An Empirical Study on Supervised and Unsupervised Fuzzy Measure Construction Methods in Highly Imbalanced Classification

Mikel Uriz, Daniel Paternain, Humberto Bustince, Mikel Galar
Institute of Smart Cities
Department of Statistics, Computer Science and Mathematics
Public University of Navarre
Pamplona, Spain
{mikelxabier.uriz,daniel.paternain,bustince,mikel.galar}@unavarra.es

***Abstract*—The design of an ensemble of classifiers involves the definition of an aggregation mechanism that produces a single response obtained from the information provided by the classifiers. A specific aggregation methodology that has been studied in the literature is the use of fuzzy integrals, such as the Choquet or the Sugeno integral, where the associated fuzzy measure tries to represent the interaction existing between the classifiers of the ensemble. However, defining the big number of coefficients of a fuzzy measure is not a trivial task and therefore, many different algorithms have been proposed. These can be split into supervised and unsupervised, each class having different learning mechanisms and particularities. Since there is no clear knowledge about the correct method to be used, in this work we propose an experimental study for comparing the performance of eight different learning algorithms under the same framework of imbalanced dataset. Moreover, we also compare the specific fuzzy integral (Choquet or Sugeno) and their synergies with the different fuzzy measure construction methods.**

***Index Terms*—ensembles, fuzzy measures, aggregations, Choquet integral**

I. INTRODUCTION

The combination of classifiers into an ensemble have proven to be a good mechanism to outperform individual classifiers when facing supervised classification problems [1]. When dealing with ensembles, two aspects are crucial: how to generate diversity among classifiers and how to combine their outputs into a single response. In this work we will focus on the latter.

Many authors have explored the aggregation of classifiers of an ensemble [1]. The simplest approach, which can be performed by some simple averaging function (such as the arithmetic mean) is not able to capture the individual importance of each classifier in the ensemble, neither to exploit the positive or negative interaction among coalitions of classifiers. For this purpose, the use of fuzzy integrals (Choquet [2] or Sugeno integrals [3]), which are based on an underlying fuzzy measure, has been proposed and explored in many papers by the fuzzy community [4], [5]. The key and possibly the hardest issue when using fuzzy integrals, is the definition of the appropriate fuzzy measure.

Analyzing the literature, one can find two main methodologies for estimating fuzzy measures for the aggregation of classifiers: supervised and unsupervised. Supervised methods base the learning on a training set of pairs (x, y), where x is the collection of evidences of a certain hypothesis. i.e., probabilities of an instance belonging to a certain class provided by the set of classifiers, and y is the desired target output. Then, an optimization procedure is established in order to conduct the learning of the fuzzy measure. Here again, there exist many alternatives to optimize the cost function, such as gradient-based iterative algorithms [6], quadratic programming [7], neural networks [8] or genetic algorithms among others.

On the other side, we find unsupervised learning algorithms, in which the estimation of the fuzzy parameters is conducted by some *a priori* information about the problem itself or the knowledge we have about the sources of information to be fused (classifiers). In this context, the estimation is in general based on both the confidence of each individual classifier and the adequacy of coalitions of classifiers [4].

Having in mind the different scenarios for the learning of the fuzzy measure, the main objective of this work is to develop an extensive empirical study comparing the aforementioned methodologies to deal with the estimation of the fuzzy measure, under the assumption that the aggregation of classifiers will be performed by a Choquet and a Sugeno integral (where possible) based on the constructed measure. In order to extract useful conclusions from the study, we will apply all the methods in a common framework, of imbalanced datasets [9].

The complete experimental study is formed of the thirty three most imbalanced datasets from KEEL dataset repository [10]. The UnderBagging ensemble method [11] is used for classifier generation and the Reduced Error Pruning with Geometric Mean [12] is applied to obtain the final ensemble. The performance of each method has been measures by the geometric mean (GM) of the performances of each class, since this metric is appropriate for dealing with imbalanced

This work was partially supported by the Public University of Navarre under the projects PJUPNA13 and PJUPNA1926.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

datasets. The results are analyzed using non-parametric stastiscal tests [13].

The remainder of the work is organized as follows. In Section II we recall the main preliminaries of the work, focused on aggregation functions, ensemble and the class imbalance problem. Later, in Section III, we explore in detail different fuzzy measure learning algorithms, both supervised and unsupervised. In Section IV, we explain the experimental framework of our study and in Section V, we show the results obtained as well as the conclusions obtained from it. We finish with the concluding remarks and future lines in Section VI.

II. Methods for Fuzzy Measure Learning

A. Aggregation functions

When facing information fusion problems, aggregation functions have been the most important mathematical tool to deal with them. An aggregation function is a function that takes n arguments and output a single value (function) which tries to represent the input set satisfying certain criteria.

Definition 1: [14]–[17]A mapping $f : [0,1]^N \to [0,1]$ is called an aggregation function if it satisfies boundary conditions, i.e., $f(0,\dots,0) = 0$ and $f(1,\dots,1) = 1$, and increasing monotonicity, i.e., if $x_i \leq y_i$ for all $i \in \{1,\dots,N\}$, then $f(x_1,\dots,x_N) \leq f(y_1,\dots,y_N)$.

Aggregation functions exhibit different behaviors. This work focuses on averaging aggregation functions (also called means), which are functions whose output is bounded by the minimum and maximum of the input vector, i.e., $\min(x_1,\dots,x_n) \leq f(x_1,\dots,x_n) \leq \max(x_1,\dots,x_n)$ for every $(x_1,\dots,x_n) \in [0,1]^n$. Examples of averaging aggregation functions are the arithmetic mean, the geometric mean, the median, the minimum and the maximum, and many others.

A special family of averaging aggregation functions are weighted aggregation functions, in which the inputs may have different importance according to a weighting vector $\mathbf{w} = (w_1,\dots,w_N)$, $w_i \in [0,1]$ and $\sum_{i=1}^{N} w_i = 1$.

A more general framework of aggregation functions that offer more flexibility than weighted aggregation functions are fuzzy measure-based fuzzy integrals, such as the Choquet or the Sugeno integral. The non-additivity of the fuzzy measure [18] that models the interaction among input sources allow to create versatile aggregation functions that consider not only the individual importance, but also the interaction among coalitions.

Definition 2: Let $\mathcal{N} = \{1,\dots,N\}$. A discrete fuzzy measure is a set function $m : 2^{\mathcal{N}} \to [0,1]$ satisfying boundary conditions $m(\emptyset) = 0$, $m(\mathcal{N}) = 1$ and monotonicity with respect to the inclusion, i.e. $m(A) \leq m(B)$ whenever $A \subset B$ for every $A, B \subseteq \mathcal{N}$.

Definition 3: Given a fuzzy measure $m : 2^{\mathcal{N}} \to [0,1]$, the discrete Choquet integral is given by

$$C_m(x_1,\dots,x_N) = \sum_{i=1}^{N} (x_{\sigma(i)} - x_{\sigma(i-1)}) m(\{\sigma(i),\dots,\sigma(N)\})$$

where $\sigma : \mathcal{N} \to \mathcal{N}$ is a permutation such that $x_{\sigma(1)} \leq \dots \leq x_{\sigma(N)}$ and $x_{\sigma(0)} = 0$ for convention.

Remark 1: Recall that the Choquet integral recovers the weighted arithmetic mean if m is additive, i.e. for any $A, B \subset \mathcal{N}$, $A \cap B = \emptyset$ then $m(A \cup B) = m(A) + m(B)$. Moreover, it recovers OWA operators if m is symmetric, i.e. for any $A, B \subseteq \mathcal{N}$, $m(A) = m(B)$ whenever $|A| = |B|$, then the Choquet integral is an OWA operator. Finally, the Choquet integral reduces to the arithmetic mean if m is symmetric and additive.

Definition 4: Given a fuzzy measure $m : 2^{\mathcal{N}} \to [0,1]$, the discrete Sugeno integral is given by

$$S_m(x_1,\dots,x_N) = \max_{i=1}^{N} \min\{x_{\sigma(i)}, m(\{\sigma(i),\dots,\sigma(N)\})\}$$

where $\sigma : \mathcal{N} \to \mathcal{N}$ is a permutation such that $x_{\sigma(1)} \leq \dots \leq x_{\sigma(N)}$.

B. Classifier ensembles

Ensembles of classifiers in Machine learning refers to the combination of several classifiers with the assumption that their combination will perform better than any single classifier in the set. Classifiers forming an ensemble need to be diverse enough so as to result in a successful combination. Although one usually has a single data source available for a given problem, several ways for constructing sets of diverse classifiers out of the same data have been developed [1]. Once the ensemble has been built, examples are classified querying all classifiers and aggregating their outputs. This aggregation phase is also known as combination or fusion [1] and is the main focus of this work.

Bagging [19] is a well-known method for generating ensemble. In Bagging, all the base classifiers are *a priori* similar and therefore, no weights are assigned to the classifiers (as it is done in Boosting). To learn each base classifier, Bagging simply takes a bootstrapped replica of the dataset, that is, samples the dataset with replacement until the same dataset size is achieved. This way, some examples will appear more than once in the replica, whereas others will not appear at all. Algorithm 1 shows the pseudo-code of this method.

Algorithm 1 Bagging

Input: S: Training set; N: Number of iterations; n: Bootstrap size; I: Weak learner

Output: Bagged classifier: $Class(x) = \arg\max_{y \in \mathcal{C}} \left(\frac{1}{N} \sum_{i=1}^{N} p_{c_i}(y|x) \right)$ where $p_{c_i}(y|x) \in [0,1]$ is the probability of x belonging to class y given by the classifier c_i

for $i = 1$ to N **do**
    $S_i \leftarrow$ RandomSampleReplacement(n,S)
    $c_i \leftarrow$ I(S_i)
end for

The most straightforward way to combine the outputs of Bagging-based classifiers is by means of weighted majority voting, where the confidences or probabilities of each classifier for each class are summed up and the class with highest confidence is predicted:

$$Class(x) = \arg\max_{y \in \mathcal{C}} \left(\frac{1}{N} \sum_{i=1}^{N} p_{c_i}(y|x) \right) \tag{1}$$

where $p_{c_i}(y|x) \in [0,1]$ is the output probability given by classifier c_i for class y and $\mathcal{C}$ is the set of classes (denote the number of classes $|\mathcal{C}|$ with C). Notice that classifiers not giving probabilities as outputs but confidence degrees in favor of each class can also be used (which would substitute $p_{c_i}(y|x)$).

Assuming that there is not independence between classifier outputs, a fuzzy measure based aggregation procedure is proposed. It is hypothesized that considering such dependencies, it will perform better than standard averging formulae.

C. The class imbalance problem

A dataset is considered to be imbalanced when the number of examples from the different classes are not evenly distributed. This scenario poses a challenge for most classifier learning algorithms [20] due to their accuracy-oriented design. In the case of two-class problems, the class of interest is usually under-represented [9] hindering classifier learning.

Ensembles specifically designed for this problem have been successfully applied to imbalanced scenarios [21], [22]. They mainly combine an ensemble learning algorithm like Bagging with preprocessing techniques. Among these ensembles, UnderBagging_RE-GM (UnderBagging with Reduced Error Pruning with Geometric Mean) [12] is a state-of-the-art method combining Bagging with random undersampling. Moreover, after learning a large pool of classifiers (100 in this work), a pruning is applied to reduce the number of classifiers to 21 (a number recommended in [23] after a thorough experimental study). Readers are referred to [12] for more details. This challenging scenario is seleted to understand whether fuzzy measure-based aggregations can make a difference in a difficult framework, but where there may still be room for improvement due to its inherent difficulties.

One key point when dealing with imbalanced datasets is that one should make use of the proper measures to evaluate the performance of the methods. Accordingly, the geometric mean (GM) [24] between the True Positive Rate ($\mathrm{TP}_{rate} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$) and the True Negative Rate ($\mathrm{TN}_{rate} = \frac{\mathrm{TN}}{\mathrm{FP+TN}}$) (the accuracy over the positive and negative classes, respectively) is selected to more appropriately measure the successes over both classes [9].

III. Related work

Fuzzy integrals, such as Choquet or Sugeno integrals, require the construction of a fuzzy measure in order to model the sources of information to be fused. When applying these integrals to a specific problem, as classification or ensemble aggregation, the coefficients of the fuzzy measure must be specified. Since it is not easy (from an expert point of view) to decide the value of each coefficient, the value of the fuzzy measure is typically computed by some algorithm. There are two different ways to learn these coefficients: supervised and unsupervised. The supervised learning algorithms are based on some previously known (ideal) values that must be fit by some optimization algorithm. On the contrary, unsupervised algorithms use some kind of external (a priori) knowledge to estimate the value of the coefficients.

In the following subsections we will detail several supervised and unsupervised learning algorithms for estimating the coefficients of a fuzzy measure.

A. Supervised learning of the fuzzy measure

1) Revised Heuristic Least Mean Squares algorithm: One of the first supervised algorithms for learning a fuzzy measure was given in [6] as an iterative gradient-based learning algorithm, which is called Heuristic Least Mean Squares algorithm (HLMS). The algorithm starts from a set of m training examples $(x^{(j)}, y^{(j)})$, where $x^{(j)}$ is a n-dimensional input vector and $y^{(j)}$ is the target output to be obtained by a fuzzy measure-based aggregation applied over $x^{(j)}$. For example, focusing on the Choquet integral, the algorithm tries to learn the parameters of a fuzzy measure such that $C_\mu(x^{(j)}) = y^{(j)}$ for every $j = 1, \ldots, m$. This requires calculating, for each training example, the individual error

$$e^{(j)} = C_\mu(x^{(j)}) - y^{(j)}.$$

From $e^{(j)}$, the corresponding coefficients of the fuzzy measure that have been involved in the calculation of $C_\mu(x^{(j)})$ are updated so as to minimize the individual error $e^{(j)}$. Finally, the monotonicity, that may be violated in the update, is corrected (see [6], [25] for more details about monotonicity correction).

The algorithm was improved in [25] by proposing a pure gradient-based update formula, as well as some improvements regarding the untouched coefficients and their influence in the monotonicity correction. This algorithm is the so called revised heuristic least mean squares algorithm (rHLMS for short).

2) Quadratic programming based optimization of fuzzy measures: The idea of using quadratic programming (QP) for learning a fuzzy measure was explored in [26]. Later, in [27], the authors recover these ideas and under the assumption that QP is appropriate for learning sparse matrices, they propose a new methodology to learn the fuzzy measure. Here again, they start from a training set $(x^{(j)}, y^{(j)})$, $j = 1, \ldots, m$ and try to minimize the Sum of Squared Error (SSE) given by

$$SSE = \sum_{j=1}^{m} (C_\mu(x^{(j)}) - y^{(j)})^2.$$

The monotonicity constraints of the coefficients of the fuzzy measure are incorporated in a compact linear algebra form, avoiding the necessity of monotonicy checking.

However, it is known that fuzzy measures learned by the minimization of SSE tend to suffer from over-fitting or converge to very complex solutions. In order to solve this problem, the authors propose to add a L_1-norm regularization term in the cost function to reduce the complexity of the obtained fuzzy measure.

3) Fuzzy measure as neural networks: Artificial Neural Networks have proven to be very useful in machine learning applications. Especially now, deep neural networks are considered to be one of the most powerful techniques for solving a great variety of problems. In [8], the authors propose to construct a neural network that is able to represent a fuzzy measure (NNFM). Due to the use of the back-propagation algorithm, the learning of the fuzzy measure can be easily done, specially if considering new programming and learning frameworks, such as PyTorch.

The main idea under the neural network is to model the coefficients of the fuzzy measure as learnable parameters. Specifically, and taking monotonicity constraints into consideration, the learnable parameters of the neural network represent, in fact, increments of the fuzzy measure. That is, a synapse represents how much a coefficient of the fuzzy measure increments when we add a new member to the coalition. Due to this way of modeling the fuzzy measure, there are no problems with monotonicity constraints during learning. Finally, positiveness of parameters are easily forced by ReLU activation functions.

B. Unsupervised learning of the fuzzy measure

The methods presented in this subsection have been specifically designed for being applied to ensemble aggregation by means of the Choquet integral. However, the underlying learning algorithm can be easily exported to any other application under the basis that we have some prior knowledge about the sources of information to be aggregated.

1) Coalition-based performance measure: The coalition-based performance (CPM) fuzzy measure learning algorithm was first given in [28] and later in [29]. Basically, the accuracy of every classifier and any combination of classifiers is collected (estimated in the training set). Then, starting from a trivial fuzzy measure in which each coefficient is set to the normalized cardinality ($m(a) = |A|/n$), the coefficients are increased or decreased according to the following behavior: if a coalition A of i classifiers (i varying from 1 to n) performs better (P_A) than the average accuracy of the coalitions of i classifiers ($\mu_{|A|}$), its corresponding coefficient is increased; otherwise, the coefficient is decreased. The increase/decrease is modeled by an hyperbolic tangent function given by:

$$m(A) = m_U(A) + \frac{tanh(100(P_A - \mu_{|A|}))}{2n},$$

where $m_U(A) = |A|/n$ is the trivial starting fuzzy measure and n is the total number of classifiers. The CPM algorithm is described in Alg. 2.

Algorithm 2 CPM

Input: $\mathcal{N}$: Set of classifiers; P_A: accuracy of any coalition $A \subseteq \mathcal{N}$.
Output: fuzzy measure: $m : 2^{\mathcal{N}} \rightarrow [0, 1]$.

Create $m_U : 2^{\mathcal{N}} \rightarrow [0, 1]$ a uniform fuzzy measure
for $A \subseteq \mathcal{N}$ **do**
  $\mu_{|A|} \leftarrow (1/\binom{n}{|A|}) \sum_{B \subseteq \mathcal{N}, |B|=|A|} P_B$
  $m_(A) = m_U(A) \frac{\tanh(100(P_A - \mu_{|A|}))}{2n}$
end for

2) A priori fuzzy measure: The A Priori Fuzzy Measure (APFM) was given in [30] and it starts from the accuracy of every classifier and any combination of classifiers (as in CPM). The accuracy is then normalized between 0.5 and 1 under the assumption that any combination performs better than a random classifier. Then, the coefficients of singletons are set to their corresponding normalized accuracy. When considering coalitions of classifiers, each coalition gets its normalized accuracy only if it is greater than the accuracy of every sub-coalition (to enforce monotonicity). Otherwise, the coalition inherits the maximum accuracy of any sub-coalition, implying that the coalition is no worse in performance than its best performing subset. The APFM algorithm is described in Alg. 3.

Algorithm 3 APFM

Input: $\mathcal{N}$: Set of classifiers; P_A: accuracy of any coalition $A \subseteq \mathcal{N}$.
Output: fuzzy measure: $m : 2^{\mathcal{N}} \rightarrow [0, 1]$.

$Max_P \leftarrow \max_{A \subseteq \mathcal{N}} P_A$
for $A \subseteq \mathcal{N}$ **do**
  $nm_A \leftarrow 1 - \frac{Max_P - P_A}{Max_P - 0.5}$
  if $|A| == 1$ **then**
    $m(A) \leftarrow nm_A$
  else
    $Max_B \leftarrow \max_{B \subsetneq A} nm_B$
    if $nm_A < nm_B$ **then**
      $m(A) \leftarrow nm_B$
    else
      $m(A) \leftarrow nm_A$
    end if
  end if
end for

3) Interaction-Sensitive Fuzzy Measure: The Interaction-Sensitive fuzzy mesure (ISFM) is based on both a confidence degree of each classifier and a pairwise similarity measure that represents the diversity of each pair of classifiers. At the beginning, the coefficient of the more confident classifier is set to its corresponding degree. Coefficients of less confident classifiers get their degree decreased inversely proportional to the similarity between them and the rest of more confident classifiers. The ISFM algorithm is described in Alg. 4.

Algorithm 4 ISFM

Input: $\mathcal{N}$: Set of classifiers; κ: confidence vector; S: similarity measure.
Output: fuzzy measure: $m : 2^{\mathcal{N}} \rightarrow [0, 1]$.

$\sigma \leftarrow$ permutation such that $\kappa_{\sigma(1)} \leq \cdots \leq \kappa_{\sigma(N)}$
for $i = 1, \ldots, N$ **do**
  $m(\{\sigma(i)\}) = \kappa_{\sigma(i)} \left(1 - \max_{j=\sigma(i)+1,\ldots,\sigma(n)} S(\sigma(i), j)\right)$
end for
for each $A \subseteq \mathcal{N}$ **do**
  $m(A) \leftarrow \sum_{i \in A} m(\{i\})$
end for
Normalize m to have $m(\mathcal{N}) = 1$

4) Modified Hüllermeier Measure: The Modified Hüllermeier Measure (MHM) was first given in [31] for an application of the Choquet integral in k-NN. However, in [4], the authors considered the same fuzzy measure with certain modifications for ensemble aggregation. In this method, having again the confidence degree of each classifier, we first construct an additive fuzzy measure where the coefficients of singletons are given by the normalized confidence degree. From this measure, the additivity is broken by both a parameter $\alpha \geq 0$ and a measure of the relative diversity of the collection of classifiers in the coalition. Thus, the coefficient is reduced with lower diversity and increased on the contrary. If α is set to zero, we recover the first additive measure. The MHM algorithm is described in Alg. 5.

Algorithm 5 MHM

Input: $\mathcal{N}$: Set of classifiers; $m' : 2^{\mathcal{N}} \rightarrow [0,1]$ original fuzzy measure; S: similarity measure; $\alpha \in [0,1]$: parameter.
Output: fuzzy measure: $m : 2^{\mathcal{N}} \rightarrow [0,1]$ associated to the instance x.

$\overline{ds} \leftarrow \max_{i \neq j \in \mathcal{N}} S(i,j) = 1 - \min_{i \neq j \in \mathcal{N}} S(i,j)$
for each $A \subseteq \mathcal{N}$ **do**
  if $|A| \leq 1$ **then**
    $div(A) \leftarrow 0$
    $rdiv(A) \leftarrow 0$
  else
    $div(A) \leftarrow \frac{2}{|A|^2 - |A|} \sum_{i<j \in A} 1 - S(i,j)$
    $rdiv(A) \leftarrow 2 \cdot div(A) \cdot \overline{ds} - 1$
  end if
  $m(A) \leftarrow m'(A)(1 + \alpha \cdot rdiv(A))$
  $m(A) = \max_{B \subseteq A} m(B)$
end for
Normalize m to have $m(\mathcal{N}) = 1$

5) Overlap index-based fuzzy measure: As in the previous method, the Overlap index-based fuzzy measure (OIFM) was firstly given in [32] for combining fuzzy rules in fuzzy rule-based classification systems. However, the usage of the measure for ensemble aggregation is quite similar. Basically, the fuzzy measure constructed is such that the coefficients of each coalition are, up to some extent, proportional to the sum of confidences of the classifiers belonging to it. In this sense, diversity is not taken into account. The overlap index used in the construction method allows one to increase flexibility and to break the additivity of the fuzzy measure due to the use of non-linear overlap indices. The OIFM algorithm is described in Alg. 6.

Algorithm 6 OIFM

Input: $\mathcal{N}$: Set of classifiers; $O : [0,1]^N \times [0,1]^N \rightarrow [0,1]$ overlap index; κ: confidence vector;
Output: fuzzy measure: $m : 2^{\mathcal{N}} \rightarrow [0,1]$.

Construct fuzzy set $E = \{(i, \kappa_i) | i = 1, \ldots, N\}$ associated to κ
for each $A \subseteq \mathcal{N}$ **do**
  Create fuzzy set E_A associated to A
  for $i = 1, \ldots, N$ **do**
    if $i \in A$ **then**
      $E_A(i) = \kappa_i$
    else
      $E_A(i) = 0$
    end if
  end for
  $m(A) \leftarrow O(E, E_A)$
end for
Normalize m so that $m(\mathcal{N}) = 1$

IV. Experimental Study

This section introduces the experimental framework considered for the empirical study.

A. General settings

This study considers the thirty three most imbalanced datasets from KEEL dataset repository [10], the most common benchmark for evaluating classifiers dealing with class imbalance [12], [21]. In Table I the total number of examples, number of features and IR (ratio between the majority and minority class examples) for each dataset are presented.

TABLE I
Summary description of the imbalanced datasets.

No.	Data-sets	#Ex.	#Atts.	IR	No.	Data-sets	#Ex.	#Atts.	IR
1	glass1	214	9	1.82	34	Glass04vs5	92	9	9.22
2	ecoli-0_vs_1	220	7	1.86	35	Ecoli0346vs5	205	7	9.25
3	wisconsin	683	9	1.86	36	Ecoli0347vs56	257	7	9.28
4	pima	768	8	1.87	37	Yeast05679vs4	528	8	9.35
5	iris0	150	4	2	38	Ecoli067vs5	220	6	10.00
6	glass0	214	9	2.06	39	Vowel0	988	13	10.10
7	yeast1	1484	8	2.46	40	Glass016vs2	192	9	10.29
11	haberman	306	3	2.78	41	Glass2	214	9	10.39
8	vehicle2	846	18	2.88	42	Ecoli0147vs2356	336	7	10.59
9	vehicle1	846	18	2.9	43	Led7digit02456789vs1	443	7	10.97
10	vehicle3	846	18	2.99	44	Glass06vs5	108	9	11.00
12	glass-0-1-2-3_vs_4-5-6	214	9	3.2	45	Ecoli01vs5	240	6	11.00
13	vehicle0	846	18	3.25	46	Glass0146vs2	205	9	11.06
14	ecoli1	336	7	3.36	47	Ecoli0147vs56	332	6	12.28
16	new-thyroid1	215	5	5.14	48	Cleveland0vs4	177	13	12.62
15	new-thyroid2	215	5	5.14	49	Ecoli0146vs5	280	6	13.00
17	ecoli2	336	7	5.46	50	Ecoli4	336	7	13.84
18	segment0	2308	19	6.02	51	Yeast1vs7	459	8	13.87
19	glass6	214	9	6.38	52	Shuttlec0vsc4	1829	9	13.87
20	yeast3	1484	8	8.1	53	Glass4	214	9	15.47
21	ecoli3	336	7	8.6	54	Pageblocks13vs4	472	10	15.85
22	page-blocks0	5472	10	8.79	55	Abalone9vs18	731	8	16.68
23	ecoli-0-3-4_vs_5	200	7	9	56	Glass016vs5	184	9	19.44
24	yeast-2_vs_4	514	8	9.08	57	Shuttlec2vsc4	129	9	20.5
25	ecoli-0-6-7_vs_3-5	222	7	9.09	58	Yeast1458vs7	693	8	22.10
26	ecoli-0-2-3-4_vs_5	202	7	9.1	59	Glass5	214	9	22.81
27	glass-0-1-5_vs_2	172	9	9.12	60	Yeast2vs8	482	8	23.10
28	yeast-0-3-5-9_vs_7-8	506	8	9.12	61	Yeast4	1484	8	28.41
30	yeast-0-2-5-7-9_vs_3-6-8	1004	8	9.14	62	Yeast1289vs7	947	8	30.56
29	yeast-0-2-5-6_vs_3-7-8-9	1004	8	9.14	63	Yeast5	1484	8	32.78
31	ecoli-0-4-6_vs_5	203	6	9.15	64	Ecoli0137vs26	281	7	39.15
32	ecoli-0-1_vs_2-3-5	244	7	9.17	65	Yeast6	1484	8	39.15
33	ecoli-0-2-6-7_vs_3-5	224	7	9.18	66	Abalone19	4174	8	128.87

The performance of each method in each dataset has been estimated using a 5-fold cross-validation repeated 5 times to account for the randomness in the partition and model construction. That is, each result is computed as the average of 25 runs. The GM is considered as performance measure, although it should be mentioned that similar conclusions may be drawn using the Area under the ROC Curve (AUC).

In accordance with other authors [13] and given that the conditions for applying parametric statistical tests may not be fulfilled, non-parametric statistical tests are used to support our comparisons. For comparing pairs of methods, the Wilcoxon test is considered, whereas the Friedman aligned-ranks test is used for multiple comparisons. In this case, if significant differences are found, Holm's *post-hoc* test is applied to check the null hypothesis of equivalence between a control method and the rest of the methods.

This experimental study is in line with previous works on the topic of ensembles for the class imbalance problem [12]. Consequently, the C4.5 decision tree [33] is used as base classifier. As has been explained, UnderBagging_RE [12] is applied, which performs a pruning phase after learning a large pool of classifiers. All the required parameters are detailed in Table II.

TABLE II
Parameters for C4.5 and UnderBagging algorithm.

Algorithm	*Parameters*
C4.5	Prune = True, Confidence level = 0.25, Confidence = Laplace Minimum number of item-sets per leaf = 2
UnderBagging	Pool of classifiers = 100, Number of final classifiers = 21 Pruning method = RE_GM

B. Summary of methods and parameter estimation

In this subsection the parameterization of the learning algorithms exposed in Section III is presented, especially those that concern the use of supervised algorithms. For unsupervised learning algorithms refer to [29] for more details of the implementation.

Due to the nature of the problem faced in this work, which is classification, a modification of the supervised algorithms is considered so as to learn two different measures (one for each class) instead of a single measure. This approach using several measures was proposed in [6] and seems to be the most adequate, since it would be the best way to handle problems with more than two classes.

rHLMS: following the indications in [6], the algorithm is slightly modified to deal with classification problems. In binary classification problems, the error to be minimized is given by

$$E^2 = \sum_{i=1}^{2}\sum_{j=1}^{m}\left|\sigma(C_{12}(x^{(j)}) - 1\right|^2$$

where

$$C_{12}(x^{(j)} = \begin{cases} C_{\mu_1}(x^{(j)}) - C_{\mu_2}(x^{(j)}) & \text{if } x^{(j)} \in C_1 \\ C_{\mu_2}(x^{(j)}) - C_{\mu_1}(x^{(j)}) & \text{if } x^{(j)} \in C_2 \end{cases}$$

and $sigma$ is the sigmoid function given by $\sigma(z) = 1/(1 + \exp -z)$.

QP: in this case the code provided by the authors[1] is used. The learning is divided into two steps. First, consider all the training examples belonging to class 1 and perform the learning. Afterwards, perform the same process with instances of class 2 for learning the second fuzzy measure.

TorchNN: the implementation of [8] is provided in Python language using PyTorch[2]. The algorithm learns a measure for each class. The calculation of the error is adapted by normalizing the output of each Choquet integral. The squared error between the normalized output of the first method and the true target is then considered.

Finally, is it is important to notice that when considering fuzzy measures obtained by supervised algorithms, the underlying aggregation function will be the Choquet integral. However, on those fuzzy measures obtained by unsupervised algorithms, both the Choquet or Sugeno integral will be considered.

V. Results and discussion

In this section we carry out the experimental study to evaluate the performance of the different fuzzy measure construction methods in the framework of imbalanced datasets. Table III presents the results obtained for each method in the comparison in terms of GM. The best result in each dataset is marked in **bold-face**. Finally, the last row summarizes the results over all datasets for each method showing the average performance.

Looking at Table III, one can observe that there are important differences among the results of the different methods. On the one hand, among unsupervised methods there are no large differences in absolute terms either considering Choquet or Sugeno, although C_{ISFM} is below the rest. With respect to supervised methods, TorchNN is performing much better than rHLMS and QP, with comparable results with unsupervised methods. Classic aggregations achieve also competitive performances in average. Overall, C_{MHM} achieves the highest accuracy, but q proper statistical analysis must be considered to better understand the differences among the methods.

A comment on the low performance of rHLMS and QP is warranted. In the previous section, it was mentioned that supervised methods are implemented in such a way that a measure for each class is learned. Doing so in TorchNN allows to learn both measures at the same time, taking their interactions into account when learning the coefficients. However, this is not easily integrated in rHLMS and QP, where the whole learning procedure would need to be altered. Consequently, this provokes favoring the majority class, hindering the final performance of the ensemble.

Continuing with the empirical comparison, a hierarchical statistical analysis of the results was undertaken. First, performing intra-family comparisons were considered. Later, inter-family comparisons among the best methods of each family were carried out. Figure 1 summarizes this study. The result of each test is presented in each intersection among methods, showing both the ranks obtained by each method and the p-valued obtained by the statistical test. The Wilcoxon test is applied when the comparison only involves pairs of methods and both the ranks and the p-value are shown. In this case, the larger the ranks, the better. Otherwise, when multiple methods are compared, Friedman aligned-ranks test is carried out, where the lower the ranks, the better. In case of statistical differences being found, the p-value of Holm's post-hoc test is presented, comparing the best methods with each one of the rest. The p-value is given in **bold-face** whenever significant differences in favor of the winning method exists (with $\alpha = 0.05$).

In the comparison of unsupervised methods, both using Choquet and Sugeno integrals, ISFM is statistically the worst performer, performing worse than the best method in each case, MHM with Choquet and OIFM with Sugeno. However, these methods do not obtain significant differences with respect to the others. Overall, MHM, OIFM and CPM can be highlighted, being the best in both cases, achieving high p-values (no significant differences with the winner). When comparing the winners of the previous comparisons, Choquet integral with MHM gets a higher number of ranks, although no statistical differences are found between the two integrals.

Moving to supervised methods, as expected, TorchNN statistically outperforms the other two methods due to the already explained reasons. In the case of classical aggregations, there seems to be a tendency in favor of the WAM, although no statistical differences are found with respect to the unweighted alternative.

Finally, comparing the results of the best method for each family: Choquet with MHM, TorchNN and WAM. Looking at the p-values, one can observe that no statistical differences are found, although ranks are in favor of Choquet with MHM. This

[1]https://github.com/B-Mur/ChoquetIntegral

[2]https://github.com/aminb99/choquet-integral-NN

TABLE III
AVERAGE GM SCORE OF EACH METHOD FOR EACH DATASET

Dataset	Unsupervised (Choquet) C_{CPM}	C_{APFM}	C_{ISFM}	C_{MHM}	C_{OIFM}	Unsupervised (Sugeno) S_{CPM}	S_{APFM}	S_{ISFM}	S_{MHM}	S_{OIFM}	Supervised (Choquet) rHLMS	QP	TorchNN	Classic AM	WAM
abalone19	0.6993	**0.7078**	0.7007	0.7037	0.7034	0.6792	0.7070	0.6916	0.6957	0.6979	0.4546	0.1913	0.6887	0.7052	0.7065
abalone9-18	0.7232	0.7092	0.7164	**0.7271**	0.7164	0.6968	0.7212	0.7074	0.7075	0.7239	0.6386	0.4704	0.7078	0.7260	0.7245
cleveland-0_vs_4	0.8405	0.7931	0.7466	**0.8625**	0.8293	0.8298	0.8118	0.7382	0.8316	0.8357	0.7284	0.6062	0.8296	0.8460	0.8485
ecoli-0-1-3-7_vs_2-6	0.7558	0.8227	0.7170	**0.8755**	0.7583	0.7461	0.8081	0.7068	0.8635	0.7357	0.7207	0.6191	0.7735	0.7953	0.7912
ecoli-0-1-4-6_vs_5	0.8839	0.8867	0.8377	0.8888	0.8855	0.8756	0.8791	0.8303	**0.8901**	0.8850	0.8412	0.7984	0.8861	0.8811	0.8826
ecoli-0-1-4-7_vs_2-3-5-6	0.8529	0.8379	0.8231	0.8428	0.8528	0.8599	0.8341	0.8144	0.8449	0.8608	0.8522	0.7667	**0.8634**	0.8520	0.8536
ecoli-0-1-4-7_vs_5-6	0.8750	0.8735	0.8518	0.8765	0.8770	0.8768	0.8757	0.8523	0.8791	0.8788	0.8793	0.8128	**0.8920**	0.8783	0.8807
ecoli-0-1_vs_5	0.8959	0.9141	0.8665	**0.9168**	0.8904	0.9071	0.8800	0.8665	0.9130	0.9009	0.8658	0.7945	0.8916	0.8925	0.8925
ecoli-0-3-4-6_vs_5	0.8787	**0.9067**	0.8864	0.8992	0.8825	0.8779	0.9026	0.8854	0.8913	0.8763	0.8372	0.7741	0.8838	0.8797	0.8791
ecoli-0-3-4-7_vs_5-6	0.8769	0.8705	0.8604	0.8673	0.8788	0.8779	0.8685	0.8600	0.8599	0.8843	0.8664	0.8201	**0.8897**	0.8775	0.8775
ecoli-0-6-7_vs_5	0.8818	0.8634	0.8468	0.8726	0.8748	0.8822	0.8600	0.8452	0.8685	0.8833	0.8680	0.8525	**0.8880**	0.8822	0.8822
ecoli4	0.9217	**0.9342**	0.8672	0.9283	0.9156	0.9169	0.8981	0.8672	0.9244	0.9147	0.9062	0.8081	0.9020	0.9182	0.9182
glass-0-1-4-6_vs_2	0.6830	**0.7266**	0.7105	0.6927	0.7079	0.6848	0.7212	0.7088	0.6900	0.6727	0.5455	0.3742	0.6885	0.7043	0.7005
glass-0-1-6_vs_2	0.6728	0.6921	0.6546	0.6851	0.6726	0.6840	0.6543	0.6317	**0.6997**	0.6811	0.5242	0.3503	0.6741	0.6709	0.6748
glass-0-1-6_vs_5	0.9464	**0.9790**	0.9561	0.9348	0.9464	0.9464	0.9784	0.9561	0.9476	0.9464	0.7844	0.8678	0.9784	0.9464	0.9464
glass-0-4_vs_5	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**	**0.9939**
glass-0-6_vs_5	0.9224	0.9939	**0.9949**	0.9781	0.9224	0.9172	0.9939	**0.9949**	0.9645	0.9224	0.9715	0.9598	**0.9949**	0.9172	0.9204
glass2	0.6683	**0.6978**	0.6969	0.6354	0.6712	0.6244	0.6718	0.6900	0.6215	0.6855	0.5673	0.3494	0.6361	0.6234	0.6576
glass4	0.9104	0.8765	0.8456	0.9039	0.9120	**0.9216**	0.8991	0.8456	0.9033	0.9126	0.7444	0.7198	0.8839	0.9185	0.9203
glass5	0.9523	0.9852	0.9634	0.9587	0.9523	0.9513	0.9847	0.9634	0.9197	0.9513	0.8632	0.9032	**0.9867**	0.9513	0.9513
led7digit-0-2-4-5-6-7-8-9_vs_1	0.8152	0.8065	0.8206	0.8146	0.8148	0.8211	0.8171	0.8206	0.8110	0.8193	**0.8477**	0.8428	0.8366	0.8133	0.8135
page-blocks-1-3_vs_4	0.9797	0.9909	**0.9943**	0.9660	0.9806	0.9764	0.9884	**0.9943**	0.9676	0.9818	0.9910	0.9772	0.9939	0.9761	0.9761
shuttle-c0-vs-c4	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
shuttle-c2-vs-c4	**1.0000**	**1.0000**	**1.0000**	0.9897	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9897	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
vowel0	0.9664	0.9599	0.9588	0.9656	0.9658	0.9676	0.9581	0.9588	0.9658	0.9676	**0.9702**	0.9569	0.9700	0.9660	0.9660
yeast-0-5-6-7-9_vs_4	0.8023	0.8100	0.8004	**0.8118**	0.8026	0.8063	0.8086	0.8032	0.8099	0.8001	0.7757	0.6540	0.7969	0.8032	0.8054
yeast-1-2-8-9_vs_7	0.7331	0.7211	0.6982	0.7335	0.7269	0.6674	0.6968	0.6865	0.6879	0.7048	0.5496	0.2729	0.6794	0.7335	**0.7347**
yeast-1-4-5-8_vs_7	0.6227	0.5754	0.5850	0.6175	0.5991	0.5258	0.5899	0.5723	0.6227	0.5983	0.3324	0.1784	0.6003	**0.6348**	0.6260
yeast-1_vs_7	0.7671	0.7565	0.7292	0.7637	0.7704	0.7391	0.7422	0.7130	**0.7719**	0.7466	0.6209	0.5075	0.7337	0.7555	0.7590
yeast-2_vs_8	0.7751	0.7767	0.7606	0.7756	0.7765	0.7419	**0.7821**	0.7387	0.7808	0.7680	0.7408	0.7415	0.7355	0.7703	0.7694
yeast4	0.8546	0.8432	0.8198	0.8535	**0.8571**	0.8524	0.8292	0.8164	0.8507	0.8521	0.8125	0.6724	0.8383	0.8522	0.8525
yeast5	0.9577	**0.9648**	0.9533	0.9573	0.9571	0.9579	0.9562	0.9533	0.9582	0.9581	0.9497	0.9140	0.9581	0.9572	0.9571
yeast6	0.8604	0.8460	0.8296	0.8458	0.8605	0.8478	0.8406	0.8252	0.8347	0.8555	0.8322	0.7283	0.8541	0.8633	**0.8637**
Mean	0.8476	0.8520	0.8329	**0.8527**	0.8471	0.8380	0.8470	0.8282	0.8473	0.8453	0.7841	0.7054	0.8464	0.8480	0.8493

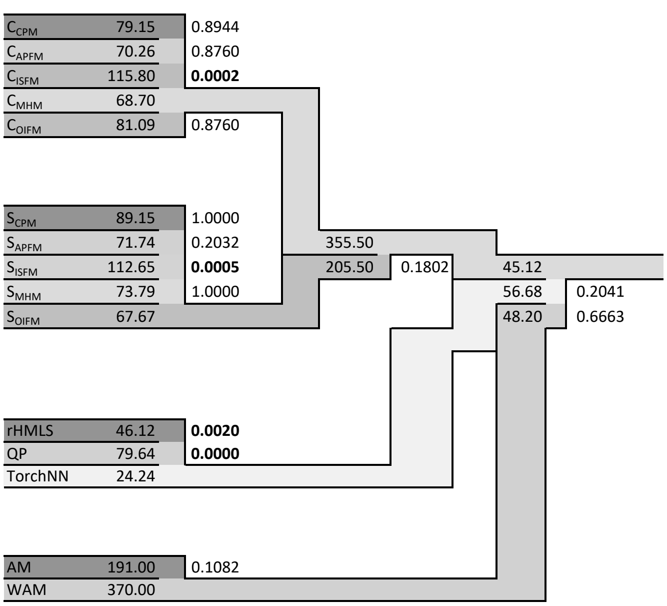

Fig. 1. Hierarchical statistical study comparing the different fuzzy measure construction methods and classical aggregation methods.

result is really interesting, as it shows that classical aggregation methods can still compete with more complex ones in certain scenarios. In this specific scenario, an ensemble of only 11 classifiers is being aggregated, due to the cost of learning supervised fuzzy measures. Moreover, we are in a challenging framework of imbalanced dataset with highly optimized classifiers learned with Bagging. For this reason, we expect other frameworks to show the goodness of fuzzy measure-based aggregations in terms of performance. Likewise, one could expect better performance from supervised methods with respect to unsupervised ones. However, these methods can easily lead to overfitting and may not be suited to work with imbalanced datasets unless their costs functions are adapted to work with this problem, as in the case with the TorchNN.

VI. CONCLUSIONS AND FUTURE WORK

In this work we have performed a study focused on different existing algorithms for the estimation and learning of a fuzzy measure, under the assumption that the fuzzy measure will be used together with a fuzzy integral to aggregate several classifiers of an ensemble.

By analyzing the literature, two different methodologies for learning the coefficients of a fuzzy measure have been found: supervised (data driven) and unsupervised (heuristic).

We have selected a representative number of algorithms which are encompassed under these two methodologies.

From the experimental study we have shown that in the current framework there are no great differences among the methods that are able to cope with the class imbalance problem. This result shows that there is still work to be done in the context of fuzzy measures-based aggregation in ensembles. We expect to found experimental frameworks better suited for these approaches: either using more classifiers or other ensemble learning models (Boosting or Gradient Boosting, for example). Supervised methods may need a separate validation set for learning, regularization strategies and adaptations to specific problems such as imbalanced ones.

REFERENCES

[1] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms, 2nd Edition*. Wiley-Interscience, 2014.

[2] G. Choquet, "Theory of capacities," *Ann. Inst. Fourier*, vol. 5, pp. 1953–1954, 1953.

[3] M. Sugeno, "Theory of fuzzy integrals and its applications," Ph.D. dissertation, Tokyo Institute of Technology, 1974.

[4] D. Štefka and M. Holeňa, "Dynamic classifier aggregation using interacion-sensitive fuzzy measures," *Fuzzy Sets and Systems*, vol. 270, pp. 25–52, 2015.

[5] G. J. Scott, K. C. Hagan, R. A. Marcum, J. A. Hurt, D. T. Anderson, and C. H. Davis, "Enhanced fusion of deep neural networks for classification of benchmark high-resolution image data sets," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, pp. 1451–1455, 2018.

[6] M. Grabisch, "A new algorithm for identifying fuzzy measures and its application to pattern recognition," in *Int. Joint Conf. of the 4th IEEE Int. Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Engineering Symposium*, march 1995, 1995, pp. 145–150.

[7] M. A. Islam, D. T. Anderson, A. J. Pinar, and T. C. Havens, "Data-driven compression and efficient learningof the choquet integral," *IEEE Transactions on Fuzzy Systems*, vol. 26, pp. 1908–1922, 2018.

[8] M. A. Islam, D. T. Anderson, A. Pinar, T. C. Havens, G. Scott, and J. M. Keller, "Enabling explainable fusion in deep learning with fuzzy integral neural networks," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2019.

[9] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*. Springer, 2018.

[10] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and 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*, vol. 17:2-3, pp. 255–287, 2011.

[11] R. Barandela, R. M. Valdovinos, and J. S. Sánchez, "New applications of ensembles of classifiers," *Pattern Analysis & Applications*, vol. 6, pp. 245–256, 2003.

[12] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets," *Information Sciences*, vol. 354, pp. 178 – 196, 2016.

[13] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, pp. 2044–2064, 2010.

[14] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Springer, 2007.

[15] G. Beliakov, H. Bustince, and A. Pradera, *A Practical Guide to Averaging Functions*, 2nd ed. Springer, 2015.

[16] T. Calvo, G. Mayor, and R. Mesiar, *Aggregation Operators. New Trends and Applications*. Physica-Verlag, 2002.

[17] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions*. Cambridge University Press, 2009.

[18] Y. Narukawa and V. Torra, "Fuzzy measure and probability distributions: distorted probabilities," *IEEE Transactions on Fuzzy Systems*, vol. 13, pp. 617–629, 2005.

[19] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[20] Q. Yang and X. Wu, "10 challenging problems in data mining research," *International Journal of Information Technology & Decision Making*, vol. 05, no. 04, pp. 597–604, 2006.

[21] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 42, pp. 463–484, 2012.

[22] J. A. Sanz, M. Galar, H. Bustince, and T. Belzunegui, "An evolutionary underbagging approach to tackle the survival prediction of trauma patients: A case study at the hospital of navarre," *IEEE Access*, vol. 7, pp. 76 009–76 021, 2019.

[23] G. Martínez-Mu$\tilde{b}$oz, D. Hernández-Lobato, and A. Suárez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 245–259, Feb 2009.

[24] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, "Strategies for learning in class imbalance problems," *Pattern Recognition*, vol. 36, no. 3, pp. 849–851, 2003.

[25] J. Murillo, S. Guillaume, E. Tapia, and P. Bulacio, "Revised hlms: A useful algorithm for fuzzy measure identifications," *Information Fusion*, vol. 14, pp. 532–540, 2013.

[26] M. Grabisch, H. Nguyen, and E. Walker, *Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference*. Springer Netherlands, 1995.

[27] D. T. Anderson, S. R. Price, and T. C. Havens, "Regularization-based learning of the choquet integral," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2014, pp. 2519–2526.

[28] M. Uriz, D. Paternain, H. Bustince, and M. Galar, "A first approach towards the usage of classifiers' performance to create fuzzy measures for ensembles of classifiers: a case study on highly imbalanced datasets," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2018, pp. 1–8.

[29] ——, "Unsupervised Fuzzy Measure Learning for Classifier Ensembles From Coalitions Performance," *IEEE ACCESS*, vol. 8, pp. 52 288–52 305.

[30] U. Agrawal, C. Wagner, J. M. Garibaldi, and D. Soria, "Fuzzy integral driven ensemble classification using a priori fuzzy measures," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–7.

[31] E. Hüllermeier, "Cho-k-nn: a method for combining interacting pieces of evidence in case-based learning," in *19th International Joint Conference on Artificial Intelligence IJCAI05*, 2005, pp. 3–8.

[32] D. Paternain, H. Bustince, M. Pagola, P. Sussner, A. Kolesárová, and R. Mesiar, "Capacities and overlap indexes with an application in fuzzy rule-based classification systems," *Fuzzy Sets and Systems*, vol. 305, pp. 70 – 94, 2016.

[33] J. R. Quinlan, *C4.5: Programs for Machine Learning*, 1st ed. San Mateo-California: Morgan Kaufmann Publishers, 1993.