HFER: Promoting Explainability in Fuzzy Systems via Hierarchical Fuzzy Exception Rules

José Ramón Trillo
DaSCI Andalusian Research Institute
University of Granada, Spain
Email: jrtrillo@ugr.es

Alberto Fernandez
DaSCI Andalusian Research Institute
University of Granada, Spain
Email: alberto@decsai.ugr.es

Francisco Herrera
DaSCI Andalusian Research Institute
University of Granada, Spain
Email: herrera@decsai.ugr.es

***Abstract*—When developing a Machine Learning model, the consideration of explainability as an additional design driver can improve its deployment into any application context. Given an audience, an eXplainable Artificial Intelligence system is one that produces details or reasons to make it's functioning clear or easy to understand. Among different paradigms that inherently support these capabilities, Fuzzy Rule Based Systems are a very accountable solution. The main issue when dealing with fuzzy systems is to select an appropriate granularity to represent (fuzzify) the input data. A low value may cause the generation of too generalist rules, causing a hinder on predictive performance, whereas a high value may lead to both overfitting and/or very complex solutions.**

To overcome this situation, we propose a novel hierarchical fuzzy classification system based on fuzzy exception rules. To do so, low granularity rules are first generated and their confidence is examined. For those cases in which the fuzzy confidence is below a quality threshold, new higher granularity rules are created to cover the "instances in conflict" for the general rule, which is still kept in the rule base. Experimental results show the achievement of a compact and interpretable final rule base while maintaining or improving the predictive performance in comparison with the baseline fuzzy rule based classification and hierarchical systems.

***Index Terms*—Explainable Artificial Intelligence (XAI), Fuzzy Rule Based Classification Systems, Hierarchical Models, Exception Rules**

I. Introduction

Nowadays, Artificial Intelligence (AI) is being the main piece of Industry 4.0 [1]. Given this increase in the interest of AI applications in many areas, it must be ensured that this type of system is applied in terms of what is known as Fairness, Accountability, Transparency, and Ethics (FATE) [2]. In fact, there exist many critical problems in which users or experts must avoid considering automated decision support systems to be blindly used. For this reason, we are witnessing a clear tendency towards embracing eXplainable AI (XAI) [3], which bets for the use of interpretable models that are straightforward to understand [4].

There are many alternatives to obtain Machine Learning solutions that can easily both interpretable and explainable. Among these, rule-based systems are a clear example, as they are actually close to a human-like cognition [5]. Fortunately, we can provide an additional degree of semantic to the use of rule-based systems by promoting a fuzzy representation in the rules' antecedent [6].

When facing any classification problem, Fuzzy Rule Based Classification Systems (FRBCS) [7] are the term used in this context. They are composed of a Knowledge Base (KB), that contains both a Rule Base (RB) and a DataBase (DB), and an inference engine. For the sake of generating an FRBCS, different learning algorithms are to be applied, being one of the widest studied one the grid-based Chi et al. approach [8], due to its simplicity, yet good performance even for difficult classification problems [9].

As for many different FRBCS learning algorithms, the Chi methodology requires the user to a priori select the granularity for the DB, i.e. the number of fuzzy labels in which every input attribute will be represented. This has a direct implication in both the coverage degree and the number of rules to be generated during the training stage [10]. On the one hand, setting a low value may cause the RB to be excessively general, not allowing to discriminate well among different class examples within a given cluster. On the other hand, if the user selects a high value, this may lead to a very complex RB that can degrade both predictive performance and interpretability. [11].

One straightforward solution to overcome the former issue is to consider the use of a fuzzy hierarchical approach [12]. Unlike alternative solutions on hierarchical structures for fuzzy modeling [13], [14], that build a model based on the cascade aggregation of "fuzzy variables", the key idea proposed in this research contribution is to generate an initial low granularity FRBCS. Then, those areas of the problem that might need a more specific description are identified, and higher granularity rules among these data clusters are created. This solution was initially developed in [15], where authors considered the computation of the fuzzy confidence of the rules as an indicator for removing the initial rule and generating new ones with higher granularity. The hitch with this approach was that since original low-granularity rules were removed, an excessive number of rules were created, even for those examples that were well-covered in the initial stage, thus leading to a lesser interpretable model.

In this work contribution, we aim to address the problem of

This work has been partially supported by the Spanish Ministry of Science and Technology under project TIN2017-89517-P, including European Regional Development Funds; and the Andalusian regional project P18-TP-5035.

achieving an FRBCS with a good trade-off between accuracy and interpretability from a similar yet different perspective. Specifically, our solution is based on the concept of exception rules [16], which are known to be specific rules that cover low dense clusters of a given class. In the case of fuzzy rules, this can be directly achieved by setting a high granularity for this type of rules, while maintaining the original low granularity ones. As such, we have named our novel approach as Hierarchical Fuzzy Exception Rules (HFER).

We must observe that this methodology implies that there will be areas of the problem with potential rules in conflict, which must be solved during the inference stage. To examine the most adequate solution, we have designed two different alternatives based on the degree of coverage of the exception rule, namely using an absolute or relative threshold. In this sense, we maintain the original hierarchical nature of the approach first proposed in [15] but providing two novel important capabilities. On the one hand, better interpretability in terms of a more compact RB. On the other hand, better explainability as the exceptional cases are explicitly considered.

We will analyze the good behavior of our novel HFER approach with respect to the baseline FRBCSs and the Hierarchical FRBCS (HFRBCS) [15] without rule selection. To do so, we have selected several benchmark datasets with a different number of instances, attributes, number of classes, and class distribution. This way, we may have a wider understanding of HFER under different case studies. As the metric of performance, we consider the standard accuracy and also the macro F1 as it allows us to obtain an average of the result of the predictive ability regardless of the class distribution.

To accomplish these goals, this work has been structured as follows. In section II, we introduce some mathematical fundamentals on FRBCS that will be the basis to define our proposal. In section III, the novel HFER algorithm is described in detail. In Section IV, we give the details on the experimental framework and the results to analyze the behavior of HFER in contrast to the baseline FRBCSs. Finally, in Section V some concluding remarks are drawn to conclude this work contribution.

II. Fundamentals on Fuzzy Rule Based Classification Systems

As we have already introduced, every FRBCS is composed of two main parts, namely the KB, which in turn includes a DB and RB, and the fuzzy reasoning method that performs the inference procedure to label new examples.

In this section, we will introduce these main components of FRBCS by considering a grid-based fuzzy rule learning algorithm [8]. Therefore, we will focus on its two main stages: (1) first, the DB definition for the fuzzy data representation (Section II-A); (2) second, the RB construction from the training input examples and the information of the initial DB (Section II-B).

A. Definition of the DB

To allow a fuzzy representation of the problem, we start by the definition of the characteristics of any dataset. To this end, the number of attributes m, and the number of instances n, are considered. Consequently, we can denote an instance as a vector $x_p = (x_{p1}, ..., x_{pm})$, $p = 1, 2, ..., n$, where x_{pi} is the attribute value of the i-th attribute in the p-th vector x_p. Besides, we denote the variable y_p is the label of an instance p. Hence, we define the class set C, we denote the number of classes with the variable h and a class as $c_g \in C$; $g = 1, ..., h$.

The DB consists of a set of K fuzzy variables that allows a smoother representation from the initial crisp-valued dataset. Having the former information, the standard structure of membership functions, which represents a fuzzy set, is calculated. For each fuzzy label of an attribute q we define a membership function, $\mu_{A_{qs}}$; $\forall q = 1, ..., m$; $s = 1, ..., K$. For this paper, the membership functions we will use are continuous, triangular, and all the membership functions of an attribute q are uniformly distributed in the universe of discourse of q. In addition, these functions verify the following characteristic:

$$\mu_{A_{qs}}(x_{pq}) \in [0, 1]; \ \forall x_{pq} \in x_p; \ \forall p = 1, ..., n$$

B. Generation of the RB

Rules can be of different types depending on the internal components. The widest case study is related to Type-I rules. This type of rules is defined as a vector of length m, where each value that belongs to the former vector is a fuzzy label. In addition, every vector has a consequent that includes the weight of the rule, and the class label that determines the output.

We assume that for each attribute we are given set fuzzy labels of cardinality K. In this way, we can denote a fuzzy term of an attribute q as A_{qs} where $s \in \{1, ..., K\}$ is the fuzzy index associated with a fuzzy label. For each instance, we can generate a fuzzy candidate rule. We can denote a fuzzy label as a_{pq} $\forall p = 1, ..., n$; $q = 1, ..., m$ and consequently, for every value in the dataset, x_{pq} we can define its associated fuzzy label, a_{pq}, as:

$$a_{pq} = argmax_{s \in \{1,...,K\}} \mu_{A_{qs}}(x_{pq}); \ x_{pq} \in x_p \\ p = 1, ..., n, \ q = 1, ..., m$$

When we have obtained the associated fuzzy label for each value of an instance p, we denote the rule associated with each instance, R_p, as the following vector:

$$R_p = (a_{pq}; \ q = 1, ..., m); \ \forall p = 1, ..., n$$

Applying the previous procedure for each of the previous instances and using a low number of fuzzy labels, we can obtain that several instances have the same rule associated. To unify them, we can define a set, RB_K, as the set of rules, R_p, not repeated whose cardinality of the set is denoted by r. In order to calculate the weight of a rule, RB_K,

RW_i; $\forall i = 1, ..., r$, we initially calculated the minimum t-norm, t_p, for each of the instances. We define t_p as:

$$t_p = \min(\mu_{a_{pq}};\ q = 1, ..., m);\ \forall p = 1, ..., n$$

t_p is an aggregation function that generates a single fuzzy membership value. This function is also included in the range $[0, 1]$. We use t_p to calculate the degree of association, denoted as T_{p,R_i}, of a instance p in a rule R_i; $\forall R_i \in RB_K$. So, we define the weight of a class $c_k \in C; k \in \{1, ..., h\}$ for a rule R_i as:

$$W_{i,c_k} = \sum_{p=1;\ c_k=y_p}^{n} T_{p,R_i}$$

Finally, we can define the class label associated to a rule, C^{RW_i}, as:

$$C^{RW_i} = \{c_t : W_{i,c_t} = max(W_{i,c_g});\ c_t, c_g \in C\}$$

$\forall R_i \in RB_K\ \exists C^{RW_i} \in C$. The weight of a rule, RW_i, is:

$$RW_i = \frac{W_{i,C^{RW_i}}}{\sum_{g=1}^{h} W_{i,c_g}};\ C^{RW_i} \in C \tag{1}$$

III. Hierarchical Fuzzy Exception Rules($HFER(K)_\chi$)

The $HFER(K)_\chi$ algorithm is an FRBCS that is intrinsically designed to set a trade-off between accuracy and interpretability. The variable K of the proposed algorithm indicates the number of linguistic labels to be used in the low granularity algorithm and the variable χ indicates the condition that an exception rule must have to be generated. For the sake of providing a clear description of this novel proposal, we decided to divide the algorithmic procedure into three parts. First, Section III-A introduces how low-granularity rules are generated. Next, Section III-B describes how new high-granularity exception rules are produced. Finally, Section III-C provides some comments on the new inference mechanism adapted to take into account the general and exception fuzzy rules.

A. First stage: baseline learning algorithm and generation of low-granularity general fuzzy rules

The preliminary phase is aimed to model the input dataset by using general rules, implying a low granularity representation for the fuzzy variables included in the DB. In order to obtain the classification rules, we will make use of the grid-based Chi algorithm with granularity K that was already introduced in Section II-B to calculate the RB, RB_K with a cardinality r.

By considering this approach, we will divide the original input space into several regions as illustrated in Fig. 1. We must take into account that, by using fuzzy rules, the borderline between each region is graded by means of the membership function values. In any case, each square that is depicted includes those examples whose membership degree is greater than or equal to 0.5 and therefore those used when creating the rule antecedent.

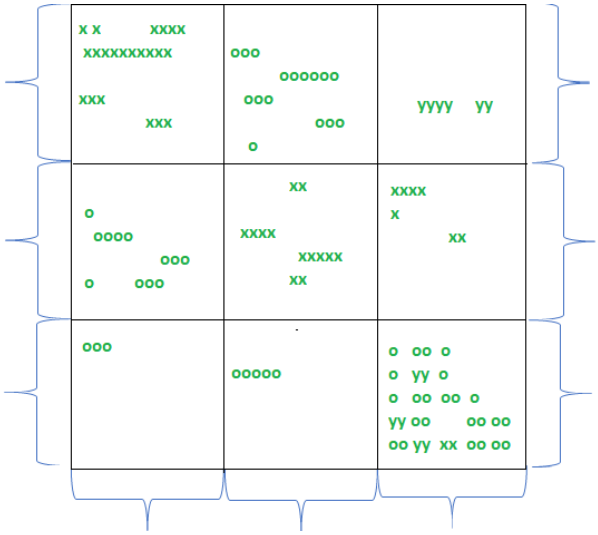

Fig. 1. Grid-based rule generation of the Chi algorithm with granularity 3 and an example of distribution of a dataset with 3 classes

B. Second stage: generation of high-granularity fuzzy exception rules

Exception rules are generated taking into account the confidence of the rules, measured as the rule weight (Eq. 1 in Section II-B).

Then, we must define a threshold, $\delta > 0$, which determines which RB_K rules should generate the exception rules. Therefore, we consider the set of rules that must generate exception rules, RB_{Ke}, as:

$$RB_{Ke} = \{R_i : RW_i < \delta;\ R_i \in RB_K; i = 1, ..., r\}$$

The set RB_{Ke} has a cardinality equal to b.

In the remainder, we provide a complete description of how fuzzy exception rules are generated:

1) Definition of the second-level DB: Initially, we calculate the set of examples that are covered by the exception rule, as pointed out below:

$$d_i = \{x_p : T_{p,R_i} \geq 0.5;\ p = 1, ..., n; R_i \in RB_{Ke}\};\ i = 1, ..., b \tag{2}$$

The value 0.5 is given by the minimum t-norm. This value is the minimum degree of membership that an instance can have with respect to the maximum degree of membership of a rule. Consequently, we denote x_{p_i} as a instance x_p belonging to the subset d_i. Besides, n_i is the number of instances of each subset. we denote a value of instance x_{p_i} as $x_{p_i q}$.

As an illustrative example, extending the one given in Fig. 1, we show in Fig. 2 that examples are selected as those from the contrary classes of the general rule obtained in the bottom-right box (marked as "y"). It must be also remarked that, since the original general rule will not be removed, examples marked as "o" will still be covered by it.

For the sake of computing novel higher granularity exception rules. We define for each attribute a set of fuzzy

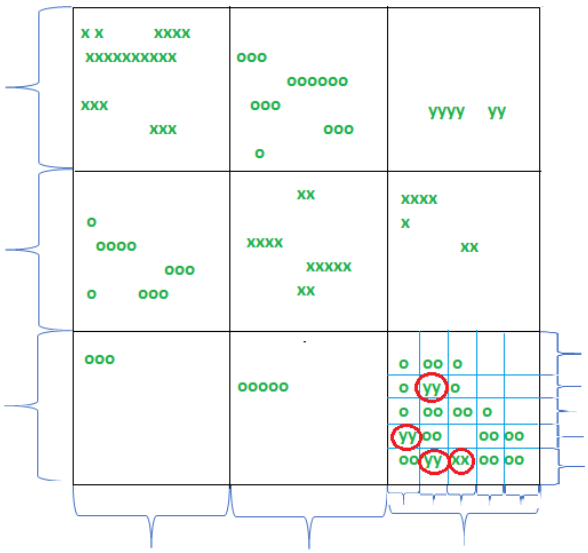

Fig. 2. Exception rule generation of the Chi algorithm with granularity 5 and an example of distribution of a dataset with 3 classes

cardinality labels $(2 * K - 1)$. In this way, we can denote a fuzzy term of an attribute q of a subset d_i as $\hat{A}_{qsi}$ where $s \in \{1, ..., (2 * K - 1)\}$ is the fuzzy index associated with a fuzzy label. We define for each fuzzy label of an attribute q of a subset d_i a membership function. We denote each belonging function as $\mu_{\hat{A}_{qsi}}$; $\forall q = 1, ..., m$; $s = 1, ..., (2 * K - 1)$; $i = 1, ..., b$. These functions have characteristics similar to the initial definition, they are continuous and triangular. However, unlike the initial definition, the membership functions of an attribute q of a subset d_i are uniformly distributed in the universe of speech of q of the subset d_i.

$$\mu_{\hat{A}_{qsi}}(x_{p_i q}) \in [0, 1]\,;\ \forall x_{p_i q} \in x_{p_i};\ p_i = 1, ..., n_i;\ i = 1, ..., b$$

2) Generation of the exception RB: For each instance, we generate a candidate rule. Thus, we can define the fuzzy label of an attribute q of a subset d_i, $\hat{a}_{p_i q}$, as:

$$\hat{a}_{p_i q} = argmax_{s \in \{1,...,(2*K-1)\}} \mu_{\hat{A}_{qsi}}(x_{p_i q});\ x_{p_i q} \in x_{p_i}$$
$$p_i = 1, ..., n_i,\ q = 1, ..., m$$

we denote for each instance a rule, R_{p_i}. Each rule is composed of the following vector:

$$R_{p_i} = (\hat{a}_{p_i q};\ q = 1, ..., m);\ \forall p_i = 1, ..., n_i;\ i = 1, ..., b$$

We denote the set of exception rules for each subset d_i as RB_{exp_i}. Every set of exception rules RB_{exp_i} is made up of the rules R_{p_i} without repetition. Thus, for any RB_{exp_i} there are no two equal rules. We represent the cardinality of each set of RB_{exp_i} rules as r_i.

To calculate the degree of association of an instance p_i for a rule R_{l_i}, $T_{p_i,R_{l_i}} \forall l_i = 1, ..., r_i$; $i = 1, ..., b$, we need to calculate the t-norm. The chosen t-norm is called the minimum t-norm, t_{p_i}, $\forall p_i = 1, ..., n_i; i = 1, ..., b$, and is defined for each instance of d_i as:

$$t_{p_i} = min(\mu_{\hat{a}_{p_i q}};\ q = 1, ..., m)$$

With these definitions we can carry out the process of calculating a rule R_{l_i}, RW_{l_i}, as follows:

We define the label of an instance p_i as y_{p_i} and the weight of a class, $c_k \in C; k \in \{1, ..., h\}$, for a rule, $R_{l_i} \in RB_{exp_{p_i}}$ as:

$$W_{l_i,c_k} = \sum_{p_i=1;\ c_k=y_{p_i}}^{n_i} T_{p_i,R_{l_i}}$$

Finally, the label of the associated class, $C^{RW_{l_i}}$, is:

$$C^{RW_{l_i}} = \{c_o : W_{l_i,c_o} = max(W_{l_i,c_g}); c_g, c_o \in C\}$$

$\forall R_{l_i} \in RB_{exp_i}$; $\exists C^{RW_{l_i}} \in C$. We explain the weight of the rule,RW_{l_i}, as:

$$RW_{l_i} = \frac{W_{l_i,C^{RW_{l_i}}}}{\sum_{g=1}^{h} W_{l_i,c_g}};\ C^{RW_{l_i}} \in C$$

C. Novel two-step inference mechanism

Two different schemes are proposed for inference, denoted as Ω and Λ. Both are included in the well-known FRM of the winning rule, but each uses the exception rules in a different way. Initially, Ω uses the generated exception rules that have a weight greater than or equal to its general rule. However, in the second version, version Λ, which is proposed, the generated rules must have a weight greater than a value β.

1) Ω version: We define the set Ω as the set of exception rules for subsets d_i, the subset of instances that are covered by the rule R_i; $R_i \in B_{Ke}$ (Eq. 2) , that have a different class label and a weight greater than or equal to its general rule.

$$\Omega = \{R_{l_i} : RW_{l_i} \geq RW_i \wedge C^{RW_{l_i}} \neq C^{RW_i};$$
$$R_i \in RB_{Ke};\ R_{l_i} \in RB_{exp_i};\ l_i = 1, ..., r_i;\ i = 1, ..., b\}$$

We denote the cardinality of the set Ω as $\hat{r}$. The inference of the instance u is done using the winning rule that prioritizes the exception rules. If the u instance belongs to several exception rules, then we use the winning rule among those exception rules:

$$y_u = \{c_J : (T_{u,R_J} * RW_J) = max\{T_{u,R_z} * RW_z\};$$
$$R_J \in \Omega;\ z = 1, ..., \hat{r}\} \quad (3)$$

If the instance u does not belong to any exception rule, then we use the winning rule with the base rules:

$$y_u = \{c_J : (T_{u,R_J} * RW_J) = max\{T_{u,R_z} * RW_z\};$$
$$R_J \in RB_K;\ z = 1, ..., r\} \quad (4)$$

2) Λ version: For the second version of the inference to take into account the 2 levels, we replace RW by the value of a β variable. The value of β chosen must be high because, in this way, we can guarantee the quality of the exception rules generated using their weight. We describe the set as:

$$\Lambda = \{R_{l_i} : RW_{l_i} \geq \beta \wedge C^{RW_{l_i}} \neq C^{RW_i};$$
$$R_i \in RB_{Ke};\ R_{l_i} \in RB_{exp_i};\ l_i = 1, ..., r_i;\ i = 1, ..., b\}$$

We denote the cardinality of the set Λ as $\ddot{r}$. The inference of the u instance belongs to several exception rules so we use the winning rule among those rules:

$$y_u = \{c_J : (T_{u,R_J} * RW_J) = max\{T_{u,R_z} * RW_z\}; \\ R_J \in \Lambda;\ z = 1, ..., \ddot{r}\} \quad (5)$$

Unfortunately, if the instance u does not belong to any exception rule, then we use the winning rule with the base rules (4).

IV. Experimental study

This section includes the experimental framework and the comparative study for determining the goodness of the proposed $HFER(K)_\chi$ algorithm. For this purpose, the content is organized as follows. First, we provide details on the selected benchmark datasets and the metrics of performance to evaluate the different versions (IV-A). The second part of this section includes the experimental results from which we may carry out our analysis and provide several lessons learned (Section IV-B).

A. Benchmark data problems and performance metrics

for the sake of using an adequate experimental framework, 15 datasets chosen for the evaluation are considered to follow different criteria in terms of number of examples, attributes or class distribution, as reported in Table I.

TABLE I
MAIN CHARACTERISTICS OF THE DATASETS USED IN THE EXPERIMENTAL STUDY. FROM LEFT TO RIGHT, NAME OF THE DATASET, NUMBER OF CLASSES, NUMBER OF EXAMPLES, NUMBER AND TYPE OF INPUT ATTRIBUTES, AND CLASS DISTRIBUTION ARE SHOWN

Name	#Class	#Examples	(R/I/N)	Class %
Iris	3	150	(4/0/0)	33.33%,33.33%,33.33%
Tae	3	151	(0/5/0)	32.45%,33.11%,34.44%
Hayes	3	132	(0/4/0)	38.64%,38.64%,22.73%
Seeds	3	210	(7/0/0)	33.33%,33.33%,33.33%
P.Indian	2	768	(2/6/0)	35.03%,64.97%
Newthy.	3	215	(4/1/0)	69.77%,16.28%,13.95%
Append.	2	106	(7/0/0)	80.19%,19.81%
Austral.	2	690	(3/11/0)	55.50%,44.50%
Bupa	2	345	(1/5/0)	42.03%,57.97%
Ecoli	8	336	(7/0/0)	42.56%,22.92%,0.6%, 0.6%,10.47%,5.95%, 1.49%,15.48%
Glass	6	214	(9/0/0)	32.71%,35.51%, 7.94%,6.07%, 4.21%,13.55%
Bank	2	1372	(4/0/0)	55.54%,44.46%
Wiscon.	2	683	(0/9/0)	65.01%,34.99%
Yeast	10	1484	(8/0/0)	16.44%,28.91%,31.20%, 2.96%,3.44%, 10.98%,2.36%, 2.02%,1.34%,0.34%
Wine	3	178	(13/0/0)	33.17%,39.89%,26.97%

In order to provide well-founded conclusions, we carry out a special validation procedure. In particular, we have partitioned the dataset using 5-folds distribution optimally balanced stratified cross-validation (DOB-SCV) [17]. This provides several advantages. First, partitions are carried out in a stratified way, which allows having the same amount of data of a class among the partitions to be created. Second, DOB-SCV distributes close-by instances among test folds, so that we avoid a possible dataset shift between training and test. Finally, the latter condition ensures that the exceptional data clusters modeled with our methodology will be also represented in test [18].

To compute the quality of the predictive ability of each FRBCS, we will use both accuracy and macro F1 metrics. The first measure is a global measure that we will use to know how our algorithm works in general. The second measure lets us know if our algorithm correctly recognizes the class regardless of the number of data the dataset has. Both metrics are computed from the standard confusion matrix, which includes the following values:

- True Positive (TP): number of correct predictions of the positive class
- True Negative (TN): number of correct predictions of the negative class
- False Positive (FP): number of incorrect predictions of the negative class, i.e. wrongly estimated as positives.
- False Negative (FN): number of incorrect predictions of the positive class, i.e. wrongly estimated as negatives.

Consequently, accuracy is obtained as the fraction between the sum of TP and TN, and the total number of instances in the dataset:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

Macro F1 is the arithmetic mean of the F1 of each class, being the individual F1 value computed as pointed out below:

$$f_{1_c} = \frac{2 * TP}{2 * TP + FP + FN};\ \forall c = 1, ..., h$$

We remember that the number of different classes in the dataset is defined by h. This equation applies to each class of the dataset. Then, the arithmetic mean is applied and macro F1 is obtained:

$$macro\ F_1 = \frac{\sum_{p=1}^{h} f_{1_p}}{h}$$

B. Experimentation and results

This part of the study is divided into two further parts. Initially, we will determine the best parameters' configuration of $HFER(K)_\chi$, specifically regarding the two versions of the inference mechanism (Section IV-B1). Then, we will carry out a thorough comparison versus the baseline FRBCS and HFRBCS (Section IV-B2).

1) Analysis of the inference mechanism of $HFER(K)_\chi$: Our algorithm has three parameters that can be chosen. The parameters are as follows:

- K: It is a natural number that represents the K parameter of the low-granularity algorithm Chi.
- χ: This categorical parameter refers to the inference scheme that can be applied to the problem, Ω and Λ.

- δ: This parameter is the threshold that the weight of the rule must exceed to avoid generating exception rules. In this work contribution, the value assigned to the variable is $\delta = 0.8$ because we consider that the rules have a weight less than δ should generate exception rules.
- β: This parameter is the threshold, in version Λ, that must exceed the weight of an exception rule in order to add the rule to the set Λ. In this contribution, we have chosen $\beta = 0.9$ because if the rule has a weight greater than or equal to β we can the quality of the rule.

Specifically, in this part of the study we will evaluate two different versions of our $HFER(K)_\chi$ proposal by considering:

- $HFER(3)_\Omega$: This version has the base algorithm has a granularity equal to 3 and the generated exception rules must have a greater weight than its general rule.
- $HFER(3)_\Lambda$: This version, like the Ω version, has the base algorithm that has a granularity equal to 3. However, exception rules must have a weight greater than or equal to the value β

Table II compares the two versions of $HFER(3)_\chi$. Subsequently, the best-performed version is compared with the other three algorithms.

The keys of the Table II is as follows:

- ACC: It is the average of the accuracy of all test datasets.
- macro F1: It is the average of the macro F1 of all test datasets.
- Rules: It is the average of the generated rules of all test datasets.
- Wins/Ties/Loses (W/T/L): It is the number of datasets that the version wins, ties, or loses. For Table II, it is calculated for accuracy and macro F1 because in the two measures mentioned they obtain the same number of datasets that win, tie, or lose the algorithms.

TABLE II
COMPARATIVE TABLE OF THE PROPOSED VERSIONS OF THE $HFER(K)_\chi$ ALGORITHM. THE VALUES SHOWN ARE AVERAGE VALUES

	$HFER(3)_\Lambda$	$HFER(3)_\Omega$
ACC	**0.7699**	0.7662
macro F1	**0.7174**	0.7163
Rules	143.7	158.6
W/T/L	5/9/1	1/9/5

The comparison between the different versions of $HFER(K)_\chi$ indicates that the absolute difference in predictive performance between the two versions is not significant. However, in terms of interpretation, there is a significant difference because $HFER(3)_\Lambda$ generates a fewer number of exception rules that $HFER(3)_\Omega$. Consequently, $HFER(3)_\Lambda$ is more efficient and more interpretable than the first version.

2) Comparative study versus baseline FRBCS and HFRBCS: In this part, we will compare our $HFER(3)_\Lambda$ version versus standard fuzzy classifiers. Specifically, we have selected the original HFRBCS with 3-5 granularity levels without rule selection, and the original Chi baseline algorithm with both 3 and 5 fuzzy labels per variable.

Experimental results are shown in both Tables III and IV, for accuracy and macro F1 metrics respectively. In addition to these performance values, we also show the number of total rules generated for each model (Rules). In the case of HFRBCS, we specifically point out the rules considered with low granularity (G_3) and high granularity (G_5). Finally, in the case of $HFER(K)_\chi$ we show the number of rules that belong to RB_K (B) and also the number of exception rules generated (E).

The proposed model, $HFER(3)_\Lambda$, obtains the best average value in both the macro F1 and accuracy measures. Focusing on the individual performance, we observe that our new proposed algorithm outperforms the baseline FRBCS in more than half of the datasets, and it is equally good in 5 problems. The comparison between the proposed algorithm and HFRBCS (Table V) it can be seen how the proposed algorithm generates fewer total rules and improves, on average, the precision and the F1 macro of the data set. Consequently, the proposed algorithm could be considered more interpretable than HFRBCS.

Focusing on the main lessons learned achieved from this study, we must focus on the following ones:

- **Number of attributes**: In the case of datasets with a larger number of attributes, the baseline grid-based learning algorithm causes an increase in the number of rules. Nevertheless, the recognition ability of $HFER(K)_\chi$ is maintained.
- **Class imbalance**: The difference between the amount of data of the classes in some of the datasets, together with the generality of the Chi algorithm with granularity 3, may cause that rules that represent minority classes have a low weight below than the threshold δ. This issue implies that, when generating the exception rules, we should focus on a higher granularity for adequate coverage with high confidence.
- **Rule weights from base algorithm**: If the base algorithm has a large number of rules whose weight (confidence) is less than δ then the number of exception rules will be higher. This is directly related to the intrinsic complexity of the problem, and therefore has a clear implication to the fairness of the model to be extracted from the raw data.

V. CONCLUSION

In this work contribution, we have proposed $HFER(K)_\chi$, a novel hierarchical FRBCS by promoting both the predictive ability and the interpretability and explainability of the system. To do so, we have made use of a hierarchical approach made of general (low granularity) fuzzy rules for a wide coverage of the input space, but also considering exception rules (high granularity) to allow representing rare cases of the problem. An experimental study, using datasets with different characteristics, has shown the goodness of $HFER(K)_\chi$ in contrast

TABLE III

RESULTS FOR ACCURACY OF THE CHI ALGORITHM WITH GRANULARITY 3, CHI ALGORITHM WITH GRANULARITY 5, HFRBCS, AND $HFER(3)_\Lambda$ IN THE TEST PARTITIONS. HIGHLIGHTED VALUES CORRESPOND TO THE BEST CASE FOR EACH DATASET. ACCURACY IS GIVEN IN RATIO INSTEAD OF THE PERCENTAGE.

Dataset Name	**CHI 3**		**CHI 5**		**HFRBCS**				$HFER(3.5)_\Lambda$			
	Acc	RULES	Acc	RULES	Acc	RULES	G_3	G_5	Acc	RULES	B	E
Iris	0.9133	15.4	**0.94**	39.2	0.8867	56.8	10.8	46	0.9133	21.8	15.4	6.4
Tae	0.518	31.4	0.537	61.4	**0.6265**	89.0	7.8	81.2	0.6099	59.4	31.4	28
Hayes	0.6435	44.6	0.6071	73.4	0.6632	63.2	13.8	49.4	**0.7003**	54.4	44.6	9.8
Newthyroid	0.8329	21.2	**0.9023**	45.2	0.8651	38.2	15.8	22.4	0.8465	37.4	21.2	16.2
Pima Indians	**0.7344**	115	0.7316	403	0.7083	384.8	30.6	354.2	**0.7344**	240.8	115	125.8
Seeds	0.8322	53.4	**0.8905**	107.4	0.881	95.0	36.4	58.6	0.8429	61.4	53.4	23.4
Appendiciti	0.8589	30.8	**0.8679**	59	0.8584	35.4	24.4	11	0.8589	35	30.8	4.2
Australian	**0.8247**	312.6	0.713	460.6	0.7798	356.4	266.2	90.2	**0.8247**	345.8	312.6	33.2
Bupa	0.5913	47.2	**0.6174**	119.8	0.5014	286 .0	8.6	277.2	0.6087	142.6	47.2	95.4
Ecoli	0.736	70.0	**0.8098**	160.4	0.6729	217.6	32.4	185.2	0.7298	114.8	70.0	44.8
Glass	0.604	42.4	0.5851	79.4	0.4771	161.6	15.6	146	**0.6275**	94.4	42.4	52.0
Bank	0.9288	30.8	0.9592	75.4	**0.9832**	379.4	17.2	362.2	0.9526	101.8	30.8	71.0
Wisconsin	**0.9224**	208.8	0.69	267	0.921	209.2	198	11.2	**0.9224**	212.8	208.8	4.0
Yeast	0.4919	105	**0.556**	228.6	0.4368	1097.0	10.2	1086.8	0.5277	495.2	105	390.2
Wine	**0.8485**	118.8	0.7023	141.8	**0.8485**	121.4	108.6	12.8	**0.8485**	121.4	118.8	2.6
Avg.	0.7521	83.16	0.7406	154.77	0.7407	239.4	53.09	186.3	**0.7699**	143.7	83.2	60.5

TABLE IV

RESULTS FOR MACRO F1 OF THE CHI ALGORITHM WITH GRANULARITY 3. CHI WITH GRANULARITY 5. HFRBCS. AND $HFER(3)_\Lambda$ IN THE TEST PARTITIONS. HIGHLIGHTED VALUES CORRESPOND TO THE BEST CASE FOR EACH DATASET. METRICS ARE GIVEN IN RATIO INSTEAD OF THE PERCENTAGE.

Dataset Name	**CHI 3**		**CHI 5**		**HFRBCS**				$HFER(3.5)_\Lambda$			
	F1	RULES	F1	RULES	F1	RULES	G_3	G_5	F1	RULES	B	E
Iris	0.9113	15.4	**0.9393**	39.2	0.8854	56.8	10.8	46	0.9113	21.8	15.4	6.4
Tae	0.5108	31.4	0.5312	61.4	**0.6204**	89	7.8	81.2	0.6063	59.4	31.4	28
Hayes	0.5736	44.6	0.4512	73.4	0.5623	63.2	13.8	49.4	**0.6237**	54.4	44.6	9.8
Newthyroid	0.709	21.2	**0.8357**	45.2	0.7713	38.2	15.8	22.4	0.7299	37.4	21.2	16.2
Pima Indians	0.6417	115	**0.6787**	403	0.5942	384.8	30.6	354.2	0.6417	240.8	115	125.8
Seeds	0.8337	53.4	**0.8918**	107.4	0.8825	95	36.4	58.6	0.8435	61.4	53.4	23.4
Appendicitis	0.7238	30.8	**0.7305**	59	0.7237	35.4	24.4	11	0.7238	35	30.8	4.2
Australian	0.8217	312.6	0.6926	460.6	**0.837**	356.4	266.2	90.2	0.8217	345.8	312.6	33.2
Bupa	0.4691	47.2	**0.5985**	119.8	0.4755	286	8.6	277.2	0.5347	142.6	47.2	95.4
Ecoli	0.6368	70	**0.7205**	160.4	0.519	217.6	32.4	185.2	0.6321	114.8	70	44.8
Glass	0.4588	42.4	0.4714	79.4	0.3433	161.6	15.6	146	**0.5041**	94.4	42.4	52
Bank	0.9263	30.8	0.9588	75.4	**0.9831**	379.4	17.2	362.2	0.952	101.8	30.8	71
wisconsin	**0.9118**	208.8	0.5319	267	0.9097	209.2	198	11.2	**0.9118**	212.8	208.8	4
Yeast	0.4239	105	**0.5158**	228.6	0.3542	1097	10.2	1086.8	0.4698	495.2	105	390.2
Wine	**0.8539**	118.8	0.7101	141.8	**0.8539**	121.4	108.6	12.8	**0.8539**	121.4	118.8	2.6
Avg.	0.6937	83.16	0.6839	154.77	0.6877	239.4	53.09	186.3	**0.7174**	143.7	83.2	60.5

TABLE V

COMPARATIVE TABLE BETWEEN $HFER(K)_\chi$ ALGORITHM AND HFRBCS ALGORITHM WITHOUT RULE SELECTION. THE VALUES SHOWN ARE AVERAGE VALUES

	$HFER(3)_\Lambda$	HFRBCS
ACC	**0.7699**	0.7407
macro F1	**0.7174**	0.6877
B or G_3	83.2	**53.16**
E or G_5	**60.5**	186.3
Total rules	**143.7**	239.4

to a standard FRBCS and an HFRBCS. It has to be noted that the improvement in the recognition of the different classes of the problem, measured by means of the F1 metric, has been achieved using a compact rule system. Specifically, the RB size of $HFER(K)_\chi$ is lower than using a higher granularity by default or applying a hierarchical approach. To sum up, we have taken a step forward in the construction of an interpretable model that prioritizes the exception rules before the base rules, allowing the recognition of minority cases and thus avoiding bias during the learning stage. This way, we stress the need for achieving accountable models for nowadays society. As future work, we must focus on two issues. First, to keep on minimizing the size of the RB to further boost the interpretability and usability of the FRBCS. To achieve this proposal, we will use the evolutionary algorithms to modify the parameters of the partitions, adjust the limitations, and also, to have a selection of rules in problems where the number increases. Second, examine in detail the areas that may require a higher level of granularity to accurately represent them.

REFERENCES

[1] M. Skilton and F. Hovsepian, *The 4th industrial revolution: Responding to the impact of artificial intelligence on business*. Springer, 2017.

[2] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82 – 115, 2020.

[3] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, 2017.

[4] X. Zhu and I. Davidson, *Knowledge discovery and data mining: challenges and realities*. Information Science Reference Hershey, PA, 2007.

[5] V. Aleven, "Rule-based cognitive modeling for intelligent tutoring systems," in *Advances in intelligent tutoring systems*. Springer, 2010, pp. 33–62.

[6] A. Fernandez, F. Herrera, O. Cordon, M. J. del Jesus, and F. Marcelloni, "Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to?" *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 69–81, 2019.

[7] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and modeling with linguistic information granules: Advanced approaches to linguistic data mining*. Berlin, Germany: Springer-Verlag, 2004.

[8] Z. Chi, H. Yan, and T. Pham, *Fuzzy algorithms: with applications to image processing and pattern recognition*. World Scientific, 1996, vol. 10.

[9] A. Fernández, S. García, M. J. del Jesús, and F. Herrera, "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets," *Fuzzy Sets and Systems*, vol. 159, no. 18, pp. 2378–2398, 2008.

[10] O. Cordón, F. Herrera, L. Magdalena, and P. Villar, "A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base," *Information Sciences*, vol. 136, no. 1-4, pp. 85–107, 2001.

[11] J. M. Alonso, C. Castiello, and C. Mencar, "Interpretability of fuzzy systems: Current research trends and prospects," in *Handbook of Computational Intelligence*, ser. Springer Handbooks, J. Kacprzyk and W. Pedrycz, Eds. Springer, 2015, pp. 219–237.

[12] J. Kerr-Wilson and W. Pedrycz, "Generating a hierarchical fuzzy rule-based model," *Fuzzy Sets and Systems*, vol. 381, pp. 124–139, 2020.

[13] R. R. Yager, "On the construction of hierarchical fuzzy systems models," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 1, pp. 55–66, 1998.

[14] L. Magdalena, "Semantic interpretability in hierarchical fuzzy systems: Creating semantically decouplable hierarchies," *Information Sciences*, vol. 496, pp. 109–123, 2019.

[15] A. Fernandez, M. J. del Jesus, and F. Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets," *International Journal of Approximate Reasoning*, vol. 50, pp. 561–577, 2009.

[16] S. Ventura and J. M. Luna, *Supervised Descriptive Pattern Mining*. Springer, 01 2018.

[17] J. G. Moreno-Torres, J. A. Sáez, and F. Herrera, "Study on the impact of partition-induced dataset shift on k-fold cross-validation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1304–1312, 2012.

[18] V. López, A. Fernández, and F. Herrera, "On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed," *Information Sciences*, vol. 257, pp. 1–13, 2014.