Re-sampling of multi-class imbalanced data using belief function theory and ensemble learning

Fares Grina[a,b], Zied Elouedi[a], Eric Lefevre[b]

[a] *Institut Superieur de Gestion de Tunis, LARODEC, Tunis, Tunisia*
[b] *Univ. Artois, UR 3926, Laboratoire de Genie Informatique et d'Automatique de l'Artois (LGI2A), Bethune, 62400, France*

Abstract

Imbalanced classification refers to problems in which there are significantly more instances available for some classes than for others. Such scenarios require special attention because traditional classifiers tend to be biased towards the majority class which has a large number of examples. Different strategies, such as re-sampling, have been suggested to improve imbalanced learning. Ensemble methods have also been proven to yield promising results in the presence of class-imbalance. However, most of them only deal with binary imbalanced datasets. In this paper, we propose a re-sampling approach based on belief function theory and ensemble learning for dealing with class imbalance in the multi-class setting. This technique assigns soft evidential labels to each instance. This evidential modeling provides more information about each object's region, which improves the selection of objects in both undersampling and oversampling. Our approach firstly selects ambiguous majority instances for undersampling, then oversamples minority objects through the generation of synthetic examples in borderline regions to better improve minority class borders. Finally, to improve the induced results, the proposed re-sampling approach is incorporated into an evidential classifier-independent fusion-based ensemble. The comparative study against well-known ensemble methods reveals that our method is efficient according to the G-Mean and F1-score measures, independently from the chosen classifier.

Keywords: Imbalanced classification, Ensemble learning, Re-sampling, Evidence theory

1. Introduction

In many real-world binary classification problems, one class tends to be heavily under-represented when it consists of far fewer observations than the other class. This results in creating a biased model with undesirable performance. Particularly, it is a scenario that occurs when a class, referred to as the minority class, is highly under-represented in the dataset, while the other class represents the majority [1].

Due to the naturally-skewed class distributions, class imbalance has been widely observed in many real-world applications, such as medical diagnosis [2], network intrusion detection [3], language translation [4], and fraud detection [5]. From a practical point of view, the minority class usually yields higher interests. For example, failing to detect a fraudulent transaction can be crucial to a banking organization.

In addition to the skewed class distribution, the complexity of the data is an important factor for classification models. Other related data imperfections are detected like class overlapping (ambiguity) and noise, as illustrated in Figure 1. The data uncertainty issue was proven to increase the difficulty for classifiers to yield good performance on imbalanced datasets [6].

In order to deal with the poor performance on imbalanced data, many strategies have been developed to deal with this issue. The proposed methods present a variety of re-sampling, classifier modifications, cost-sensitive learning, or ensemble approaches [7]. Data re-sampling is one of the simplest yet efficient strategies to deal with imbalanced classification. These methods typically aim at re-balancing the data at the preprocessing level. This gives it the advantage of versatility, for the reason that it is classifier-independent. More recently, ensemble learning is incorporated and combined with different strategies, such as re-sampling. Ensembles are used to improve a single classifier by combining several base classifiers (also called weak learners) that outperform every independent one.

Nevertheless, learning from multi-class imbalanced datasets has not been as heavily researched. As a consequence, proposed methods which are designed to directly be applied on binary cases, cannot be easily adapted to multi-class imbalanced scenarios. This is due to the complexity of multi-class relationships compared to two-class problems. Rather than directly applying these methods to the multiple classes, many methods focus on decomposing multi-class problems into binary ones. For instance, the One-Versus-One (OVO) approach [8] is a decomposition scheme developed to modify the

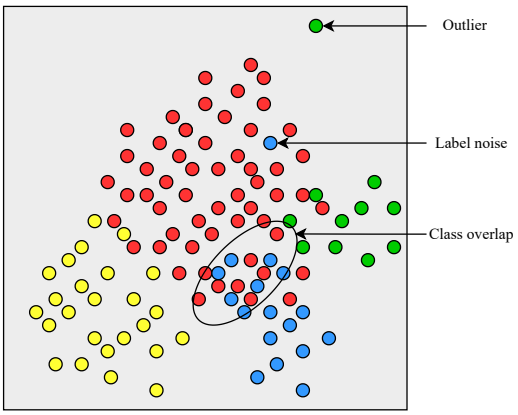


Figure 1: Challenges in multi-class imbalanced datasets.

multi-class problem into multiple binary sub-problems, one for each pair of classes. Even though binarization is simple and straightforward approach for learning from multiple class problems, it may lead to some regions being ignored and left unlearned. Specifically, when there is high data uncertainty, such as ambiguity created by high overlapping, and noise. These issues cannot be dealt with using decomposition techniques. In addition, most of the proposed methods have been observed to suffer from limitations, such as the presence of other data imperfections (i.e. high class overlapping, noise and outliers).

Recently, belief function theory was used for imbalanced classification [9, 10, 11]. Specifically, a hybrid re-sampling approach [11] was suggested to deal with multi-class imbalanced datasets in the presence of uncertainty (data ambiguity and noise). This approach uses belief function theory to represent class memberships, before combining oversampling and undersampling. Utilizing this evidential representation provides us with more information in order to better choose the locations of newly generated objects and which majority instances to remove. Under the belief function framework, it is possible to assign masses towards non-singleton sets, which makes it easier to represent memberships to intersections. This property facilitates the representation of belonging to class overlapping regions. Then, an evidential version of SMOTE [12] is performed on the minority classes, and evidential undersampling on majority classes.

In this paper, we extend the latter method [11] by incorporating it into an ensemble learning framework to create an evidential ensemble-based re-

sampling method for multi-class imbalanced data. The motivation behind this extension is the choice of some parameters, which can be seen as its main drawback, since the behavior of this algorithm is quite dependent on the assumption of how much ambiguity is present. Indeed, it is rather difficult to know the exact amount of ambiguity or class overlapping present in the data. To fix this issue, we integrate this evidential method into the process of a bagging ensemble. The goal is to train multiple base classifiers using different subsets created by our evidential re-sampling. As a result, different assumptions of how ambiguous the dataset is, will promote diversity among the generated classifiers. Finally, we use a classifier fusion approach based on the belief function theory. To ensure the versatility of our contribution, we made sure that our fusion mechanism does not depend on one specific classifier.

The remainder of this paper will be divided as follows. Subsection 2.1 presents related works in re-sampling and ensemble learning. Belief function theory will be recalled in Subsection 2.2. Section 3 details each step of our contribution, that is, the evidential mechanism used for re-sampling and the classifier combination method. Experimental evaluation and discussion are conducted in Section 4. Our paper ends with a conclusion and an outlook on future work in Section 5.

2. Preliminaries

2.1. Imbalanced data classification

In this section, we mainly review existing works relative to re-sampling and ensemble learning, and finally, some of multi-class imbalance approaches are presented.

2.1.1. Re-sampling for dealing with class imbalance

Data re-sampling is one of the most common approaches for dealing with imbalanced classification [7]. In fact, data re-sampling deals with class imbalance at the preprocessing level by changing the class distribution of the training set. As a result, it alleviates the effects of distribution skewness of the learning process. These methods can be further categorized into three groups, namely:

- **Oversampling**: These techniques introduce new minority synthetic samples to re-balance the dataset. The most straightforward method

is random oversampling (ROS), which consists of selecting minority observations in the original data set and simply replicating them. Although it appears to be technically effective since the class balance is adjusted, it can lead to overfitting [13]. To cope with overfitting, the Synthetic Minority Oversampling Technique (SMOTE) was suggested in [12]. Unlike ROS, SMOTE generates new synthetic samples by interpolating among several minority objects that are close to each other. However, many studies [14, 15] have shown SMOTE's drawbacks which are potential amplification of noise and overlapping already present in the data. SMOTE's improvements include Borderline-SMOTE [16], which identifies borderline minority class examples to generate new samples. Clustering-based oversampling techniques were also proposed [17, 15] to smartly select the regions where to generate new points. Safe-Level-SMOTE [18] is another technique which assigns weight degrees to samples based on the region they are located in. The weight is computed using nearest neighbour minority instances. Then SMOTE is performed only on samples that are labeled safe by the algorithm.

- **Undersampling**: These approaches create a subset of the original dataset by removing some majority class instances. Like random oversampling, the naive undersampling technique is to randomly remove majority objects, which may potentially remove meaningful information from the dataset. Therefore, other techniques have been suggested to smartly remove unwanted majority class instances. Commonly, traditional filtering techniques have been used to perform undersampling. For example, Neighborhood Cleaning Rule (NCL) discards majority class instances using the Edited Nearest Neighbors (ENN) introduced in [19]. Similarly, Tomek Links (TL) [20] is occasionally used as an undersampling method. Clustering has also been used for undersampling in a number of occasions [21, 22], to optimize the selection process of majority instances to eliminate.

- **Hybrid**: This strategy combines both oversampling and undersampling in order to re-balance the dataset. Typically, SMOTE is paired with an undersampling procedure to fix its drawbacks. For instance, SMOTE-ENN and SMOTE-TL were suggested in [23] to combine SMOTE with ENN and TL respectively. SMOTE-RSB*

[24] is a method which combines SMOTE for oversampling with the Rough Set Theory [25] as a cleaning technique. In SMOTE-IPF [26], SMOTE is firstly executed, and then the Iterative-Partitioning Filter (IPF) [26] is performed to remove noisy original examples, and those introduced by SMOTE. Authors in [27] suggested a combination of a SMOTE-like algorithm with a cleaning procedure to reduce the effects of overlapping. Similarly, the class overlap issue is touched upon in [6] combining a soft clustering method with Borderline-SMOTE.

2.1.2. Ensemble learning in imbalanced classification

The main idea behind ensembles is to improve a single classifier by combining the results of multiple classifiers that outperform every independent one. Directly applying ensemble classifiers, such as random forest (RF) [28] to imbalanced data, has been a popular choice to deal with class imbalance [29]. This paper focuses on re-sampling-based ensembles, which combines ensemble learning with re-sampling techniques to tackle class imbalance. Most works considered the use of bagging, boosting, or a combination of the two.

Bagging builds ensembles using the concept of *independent learning*. This strategy trains the base classifiers independently from each other, and uses data re-sampling to introduce diversity into the predictions of the models. While boosting learns of the misclassification of previous iterations by adapting the importance of misclassified objects in future iterations.

Random undersampling is popularly used with ensembles [30]. SMOTEBagging and UnderBagging were suggested in [31]. The former integrates SMOTE's oversampling into the bagging algorithm, with an adaptive way of computing the re-sampling rate, while Underbagging does the same using random undersampling. In order to optimize the model performance, a hybrid re-sampling technique was combined with bagging [32].

Boosting-based ensembles have also been proposed for the class imbalance issue. Similar to bagging-based ensembles, these methods merge data re-sampling techniques into boosting algorithms, more specifically the AdaBoost algorithm [33]. SMOTEBoost [34] performs SMOTE during each boosting iteration in order to generate minority objects. RUSBoost [35] is also similar to SMOTEBoost, but it eliminates instances from the majority class by random undersampling in each iteration. Evolutionary algorithms were also used to create a boosting-based algorithm [36]. SMOTEWB [37] is another

boosting ensemble, which combines SMOTE with a noise detection method, into a boosting framework.

Some methods have used hybrid approaches involving both boosting and bagging, such as EasyEnsemble and BalanceCascade [38].

2.1.3. Learning from multi-class imbalance

Many of the previously presented methods cannot directly handle the multi-class imbalance. Hence, significant effort has been recently invested to tackle the multi-class imbalanced classification. Rather than directly applying these methods to the multiple classes, many methods focus on decomposing multi-class problems into binary ones. For instance, the One-Versus-One (OVO) approach [8] is a decomposition scheme developed to modify the multi-class problem into multiple binary sub-problems, one for each pair of classes. Each sub-problem is trained on a binary classifier, ignoring the remaining instances not belonging the pair. Similarly to OVO, One-Versus-All (OVA) [39] is another decomposition framework which transforms the multi-class data into multiple binary sub-problems. However, OVA trains a classifier for each class in the training dataset. Other variants of these methods were also suggested [40, 41, 40, 42], mostly to improve the combination of classifiers decisions. Even though binarization is simple and straightforward approach for learning from multiple class problems, it may lead to some regions being ignored and left unlearned. Specifically, when there is high data uncertainty, such as ambiguity created by high overlapping, and noise. These issues cannot be dealt with using decomposition techniques.

2.2. Belief function theory

Belief function theory [43, 44, 45], also called the evidence theory or Dempster-Shafer theory (DST), is a flexible and well-founded framework to represent and combine uncertain information. The frame of discernment denotes a finite set of M exclusive possible propositions, e.g., possible class labels for an object in a classification problem. The frame of discernment is denoted as follows:

$$\Omega = \{w_1, w_2, ..., w_M\} \tag{1}$$

A basic belief assignment (also referred to as *bba*) represents the amount of belief given by a source of evidence, committed to 2^{Ω}, that is, all subsets of the frame including the whole frame itself. Formally, a *bba* is represented

by a mapping function $m : 2^{\Omega} \rightarrow [0, 1]$ such that:

$$\sum_{A \in 2^{\Omega}} m(A) = 1 \tag{2}$$

Each mass $m(A)$ quantifies the amount of belief allocated to a event A of Ω. A *bba* is called unnormalized if the sum of its masses is not equal to 1, and should be normalized under a closed-world assumption [46]. A focal element is a subset $A \subseteq \Omega$ where $m(A) > 0$.

The *Plausibility* function is another representation of knowledge defined by *Shafer* [44] as follows:

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \quad \forall \ A \in 2^{\Omega} \tag{3}$$

$Pl(A)$ represents the total possible support for A and its subsets.

To combine several *bba*s, *Dempster*'s rule [43] is a popular choice. Let m_1 and m_2 two BBAs defined on the same frame of discernment Ω, their combination based on *Dempster*'s rule gives the following *bba*:

$$m_1 \oplus m_2(A) = \begin{cases} \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)} & for\ A \neq \emptyset\ and\ A \in 2^{\Omega}. \\ 0 & for\ A = \emptyset. \end{cases} \tag{4}$$

In this paper, our intuition behind the use of belief function theory is to represent the different ambiguities present in the data such as overlapping and outliers. This helps us categorize the different observations based on their locations.

3. Ensemble-based evidential hybrid Re-Sampling for multi-class imbalance

Recently, a hybrid re-sampling approach [11] was suggested to deal with multi-class imbalanced datasets in the presence of uncertainty (data ambiguity and noise). This method combines oversampling and undersampling to re-balance multi-class datasets.

For all classes with a number of objects higher than the mean s, the assigned memberships are used to smartly perform undersampling. Our version of undersampling has an adaptive behavior, since the number of

removed objects depends on the amount of overlap and noise present in the corresponding majority class. However, each majority class size should not get inferior to the calculated mean s.

In case of all classes with a number of objects lower than the mean s, we use the calculated evidential memberships in order to perform oversampling in the borders of the minority class. Similarly to undersampling, our version of oversampling adapts to each class and generates synthetic minority instances only in the wanted locations. The only stopping criterion is not exceeding the mean s.

In this paper, we improve the prior approach by incorporating it into a bagging ensemble, to provide an Ensemble-based Evidential Hybrid Resampling method (E-EVRS). The main idea is to create diverse re-sampled subsets using different assumptions of ambiguity. This will add diversity to the resulting model, by combining various decision boundaries created by each base learner. Our method is illustrated in Figure 2. Furthermore, we present a general pseudocode of the proposed method in Algorithms 1 and 2, where we formally describe the high-level parts of our approach.

Before training each base classifier, the evidential process starts by assigning a soft label structure to each observation. In order to ensure diversity, a new assumption of ambiguity is selected for each classifier. Then, the selections of instances to eliminate and locations of generated objects are made based on rules. The idea is to use the evidential structure in order to better choose the locations of newly generated objects and which majority instances to remove. The reason behind this is to avoid the loss of important majority data, in case of undersampling, and emphasize the borders of the minority class in case of oversampling. After performing oversampling and undersampling, each subset will be used to train a base model. It is important to note that one can use any base classifier. Finally, we accomplish classifier fusion using an evidential combination, to create the final learning model.

3.1. Creating soft evidential labels

Our proposed approach starts by computing the centroids of each class and meta-class (the overlapping region), then creating a *bba* based on the distance between each object and each centroid. The usage of class centroids, instead of other methods like nearest neighbor-based techniques, has a relatively low computational complexity. Indeed, nearest neighbor-based techniques require the computation of nearest neighbors for each instance, which is not applicable when the number of samples is big. While in our case, instead

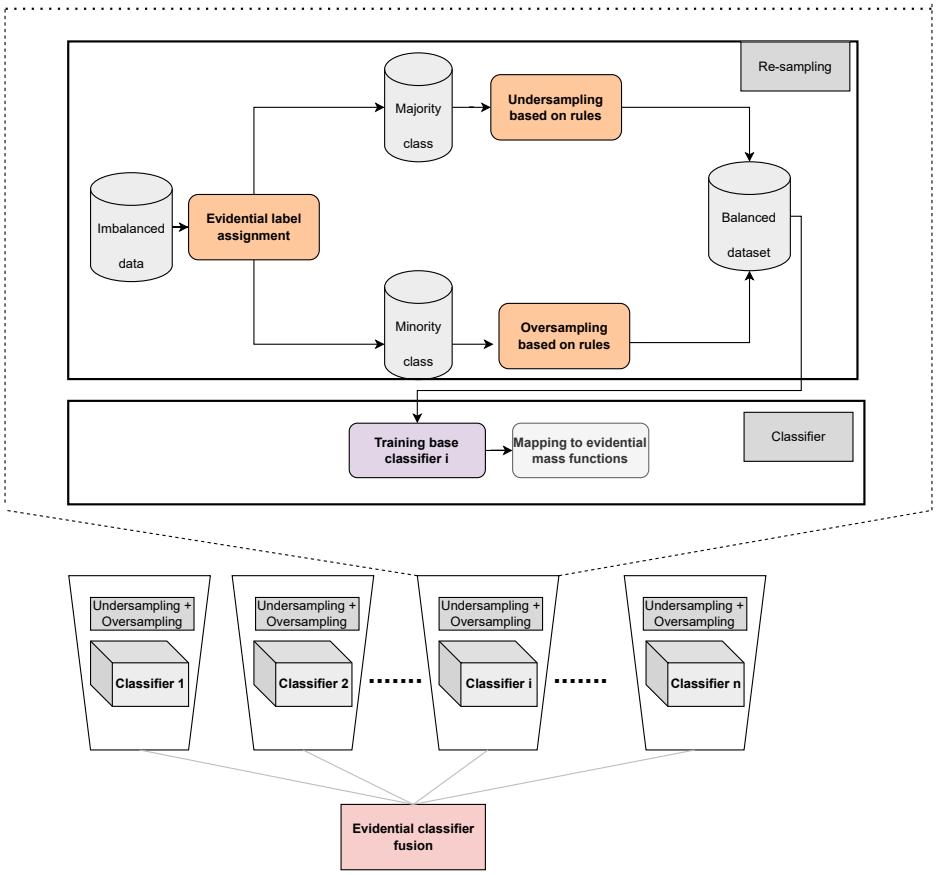


Figure 2: Evidential Ensemble-based re-sampling algorithm (E-EVRS)

Algorithm 1 Multi-class re-sampling using belief function theory and ensemble learning

Input: number of base learners n_{clf}, a training set D, a validation set V
Output: a rebalanced dataset D and class predictions
Split D into D_1, D_2, D_3, ..., D_n where D_i is a single class
$s \leftarrow$ mean of all class sizes
$\beta \leftarrow 0$
for $j \leftarrow 1$ to n_{clf} **do** ▷ Ensemble iterations
    $\beta \leftarrow \beta + 1/n_{clf}$
    Evidential labeling:
    for each instance in D **do**
        Assign a mass function using β
        $maxm \leftarrow$ subset with highest mass
        $plmax \leftarrow$ singleton with maximum plausibility
    end for
    for each D_i, $i = 1, 2, \ldots, n$ **do**
        $D \leftarrow$ Sort D according to Eq. 10 in a descending order
        if $length(D_i) > s$ **then**
            Evidential undersampling
            while $length(D_i) > s$ **or** end of D_i reached **do**
                $x \leftarrow$ Select one instance from D_i
                if $|maxm| > 1$ **then**
                    Remove x from D_i
                end if
            end while
            for each instance in D_i **do**
                if $plmax \neq class(D_i)$ **then**
                    Remove instance from D_i (considered as label noise or outlier)
                end if
            end for
        else if $length(D_i) < s$ **then**
            Evidential oversampling
            for each instance in D_i **do**
                if $|maxm| > 1$ **and** $plmax == class(D_i)$ **then**
                    select as borderline minority object
                end if
                apply SMOTE on selected instances to create $s - |D_i|$ synthetic objects
            end for
        end if
    end for
    Training the base classifier
    $Clf_j \leftarrow$ Train classifier on D
end for
$predictions \leftarrow ClassifierFusion(Clf, V)$ *# See Algorithm 2*
return D, $predictions$

Algorithm 2 Base classifier fusion using belief function theory

```
function ClassifierFusion(base classifiers Clf, validation set V)
    for each instance in V do
        for each Clf_j, j = 1, 2, ..., n_clf do
            create a probability distribution for class prediction
            mapping to evidential mass function
        end for
        apply Dempster's rule to combine Clf_j, j = 1, 2, ..., n_clf
        predictions ←add final prediction using maximum plausibility
    end for
    return predictions
end function
```

of nearest neighbors, class centroids are used to quantify class memberships. This makes our method effective for dealing with large scale data thanks to its relatively low computational and complexity burden.

The class centroids are calculated by the mean value of the training set in the corresponding class. Regarding the overlapping regions represented by meta-classes, the centroids are defined by the barycenter of the involved class centroids as follows:

$$C_U = \frac{1}{|U|} \sum_{\omega_i \in U} C_i \tag{5}$$

where ω_i are the classes in U, U represents the meta-class, and C_i is the corresponding centroid.

After creating the centroids, we assign to each instance a soft evidential label represented by a *bba* over the frame of discernment $\Omega = \{\omega_1, ..., \omega_M, \omega_0\}$, where the M classes are represented. The proposition ω_0 is included in the frame of discernment to represent the outlier, i.e., assignment of objects that are far from any class in the data. It is important to note that not all meta-classes should be considered as potential focal elements. In fact, some classes do not overlap, and so no object needs to be assigned to the meta-class involving them. As enforced in [47], the meta-class centroid should be closer to the centroids of its involved classes than to other incompatible classes' centroids. By doing so, we greatly reduce the computational complexity of the algorithm, since only the necessary mass computations will be calculated.

If it's not the case, it will not be considered as effective for the computation of *bbas*. This rule reduces the number of focal elements, and thus, the computational complexity of the algorithm.

Let x_s be an instance belonging to the training set. The idea is that each class or meta-class centroid represents a piece of evidence to the evidential membership of x_s. Accordingly, the mass values for each focal element in regard to x_s's memberships should depend on $d(x_s, C)$, that is, the distance between the respective class centroid C and x_s. The farther the centroid is, the lower the mass value for the corresponding class. By analogy, the closer x_s is to a class/meta-class centroid, the more likely it belongs to it. Hence, the initial unnormalized masses should be represented by decreasing distance based functions. We use the Mahalanobis distance [48], in this work, as recommended by [47] in order to deal with anisotropic datasets. Meta-classes U are chosen based on the constraint given above. The unnormalized masses are calculated accordingly:

$$\hat{m}(\{\omega_i\}) = e^{-d(x_s, C_i)} \tag{6}$$

$$\hat{m}(U) = e^{-\gamma\, \lambda\, d(x_s, C_U)}, \qquad for \;\; |U| \geq 1 \tag{7}$$

$$\hat{m}(\{\omega_0\}) = e^{-t} \tag{8}$$

where $\lambda = \beta\, |U|^{\alpha}$. A value of $\alpha = 1$, which penalizes the meta-classes with high cardinalities, is fixed as recommended to obtain good results on average, and β is a parameter such that $0 \leq \beta \leq 1$. It is used to tune the number of objects committed to the overlapping region. The value of γ is equal to the ratio between the maximum distance of x_s to the centroids in U and the minimum distance. It is used to measure the degree of distinguishability among classes in U. The smaller γ indicates a poor distinguishability degree between the classes of U for x_s. The outlier class ω_0 is taken into account in order to deal with objects far from all classes, and its mass value is calculated according to an outlier threshold t.

Finally, the unnormalized belief masses $\hat{m}$ are normalized as follows:

$$m(A) = \frac{\hat{m}(A)}{\sum_{B \subseteq \Omega} \hat{m}(B)} \tag{9}$$

3.2. Evidential adaptive undersampling

As mentioned above, this is dedicated to the classes whose size is higher than the mean size, corresponding to a majority. The created *bbas* are used here to determine whether an object is necessary for the learning phase or not. The logic behind our idea is to discard the samples which have a high uncertainty, that is, samples which present a relatively higher difficulty to

correctly classify. These types of instances involve high ambiguity (class overlapping samples), outliers, and label noise. The evidential membership is used to detect those samples.

3.2.1. Class overlapping

In this framework, overlapping objects have high masses assigned to meta-class focal elements, i.e., non-singleton propositions. For instance, a sample with the maximum mass assigned to $U = \{\omega_1, \omega_2, \omega_3\}$ means that it is located in the region intersecting the three classes ω_1, ω_2, and ω_3. This specific instance can be removed in the undersampling phase, in order to reduce the data ambiguity and reduce majority classes' sizes, at the same time.

Some control over the number of examples removed should be set up. Hence, the selected objects for undersampling are sorted in a descending order based on the average mass value attributed to non-singleton elements $\bar{\mu}$. Formally, for a selected object x_i:

$$\bar{\mu}_{x_i} = \frac{\sum_{|A|>1} m(A)}{k}, \qquad A \in 2^{\Omega} \tag{10}$$

where k represents the number of non-singleton focal elements. In other words, the more ambiguous objects (higher imprecision) are firstly removed until the size of the corresponding majority class reaches the mean s.

Regarding majority objects whose highest mass is not assigned to a non-singleton proposition (meta-class), we can safely say that they are not located in an overlapping region. However, they could be situated far from all classes (outlier), or in a different class (label noise). To further detect those types of samples, the maximum plausibility $Pl_{max} = max_{\omega \in \Omega} Pl(\{\omega\})$ is used, which is a popular choice for decision making in belief function theory [49].

3.2.2. Label noise

Normally, a safe object should have the maximum plausibility assigned to its label. Otherwise, it could be considered as located in another class, which could be described as label noise. Following this logic, each object, with the maximum plausibility assigned to another label than its own, is discarded from the dataset.

3.2.3. Outliers

This type of objects is located far from any class in the data. Usually, this could be described as the state of ignorance in our framework. Thus,

objects with maximum plausibility assigned to ω_0, i.e., $Pl_{max} = Pl(\{\omega_0\})$, are removed from the dataset.

3.3. Evidential adaptive oversampling

In order to strengthen the presence of minority classes in the dataset, an oversampling phase is added to empower the borders of each minority class. Our objective is to emphasize the borders of each minority class, much like other oversampling techniques such as BorderlineSMOTE [16]. Another aspect of our approach is avoiding oversampling noisy examples and outliers, which can potentially add more unwanted noise to the dataset.

The previously computed *bbas* are used in this phase to smartly pick the regions where synthetic minority objects should be created. Minority instances are sorted into three probable categories, similar to the cleaning step: overlapping, label noise, or outlier. If an object does not correspond to one of the three categories, it is considered as located in a safe region and is not selected for the creation of new synthetic objects. The same is valid for label noise and outliers. Indeed, selecting noisy objects and outliers to generate new samples could lead to overgeneralization, which is a significant drawback of many oversampling techniques [16].

Our evidential approach to oversampling consists of generating synthetic minority data near the borderline objects of the minority class. The idea is to empower the minority class borders in order to avoid the misclassification of difficult objects. Formally speaking, only objects whose highest mass is committed towards an overlapping region are selected for oversampling. This procedure also helps us avoid selecting objects which are committed towards label noise and outlier. Indeed, selecting those objects would amplify the problems already present in the dataset. Then, for each selected examples, one of its k nearest neighbors is used to generate a new synthetic minority object by interpolation.

As mentioned above, the number of generated examples is also controlled and the size of each minority class should not exceed the mean s. In fact, the objects in the corresponding minority class are sorted in descending order based on Eq. 10. The idea behind this is to give priority towards minority objects with higher uncertainty in order to generate synthetic object in difficult-to-classify locations.

3.4. Base classifier learning and combination

Our approach achieved good performance in multi-class imbalanced classification tasks because it aims at improving the visibility of the minority class, by efficiently re-balancing the data in presence of uncertainty [11]. However, the performance is highly influenced by the selected value for the parameter β, which controls the amount to eliminate from the ambiguous region, and the amount to generate. As a result, very different subsets are created, as seen in Figure 3. The figure shows the results of hybrid re-sampling performed on a real 3-class imbalanced example, before training a Support Vector Machine (SVM) classifier [50]. As illustrated, the re-sampled subsets can yield very diverse decision boundaries, depending on different ambiguity assumptions. To tackle this issue, our evidential re-sampling method is included into a bagging ensemble. For each iteration, a different value of the parameter β is selected.

Finally, each subset is used to train a base classifier. Any classifier could be used with our framework, as long as it can yield a probability distribution as result. Since each classifier is trained independently in bagging ensembles, we make the assumption that each model's output is an independent piece of evidence. Henceforth, we can apply Dempster's rule of combination presented in Eq. 4, as suggested in [51]. In our case, the output of each base classifier should be represented by mass functions. For this purpose, we propose to use the inverse pignistic transform [52] in order to convert the probabilistic output of the classifier to mass functions. As a result, a mass function is created for each base learner. Thus, the Dempster rule of combination can be applied to create a final combined mass function. Finally, the decision is made by choosing the singleton with the maximum plausibility $Pl_{max} = max_{\omega \in \Omega} Pl(\{\omega\})$.

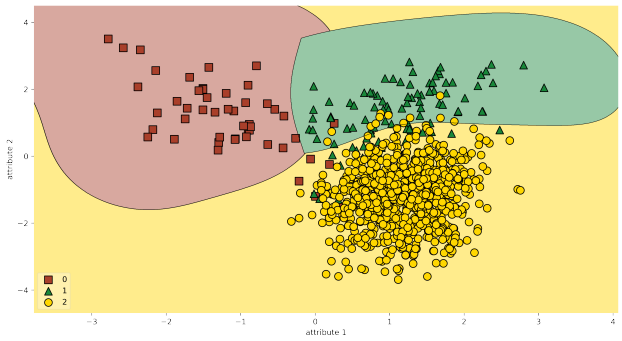
4. Experimental study

In this section, we will firstly detail the setup of the conducted experiments in subsection 4.1. Lastly, we will present the results and discuss them in subsection 4.2.

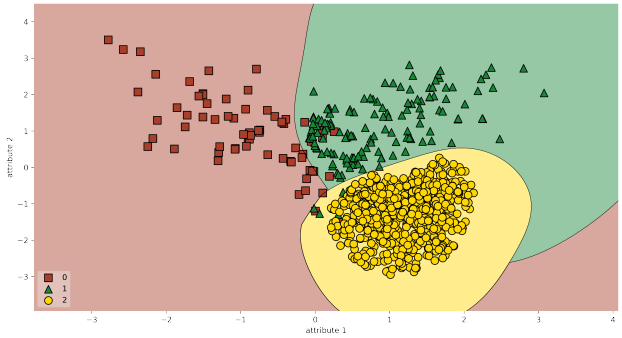
4.1. Experimental setup

4.1.1. Datasets.

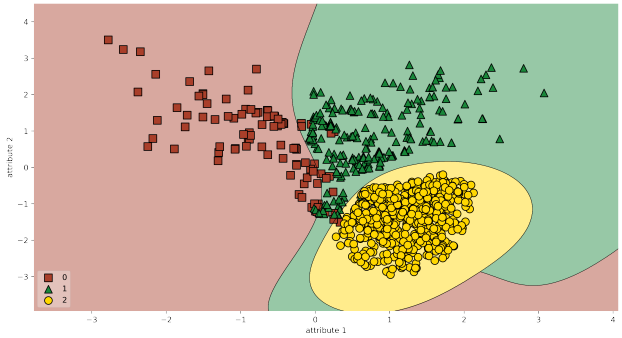
A total of 11 imbalanced datasets were selected from the KEEL repository [53]. The datasets are further detailed in Table 1. The imbalance ratio



(a) SVM's decision boundaries on original distribution without re-sampling.



(b) SVM's decision boundaries on re-sampled data using our approach with $\beta = 0.7$.



(c) SVM's decision boundaries on re-sampled data using our approach with $\beta = 0.2$.

Figure 3: Comparing the resulted decision boundaries by SVM after performing our evidential re-sampling on different amounts of overlap.

Table 1: Description of the imbalanced datasets selected from the KEEL repository.

Datasets	IR	Features	Samples	#Class
wine	1.48	13	178	3
vehicle	1.20	18	846	4
contraceptive	1.89	9	1473	3
dermatology	5.55	34	366	6
balance	5.88	4	625	3
new-thyroid	5	5	215	3
zoo	10.25	16	101	7
thyroid	39.17	21	720	3
pageblocks	164	10	548	5
yeast	92.6	8	1484	10
shuttle	4558.6	9	58,000	7

(IR) is computed as the proportion of the number of majority examples to the number of minority ones $IR = \frac{\#majority}{\#minority}$. The variations of the different parameters (IR, features, and size) allowed for experimenting in different real world settings. In this case, the class with maximum number of examples is the majority class, and the class with the minimum number of examples is the minority one

4.1.2. Reference methods and parameters.

A variety of techniques is studied in this paper: RUSBoost [35], RUSBagging [31], Mahalanobis Distance Oversampling (MDO) [54], Static-SMOTE (S-SMOTE) [55] and the basic version of SMOTE paired with the One-Versus-One strategy (SMOTE-MC). Note that RF, RUSBoost, and RUSBagging are techniques based on ensemble learning, whereas S-SMOTE, MDO, and SMOTE-MC are smote-based methods specifically designed for multi-class imbalance. Parameter selection was conducted independently for each data partition using 3-fold cross validation on the training data in order to select the most performing parameters for each method. The following parameters were considered:

- **E-EVRS**:
  $\alpha = 1$ as recommended in [47]

$t = [2, 5]$ as recommended in [47]
k (oversampling) $= [3, 6]$
Number of base learners $= [10, 20, \ldots, 100]$
β: as discussed above in subsection 3.4, a new value is assigned for each base learner in order to have a diverse ensemble. The values are selected based on the number of base learners. For instance, if the number of base classifiers is 10, the selected values for β are $[0.1, 0.2, 0.3, \ldots, 1]$.

- **RUSBoost**:
  Number of base learners $= [10, 20, \ldots, 100]$
  re-sampling ratio $\in [50, 100, 150, 200]$

- **RUSBagging**:
  Number of base learners $= [10, 20, \ldots, 100]$
  re-sampling ratio $\in [50, 100, 150, 200]$

- **MDO**:
  $K1 = [1, 2, \ldots, 10]$
  re-sampling ratio $\in [50, 100, 150, 200]$

- **S-SMOTE**:
  $k - nn = [3, 6]$
  re-sampling ratio $\in [50, 100, 150, 200]$

- **SMOTE-MC**:
  $k - nn = [3, 6]$
  re-sampling ratio $\in [50, 100, 150, 200]$

4.1.3. Base classifiers

The classifiers used as base classifiers for all ensembles and re-sampling algorithms are the Support Vector Machine (SVM) classifier and the decision tree classifier. For both classification methods, we use the implementations from the Scikit-learn library [56] with default parameters unless stated otherwise. For the case of our method E-EVRS, SVM's output is converted into probability distributions using Platt scaling [57]. As for the decision tree, we use the CART implementation without pruning and collapsing. However, the minimum impurity decrease was set to 0.05 instead of the default 0.0 to function as an early stop in the training phase in order to improve the probability estimation of the trees, which we need for E-EVRS.

4.1.4. Metrics and evaluation strategy

To appropriately evaluate the different algorithms in imbalanced scenarios, we use the G-Mean (GM) [58] and the F1-score, which are popular measures for evaluating classifiers in imbalanced learning, as the standard classification accuracy is not suitable for imbalanced learning. The evaluation measures used, in this paper, are mathematically formulated as follows:

$$\text{G-Mean} = \sqrt{sensitivity \times specificity} \tag{11}$$

$$F1 - score = \frac{2 \times sensitivity \times precision}{sensitivity + precision} \tag{12}$$

with:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{13}$$

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative} \tag{14}$$

$$Specificity = \frac{TrueNegative}{TrueNegative + FalsePositive} \tag{15}$$

In our case (multi-class), the G-Mean score was obtained as a higher root of the product of sensitivity for each class. While F1-scores are calculated for each class separately by means of One-v-All strategy.

To avoid inconsistency of results, the performance estimation of each classifier is obtained by means of a 5-fold stratified cross validation, and their results are averaged. Instead of dividing the dataset into 10 folds, 5 folds are considered in order to dispose of a sufficient quantity of minority class samples in the test partitions. Finally, statistical comparisons were carried out using the Wilcoxon's signed rank tests [59] to further evaluate the significance of the results.

4.2. Results and discussion

In this section, we discuss the performance of our method compared to the other algorithms. Tables 2, 3, 4 and 5 show the averaged score for the G-Mean and F1-score for SVM and CART and the respective standard deviation in parentheses. The best average score is marked in bold. Using SVM, E-EVRS performed the best in 6 out of 11 datasets in terms of the G-Mean metric, in 7 out of 11 datasets in terms of F1-score. While the performance of our method with the decision tree classifier was the best in 8 out of 11

Table 2: G-Mean results for KEEL datasets using different re-sampling techniques with SVM as base classifier

Datasets	RUSBagging	RUSBOOST	SMOTE-ALL	S-SMOTE	MDO	E-EVRS
wine	0.956 0.045	0.981 (0.031)	0.962 (0.038)	0.967 (0.063)	**0.982** (0.038)	0.980 (0.048)
dermatology	0.960 (0.039)	0.910 (0.047)	0.956 (0.034)	0.960 (0.028)	0.957 (0.039)	**0.976** (0.029)
balance	0.566 (0.198)	0.783 (0.100)	0.865 (0.045)	0.762 (0.074)	0.536 (0.014)	**0.885** (0.066)
newthyroid	0.895 (0.114)	0.855 (0.141)	**0.966** (**0.061**)	0.941 (0.064)	0.862 (0.094)	0.963 (0.036)
contraceptive	0.488 (0.048)	0.525 (0.035)	**0.547** (**0.040**)	0.541 (0.034)	0.520 (0.030)	0.545 (0.033)
thyroid	0.952 (0.010)	0.968 (0.039)	0.579 (0.351)	0.557 (0.291)	0.549 (0.102)	**0.987** (**0.008**)
pageblocks	0.187 (0.375)	0.363 (0.447)	0.388 (0.476)	0.257 (0.394)	0.223 (0.375)	**0.592** (**0.466**)
vehicle	0.706 (0.059)	0.604 (0.064)	0.741 (0.042)	0.750 (0.033)	0.742 (0.032)	**0.845** (**0.040**)
zoo	0.500 (0.500)	**0.800** (**0.400**)	0.487 (0.500)	0.679 (0.449)	0.400 (0.490)	0.789 (0.445)
yeast	0.460 (0.180)	0.576 (0.275)	0.554 (0.166)	0.415 (0.231)	0.544 (0.190)	**0.592** (**0.223**)
shuttle	0.894 (0.298)	0.604 (0.367)	**0.982** (**0.015**)	0.937 (0.025)	0.542 (0.223)	0.976 (0.026)
Mean Rank	4.528	3.583	2.611	3.972	4.722	**1.528**

datasets for G-Mean, and in 7 out of 11 F1-score. The tables indicate that our algorithm consistently produced the best results, in terms of the G-Mean and F1-score metrics, when applied to these benchmarking datasets. To derive the rank order, cross-validated scores are used, assigning rank one to the best performing and rank six to the worst performing technique. The mean ranking results for each combination of metric and classifier are shown in Figure 4. This shows that the proposed method outperforms other methods with regard to all evaluation metrics. Notably, the technique's superiority can be observed independently of the classifier.

Additionally, the results show that our method performs relatively well when the dataset has a high imbalance ratio. For instance, E-EVRS performs the best in all cases, independently from the classifier and metric, for the dataset *pageblocks*. For the datasets *shuttle* and *thyroid*, it presented better performance in 3 out of 4 cases. Since these datasets have very high imbalance ratios, we can safely say that our approach is robust against severe imbalanced cases.

The two chosen metrics consider the accuracy of both classes, since, as defined in Eq.11, G-Mean is the square root of the product between the true negative rate (i.e., specificity), and the true positive rate (i.e., sensitivity). Meanwhile, the F1-score is based on precision and sensitivity. Therefore, we can initially argue that our proposal E-EVRS indeed improves the learning on the minority class while keeping the accuracy for the majority one.

In order to further assess the significance of the reported results, Tables 6 and 7 presents the statistical analysis made by Wilcoxon's signed ranks

Table 3: F1-score results for KEEL datasets using different re-sampling techniques with SVM as base classifier

Datasets	RUSBagging	RUSBOOST	SMOTE-ALL	S-SMOTE	MDO	E-EVRS
wine	0.956 (0.041)	0.978 (0.033)	**0.982** (**0.037**)	0.965 (0.067)	**0.982** (**0.037**)	0.976 (0.032)
dermatology	0.961 (0.036)	0.923 (0.036)	0.966 (0.030)	0.972 (0.026)	0.959 (0.035)	**0.979** (0.027)
balance	0.611 (0.039)	0.742 (0.070)	0.798 (0.056)	0.656 (0.066)	0.625 (0.014)	**0.812** (**0.061**)
newthyroid	0.904 (0.077)	0.896 (0.094)	0.955 (0.044)	0.947 (0.051)	0.914 (0.060)	**0.956** (**0.030**)
contraceptive	0.492 (0.042)	0.519 (0.031)	0.536 (0.032)	0.532 (0.032)	0.518 (0.027)	**0.539** (**0.033**)
thyroid	0.844 (0.079)	**0.914** (**0.067**)	0.835 (0.087)	0.748 (0.073)	0.701 (0.102)	0.902 (0.121)
pageblocks	0.626 (0.132)	0.636 (0.190)	0.688 (0.168)	0.517 (0.134)	0.461 (0.102)	**0.702** (**0.183**)
vehicle	0.732 (0.049)	0.650 (0.053)	0.775 (0.035)	0.774 (0.028)	0.767 (0.028)	**0.779** (**0.032**)
zoo	0.815 (0.196)	**0.930** (**0.151**)	0.825 (0.203)	0.868 (0.176)	0.802 (0.172)	0.918 (0.136)
yeast	0.440 (0.072)	0.269 (0.062)	0.596 (0.059)	**0.609** (**0.069**)	0.576 (0.057)	0.598 (0.061)
shuttle	0.565 (0.059)	0.550 (0.166)	0.830 (0.026)	0.736 (0.030)	0.549 (0.052)	**0.864** (**0.032**)
Mean Rank	4.694	4.528	2.472	2.833	4.833	**1.528**

Table 4: G-Mean results for KEEL datasets using different re-sampling techniques with decision tree as base classifier

Datasets	RUSBagging	RUSBOOST	SMOTE-ALL	S-SMOTE	MDO	E-EVRS
wine	0.976 (0.032)	**0.980** (**0.031**)	0.969 (0.042)	0.959 (0.046)	0.976 (0.032)	0.973 (0.066)
dermatology	0.942 (0.051)	0.910 (0.047)	0.940 (0.285)	0.947 (0.040)	0.950 (0.034)	**0.954** (**0.047**)
balance	0.885 (0.035)	0.783 (0.100)	0.718 (0.265)	0.861 (0.042)	0.638 (0.246)	**0.897** (**0.034**)
newthyroid	0.949 (0.055)	0.855 (0.141)	0.929 (0.080)	0.946 (0.056)	0.929 (0.073)	**0.952** (**0.060**)
contraceptive	**0.548** (**0.045**)	0.525 (0.035)	0.546 (0.036)	0.539 (0.046)	0.516 (0.039)	0.538 (0.042)
thyroid	0.984 (0.009)	0.978 (0.039)	0.931 (0.082)	0.965 (0.064)	0.569 (0.468)	**0.990** (**0.006**)
pageblocks	0.452 (0.453)	0.363 (0.447)	0.470 (0.455)	0.555 (0.458)	0.499 (0.347)	**0.558** (**0.447**)
vehicle	0.730 (0.043)	0.604 (0.064)	**0.760** (**0.054**)	0.750 (0.045)	0.754 (0.039)	0.745 (0.051)
zoo	0.300 (0.458)	**0.700** (**0.400**)	**0.700** (**0.458**)	0.695 (0.455)	0.300 (0.458)	**0.700** (**0.458**)
yeast	0.610 (0.180)	0.522 (0.135)	0.658 (0.174)	0.615 (0.231)	0.607 (0.151)	**0.704** (**0.039**)
shuttle	0.804 (0.458)	0.584 (0.400)	0.831 (0.458)	0.745 (0.455)	0.598 (0.458)	**0.854** (**0.458**)
Mean Rank	3.306	5.111	2.778	3.472	4.500	**1.500**

Table 5: F1-measure results for KEEL datasets using different re-sampling techniques with decision tree as base classifier

Datasets	RUSBagging	RUSBOOST	SMOTE-ALL	S-SMOTE	MDO	E-EVRS
wine	0.973 (0.037)	0.978 (0.037)	0.968 (0.035)	0.956 (0.048)	**0.978** (**0.037**)	0.962 (0.070)
dermatology	0.951 (0.053)	0.923 (0.036)	0.944 (0.033)	0.947 (0.040)	**0.952** (**0.031**)	**0.952** (**0.031**)
balance	0.778 (0.045)	0.742 (0.070)	0.821 (0.091)	0.758 (0.050)	0.806 (0.091)	**0.849** (**0.085**)
newthyroid	0.945 (0.042)	0.896 (0.094)	0.939 (0.067)	0.938 (0.053)	0.939 (0.076)	**0.949** (**0.053**)
contraceptive	**0.541** (**0.041**)	0.519 (0.031)	0.538 (0.037)	0.529 (0.043)	0.535 (0.035)	0.539 (0.041)
thyroid	0.864 (0.066)	0.914 (0.067)	0.877 (0.045)	0.917 (0.059)	0.719 (0.279)	**0.934** (**0.049**)
pageblocks	0.680 (0.099)	0.636 (0.190)	0.686 (0.131)	0.665 (0.138)	0.550 (0.160)	**0.700** (**0.137**)
vehicle	0.758 (0.035)	0.650 (0.053)	0.776 (0.040)	0.774 (0.038)	0.776 (0.034)	**0.771** (**0.034**)
zoo	0.810 (0.133)	**0.930** (**0.151**)	0.927 (0.113)	0.919 (0.110)	0.732 (0.203)	0.928 (0.110)
yeast	0.440 (0.072)	0.469 (0.062)	0.495 (0.051)	**0.589** (**0.069**)	0.576 (0.057)	0.563 (0.151)
shuttle	0.814 (0.059)	0.550 (0.166)	0.731 (0.026)	0.736 (0.030)	0.549 (0.052)	**0.869** (**0.095**)
Mean Rank	3.722	4.722	3.500	3.111	3.722	**2.000**

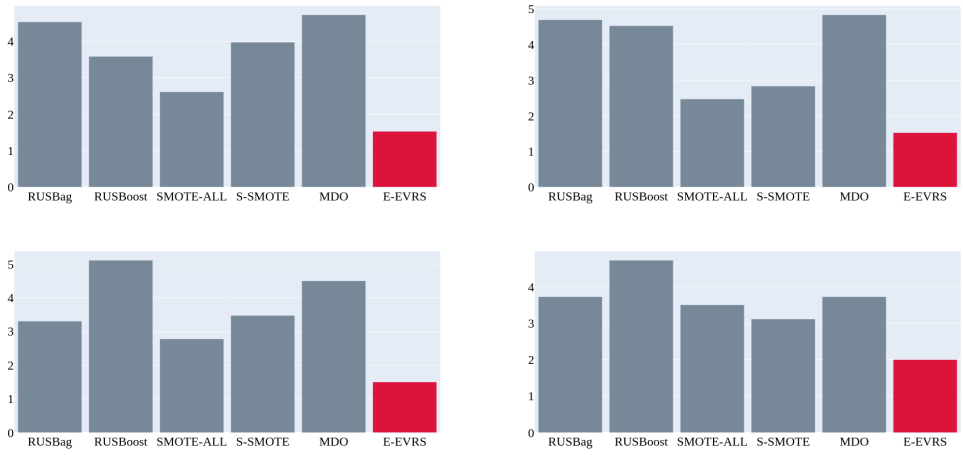


Figure 4: Mean rankings of evaluated techniques for different classifiers and metrics

test. $R+$ represents the sum of ranks in favor of E-EVRS, while $R-$ reflects the sum of ranks in favor of the other reference methods, and p-values are computed for each comparison. As shown in Tables 6 and 7, all p-values are lower than 0.1. Thus, one can say that our method significantly outperformed the other techniques using SVM and CART, for both selected metrics, with a significance level of $\alpha = 0.1$.

The reported results show that E-EVRS performed significantly better than the compared methods in complex imbalanced datasets. This is likely due to the fact that our method combines the two approaches of oversampling and undersampling, with an adjusted amount of re-sampling for each dataset. Unlike traditional undersampling and oversampling techniques, our approach avoids excessive removal of majority objects (loss of important information) and generation of minority examples (overgeneralization). This can be shown by the robustness of our approach against severely class imbalanced datasets such as *pageblocks*, *thyroid*, and *shuttle*.

5. Conclusion

In this paper, we proposed an evidential ensemble-based re-sampling method (E-EVRS), in which we use evidence theory to handle class imbalanced. The goal is to re-balance the dataset by creating synthetic

Table 6: Pairwise comparisons of obtained G-Mean and F1-score for SVM as base classifier based on Wilcoxon's signed ranks test.

	G-Mean			*F1-score*		
Comparisons	$R+$	$R-$	p-value	$R+$	$R-$	p-value
E-EVRS vs RUSBagging	66.0	0.0	0.0009765	66.0	0.0	0.001953125
E-EVRS vs RUSBOOST	63.0	3.0	0.00488281	60.0	5.0	0.009765625
E-EVRS vs SMOTE-ALL	60.0	6.0	0.01367181	63.5	2.5	0.004882812
E-EVRS vs S-SMOTE	60.0	0.0	0.00097656	61.0	5.0	0.009765625
E-EVRS vs MDO	65.0	1.0	0.0019531	65.0	1.0	0.001953125

Table 7: Pairwise comparisons of obtained G-Mean and F1-score for decision tree as base classifier based on Wilcoxon's signed ranks test.

	G-Mean			*F1-score*		
Comparisons	$R+$	$R-$	p-value	$R+$	$R-$	p-value
E-EVRS vs RUSBagging	60.5	5.5	0.01367187	60.0	6.0	0.013671875
E-EVRS vs RUSBOOST	65.0	1.0	0.00691042	63.0	3.0	0.0048828125
E-EVRS vs SMOTE-ALL	60.0	6.0	0.0284168	59.0	7.0	0.01855468
E-EVRS vs S-SMOTE	61.5	4.5	0.009765625	57.0	9.0	0.0322265625
E-EVRS vs MDO	62.0	4.0	0.0068359375	55.0	11.0	0.092600697

minority examples near ambiguous samples, and improve the visibility of the minority class by removing unwanted examples, such as noisy and overlapped observations. This technique is incorporated into a bagging ensemble framework, in order to diversify the created subsets. Therefore, it is more likely to improve the final decision boundary of the classifier.

Finally, the research conducted on benchmark datasets confirmed the effectiveness of the proposed solution. Our experimental study demonstrates that integrating evidential re-sampling into ensemble learning, could result in diversity of base models, which improves the learning performance. Further investigations can include the integration of our evidential re-sampling into a boosting-based ensemble algorithm.

References

[1] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[2] T. Huynh, A. Nibali, Z. He, Semi-supervised learning for medical image

classification using imbalanced training data, Computer Methods and Programs in Biomedicine (2022) 106628.

[3] Y. Fu, Y. Du, Z. Cao, Q. Li, W. Xiang, A deep learning model for network intrusion detection with imbalanced data, Electronics 11 (6) (2022) 898.

[4] X. Li, H. Gong, Robust optimization for multilingual translation with imbalanced data, Advances in Neural Information Processing Systems 34 (2021).

[5] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, Q. He, Pick and choose: a gnn-based imbalanced learning approach for fraud detection, in: Proceedings of the Web Conference 2021, 2021, pp. 3168–3177.

[6] P. Vuttipittayamongkol, E. Elyan, A. Petrovski, On the class overlap problem in imbalanced data classification, Knowledge-based systems 212 (2021) 106631.

[7] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: Review of methods and applications, Expert Systems with Applications 73 (2017) 220–239.

[8] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Advances in neural information processing systems 10 (1997) 507–513.

[9] J. Niu, Z. Liu, Imbalance data classification based on belief function theory, in: International Conference on Belief Functions, Springer, 2021, pp. 96–104.

[10] F. Grina, Z. Elouedi, E. Lefevre, Evidential undersampling approach for imbalanced datasets with class-overlapping and noise, in: International Conference on Modeling Decisions for Artificial Intelligence, Springer, 2021, pp. 181–192.

[11] F. Grina, Z. Elouedi, E. Lefevre, Evidential Hybrid Re-sampling for Multi-class Imbalanced Data, in: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, 2022, pp. 612–623.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[13] N. Japkowicz, Class imbalances: are we focusing on the right issue, in: Workshop on Learning from Imbalanced Data Sets II, Vol. 1723, 2003, p. 63.

[14] J. A. Sáez, J. Luengo, J. Stefanowski, F. Herrera, SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering, Information Sciences 291 (C) (2015) 184–203.

[15] G. Douzas, F. Bacao, F. Last, Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE, Information Sciences 465 (2018) 1–20.

[16] H. Han, W. Y. Wang, B. H. Mao, Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, Lecture Notes in Computer Science 3644 (PART I) (2005) 878–887.

[17] L. Ma, S. Fan, CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests, BMC Bioinformatics 18 (1) (2017) 1–18.

[18] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5476 LNAI (2009) 475–482.

[19] D. L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man, and Cybernetics (3) (1972) 408–421.

[20] T. Ivan, Two modification of cnn, IEEE transactions on Systems, Man and Communications, SMC 6 (1976) 769–772.

[21] C.-F. Tsai, W.-C. Lin, Y.-H. Hu, G.-T. Yao, Under-sampling class imbalanced datasets by combining clustering analysis and instance selection, Information Sciences 477 (2019) 47–54.

[22] N. Ofek, L. Rokach, R. Stern, A. Shabtai, Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem, Neurocomputing 243 (2017) 88–102.

[23] G. Batista, R. Prati, M.-C. Monard, A study of the behavior of several methods for balancing machine learning training data, SIGKDD Explorations 6 (2004) 20–29.

[24] E. Ramentol, Y. Caballero, R. Bello, F. Herrera, SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory, Knowledge and Information Systems 33 (2) (2012) 245–265.

[25] Z. Pawlak, Rough sets, International journal of computer & information sciences 11 (5) (1982) 341–356.

[26] T. M. Khoshgoftaar, P. Rebours, Improving software quality prediction by noise filtering techniques, Journal of Computer Science and Technology 22 (3) (2007) 387–396.

[27] M. Koziarski, M. Woźniak, B. Krawczyk, Combined cleaning and resampling algorithm for multi-class imbalanced data with label noise, Knowledge-Based Systems 204 (2020) 106223.

[28] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[29] A. More, D. P. Rana, Review of random forest classification techniques to resolve data imbalance, in: 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM), IEEE, 2017, pp. 72–78.

[30] B. C. Wallace, K. Small, C. E. Brodley, T. A. Trikalinos, Class imbalance, redux, in: 2011 IEEE 11th international conference on data mining, IEEE, 2011, pp. 754–763.

[31] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: 2009 IEEE Symposium on Computational Intelligence and Data Mining, IEEE, 2009, pp. 324–331.

[32] I. Jung, J. Ji, C. Cho, Emsm: Ensemble mixed sampling method for classifying imbalanced intrusion detection data, Electronics 11 (9) (2022) 1346.

[33] Y. Freund, R. E. Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.

[34] N. V. Chawla, A. Lazarevic, L. O. Hall, K. W. Bowyer, Smoteboost: Improving prediction of the minority class in boosting, in: European conference on principles of data mining and knowledge discovery, Springer, 2003, pp. 107–119.

[35] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: A hybrid approach to alleviating class imbalance, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 40 (1) (2009) 185–197.

[36] B. Krawczyk, M. Galar, Ł. Jeleń, F. Herrera, Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy, Applied Soft Computing 38 (2016) 714–726.

[37] F. Sağlam, M. A. Cengiz, A novel smote-based resampling technique trough noise detection and the boosting procedure, Expert Systems with Applications 200 (2022) 117023.

[38] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2) (2008) 539–550.

[39] R. Rifkin, A. Klautau, In defense of one-vs-all classification, Journal of machine learning research 5 (Jan) (2004) 101–141.

[40] Y. L. Murphey, H. Wang, G. Ou, L. A. Feldkamp, Oaho: an effective algorithm for multi-class learning from imbalanced data, in: 2007 International Joint Conference on Neural Networks, IEEE, 2007, pp. 406–411.

[41] N. Garcia-Pedrajas, D. Ortiz-Boyer, Improving multiclass pattern recognition by the combination of two strategies, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (6) (2006) 1001–1006.

[42] T. G. Dietterich, G. Bakiri, Error-correcting output codes: A general method for improving multiclass inductive learning programs, in: AAAI, 1991, pp. 572–577.

[43] A. P. Dempster, A generalization of bayesian inference, Journal of the Royal Statistical Society: Series B (Methodological) 30 (2) (1968) 205–232.

[44] G. Shafer, A mathematical theory of evidence, Vol. 42, Princeton university press, 1976.

[45] P. Smets, The Transferable Belief Model for Quantified Belief Representation, Springer Netherlands, Dordrecht, 1998, pp. 267–301.

[46] P. Smets, The nature of the unnormalized beliefs encountered in the transferable belief model, in: Uncertainty in artificial intelligence, Elsevier, 1992, pp. 292–297.

[47] Z.-G. Liu, Q. Pan, J. Dezert, G. Mercier, Credal classification rule for uncertain data based on belief functions, Pattern Recognition 47 (7) (2014) 2532–2541.

[48] P. C. Mahalanobis, On the generalized distance in statistics, Vol. 2, National Institute of Science of India, 1936, pp. 49–55.

[49] T. Denoeux, Analysis of evidence-theoretic decision rules for pattern classification, Pattern recognition 30 (7) (1997) 1095–1107.

[50] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, 1992, pp. 144–152.

[51] B. Quost, M.-H. Masson, T. Denœux, Classifier fusion in the Dempster–Shafer framework using optimized t-norm based combination rules, International Journal of Approximate Reasoning 52 (3) (2011) 353–374.

[52] D. Dubois, H. Prade, P. Smets, A definition of subjective possibility, International journal of approximate reasoning 48 (2) (2008) 352–364.

[53] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, F. Herrera, Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2010) 255–287.

[54] L. Abdi, S. Hashemi, To combat multi-class imbalanced problems by means of over-sampling techniques, IEEE transactions on Knowledge and Data Engineering 28 (1) (2015) 238–251.

[55] F. Fernández-Navarro, C. Hervás-Martínez, P. A. Gutiérrez, A dynamic over-sampling procedure based on sensitivity for multi-class problems, Pattern Recognition 44 (8) (2011) 1821–1833.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[57] J. Platt, Probabilistic outputs for svms and comparisons to regularized likehood methods, in: Advances in Large Margin Classifiers, MIT Press, 1999.

[58] R. Barandela, R. M. Valdovinos, J. S. Sánchez, New applications of ensembles of classifiers, Pattern Analysis & Applications 6 (3) (2003) 245–256.

[59] F. Wilcoxon, Individual comparisons by ranking methods, in: Breakthroughs in statistics, Springer, 1992, pp. 196–202.