Applications of Autonomous Learning Multi Model Systems to Binary Classification on Imbalanced Datasets

Rodrigo Saragoça Boal Ventura
rodrigo.boal.ventura@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
November 2021

***Abstract*—Imbalanced datasets present one of the most complex challenges for modern machine learning classification methods. Furthermore, despite imbalanced datasets being widespread in many different and relevant areas, traditional classification methods often show a large bias towards the over represented classes, and often show a poor prediction performance on the underrepresented classes. In this article, the problem of imbalanced datasets is approached using the Zero-Order Autonomous Learning Multi-Model (ALMMo-0) classifier, an AnYa-type fuzzy rule-based system, which is non-iterative, non-parametric and fully autonomous. All the tests are performed on imbalanced versions of benchmark datasets. Firstly, the performance of ALMMo-0 classifiers is compared to benchmark classification methods. In particular, the performance is compared to traditional classification methods, as well as other fuzzy rule-based systems, and is shown to out-perform the benchmark methods on highly imbalanced datasets. Secondly, based on the empirical results obtained with the original ALMMo-0 algorithm, two modifications are proposed that seek to improve the minority class detection performance: a one class classifier adaptation of the original ALMMo-0 that requires no extra parameters to be estimated, and a weighted class confidence strategy that assigns different weights to each class and optimizes the weight values using Bayesian optimization. The proposed methods are shown to out-perform the original ALMMo-0 regarding the minority class prediction performance. In particular, the one class classifier is shown to outperform for high imbalances, while the weighted class confidence strategy is shown to outperform for low and high imbalances, depending on the chosen cost function for the weight optimization.**

***Index Terms*—ALMMo-0 Classifier, Imbalanced Datasets, Binary Classification, One Class Classification, Bayesian Optimization**

I. Introduction

One particular type of challenge that is still incredibly relevant in the field of machine learning is classification tasks on highly imbalanced datasets. Imbalanced datasets are classification datasets where some classes are severely underrepresented compared to the remaining classes. As such, they pose a very challenging task to many traditional classification methods, as these often completely disregard the minority classes, which are usually the classes of most interest.

Furthermore, the problem of imbalanced datasets is particularly relevant, as such datasets occur in a wide variety of areas and applications. Examples include disease diagnosis, fraud detection, cybersecurity and image recognition. Therefore, not only is the problem of imbalanced datasets very prevalent, but it also occurs in crucial areas of high relevance and interest.

Attending to the high interest in this particular issue, a lot of attention has been devoted to develop new approaches and strategies that help reducing the effects of high class imbalances. As such, a wide variety of methods have been proposed with different degrees of success and applicability (1).

However, although many of the state of the art methods for addressing high class imbalances have achieved better prediction performances on such datasets, most of these methods still yield black box models that are not explainable in human terms. This problem, known as interpretability (2), is a very pertinent and ongoing issue in machine learning, that refers to obtaining models that not only achieve good prediction performances, but that are also interpretable by a human expert that can understand the model reasoning to decide on a certain class.

Another problem that affects many of the traditional classification methods is concerned with streaming data environments, where online learning and adapting to constantly changing (non-stationary) environments is required. This is due to the complexity of some of the state of the art classification methods that causes the training and prediction times to often be unacceptable for such streaming data applications.

Attending to these commons issues that affect traditional, as well as state of the art classification methods, this article approaches the problem of imbalanced datasets from the perspective of Evolving Fuzzy Inference Systems (EFIS) (3). EFIS are fuzzy inference systems (FIS) that incorporate mechanisms that allow for adapting to abrupt changes in the streaming data, while remaining computationally lightweight in terms of both memory and processing requirements. Therefore, as fuzzy rule based (FRB) systems, EFIS are much more interpretable than traditional machine learning methods, as they are based on fuzzy logic [19]. Furthermore, they are also well suited for the growing challenge of streaming data environments.

In 2012, a new type of EFIS architecture was introduced (4), using AnYa type fuzzy rules and the principles first introduced with the Empirical Data Analysis (EDA) framework (5). Contrarily to traditional FRB systems, such as the Mamdani and Takagi-Sugeno fuzzy systems, AnYa type fuzzy rules introduced a significantly simplified alternative to define

the antecedent part of the rules, by replacing the parametric membership functions with non-parametric data clouds.

Attending to the non-parametric nature of the rule antecedents, AnYa type systems are usually less computationally expensive to train than other FRB systems, making them very adequate to online learning tasks and streaming data applications. Despite this being the original task for which AnYa type systems were introduced, they have since been successfully applied to static classification and regression problems with promising results.

In this article, the Zero-Order Autonomous Learning Multi Model (ALMMo-0) classifier (6) is used to approach the problem of highly imbalanced datasets. The ALMMo-0 classifier is an AnYa-type FIS, meaning that it is non-parametric and lightweight when compared to traditional fuzzy systems, as its antecedents are cloud based. Furthermore, they are also non-iterative and fully autonomous, making them ideal for streaming data environments.

Despite its simplicity, ALMMo-0 classifiers have been shown to outperform benchmark classification methods, as well as traditional fuzzy classifiers. More importantly for the topic of this thesis, ALMMo-0 classifiers model each class independently, reducing the bias towards the underrepresented classes

In (7), ALMMo-0 classifiers were used to classify different heart disorders using sound data. The main goal of the proposed approach was to build highly accurate models, while also providing interpretable and explainable rules that may be used by experts to better diagnose hear disorders. Results have shown that ALMMo-0 classfiers were able to obtain better results than state-of-the-art methods, and also that the performance of the original algorithm could be slightly improved by adding a standardization and normalization pre-processing layer.

The simplicity of ALMMo-0 classifiers makes them particularly suitable to be used as the base learners in ensemble learning methods, and also allows the creation of more lightweight and interpretable deep learning architectures. In (8), the Multi-Layer Multi-Model Images Classifier (MICE) ensemble was first introduced, as a fast deep learning network for handwriting recognition. The MICE ensemble was proposed in order to allow ALMMo-0 classifiers to be applied to more complex tasks such as image recognition, and consists of multiple ALMMo-0 base learners trained in parallel. The ensemble prediction is accomplished using a winner-takes-all voting strategy that takes each one of the learners predictions. The MICE ensemble also introduced the first ALMMo deep learning architecture, by adding a pre-processing layer that extracts GIST and HOG features from the raw image input. Each one of the ensemble base classifiers is then trained using a unique subset of the extracted features.

II. Imbalanced Datasets

Imbalanced datasets are classification datasets in which the objects are not equally distributed among the problem classes. One of the main implications of training classifiers on highly imbalanced datasets is that, broadly speaking, the final model will often show a bias towards classes that have more available samples, and in some cases may even completely ignore the underrepresented classes. This is due to the fact that most classifiers were designed assuming that class imbalance either does not exist, or if it exists, it is not severe enough to affect the classifier performance.

One type of imbalanced datasets that are particularly relevant are imbalanced binary classification datasets, where one of the classes (typically the negative) is significantly overrepresented and is known as the majority class. The other class (typically the positive) is known as the minority class. Imbalanced binary datasets are particularly relevant since multi-class problems may be decomposed into several binary problems, using either a one-vs-one or a one-vs-all approach.

Attending to the already discussed prevalence and relevance of imbalanced datasets, a wide variety of approaches have been proposed in the literature (9), and as such, the methods hereby presented pretend to be a broad overview of the different types of approaches that have been successfully applied in different problems. Broadly speaking, one can define two different types of approaches to imbalanced datasets: external (or data-level) approaches, and internal (or algorithm-level) approaches.

External approaches include all methods that do not modify in any way the learning algorithm, meaning that these methods are generally applicable to any classifier. Examples of such approaches include ressampling methods, feature selection methods and ensemble learning techniques.

Internal approaches include all methods that modify the learning algorithm, in order to improve the minority class detection. These include methods such as cost sensitive methods, threshold moving techniques and one class classification.

III. Zero-Order Autonomous Learning Multi-Model Classifiers

The Zero-Order Autonomous Learning Multiple Model (ALMMo-0) classifier is a non-parametric, non-iterative and fully autonomous AnYa type fuzzy rule-based (FRB) multi-class classifier. The structure of ALMMo-0 classifiers is composed of one sub-model per class, each one being trained only on its class samples.

Each one of the class sub-models is based on multiple AnYa type fuzzy rules with one data cloud associated to each rule's antecedent. These rules assume the structure shown in 1, where x is a sample, f_j^i is the focal point associated to the j^{th} rule of the i^{th} class classifier, and $Label^i$ is the respective class label.

$$\textbf{IF } x \sim f_j^i \textbf{ THEN } Label^i \tag{1}$$

Furthermore, each sub-model is trained independently of each other since only its class samples are used to iteratively create the clouds, meaning that the cloud structure for each class sub-model is not affected by the cloud structures of the remaining sub-models.

Therefore, the fully trained ALMMo-0 classifier is composed of multiple FRB systems, each one with its set of AnYa type rules. When classifying a new sample, every rule in each

one of the class-models will output a confidence score, denoted as λ_j^i.

The confidence score is a defined as a function of the distance between one data sample and one cloud focal point, as shown in 2. In the original article, the Euclidean distance is used, without any loss of generality.

$$\lambda_j^i = \exp -\frac{1}{2}\left\|x - f_j^i\right\|^2 \tag{2}$$

In order to assign a label to the sample, one must devise some form of rule based on the confidence scores of each sub-model. In ALMMo-0 classifiers, the winner takes all strategy is used. First, the maximum confidence score of each class sub-model $\lambda_{j^*}^i$ is found, and then, the winner takes all principle is used and the class sub-model with the highest confidence score assigns its label:

$$\hat{y} = \underset{i=1,2,\ldots,L}{\arg\max}(\lambda_{j^*}^i) \tag{3}$$

This classification strategy means that the data clouds effectively create Voronoi tessellation of the data space, dividing it into different sub-regions that belong to different classes. As mentioned, these data clouds are non parametric and no prior assumptions about the data are made.

Furthermore, the training algorithm that creates the clouds is non iterative and lightweight, meaning that ALMMo-0 classifiers are particularly adequate for streaming data applications, even though no specific mechanisms to adapt to abrupt changes in the data are proposed.

Let $\mathbf{x_k^i}$ be the k^{th} sample from the i^{th} class. The sample is first norm normalized, as shown in 4, effectively removing one degree of freedom from the data space and projecting the sample to a unit radius hyper sphere centered around the origin.

$$\mathbf{x_k^i} \leftarrow \frac{\mathbf{x_k^i}}{\left\|\mathbf{x_k^i}\right\|} \tag{4}$$

If the sample is the first sample of its class, $\mathbf{x_1^i}$, then the respective sub-model parameters are initialized using (5).

$$\begin{cases} N^i \leftarrow 1 \\ \mu^i \leftarrow x_1^i \\ X^i \leftarrow \left\|x_1^i\right\|^2 \end{cases} \tag{5}$$

Where N^i is the number of samples used to train the sub-model, μ^i is the sub-model mean, and X^i is the sub-model average scalar product. Then, the sample is used to create the first cloud of its class sub-model, using (6).

$$\begin{cases} R^i \leftarrow 1 \\ M_1^i \leftarrow 1 \\ f_1^i \leftarrow x_1^i \\ X_1^i \leftarrow \left\|x_1^i\right\|^2 \\ r_1^i \leftarrow r_0 \end{cases} \tag{6}$$

Where R^i is the total number of rules in the sub-model, M_1^i is the number of samples used to update the cloud, f_1^i is the cloud's focal point, X_1^i is the cloud's average scalar product, and r_1^i is the cloud´s radius. This radius effectively describes a confidence score threshold around the focal point and its value is not a problem specific parameter. In this thesis, the initial cloud radius r_0 is assumed to be $r_0 = \sqrt{2(1-\cos 15)}$, as specified in the original article.

If the newly arrived sample $\mathbf{x_k^i}$ is not the first sample of its class, the sub-model parameters N^i, μ_k^i, and X_k^i are recursively updated using (7).

$$\begin{cases} N^i \leftarrow N^i + 1 \\ \mu^i \leftarrow \frac{N^i-1}{N^i}\mu^i + \frac{x_k^i}{N^i} \\ X^i \leftarrow \frac{N^i-1}{N^i}X^i + \frac{1}{N^i}\left\|x_k^i\right\|^2 \end{cases} \tag{7}$$

The algorithm then checks for density anomalies in the sub-model, ie, regions of the data space where the unimodal density is either too high (high concentration of clouds) or too low (low concentration of clouds). This is accomplished by computing the unimodal density between the sample x_k^i and the sub-model mean μ^i, using (8), and comparing it to the unimodal densities between each cloud focal point in the sub-model f_j^i, computed using (9).

$$D(x_k^i, \mu^i) = \frac{1}{1 + \frac{\left\|x_k^i - \mu^i\right\|^2}{X^i - \|\mu^i\|^2}} \tag{8}$$

$$D(f_j^i, \mu^i) = \frac{1}{1 + \frac{\left\|f_j^i - \mu^i\right\|^2}{X^i - \|\mu^i\|^2}} \tag{9}$$

The maximum and minimum focal point densities are compared to the sample density, defining the first condition in the algorithm, shown in (10).

$$\begin{aligned} D(x_k^i, \mu^i) \; &> \; \max_{j=1,2,\ldots R^i}(D(f_j^i, \mu^i)) \vee \\ D(x_k^i, \mu^i) \; &< \; \min_{j=1,2,\ldots R^i}(D(f_j^i, \mu^i)) \end{aligned} \tag{10}$$

If Condition 1 is verified, then there is a density anomaly (either too high or too low) and a new cloud is created around the sample x_k^i, using (11).

$$\begin{cases} R^i \leftarrow R^i + 1 \\ M_j^i \leftarrow 1 \\ f_j^i \leftarrow x_k^i \\ X_j^i \leftarrow \left\|x_k^i\right\|^2 \\ r_j^i \leftarrow r_0 \end{cases} \tag{11}$$

Otherwise, if Condition 1 is not verified, then no density anomaly exists and a second condition, based on distance instead of density, is checked to assess if a nearby cloud already exists. First, the distances between the sample x_k^i and each focal point in the sub-model are computed, and then the nearest cloud focal point f_{j*}^i is found, using equation 12.

$$f_{j*}^i = \underset{j=1,2,\ldots R^i}{\arg\min}\left(\left\|x_k^i - f_j^i\right\|\right) \tag{12}$$

If the distance to the nearest focal point f_{j*}^i is less than its radius r_{j*}^i, then the sample is considered to be close enough

and the cloud is updated. This distance criteria is expressed by Condition 2, as shown in (13).

$$\left\|x_k^i - f_{j*}^i\right\| \leq r_{j*}^i \tag{13}$$

If Condition 2 is met, the nearest cloud parameters are updated using (14).

$$\begin{cases} M_{j*}^i \leftarrow M_{j*}^i + 1 \\ f_{j*}^i \leftarrow \frac{M_{j*}^i - 1}{M_{j*}^i} f_{j*}^i + \frac{x_k^i}{M_{j*}^i} \\ X_{j*}^i \leftarrow \frac{M_{j*}^i - 1}{M_{j*}^i} X_{j*}^i + \frac{1}{M_{j*}^i} \left\|x_k^i\right\|^2 \end{cases} \tag{14}$$

Otherwise, if Condition 2 is not met, a new cloud is created using (15).

$$\begin{cases} R^i \leftarrow R^i + 1 \\ M_j^i \leftarrow 1 \\ f_j^i \leftarrow x_k^i \\ X_j^i \leftarrow \left\|x_k^i\right\|^2 \\ r_j^i \leftarrow r_0 \end{cases} \tag{15}$$

The algorithm then proceeds to the next sample. The complete learning process is summarized below in Algorithm 1.

Algorithm 1: ALMMo-0 Training Algorithm

```
for each class i do
    for x_k^i in class i do
        Normalize x_k^i using (4);
        if sub-model i is empty then
            Initialize sub-model using (5) and (6);
        else
            Update sub-model parameters using (7);
            Compute sample density using (8);
            Compute focal densities using (9);
            if Condition 1 (10) is met then
                Create new cloud using (11);
            else
                Find nearest cloud using (12);
                if Condition 2 (13) is met then
                    Update nearest cloud using (14);
                else
                    Create new cloud using (15);
                end
            end
        end
    end
end
```

IV. Proposed Modifications

A. One Class Classifier

In one-class classification (10), models are trained using only the normal (majority) class data with the goal of training a model that distinguishes between normal system behaviour and abnormal behaviour. This task, also known as outlier detection or novelty detection, has been successfully applied to highly imbalanced datasets. Concretely, several clustering-based one-class classification models have been introduced, and their general principles can be applied to ALMMo-0 classifiers.

One could conceive a one-class adaptation of the original algorithm, by only training the majority class sub-model, and therefore creating only one set of rules (clouds), that describe the normal behaviour of the system. However, one must also define some classification criterion, that is based on the cloud structure of ALMMo-0 classifiers.

Recalling the training process described by Algorithm 1, Condition 2 defines a proximity criterion, in which the cloud radius is set as a decision boundary to assess whether or not a sample is within a cloud region of influence. Therefore, one could use this criterion to classify a sample, by classifying samples within a cloud's region of influence as normal, and abnormal otherwise. Figure 1 illustrates this classification strategy.

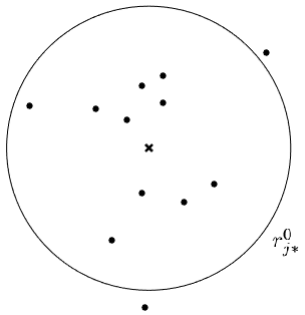

Fig. 1. Data cloud with its support samples and the radius defining a proximity criterion

Formally, this classification criterion is as shown in (16), where $\hat{y}$ is the class prediction, x is a sample, f_{j*}^0 is the closest majority class focal point, and r_{j*}^0 is its respective cloud´s radius.

$$\begin{cases} \hat{y} = 0 & \left\|x - f_{j*}^0\right\| < r_{j*}^0 \\ \hat{y} = 1 & otherwise \end{cases} \tag{16}$$

Furthermore, the clouds radius are already iteratively updated, as part of the training algorithm, meaning that no extra parameters must be estimated to apply this classification criteria.

This simple modification of the original algorithm seeks to be a lightweight approach to improve the performance of the original classifier on imbalanced binary classification. Comparing to the original classifier, no extra parameters must be estimated, and the minority class sub-model is not trained, meaning that the resultant model has fewer rules (clouds) and is slightly less complex.

Attending to the simpleness of the proposed modification, the only required change to the training algorithm is to only train the majority class sub-model, using the original training algorithm. The algorithm starts by finding the number of samples for each class. The class with the largest number of samples (majority class) is set as the default (normal) class. Thus, the minority class sub-model is not trained and it is effectively removed from the model.

B. Weighted Class Confidence

As already discussed, ALMMo-0 classifiers are defined by a set of sub-models, each one modelling its class with a set of AnYa type fuzzy rules. The classification strategy is, as already described, accomplished using the winner takes all approach. This approach takes each one of the sub-models confidence levels, and assigns the class of the sub-model with the highest confidence.

However, as already mentioned, ALMMo-0 also suffer from a bias towards the majority class because of its larger number of clouds that will often capture minority samples and misclassify the sample as false negatives. Furthermore, the misclassified minority samples often have a confidence level λ^1 that is only ever so slightly smaller than the confidence level for the closest majority cloud λ^0.

This suggests that one could introduce a weighting strategy that assigns different weights to the different class sub-models. Returning to the specific case of binary classification, if one introduces two complementary weights, denoted as ω_0 and ω_1, that weight the class confidence levels, λ^0 and λ^1, a winner takes all strategy may defined as in (17).

$$\hat{y} = \arg\max(\omega_0 \lambda^0_{j^*}, \omega_1 \lambda^1_{j^*}) \tag{17}$$

For the case where $\omega_0 = \omega_1$, this approach is effectively equivalent to the original winner takes all strategy. Even though only the relative values of the weights is important, for clarity and intrepretability sake, the weights are defined as in (18).

$$\begin{cases} 0 \leq \omega_0 \leq 1 \\ 0 \leq \omega_1 \leq 1 \\ \omega_0 + \omega_1 = 1 \end{cases} \tag{18}$$

Attending to this definition, only one of weights needs to be estimated, since the other weight is simply its complement. This means that the weighted confidence strategy has similarities to threshold moving approaches, where also one parameter, the decision threshold, is optimized.

In fact, if one defines the normalized activations, denoted as γ_0 and γ_1, as shown in (19), the classification criteria can be defined as in (20).

$$\begin{cases} \gamma_0 = \frac{\lambda^0}{\lambda^0 + \lambda_1} \\ \gamma_1 = \frac{\lambda^1}{\lambda^0 + \lambda_1} \end{cases} \tag{19}$$

$$\begin{cases} \hat{y} = 0 & \gamma_1 < 0.5 \\ \hat{y} = 1 & otherwise \end{cases} \tag{20}$$

Therefore, the normalized majority class confidence level, γ_1, acts as the classification decision threshold, meaning that the proposed weighted confidence strategy is effectively equivalent to adapt the ALMMo-0 classifier to a regressor, and then optimizing a decision threshold.

Nonetheless, since ALMMo-0 classifiers have been discussed as such, the approach hereby presented is always discussed from the weighted confidence perspective, in line with weighted voting strategies used in ensemble learning. Figure 2 illustrates this ensemble interpretation of the ALMMo-0 architecture.

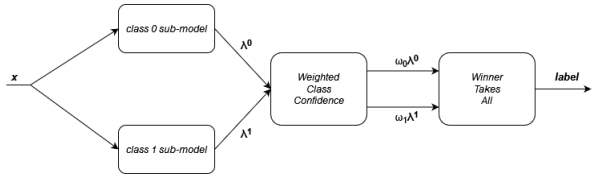

Fig. 2. Diagram of the modified ALMMo-0 classifier, in which the class confidence levels are weighted before the class decision

Finally, the question of how to optimize the class weights still remains to be answered. Similarly to what is usually done in threshold moving methods (11), the training algorithm is divided into two stages. In the first stage, the model is trained normally, using both class samples. In the second stage, the threshold value is optimized, usually using the classifier performance on a separate validation set, that was not used during the first training stage.

One crucial aspect that must be defined beforehand is the criteria used to assess the classifier performance on the validation set. Generally speaking, one may choose any relevant classification metric (12) to define a cost function that is only a function of ω_1. This cost function, denoted as F, is defined as in (21).

$$F(\omega_1) = F(\hat{y}(\omega_1), y) \tag{21}$$

Where F encodes some type of metric that is itself only a function of the predictions $\hat{y}$ and the actual labels y. Thus, the cost function is computed by first computing the predictions $\hat{y}$ for a given minority class weight ω_1, and then computing the classification metrics with the obtained results.

The cost function F may multiple local minimums, and therefore gradient methods may get stuck at one of the local minimums and yield a sub-optimal solution. Furthermore, in order to compute the cost function for a given ω_1, the class predictions $\hat{y}$ must first be computed, which even though is not prohibitively slow, may be slow enough to make grid search methods not applicable to online training environments.

Attending to the cost function characteristics and also to the need to minimize the number of times the cost function is sampled, the proposed method for the weight optimization is Baeysian optimization (13), which has been applied to a wide range of problems and, in particular, to hyper parameter optimization in machine learning models.

Bayesian optimization is an optimization technique that uses Bayes Theorem to direct the search in an efficient and effective manner, compromising between exploring unexplored regions of the search space, and refining the search in regions with high likelihood of containing a new minimum.

As mentioned, the training algorithm can be divided in two stages. The first stage, in which the training set is used to train both class sub-models, is completely similar to the one described for the original algorithm, as described in Algorithm 1.

The second stage is the optimization of the class confidence weights. In order to define the optimization problem, first the search interval for the value of ω_1 must be defined. As mentioned, in order to avoid over fit of the training data, a separate validation set is used to assess the performance of the unweighted model and optimize the weight values.

Since, as mentioned, the proposed algorithm is meant for the specific case of imbalanced binary classification, and since the minority class is known beforehand, in order to increase the bias towards the minority class, the minority class weight must be greater than 0.5, which corresponds to the unweighted model. If no performance improvements are found in the validation set, both weights keep the same value of 0.5.

Therefore, only the misclassified minority class samples of the validation set are used to define the search space. First, the validation set class predictions are computed for the unweighted model. Then, for each false negative, both class confidence levels, denoted as λ^0 and λ^1, respectively, are used to compute the minimum minority class weight that would correctly classify the misclassified sample, denoted as ω_1^*, and computed as shown in 22.

$$\omega_1^* = \frac{\lambda^0}{\lambda^0 + \lambda^1} \tag{22}$$

These candidate minority class weights are computed for each false negative and grouped in a vector of candidate weights, denoted as ω_1^{cands}. The search interval is then defined as in 23.

$$\omega_1 \in [0.5\ ,\ \max(\omega_1^{cands})] \tag{23}$$

By setting the maximum search value as the maximum of all the minority class candidate weights, its is guaranteed that the optimization can lead to a weight that correctly classifiers all the minority class samples that were misclassified by the unweighted model. Furthermore, since the initial cost is computed for the unweighted model, it is guaranteed that the performance of the model will never degrade relative to the original model.

Having the search interval defined, it is now possible to formulate the optimization problem, as shown in (24).

$$\begin{aligned} \text{Minimize} \quad & F(\omega_1) = F(\hat{y}(\omega_1), y) \\ \text{subject to} \quad & 0.5 \ \leq\ \omega_1\ \leq \max(\omega_1^{cands}) \end{aligned}$$

As already mentioned, the cost function $F(\omega_1)$ should encode an appropriate classification metric, or a combination of different classification metrics. However, it is crucial to mention that the goal of this optimization is not necessarily to improve the performance of the model in terms of the metrics encoded by the cost function. Since the validation set is used for the weight optimization, the optimal value will only be optimal for the validation data, meaning that it does not guarantee optimal in the test set.

V. Results

In order to assess the performance of both the original algorithms, as well as the proposed methods, six widely known and commonly used benchmark datasets were chosen. The chosen benchmark datasets pretend to be as diverse as possible, particularly regarding both the number and type of features. Furthermore, in order to study performances for different degrees of class imbalance, the datasets were ressampled to artificially obtain datasets with incrementally low minority class ratios. Thus, the chosen original datasets could not have too few samples or be overly imbalanced.

Attending to these restrictions the chosen datasets are the Australian Credit Approval (Australian) (14), German Credit (German) (15), Mammographic Mass (Mammographic) (16), Pima Indians Diabetes (Pima) (17), Diagnostic Wisconsin Breast Cancer (WBCD) (18) and Original Wisconsin Breast Cancer (WBCO) (19). The general characteristics of the chosen datasets are shown in Table I.

Dataset	Imbalance	Features		
		Real	Integer	Categorical
Australian	44.5%	3	5	6
German	30.0%	0	7	13
Mammographic	48.6%	0	5	0
Pima	34.9%	2	6	0
WBCD	37.3%	30	0	0
WBCO	35.0%	0	9	0

TABLE I
BENCHMARK DATASETS CHARACTERISTICS

It is important to recall that throughout this thesis, class imbalance is defined as the percentage of minority samples, and it is expressed as shown in (24), where N_{min} is the number of minority class samples, and N_{maj} is the number of majority class samples.

$$Imbalance = \frac{N_{min}}{N_{maj} + N_{min}} \tag{24}$$

In order to study the performance for the different methods and for different class imbalances, the benchmark datasets were ressampled to obtain class imbalances of 20%, 10%, 5% and 1%. The ressampled datasets are described in Table II.

For each test, 5-fold validation was used, meaning that 5 tests are performed, each one using a different fold as the testing set, and the remaining ones (80%) as the training set. Recalling the training algorithm proposed for the weighted class confidence ALMMo-0, the validation set was randomly sampled from the training folds, for each test. In order to achieve a 20% test, 10% validation, 70% train ratio, the validation set corresponds to 12.5% of the training set.

Attending to the random nature of the resampling procedure, it was repeated 5 times for each one of the original datasets, creating 5 different datasets, that were then used to obtain the different class imbalances, as already described. Thus, a total of 25 tests were performed for each method and for each resampled dataset.

Regarding the choice of the benchmark methods, and since this thesis seeks to compare the original ALMMo-0 performance to both common classification benchmark methods and other FRB systems, two groups of methods were chosen as benchmarks.

The first group of methods, composed of Logistic Regression (LR), Support Vector Machine (SVM), and shallow

Dataset	Imbalance	Majority Samples	Minority Samples
Australian	20%	5831	1458
	10%	5831	648
	5%	5831	307
	1%	5831	58
German	20%	5698	1424
	10%	5698	633
	5%	5698	300
	1%	5698	57
Mammographic	20%	7655	1914
	10%	7655	850
	5%	7655	403
	1%	7655	77
Pima	20%	5090	1272
	10%	5090	565
	5%	5090	268
	1%	5090	51
WBCD	20%	4026	1006
	10%	4026	447
	5%	4026	212
	1%	4026	40
WBCO	20%	4539	1135
	10%	4539	504
	5%	4539	239
	1%	4539	45

TABLE II
RESSAMPLED BENCHMARK DATASETS CHARACTERISTICS

Neural Network (NN), are commonly used benchmark methods used for classification tasks. The chosen methods are all relatively simple methods compared to state of the art methods, since the ALMMo-0 classifier is also a fairly simple and lightweight structure.

The second group of methods was chosen with the intent of comparing the original ALMMo-0 to both traditional FIS, as well as other more complex AnYa type FRB systems. For traditional FIS, the adaptive neuro-fuzzy inference system (ANFIS) was chosen. ANFIS are a type of artificial neural network based on a first-order Takagi-Sugeno inference system. Regarding AnYa type FRB systems, the natural choice is the ALMMo regressor, which is also a first-order FIS. Since ALMMo regressors share a similar antecedent structure, comparing their performance to the ALMMo-0 classifier is of particular interest.

Regarding the choice of cost functions for the proposed ALMMo-0 weighted class method, four pertinent and commonly used classification metrics were selected. These classification metrics are the geometric mean (GM), F1-Score (F1), Cohen's kappa coefficient (KC), and the Matthews correlation coefficient (MC).

A. ALMMo-0 and Benchmark Methods

Starting with the minority class prediction performance for low imbalance datasets, the results clearly suggest that, in general, ALMMo-0 models outperform LR, NN and ANFIS, while showing more mixed results when comparing to SVM and ALMMo.

Supporting this conclusion are the recall performances, which show that LR, NN and ANFIS under-perform on most of the tests. Recall performances are more mixed for SVM and ALMMo, showing a slight advantage.

Regarding the impact of the better recall performances on the majority class detection, recall results show that the ALMMo-0 classifier out-performs all the benchmark models. This might seem surprising, since better recall performances often result in worse precision performances. However, observing the specificity results, which show that the ALMMo-0 either matches or under-performs all the other methods with the exception of the ALMMo regressor, it becomes clear that the the benchmark methods generally ignored the minority class, meaning that near perfect specificity results were achieved without any positive class predictions.

Another interesting set of results are the accuracy performances, which show a general advantage of the ALMMo-0 classifier, outperforming all the benchmark methods except SVM. This further suggests that the better minority class detection verified for the ALMMo-0 did not significantly impact the overall prediction performance.

Regarding the remaining metrics, the results clearly suggest that the ALMMo-0 classifier generally outperforms the benchmarks methods, with the exception of the SVM, which shows more mixed results. This is expectable, since the geometric mean and F1-Score benefit from higher recall scores. Regarding the Kappa coefficient and the Matthews coefficient, the better results are also not surprising, since these metrics benefit from better minority class detection. Furthermore, these results once again suggest that the ALMMo-0 classifier is able to achieve better minority class detection without overly compromising the overall classification performance.

Regarding the minority class prediction performance for high imbalance datasets, it is very clear that the ALMMo-0 classifier generally outperforms all the benchmark methods. Once again, this conclusion is clearly supported by the recall results, which show that ALMMo-0 outperforms all the other models.

Furthermore, the results also show that the ALMMo-0 classifier outperforms all the benchmark models in terms of precision, while underperfoming in terms of specificity. The reason for this is that once again, the benchmark models disregard the minority class samples, meaning the number of positive class predictions is generally very low, or even zero.

Regarding the accuracy results, the ALMMo-0 classifier is out-performed by all the benchmark methods. The reason for this is simply that for highly imbalanced datasets, the accuracy is far from being a good assessment of the overall prediction performance of the models.

Regarding the remaining metrics, the results further support the overall better minority class prediction performance of ALMMo-0 models, since it outperforms all the benchmark methods, for each one of the metrics. Therefore, it once again becomes clear that ALMMo-0 models achieve better minority class prediction performance without compromising the overall prediction performance.

Attending to these results, it is clear that the proposed method is particularly well suited for highly imbalanced datasets, as it is able to achieve considerably better minority class prediction performance without overly compromising the overall classification performance.

All the discussed results are summarized by Table III.

Metric	Class Imbal.	Method				
		LR	SVM	NN	ANFIS	ALMMo
Accuracy	LOW	3/3/6	6/2/4	3/3/6	1/2/9	0/3/9
	HIGH	5/4/3	8/4/0	5/3/4	4/2/6	4/6/2
Recall	LOW	2/3/7	5/3/4	4/1/7	2/0/10	6/2/4
	HIGH	1/2/9	3/1/8	1/2/9	0/1/11	2/2/8
Precision	LOW	3/2/7	4/3/5	3/2/7	1/4/7	0/2/10
	HIGH	1/3/8	2/5/5	1/3/8	0/3/9	1/1/10
Specificity	LOW	5/2/5	5/2/5	7/0/5	7/1/4	2/3/7
	HIGH	8/2/2	8/4/0	8/2/2	7/3/2	8/3/1
G-Mean	LOW	3/2/7	5/3/4	4/1/7	2/0/10	4/3/5
	HIGH	1/2/9	4/0/8	1/2/9	0/1/11	2/2/8
F1-Score	LOW	3/1/8	4/5/3	3/2/7	1/1/10	1/2/9
	HIGH	1/2/9	4/0/8	1/1/10	0/0/12	0/1/11
Kappa	LOW	3/1/8	4/5/3	3/2/7	1/1/10	1/2/9
	HIGH	1/2/9	4/0/8	1/1/10	0/0/12	0/1/11
Matthews	LOW	3/1/8	6/3/3	3/2/7	1/1/10	1/2/9
	HIGH	1/2/9	4/2/6	1/1/10	0/0/12	0/1/11

TABLE III
WIN/TIE/LOSS COUNTS OF THE BENCHMARK METHODS AGAINST THE ALMMO-0 CLASSIFIER

B. ALMMo-0 and Proposed Modifications

Metric	Class Imbal.	Method				
		1CC	WCC			
			GM	F1	KC	MC
Accuracy	LOW	1/2/9	0/3/9	1/6/5	0/6/6	0/6/6
	HIGH	1/3/8	0/1/11	0/6/6	0/7/5	0/8/4
Recall	LOW	3/1/8	10/2/0	3/9/0	2/10/0	2/10/0
	HIGH	9/1/2	10/2/0	4/8/0	4/8/0	4/8/0
Precision	LOW	2/2/8	0/1/11	0/4/8	0/4/8	0/3/9
	HIGH	0/3/9	0/0/12	0/6/6	0/6/6	0/6/6
Specificity	LOW	3/1/8	0/1/11	0/6/6	0/7/5	0/8/4
	HIGH	0/3/9	0/1/11	0/4/8	0/6/6	0/7/5
G-Mean	LOW	2/2/8	6/6/0	2/10/0	2/10/0	2/10/0
	HIGH	9/1/2	8/4/0	2/10/0	4/8/0	3/9/0
F1-Score	LOW	2/0/10	0/3/9	2/7/3	1/8/3	1/8/3
	HIGH	5/4/3	0/1/11	1/8/3	1/8/3	1/7/4
Kappa	LOW	2/0/10	0/3/9	2/7/3	1/8/3	1/7/4
	HIGH	5/4/3	0/1/11	2/7/3	1/7/4	1/7/4
Matthews	LOW	2/0/10	0/3/9	1/8/3	0/9/3	1/7/4
	HIGH	4/5/3	0/2/10	1/7/4	1/9/2	1/7/4

TABLE IV
WIN/TIE/LOSS COUNTS OF THE PROPOSED METHODS AGAINST THE ALMMO-0 CLASSIFIER

*1) **One Class Classifier (1CC)**:* Regarding the accuracy, it is clear that, in general, the one class classifier significantly under performs for all datasets and class imbalances. Furthermore, the results show no significant relation between accuracy performance and the imbalance ratio.

In terms of recall, it is clear that the proposed method tends to outperform at higher imbalances, while underperforming at lower imbalances. Furthermore, the results show a clear relation between recall performance and class imbalance, as the recall performance tends to improve as the class imbalance becomes more pronounced.

Precision performance results clearly show that the proposed method significantly under performs for all datasets and class imbalances. This is expected, as the increase in recall performance implies a decrease in precision performance, as the reduction of false negatives leads to an increase of false positives. Furthermore, results show that the precision performance deteriorates for more pronounced class imbalances, following the opposite behaviour that was verified for the recall performance.

Regarding specificity, results show a slight decrease in performance for all datasets and class imbalances. Furthermore, there is also no clear relation between the specificity performance and the imbalance ratio. However, it is very clear that the decrease in specificity performance is always less substantial than the decrease in precision performance, since the number of false positives is much smaller than the number of true negatives.

Performance in terms of the geometric mean follows a similar behaviour to the one discussed for recall, as the proposed method only outperforms the original algorithm for high imbalances. These results are simple to justify, since, as discussed, no substantial changes in specificity performance were found and, therefore, the geometric mean performance is essentially controlled by the recall performance.

Regarding the F1-Score, the results show a similar behaviour to what was observed for the geometric mean, although the results are more mixed and dependent on the dataset. These results are justified by the recall and precision performances, clearly suggesting that for high imbalances, the recall improvements outweigh the drops in precision performance. Furthermore, the fact that the F1-Score gives equal importance to recall and precision further suggests that the proposed method provides a good compromise between minimizing false negatives and not excessively increasing the number of false positives.

Regarding the Kappa and Matthews coefficients, the results shows that the proposed method under-performs for low imbalances, while showing mixed and highly dataset dependent results for high imbalances. Nonetheless, since there is no significant performance change regarding these metrics, and since the recall performance clearly improved, it is clear that for high imbalances the proposed method provides a clear improvement on the minority class prediction performance without compromising the overall classification performance.

Attending to the results presented so far for the individual metrics, one can arrive at some general conclusions about the proposed one class classifier. Firstly, it is clear that there is an improvement in the minority class detection for high class imbalances, as is clearly shown by the recall improvements. It is also clear that the minimization of false negatives does not exaggeratedly increase the number of false positives. However, one could still argue that, in certain contexts and applications, the increase in false positives may still be too large and may not justify the reduction of false negatives.

*2) **Weighted Class Confidence (WCC)**:* Accuracy performance results show that, similarly to the one class classifier, the proposed method generally under performs for all datasets and class imbalances. However, contrarily to what was observed for the one class classifier, the results also show a clear relation between higher class imbalances and higher accuracy penalties.

Furthermore, the accuracy results also suggest that the performance for GM is clearly distinct from the other cost functions, as it is clear that GM yields the highest drops in accuracy, not only when compared to original ALMMo-0, but

also when compared to the one class classifier. Cost functions F1, KC and MC show a general small under performance, with an overall moderate drop for high imbalances.

Recall results show that the proposed method generally outperforms across all datasets and, in particular, high class imbalances. This is in contrast with what was observed for the one class classifier, which has shown moderate recall drops for low class imbalances. Therefore, the results suggest that the weighted class confidence method is, in general, more adequate for lower class imbalances than the proposed one class classifier.

Furthermore, the recall performances show once again a clear difference between GM and the remaining cost functions, as GM not only significantly outperforms the other cost functions, but also significantly outperforms the one class classifier. This is expected, as GM has also shown the largest drop in accuracy, implying that GM achieves the largest reduction in false negatives, at the cost of having the largest increase in false positives. Cost functions F1, KC and MC all show comparable results, achieving a more subtle false negative minimization, while also not exaggeratedly increasing the number of false positives.

As expected, precision results show that the proposed method under-performs for all cost functions, and also support what was concluded from the recall performances, as GM shows the largest drops, particularly at higher imbalances. Cost functions F1, KC and MC show less significant drops in precision since, as already mentioned, they also achieve more modest recall improvements.

The specificity results further support what was concluded from the previous metrics, showing that, as it was the case for the one class classifier, there is a general under performance for all datasets and class imbalances. Furthermore, cost functions F1, KC and MC once again show comparable results which are also comparable to the one class classifier. Cost function GM is the only method that shows a significant drop in specificity, particular at higher class imbalances since, as was already discussed, it significantly increases the number of false positives.

Geometric mean performances suggest a very similar behaviour to the one observed for the one class classifier. Cost functions F1, KC and MC achieve moderate improvements at higher class imbalances, in line with their recall performance, while still under performing the one class classifier. Cost function GM shows once again the largest improvement, since it also achieves the largest recall improvements.

It is important to mention that, despite GM using the geometric mean to optimize the class weighting, this is most likely not the reason why it outperforms the other methods, since, as it will be discussed for F1, KC and MC, the results show no clear correlation between the cost function metric and the actual model performance on that same metric. Thus, the most likely explanation for the substantial performance increase that GM shows for the geometric mean is that, since the specificity suffers only moderate drops, using the geometric mean as the cost function is essentially equivalent to optimizing only the recall performance, meaning that no compromise between false negatives and false positives is considered.

Observing the results for F1-Score, Kappa and Matthews, it once again becomes clear that cost function GM and the other cost functions yield different results. Cost function GM shows larger drops than the other cost functions and the one class classifier, once again due to the large increase in false positives that causes an overall classification performance penalization.

Attending to these results, it is clear that the proposed method is not only able to achieve better minority class detection, but also that the different cost functions yield models with different biases towards the minority class. In particular, the results suggest that the GM cost function shows the strongest performance on low imbalances, while the remaining cost functions may be better suited for high imbalances, as they provide a more balanced compromise between recall and precision performances.

VI. Conclusion

This thesis proposed two main goals: studying and comparing the performance of ALMMo-0 classifiers to other relevant benchmark methods, and proposing methods based on the original ALMMo-0 that seek to mitigate the effect of highly imbalanced datasets.

Regarding the first point, the results clearly suggest that ALMMo-0 classifiers achieves better minority class prediction performance when compared to benchmark methods such as LR, SVM and NN, when compared to traditional FRB systems such as the ANFIS, and also when compared to the first-order ALMMo regressor.

Furthermore, the selected benchmark methods were left unmodified, meaning that the original algorithms, as well as the datasets, were left unmodified. Thus, the results also suggest that these original algorithms in general show very poor performances on highly imbalanced datasets, and, in many cases, completely ignore the minority class.

Regarding the second point, two modifications of the original ALMMo-0 algorithm were proposed: a one class classifier, and a weighted class confidence approach. The results suggest that both methods achieve, to different extents, better minority class detection that the original ALMMo-0 classifier.

Regarding the proposed one class classifier, the results suggest that it outperforms the original ALMMo-0 on datasets that display high class imbalances. This conclusion is supported by the overall higher recall scores, as well as the higher GM, F1, KC and MC values. Furthermore, the overall better minority class detection is achieved without increasing too much the number of false positives.

The proposed one class classifier is also a remarkably simple modification of the original algorithm, as it simply removes the minority class sub-model, leaving the rest of the training algorithm unmodified. Since it uses the clouds radius as the classification decision threshold, which are parameters that already must be estimated during the training process, the proposed method does not introduce any complexity to the original algorithm.

Regarding the proposed weighted class confidence approach, the results suggest that the different cost functions

yield models with remarkably different characteristics. In particular the GM cost function tends do yield models with higher biases towards the minority class, meaning that it gives more weight to positive class samples. As such, the weighted class confidence approach using the GM cost function is particularly adequate for datasets that show a moderate class imbalance, as the resultant models show significant increases in recall and acceptable drops in precision. However, on highly imbalanced, a large drop in precision is often observed, meaning that the models may not be applicable to some domains.

The remaining cost function, F1, KC and MC, yield slightly different results but, in general, generate models with a more subtle bias towards the minority class. In particular, for highly imbalanced datasets, these cost functions yield models that still achieve significant recall improvements, but also do not exaggeratedly compromise the majority class detection.

Still regarding the proposed weighted class confidence approach, some issues were observed, particularly in the optimization process. As evidenced by the relatively similar results for the F1, KC and MC cost functions, it is clear that for small validation sets, the optimisation procedure will not be able to differentiate between the different cost functions, as the small number of validation samples will often force the process to converge towards the same weight value that minimizes the different cost functions.

Furthermore, the optimization results also show that for smaller validation sets, it is often observed that the model remains unweighted, meaning that increasing the minority class weight did not translate into a minimization of the cost function.

Nonetheless, the results obtained for the proposed methods are still very encouraging, as they offer, in general, different models that outperform the original ALMMo-0, which itself also was shown to outperform the selected benchmark methods.

References

[1] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0192-5

[2] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," *Proceedings - 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018*, pp. 80–89, 2019.

[3] I. Škrjanc, J. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey," *Information Sciences*, vol. 490, pp. 344–368, 2019.

[4] P. Angelov and R. Yager, "A new type of simplified fuzzy rule-based system," *International Journal of General Systems*, vol. 41, no. 2, pp. 163–185, 2012.

[5] P. Angelov, X. Gu, D. Kangin, and J. Principe, "Empirical data analysis: A new tool for data analytics," *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*, pp. 52–59, 2017.

[6] P. Angelov and X. Gu, "Autonomous learning multi-model classifier of 0-Order (ALMMo-0)," *IEEE Conference on Evolving and Adaptive Intelligent Systems*, vol. 2017-May, pp. 22–28, 2017.

[7] E. Soares, P. Angelov, and X. Gu, "Autonomous Learning Multiple-Model zero-order classifier for heart sound classification," *Applied Soft Computing Journal*, vol. 94, p. 106449, 2020. [Online]. Available: https://doi.org/10.1016/j.asoc.2020.106449

[8] P. Angelov and X. Gu, "MICE: Multi-layer Multi-model Images Classifier Ensemble," 2017.

[9] A. Singh and A. Purohit, "A Survey on Methods for Solving Data Imbalance Problem for Classification," *International Journal of Computer Applications*, vol. 127, no. 15, pp. 37–41, 2015.

[10] C. X. Ling and V. S. Sheng, "Cost-Sensitive Learning and the Class Imbalance Problem," *Encyclopedia of Machine Learning*, no. January 2010, pp. 231–235, 2008. [Online]. Available: http://www.springer.com/computer/ai/book/978-0-387-30768-8%5Cnhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.164.4418&rep=rep1&type=pdf

[11] G. Collell, D. Prelec, and K. R. Patil, "A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data," *Neurocomputing*, vol. 275, pp. 330–340, 2018. [Online]. Available: https://doi.org/10.1016/j.neucom.2017.08.035

[12] S. S. Mullick, S. Datta, S. G. Dhekane, and S. Das, "Appropriateness of performance indices for imbalanced data classification: An analysis," *Pattern Recognition*, vol. 102, p. 107197, 2020. [Online]. Available: https://doi.org/10.1016/j.patcog.2020.107197

[13] P. I. Frazier, "A tutorial on bayesian optimization," 2018.

[14] "Australian dataset," https://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval).

[15] "German dataset," https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

[16] "Mammographic dataset," https://archive.ics.uci.edu/ml/datasets/mammographic+mass.

[17] "Pima dataset," https://www.kaggle.com/uciml/pima-indians-diabetes-database.

[18] "Wisconsin breast cancer (diagnostic) dataset," https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic).

[19] "Wisconsin breast cancer (original) dataset," https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original).