMR F1 with scores 0.93, 10.87, and 0.055. The worst performer, out of 14 considered techniques, has been ADA F1. In terms of the AUC measure two leading methods have not changed with the DOMR AUC obtaining a mean scores of 0.94, 11.71, and 0.054 followed by the DOM AUC with scores 0.9368, 11.11, and 0.055.The worst performer for the considered measure has been LAM AUC. The picture has not changed much in the case of G measure. Again DOMR G and DOM G have been leading with LAM G offering the worst performance Satisfactory performance, in addition to the leaders, with mean F1 score above 0.9 obtained in the reported experiment, has been achieved by RAD F1, ADA F1, CSM F1, FWS F1 and RNN F1 methods. Similar group closely following the leaders with the AUC measure above 0.9 have consisted of RNNR, ADAR, RADR, CSMR, FWSR, RAD, and RNN. Similar group of methods with satisfactory performance above 0.9 for the G performance measure, included RADR, RNNR, ADAR, CSMR, FWSR, and RAD. For the F1 performance measure, the Nemenyi post hoc test allows us to draw the following conclusion true at the significance level of 0.05: DOMR, CSMR, and DOM perform significantly better than ADA, CSM, FWS, RNNR, LAMR, FWSR, RADR, LAM, and ADAR. For the AUC performance measure the Nemenyi post hoc test allows us to draw the following conclusion true at the significance level of 0.05: DOM and DOMR perform significantly better than ADAR, RADR, CSMR, FWSR, RAD, RNN, CSM, ADA, FWS, LAMR, and LAM. For the G performance measure, the Nemenyi post hoc test allows us to draw the following conclusion true at the significance level of 0.05: DOM and DOMR perform significantly better than ADAR, CSMR, FWSR, RAD, CSM, RNN, ADA, FWS, LAMR, and LAM. Overall, DOMR, and DOM outperform in terms of the number of wins among all the remaining balancing techniques considered in this paper.

The reported experiment has also confirmed that the proposed approach for reducing the number of examples in the majority set is usually advantageous considering the classification performance.

V. CONCLUSION

The main contribution of the paper is proposing and validating an approach for mining two classes imbalanced datasets, based on oversampling where non-dominated synthetic examples are generated, and undersampling of the majority class examples. Non-dominated synthetic examples are generated using two criteria - real-valued classification potential, and distance from the borderline between minority and majority instances, while majority class examples are reduced by discarding those which are closest to the centroid of the minority class data.

The proposed method has been validated in an extended computational experiment with 70 examples from the Keel repository of imbalanced datasets serving as the testbed. In the experiment, the performance of our method named DOMR has been compared with the performance of 6 other oversampling approaches known to offer good performance when dealing with imbalanced datasets. DOMR, in nearly all cases, outperformed statistically all other methods. The finding is true for each of the 3 considered performance metrics including the F1 measure, the area under the receiver operating characteristic curve, and the geometric mean. In the subsequent ranking of methods, the version of the proposed approach without the undersampling part takes the second place followed by Combined Synthetic Oversampling and Undersampling Technique, Feature Weighted Synthetic Minority Oversampling Technique, and Hybrid Resampling Method based on SMOTE and reverse k-nearest neighbors. The last two methods on 4th and 5th place depending on the performance measure used.

Another important finding concerns the beneficial role of the proposed undersampling algorithm. In a vast majority of cases supplementing the original oversampling method with the proposed undersampling algorithm brings an improvement in performance providing the reduction is set at a moderate level.

Directions of future research could include an extension of the two-classes version into multiple classes imbalanced dataset mining. We would also like to investigate the influence of using different learners from the performance point of view. Another possible direction of research is considering more than two criteria when producing synthetic examples within an oversampling process.

In this paper, we did not study numerous possible extensions of the proposed approach. Applying specialized techniques for feature selection, engineering, as well as ensemble methods like bagging, boosting, or stacking could bring further benefits in terms of learner performance. Techniques for dealing with missing data could be also used when needed. Such extensions would not, however, change the core of the proposed method.

REFERENCES

[1] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, Feb. 2009.

[2] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE—Majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.

[3] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Advances in Knowledge Discovery and Data Mining*, T. Theeramunkong, B. Kijsirikul, N. Cercone, and T.-B. Ho, Eds. Berlin, Germany: Springer, 2009, pp. 475–482.

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

[5] H. Chen, T. Li, X. Fan, and C. Luo, "Feature selection for imbalanced data based on neighborhood rough sets," *Inf. Sci.*, vol. 483, pp. 1–20, May 2019.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[7] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera," *Learning from Imbalanced Data Sets*. Cham, Switzerland: Springer, 2018.

[8] Y.-G. Fu, J.-F. Ye, Z.-F. Yin, L.-J. Chen, Y.-M. Wang, and G.-G. Liu, "Construction of EBRB classifier for imbalanced data based on fuzzy C-means clustering," *Knowl.-Based Syst.*, vol. 234, Dec. 2021, Art. no. 107590.

[9] M. Gao, X. Hong, S. Chen, and C. J. Harris, "Probability density function estimation based over-sampling for imbalanced two-class problems," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2012, pp. 1–8.

[10] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.

[11] H. Han, W. Wang, and B. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput. (ICIC)*, in Lecture Notes in Computer Science, D.-S. Huang, X.-P. S. Zhang, and G.-B. Huang, Eds. Hefei, China. Berlin, Germany: Springer, Aug. 2005, pp. 878–887.

[12] 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 Netw., IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 1322–1328.

[13] A. Islam, S. B. Belhaouari, A. U. Rehman, and H. Bensmail, "KNNOR: An oversampling technique for imbalanced datasets," *Appl. Soft Comput.*, vol. 115, Jan. 2022, Art. no. 108288.

[14] J. Jedrzejowicz and P. Jedrzejowicz, "Bicriteria oversampling for imbalanced data classification," *Proc. Comput. Sci.*, vol. 207C, pp. 239–248, Jan. 2022.

[15] P. Jedrzejowicz, "Oversampling for mining imbalanced datasets: Taxonomy and performance evaluation," in *Proc. 14th Int. Conf. Comput. Collective Intell. (ICCCI)*, in Lecture Notes in Computer Science, vol. 13501, N. T. Nguyen, Y. Manolopoulos, R. Chbeir, A. Kozierkiewicz, and B. Trawinski, Eds. 2022, Hammamet, Tunisia. Cham, Switzerland: Springer, Sep. 2022, pp. 322–333.

[16] M. Koziarski, "CSMOUTE: Combined synthetic oversampling and undersampling technique for imbalanced data classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Shenzhen, China, Jul. 2021, pp. 1–8.

[17] M. Koziarski, "Potential anchoring for imbalanced data classification," *Pattern Recognit.*, vol. 120, Dec. 2021, Art. no. 108114.

[18] M. Koziarski, B. Krawczyk, and M. Woźniak, "Radial-based oversampling for noisy imbalanced data classification," *Neurocomputing*, vol. 343, pp. 19–33, May 2019.

[19] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.

[20] P. Liang, W. Li, and J. Hu, "Oversampling the minority class in a multi-linear feature space for imbalanced data classification," *IEEJ Trans. Electr. Electron. Eng.*, vol. 13, no. 10, pp. 1483–1491, Oct. 2018.

[21] S. Maldonado, C. Vairetti, A. Fernandez, and F. Herrera, "FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108511.

[22] J. Mathew, C. K. Pang, M. Luo, and W. H. Leong, "Classification of imbalanced data by oversampling in kernel space of support vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4065–4076, Sep. 2018.

[23] I. Nekooeimehr and S. K. Lai-Yuen, "Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets," *Expert Syst. Appl.*, vol. 46, pp. 405–416, May 2016.

[24] T. Pan, J. Zhao, W. Wu, and J. Yang, "Learning imbalanced datasets based on SMOTE and Gaussian distribution," *Inf. Sci.*, vol. 512, pp. 1214–1233, Feb. 2020.

[25] M. Pérez-Ortiz, P. A. Gutiérrez, P. Tino, and C. Hervás-Martínez, "Oversampling the minority class in the feature space," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1947–1961, Sep. 2016.

[26] S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane, and N. Japkowicz, "Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Singapore, Nov. 2018, pp. 447–456.

[27] X. Wang, J. Xu, T. Zeng, and L. Jing, "Local distribution-based adaptive minority oversampling for imbalanced data classification," *Neurocomputing*, vol. 422, pp. 200–213, Jan. 2021.

[28] X. Yi, Y. Xu, Q. Hu, S. Krishnamoorthy, W. Li, and Z. Tang, "ASN-SMOTE: A synthetic minority oversampling method with adaptive qualified synthesizer selection," *Complex Intell. Syst.*, vol. 8, no. 3, pp. 2247–2272, Jun. 2022.

[29] A. Zhang, H. Yu, Z. Huan, X. Yang, S. Zheng, and S. Gao, "SMOTE-R*k*NN: A hybrid re-sampling method based on SMOTE and reverse *k*-nearest neighbors," *Inf. Sci.*, vol. 595, pp. 70–88, May 2022.

JOANNA JEDRZEJOWICZ received the degree in mathematics from Warsaw University, the Ph.D. degree from the University of Gdańsk, Poland, and the Habilitation degree in computer science from the Poznan University of Technology. She is currently a Professor of informatics with the Faculty of Mathematics, Physics and Informatics, University of Gdańsk. Her research interests include artificial intelligence, complexity theory, and formal languages. She has published in this area over 40 articles in the international scientific journals and proceedings, two books, and several textbooks for students. She is a program committee member in a number of international scientific conferences.

PIOTR JEDRZEJOWICZ received the degree, Ph.D., and Habilitation degrees from the University of Gdańsk. He is currently a Professor of information systems and the Head of the Information Systems Department, Gdynia Maritime University. During his career, he has been a Visiting Professor in Germany, U.K., China, and Sweden, and a Research Fellow with the School of Computer Science, McGill University, Montreal. He has published four books and over 200 articles in the international scientific journals and proceedings. His research interests include artificial intelligence, machine learning, agent technology, and decision support systems. He is an Elected Member of the Committee of the Computer Science and the Polish Academy of Science. He is a member of the Scientific Council and the Polish Society of Artificial Intelligence. He has been leading several research projects and serving as the general chair, the program chair, and a program committee member in numerous international scientific conferences. He has been invited to edit several special numbers of the international scientific journals.