Building Explanations for Fuzzy Decision Trees with the ExpliClas Software

Jose M. Alonso*, Pietro Ducange†, Riccardo Pecori‡, Raúl Vilas*

* Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain
josemaria.alonso.moral@usc.es
raul.vilas@rai.usc.es

† Dep. of Information Engineering, University of Pisa, Pisa, Italy
pietro.ducange@unipi.it

‡ Dep. of Engineering, University of Sannio, Benevento, Italy
rpecori@unisannio.it

***Abstract*—Fairness, Accountability, Transparency and Explainability have become strong requirements in most practical applications of Artificial Intelligence (AI). Fuzzy sets and systems are recognized world-wide because of their outstanding contribution to model AI systems with a good interpretability-accuracy trade-off. Accordingly, fuzzy sets and systems are at the core of the so-called Explainable AI. ExpliClas is a software as a service which paves the way for interpretable and self-explainable intelligent systems. Namely, this software provides users with both graphical visualizations and textual explanations associated with intelligent classifiers automatically learned from data. This paper presents the new functionality of ExpliClas regarding the generation, evaluation and explanation of fuzzy decision trees along with fuzzy inference-grams. This new functionality is validated with two well-known classification datasets (i.e., Wine and Pima), but also with a real-world beer-style classifier.**

***Index Terms*—Fuzzy Systems Software, Open Source Software, Software as a Service, Fuzzy Rule-based Systems, Explainable AI**

I. Introduction

The European Commission (EC) [1] has deemed Artificial Intelligence (AI) as the most strategic technology of XXI century. In addition, the EC states that AI must be carefully developed and applied in agreement with the European values and fundamental rights as well as ethical principles such as accountability and transparency. Accordingly, the EC highlights the need to increase resources and to speed up research on Explainable AI (XAI).

Nowadays, learning interpretable AI-based systems from data is one of the major challenges of XAI [2]. Actually, given the world-wide popularity of deep learning methods, more and more black-box models are learned from data and they are able to solve many varied and complex problems. However, all these models lack explanation ability when interacting with humans. This is the reason why many researchers focus on how to open black-box models [3], while others opt for designing and using interpretable models instead [4].

It is worthy to note that researchers in the field of fuzzy sets and systems have addressed the problem of carefully designing interpretable fuzzy systems for years [5], [6]. In addition, a recent survey [7] shows how about 30% of publications in XAI prior to 2018 came from authors well recognized in the field of fuzzy logic. This is mainly due to the fact that interpretability is deeply rooted in the fundamentals of fuzzy sets and systems since the seminal ideas of Prof. Zadeh [8]. Moreover, since interpretability and accuracy are usually conflicting goals by nature, many researchers have applied multi-objective evolutionary algorithms with the aim of designing fuzzy systems with a good interpretability-accuracy balance [9].

Other social and cognitive aspects (e.g., comprehensibility or human-machine interaction) pay also an important role in XAI [10]. Effective explanations are expected to be multi-modal (i.e., a mixture of texts with graphs and/or sounds ready to be conveyed to users through alternative channels), but also natural (i.e., similar to those provided by humans) and easy to understand, no matter the user's background.

Unfortunately, there is a lack of software for generation and evaluation of XAI explanations. Even though there is general purpose data mining software (e.g., Weka [11], R packages [12] or Python libraries [13]), only a few programs (e.g., LIME [14] or LORE [15]) are released as open source software for XAI. In the case of fuzzy systems software [16], there are tools for designing interpretable fuzzy models (e.g., FisPro [17] or GUAJE [18]).

ExpliClas is a web service for generation and evaluation of multi-modal XAI explanations related to Weka classifiers and it was introduced in [19]. ExpliClas provides users with global and local explanations. The global view of the classifier refers to quality indicators (e.g., classification ratio) as well as to structural properties (e.g., number of features, tree size, rule length, etc.). In addition, the local view pays attention to the classification of single data instances. The first version of ExpliClas included four algorithms implemented in Weka [11]: three decision tree algorithms [20] (J48, REPTree, and RandomTree) and one fuzzy algorithm (FURIA [21]).

In this paper, we present the new version of ExpliClas which includes the implementation of the Fuzzy Hoeffding Decision Tree (FHDT) [22], which is a fuzzy extension of the original Hoeffding Decision Tree (HDT) [23]. In addition, we have enhanced the graphical user interface with the aim of becoming even more user friendly, by also redesigning the panel for visualization of fuzzy rules. Moreover, we have added a

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

new software module for the generation and visualization of fuzzy inference-grams (FINGRAMs) [24]. This is aimed at visualizing rule interaction at inference level. To do so, fuzzy rule-based systems (FRBSs) are displayed in the form of social networks, where nodes correspond to rules and edge values are rule co-firing degrees.

The rest of the manuscript is organized as follows. Section II introduces some preliminary concepts sketching briefly the FHDT algorithm and revisiting the concept of FINGRAM. Section III presents implementation details. Section IV goes with the experimental evaluation of the proposal and provides illustrative examples. Finally, Section V concludes the paper.

II. Preliminaries

A. Fuzzy Hoeffding Decision Trees

A Fuzzy Hoeffding Decision Tree (FHDT) is a multi-way decision tree, i.e., unlike binary trees, more than two decision branches are allowed at each node. Indeed, its structure resembles, as all decision trees, a directed acyclic graph, with internal nodes representing a test on a feature, branches denoting the outcome of the test, and terminal nodes (leaves) containing instances belonging to one or more class labels. Namely, each splitting node has T_f exiting branches, equal to the number of fuzzy sets (i.e., linguistic terms) for each input variable X_f. In a general scenario, class $C_k \in \Gamma$ of each leaf L_h is associated with a weight $w_{h,k}$, which determines the strength of class C_k in the leaf node. In FHDT, the weight is the Fuzzy Cardinality (FC), which is calculated as follows:

$$w_{h,k} = \sum_{i=1}^{N_{h,k}} \mu_{B_h}(x_{h,k}), \tag{1}$$

where $N_{h,k}$ is the number of instances in L_h and $\mu_{B_h}(x_{h,k})$ is the membership degree of each instance x_h to decision fuzzy set B_h on the branch leading to L_h.

The features, over which the tests are performed in each internal node, can repeat or not; moreover, they can be ranked and selected according to the Fuzzy Information Gain (FIG) as discussed in [25].

In classical decision trees, each node mirrors a single crisp set and each leaf owns only one class label; thus, a certain unlabeled instance $\hat{\mathbf{x}}$ activates only one path and is assigned to only one class. In the case of FHDT, each node is represented by a fuzzy set. As a consequence, $\hat{\mathbf{x}}$ can be assigned to various classes $C_k \in \Gamma$ by following different activation paths, made of nodes from the root to possibly many leaves with different strengths of activation, called *matching degrees*. Then, the chosen output class label is obtained by means of the *weighted vote* approach, by maximizing the total strength of vote. This vote is obtained by summing up all the association degrees of the instance with each class in the reached leaves. The association degree is computed as the product of the aforementioned weight $w_{h,k}$ per class and the above-mentioned matching degree.

As described in [22], the FHDT is built by means of an iterative learning procedure composed of two main steps:

- *The update of the statistics in the nodes and leaves*. Starting from the root, the learning procedure determines the leaves reached by the current data instance and its statistics for the calculation of the fuzzy information gain are updated. As mentioned previously, a data instance can activate more than one branch, given that it has different membership degrees for each fuzzy set in the fuzzy partition built over the feature in the considered node. While traversing the nodes in a path, the FCs of all classes in the leaf node are updated, according to the matching degree of the instance at that node.
- *The growth of the tree*. A set of conditions are checked to determine whether to split or not a current leaf (CL). First, the FCs per class are summed up to compute the total fuzzy number of instances in CL, denoted as *Total Fuzzy Cardinality* (TFC_{CL}). If this is lower than the Grace Period (GP), the splitting is not permitted. Otherwise, features are ranked according to the FIG, computed for each feature X_f. Then, the Fuzzy Hoeffding Bound for the current leaf (FHB_{CL}) is computed as follows:

$$FHB_{CL} = \sqrt{\frac{R^2 \ln(1/\delta)}{2TFC_{CL}}}, \tag{2}$$

where R is the log_2 of the number of classes contained in CL, and δ is the Split Confidence. If the difference between the fuzzy information gains of the first two attributes in the ranking is higher than FHB_{CL} or FHB_{CL} is lower than a Tie Threshold (TT), then the splitting of CL is allowed by employing the attribute with the highest FIG. Indeed, the splitting is only done if a further check is passed, i.e., the percentage of instances in each post-split branch is higher than the Minimum Fraction of Weight along two branches (MFW). For details on the calculation of FC and TFC go to [25].

It is worth noting that the incremental learning procedure depends on a set of parameters (GP, δ, TT, and MFW), which we have summarized above.

B. Fuzzy Inference-grams

A Fuzzy Inference-gram (FINGRAM) is a graphical representation of an FRBS at inference level and it was first introduced in [24] and later enhanced in [26]. The term FINGRAM was coined by analogy with the term scientogram previously defined by Vargas-Quesada and Moya-Anegón for visualizing the structure of science [27].

In practice, a fuzzy rule base (RB) can be seen as a population made up of a set of individual rules which are competing and collaborating among them with the aim of yielding good generality-specificity and interpretability-accuracy tradeoffs. Therefore, a FINGRAM can be depicted as a social network where nodes (individuals of the population, i.e., single fuzzy rules) interact at inference level. Moreover, the edges, which graphically connect nodes in the network, actually represent the rule co-firing degree (i.e., the degree of activation of two rules simultaneously fired by a given data instance) among the involved fuzzy rules.

The procedure for generation and visualization of FINGRAMs comprises three main steps:

- *Network Generation*. Given an FRBS containing N rules and a dataset, first we compute the adjacency matrix M of the graph to represent the network. M is an $N \times N$ matrix with $m_{ij} \in [0,1]$ being the rule co-firing degree of all pairs of rules (R_i and R_j) averaged for all data instances. We can choose among different co-firing metrics depending on the nature of the rules (e.g., classification, regression or association rules). For example, the most basic co-firing metric for fuzzy rule-based classifiers is adapted from the co-citation metric defined by Salton and Bergmark for scientograms [27]. It is computed as follows [26]:

$$m_{ij} = \begin{cases} \frac{|D_i \cap D_j|}{\sqrt{|D_i| \cdot |D_j|}} & , \quad i \neq j \\ 0 & , \quad i = j \end{cases} \tag{3}$$

  where $|D_i|$ counts the number of data instances which are covered by rule R_i (i.e., the rule firing degree of R_i for a given data instance is greater or equal than a given threshold), while $|D_i \cap D_j|$ counts the number of data instances which are covered simultaneously by both R_i and R_j.
- *Network Scaling*. Due to the nature of fuzzy rules, i.e., several rules are likely to simultaneously fire with different activation degrees for a given data instance, the graph associated to matrix M is likely to be dense and difficult to analyze. Fortunately, we can choose among several scaling algorithms (e.g., thresholding or Pathfinder) with the aim of filtering out some of the less relevant edges in the graph before printing and exploring the generated FINGRAM. In short, these algorithms consider different metrics of similarity, correlation or distance in order to prune a given graph. In particular, we recommend to use the Pathfinder algorithm [28], which is able to efficiently scale FINGRAMs while preserving the most important edges, and makes easier to visualize the underlying structure of interactions among rules.
- *Network Drawing*. Depending on the selected co-firing metric m_{ij}, the network may be drawn in the form of a directed or undirected graph. We can choose among several algorithms [27] for drawing pleasant graphs in accordance with aesthetic criteria (e.g., maximizing the use of the available space or minimizing the number of crossed edges). In particular, we recommend the use of the Kamada–Kawai algorithm [29] for drawing FINGRAMs of fuzzy rule-based classifiers.

FINGRAMs can be generated and visualized with the standalone FingramsGenerator software [30]. In addition, other tools (e.g., KEEL [31] or GUAJE [18]) include specific modules for dealing with FINGRAMs.

III. ExpliClas Software

The architecture of ExpliClas was introduced in [19] and comprises two main parts, namely the API and the Web Client:

- *The API REST*[1] offers the following services:
  - **Session** manages the access of users to the rest of services. Users do not need to sign up but they are automatically assigned a token for each new session.
  - **Dataset** manages all operations related to datasets. For the sake of simplicity, several datasets are preloaded and split into training and test sets. Anyway, users are allowed to upload other datasets. Notice that we adopt the Weka *arff* dataset format [11].
  - **Builder** is in charge of learning classification models from the datasets previously loaded. Five classification algorithms are already available (J48 [20], RandomTree [20], REPTree [20], FURIA [21] and FHDT [32]). All of them are implemented for Weka [11]. The ExpliClas Builder service acts as a wrapper of Weka and facilitates the generation of classifiers even by non-expert users. Of course, advanced users may modify the given parameters (see Fig. 1). For example, the field "fuzzySets" lets users to set the number of fuzzy sets for the fuzzy partition of each feature in the dataset. It is also possible to provide "Centroids" as a list of values associated to each fuzzy set (except for the first and last fuzzy sets in the partition which are anchored to zero and one, respectively). It is worth noting that ExpliClas validates the provided parameters and assists the user to set valid values if needed.

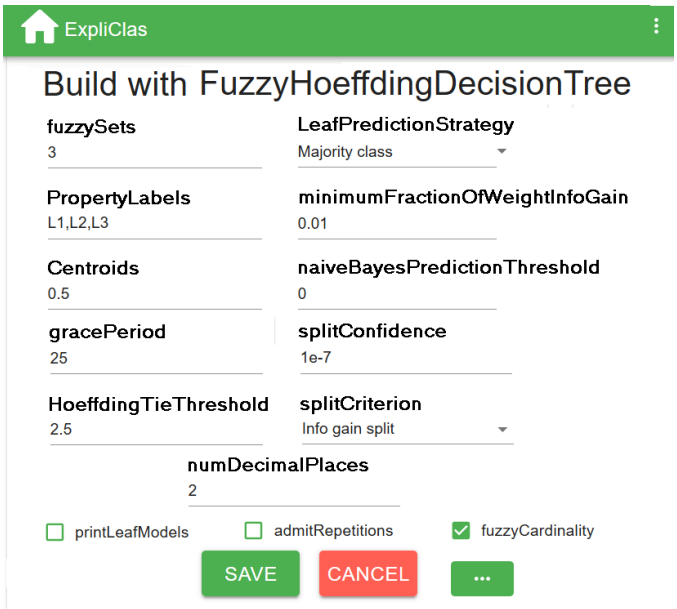

Fig. 1. ExpliClas screenshot for building FHDT

  - **Classifier** manages the inference process associated to the classifiers previously generated. Given a data instance, crisp trees are traversed from root to leaves with the aim to identify the output class. In the

[1]ExpliClas API: https://demos.citius.usc.es/ExpliClasAPI/

case of FURIA and FHDT, a fuzzy inference engine computes the output class in terms of the rules that are fired by the given data instance.
  - **Explainer** generates global and local multi-modal (i.e., graphical + textual) explanations. Users are provided with graphical representations associated with decision trees and fuzzy rules on top of the screen. Moreover, textual explanations are given at the bottom to facilitate the understanding of the related graphics. On the one hand, global explanations pay attention to the structure and quality, in terms of confusion matrix, of the model. On the other hand, local explanations pay attention to the classification of single data instances.
  - **Fingrams** deal with the graphical representation of fuzzy rule-based classifiers (i.e., FURIA and FHDT) at inference level. This service wraps the Fingrams-Generator [30] software to make transparent to users the generation, scaling and drawing of FINGRAMs. The generated *svg* files are visualized through the Web Client. The visual analysis of FINGRAMS facilitates the identification of the most relevant rules as well as of potential redundancies and/or inconsistencies that may be fixed in order to improve the interpretability-accuracy tradeoff of fuzzy classifiers, automatically learned from data.
- *The Web Client*[2] is actually a user-friendly dynamic dashboard for XAI. It permits users to load datasets, generate classifiers from data, and analyze the behavior of the generated classifiers. In addition, users can download log *txt* files as well as configuration *json* files with details about the Strong Fuzzy Partitions (SFPs) used for linguistic approximations of numerical intervals, decision trees, fuzzy rule-based classifiers, and so on.

ExpliClas assists the users along the whole pipeline from row data to explanations in natural language. It is worthy to note that crisp decision trees (J48, RandomTree and REPTree), likewise FURIA, manage local semantics. This means that each splitting condition in a node of a tree, or each fuzzy set in a FURIA rule, are generated with the focus only on accuracy, while disregarding the interpretability of the whole model. As a result, they lack linguistic interpretability. To overcome this drawback, ExpliClas creates a linguistic layer which is endowed with global semantics on top of these models. More precisely, for each feature in the given dataset, ExpliClas creates a SFP. By default, a SFP is made up of three linguistic terms (*Low*, *Medium*, and *High*) but it is editable by the user. SFPs were first introduced by Ruspini [33] and they satisfy all mathematical properties (e.g., coverage, distinguishability, etc.) required for designing interpretable fuzzy partitions [34]. They are formally defined as follows:

$$\forall x \in U : \sum_{L \in T} \mu_L(x) = 1 \tag{4}$$

[2]ExpliClas Web Client: https://demos.citius.usc.es/ExpliClas/

where L represents each of the T linguistic terms associated with a linguistic variable X with a fuzzy partition defined in the universe of discourse U; and $\mu_L(x)$ is the membership degree of value x in relation with the fuzzy set which characterizes L.

In practice, the goodness and expressiveness of textual explanations, associated with models with local semantics, rely on the goodness of the associated linguistic approximations. No matter if we consider a branch of a crisp tree or a FURIA rule, it can be translated into a conjunction of constraints $(A_1 \dots A_Z)$ that a given data instance should satisfy to be classified as belonging to class C_k (i.e., the leaf of the branch or the conclusion of the fuzzy rule):

$$\textbf{IF } A_1 \textbf{ AND } \dots \textbf{ AND } A_Z \textbf{ THEN } C_k \tag{5}$$

where A_i = [a_{i1},a_{i2}], with $i \in$ [1, Z] and Z $\in$ [1,F] being F the number of features in the dataset, is a numerical interval which turns out of one or more in-equations in a tree (e.g., $a_{i1} \leq a$ and $a \leq a_{i2}$ with $a \in U_i$ that is the universe of discourse of feature i); or it corresponds to the $0.5 - cut$ in a FURIA fuzzy set. For example, given three features ($Color \in [0, 45]$, $Bitterness \in [8, 250]$, and $Strength \in [0.039, 0.136]$) the following decision tree (printed in Weka format) is translated into the 4 rules listed below:

```
Color <= 6
|   Bitterness <= 26: 1 (50.0)
|   Bitterness > 26: 3 (49.0)
Color > 6
|   Strength <= 0.057: 2 (50.0/1.0)
|   Strength > 0.057: 4 (2.0)
```

IF $Color$ is in A_1=[0,6] **AND** $Bitterness$ is in A_2=[8,26] **THEN** C_1
IF $Color$ is in A_1=[0,6] **AND** $Bitterness$ is in A_2=[26,250] **THEN** C_3
IF $Color$ is in A_1=[6,45] **AND** $Strength$ is in A_3=[0.039,0.057] **THEN** C_2
IF $Color$ is in A_1=[6,45] **AND** $Strength$ is in A_3=[0.057,0.136] **THEN** C_4

ExpliClas approximates each A_i by the closest linguistic term L in T (i.e., by the linguistic term associated with the fuzzy set with the most similar $0.5 - cut$ interval in the given SFP for feature i). The similarity $S(A, L)$ between the two numerical intervals A and L is computed as follows:

$$\forall L \in T : S(A, L) = \frac{A \cap L}{A \cup L} \in [0, 1], \tag{6}$$

being 1 in case A perfectly matches L, and 0 if both intervals are disjoint. In case that given an interval A, two different but consecutive linguistic terms L_l and L_{l+1} yielded the same similarity value (i.e., $S(A, L_l) = S(A, L_{l+1}) = 0.5$) then the linguistic approximation of A would be "L_l or L_{l+1}" (e.g., "*Low* or *Medium*"). Following with the previous illustrative example, if $Color$ were defined by a uniform SFP with two terms (*Low* and *High*), defined by two triangular fuzzy sets, then the two numerical intervals defined by the $0.5 - cut$ would be $Low = [0, 22.5]$ and $High = [22.5, 45]$. Then, $S(A_1, Low) = 6/22.5 = 0.266$ and $S(A_1, High) = 0$, with $A_1 = [0, 6]$. So, Low would be selected as the linguistic approximation of A_1.

It is worthy to note that this linguistic approximation is not needed in the case of FHDT which is endowed with global semantics (i.e., $\forall A \ \exists L \in T : S(A, L) = 1$) which favors the

semantic interpretability and explainability of the generated models. All the related documentation and source code are available at:

https://gitlab.citius.usc.es/jose.alonso/xai

IV. Experiments

The aim of this section is to show the new functionality implemented in ExpliClas. Firstly, in Subsection IV-A, we present the experimental setting. Secondly, in Subsection IV-B, we delve into the advantages and drawbacks of applying the linguistic approximation approach to build natural explanations associated to models characterized by local semantics, i.e., crisp decision trees and FURIA. Thirdly, in Subsection IV-C, we pay attention to different configurations associated with the FHDT and how they impact in getting a good interpretability-accuracy tradeoff. In addition, we show the naturalness and expressiveness of explanations supported by global semantics embedded into FHDT models. Finally, we illustrate how FINGRAMs are helpful to analyze FHDT models at inference level.

A. Experimental setting

In the experimental analysis, we have considered two well-known datasets (WINE and PIMA) which are taken from the UCI machine learning repository [35]. On the one hand, WINE contains 178 data instances. They represent the results of a chemical analysis of wines from grapes grown in the same Italian region but derived from three different cultivars. The classification task consists in identifying one out of 3 types of wines in terms of 13 features extracted from the previous chemical analysis. On the other hand, PIMA contains 768 data instances associated with subjects with Pima Indian heritage. In this case, the classification task is binary. It consists in detecting whether the subject shows sign of diabetes according to 8 features defined by the World Health Organization.

In addition, we have considered a real-world dataset (BEER) which was first introduced in [36]. BEER contains 400 data instances, with 50 instances associated with each one of 8 beer styles (Blanche, Lager, Pilsner, IPA, Stout, Barleywine, Porter, and Belgian Strong Ale). The classification task consists in recognizing one out of the aforementioned 8 beer styles in terms of 3 features (Color, Bitterness and Strength), carefully defined by a brewery worker. As described in [36], each feature is characterized by a SFP with trapezoidal fuzzy sets. Given a feature $X \in \{Color, Bitterness, Strength\}$, each linguistic term L_x associated with X is described by a trapezoidal fuzzy set in terms of 4 parameters $[0/a, 1/b, 1/c, 0/d]$. Obviously, only 3 parameters are required in case of the first and last fuzzy sets in a SFP, because they have semi-trapezoidal shape. Notice that for the sake of simplicity, we have represented each parameter by $\mu(x)/x$, being $\mu(x)$ the membership degree of value x and $x \in U$ (U is the universe of discourse of X):

Color $\in [0, 45]$: *Pale* [1/0, 1/2, 0/4], *Straw* [0/2, 1/4, 1/7, 0/8], *Amber* [0/7, 1/8, 1/18, 0/20], *Brown* [0/18, 1/20, 1/28, 0/30], *Black* [0/28, 1/30, 1/45].
Bitterness $\in [8, 250]$: *Low* [1/8, 1/20, 0/22], *Low-Medium* [0/20, 1/22, 1/30, 0/35], *Medium-High* [0/30, 1/35, 1/45, 0/50], *High* [0/45, 1/50, 1/250].
Strength $\in [0.039, 0.136]$: *Session* [1/0.039, 1/0.05, 0/0.055], *Standard* [0/0.05, 1/0.055, 1/0.065, 0/0.07], *High* [0/0.065, 1/0.07, 1/0.085, 0/0.095], *Very high* [0/0.085, 1/0.095, 1/0.136].

We have considered all algorithms implemented in ExpliClas: three crisp decision trees (J48, RandomTree, and REPTree) and two fuzzy rule-based classifiers (FURIA and FHDT). In addition, just for comparison purposes, we have also taken into account the Weka implementation of the crisp Hoeffding decision tree (HDT). Moreover, as baseline from the point of view of accuracy, we have selected the Random Forest (RF) [37] algorithm (implemented in Weka as well) because, as explained in [38], this algorithm is able to get high accuracy in most classification problems.

We have applied the 10-fold cross-validation provided by Weka [11] and reported the following quality metrics:

- for **Accuracy**: the ratio of correctly classified instances (RCCI), the root mean square error (RMSE), Precision, Recall, and F-Measure.
- for **Interpretability**: number of leaves/rules (NR), total rule length (TRL), number of concepts (NC), linguistic similarity (LS) between the concepts used in the model and those managed by human experts.

In case of decision trees, we first translate the tree branches into IF-THEN rules (see Eq. 5) and then we compute the interpretability metrics previously enumerated. TRL accounts for the total number of conditions A_i in all the rules. NC computes the number of distinct conditions which appears in the RB, i.e., we assume each condition to represent a concept and we count the number of different concepts in the RB.

For each single condition in a rule, we compute the LS with the closest linguistic term in the partition that is taken as a reference (see Eq. 6). In the case of BEER, we have considered as a reference both expert partitions and SFPs uniformly distributed in the universe of discourse associated with each feature. In the case of WINE and PIMA, since we don't have expert partitions, only uniform SFPs are taken into account. We have experimented with different numbers of fuzzy sets (2, 3, 5, 7) in the uniform SFPs. Two is chosen because crisp trees are binary trees, so each condition yields to two node children or one leaf node. Three, five, and seven are odd numbers smaller than nine. We have chosen those numbers because, according to psychologists (see [39], [40]), human information processing capability is limited to 7 ± 2 distinct concepts for a given feature. For example, LS-2 means we compute linguistic similarity with respect to linguistic approximations made of 2 fuzzy sets for each feature in the dataset. Then, we compute the LS for a given rule as the average value of the LS computed for all conditions in the rule premise. At the end, the LS for the whole model is computed as the average value of the LS computed for all rules.

Tables I and II summarize the results for the three datasets under study. In the case of the algorithm FHDT, we have tested the influence of two parameters. Namely, the number of fuzzy sets per feature and allowing (T=true) or not (F=false) the repetition of features when growing the tree. Thus, for example, FHDT-2-T means the algorithm creates

TABLE I
COMPARATIVE ANALYSIS OF INTERPRETABILITY-ACCURACY TRADEOFF FOR THE CONSIDERED ALGORITHMS (WINE AND PIMA DATASETS)

Dataset	Algorithm	Accuracy					Interpretability						
		RCCI (%)	RMSE	Precision	Recall	F-Measure	NR	TRL	NC	LS-2	LS-3	LS-5	LS-7
WINE	RF	**98.315**	**0.128**	**0.984**	**0.983**	**0.983**	-	-	-	-	-	-	-
	J48	93.820	0.202	0.938	0.938	0.938	5.2	12.9	8.4	0.604	0.601	0.508	0.373
	RandomTree	93.258	0.212	0.933	0.933	0.932	11.2	39.8	39.8	0.645	0.601	0.466	0.354
	REPTree	93.258	0.202	0.933	0.933	0.932	4.6	10.4	10.4	0.654	0.626	0.496	0.359
	HDT	88.202	0.249	0.882	0.882	0.882	5.7	13.9	13.9	0.636	0.589	0.476	0.375
	FHDT-2-F	75.281	0.425	0.789	0.753	0.752	**2.9**	**4.7**	**3.8**	**1.000**	0.667	0.400	0.286
	FHDT-2-T	64.607	0.427	0.679	0.646	0.649	5.6	9.8	4.5	0.733	0.489	0.293	0.209
	FHDT-3-F	87.640	0.339	0.879	0.876	0.877	7	13.4	9	0.509	**1.000**	0.600	0.429
	FHDT-3-T	87.640	0.3257	0.878	0.876	0.876	9.6	18.6	10.2	0.458	0.905	0.543	0.388
	FHDT-5-F	89.326	0.263	0.894	0.893	0.893	11.8	20.4	13.6	0.342	0.483	**1.000**	0.620
	FHDT-5-T	**90.960**	**0.243**	**0.910**	**0.910**	**0.910**	14.5	26.5	13.6	0.342	0.482	**1.000**	0.619
	FHDT-7-F	86.441	0.269	0.868	0.864	0.865	18.2	31.4	20.2	0.262	0.372	0.519	**1.000**
	FHDT-7-T	89.830	0.254	0.900	0.898	0.898	20.4	36.2	20.4	0.262	0.367	0.521	**1.000**
	FURIA	94.944	0.182	0.950	0.949	0.950	6.3	12.4	12.4	0.588	0.577	0.521	0.421
PIMA	RF	75.781	**0.403**	0.754	0.758	**0.755**	-	-	-	-	-	-	-
	J48	73.828	0.446	0.735	0.738	0.736	19.2	77	34	0.593	0.535	0.441	0.351
	RandomTree	68.099	0.565	0.684	0.681	0.682	135	760.5	235.4	0.517	0.492	0.443	0.401
	REPTree	75.260	0.429	0.747	0.753	0.748	14.5	50.9	23.3	0.590	0.541	0.466	0.405
	HDT	70.573	0.446	0.700	0.706	0.702	21.2	82.2	38.2	0.538	0.502	0.411	0.363
	FHDT-2-F	65.104	0.465	-	0.651	-	**4.1**	**10**	**6.2**	**1.000**	0.667	0.4	0.286
	FHDT-2-T	65.104	0.465	-	0.651	-	20.7	59.3	8.8	0.736	0.491	0.294	0.210
	FHDT-3-F	69.922	0.441	0.744	0.699	0.625	6.4	12.3	8.1	0.520	**1.000**	0.6	0.429
	FHDT-3-T	72.266	0.429	0.727	0.723	0.686	34.3	82.8	12.4	0.421	0.844	0.506	0.362
	FHDT-5-F	73.890	0.423	0.738	0.739	0.714	19.6	47.9	23.4	0.330	0.505	**1.000**	0.638
	FHDT-5-T	75.785	0.409	**0.757**	0.758	0.740	54.4	149.6	29.8	0.294	0.442	0,891	0.562
	FHDT-7-F	72.881	0.421	0.719	0.729	0.715	32.6	75.2	37.5	0.253	0.370	0.561	**1.000**
	FHDT-7-T	**75.984**	**0.401**	0.754	**0.760**	**0.751**	77.2	213.8	40.3	0.248	0.365	0.548	0.988
	FURIA	74.479	0.473	0.737	0.745	0.734	7.8	17.5	17	0.649	0.583	0.484	0.405

TABLE II
COMPARATIVE ANALYSIS OF INTERPRETABILITY-ACCURACY TRADEOFF FOR THE CONSIDERED ALGORITHMS (BEER DATASET)

Algorithm	Accuracy					Interpretability							
	RCCI (%)	RMSE	Precision	Recall	F-Measure	NR	TRL	NC	LS-2	LS-3	LS-5	LS-7	LS-EXP
RF	**96.250**	**0.087**	**0.962**	**0.963**	**0.962**	-	-	-	-	-	-	-	-
J48	95.000	0.108	0.950	0.950	0.950	9.8	23.4	15.5	0.526	0.598	0.542	0.473	0.585
RandomTree	94.000	0.123	0.940	0.940	0.940	27.6	75.4	42.9	0.419	0.445	0.439	0.426	0.462
REPTree	95.250	0.107	0.953	0.953	0.952	8	18	12.9	0.580	0.602	0.517	0.459	0.565
HDT	92.000	0.145	0.920	0.920	0.920	10.8	25.1	16.7	0.482	0.549	0.511	0.473	0.526
FHDT-2-F	37.500	0.316	0.450	0.375	0.309	**2.6**	**3.8**	**3.2**	**1.000**	0.667	0.400	0.286	0.448
FHDT-2-T	36.000	0.317	0.413	0.360	0.332	10.9	16.6	4.7	0.450	0.300	0.180	0.128	0.212
FHDT-3-F	66.000	0.289	0.718	0.660	0.636	6	11.5	7.5	0.542	**1.000**	0.600	0.429	0.620
FHDT-3-T	72.000	0.273	0.774	0.720	0.705	18.9	37.6	12.2	0.401	0.721	0.433	0.309	0.444
FHDT-5-F	79.950	0.252	0.823	0.799	0.792	10.9	19.1	12.4	0.363	0.471	**1.000**	0.611	**0.791**
FHDT-5-T	86.146	0.210	0.870	0.861	0.864	25.6	52.3	15.3	0.331	0.446	0.934	0.575	0.710
FHDT-7-F	87.154	0.225	0.882	0.872	0.873	15.7	26.6	17.7	0.267	0.389	0.522	**1.000**	0.485
FHDT-7-T	95.214	**0.158**	**0.953**	**0.952**	**0.952**	35.3	68.8	19.8	0.251	0.361	0.520	0.971	0.482
FHDT-544-F	76.000	0.257	0.802	0.760	0.739	9.2	14.8	10.6	0.401	0.507	0.897	0.593	0.582
FHDT-544-T	83.750	0.224	0.852	0.838	0.840	22.5	44.9	14.1	0.381	0.476	0.745	0.528	0.514
FHDT-EXP-F	89.750	0.228	0.902	0.898	0.898	11	19.4	13	0.417	0.497	0.858	0.586	0.707
FHDT-EXP-T	**94.750**	0.174	0.949	0.948	0.947	23.7	51.4	13.8	0.427	0.513	0.791	0.584	0.684
FURIA	95.750	0.097	0.957	0.958	0.957	14.6	30.8	29.8	0.483	0.525	0.490	0.488	0.507

SFPs with two fuzzy sets uniformly distributed in the universe of discourse of each feature. T means repetition of features is allowed. In the case of BEER, FHDT-544-F means that we considered different number of fuzzy sets per feature (5 for Color, 4 for Bitterness and 4 for Strength) in accordance with the expert partitions taken as reference. F means repetition of features is not allowed. We used EXP instead of 544 to identify non-uniform SFPs closer to those defined by the expert. Notice that even though expert partitions previously defined manage fuzzy sets with trapezoidal shape the current implementation of FHDT only manages triangular fuzzy sets. Therefore, we defined EXP partitions in FHDT-EXP as follows (with centroids in brackets):

Color (FHDT-EXP): *Pale* (0), *Straw* (5.5), *Amber* (13), *Brown* (24), *Black* (45).
Bitterness (FHDT-EXP): *Low* (8), *Low-Medium* (26), *Medium-High* (40), *High* (250).
Strength (FHDT-EXP): *Session* (0.039), *Standard* (0.06), *High* (0.0775), *Very high* (0.136).

The following sections discuss the main findings derived from these experiments.

B. Analysis of linguistic explanations associated with models with local semantics

As it can be appreciated in Tables I and II, the most accurate models are usually built with RF and FURIA. In addition, REPTree generates models with a good interpretability-accuracy tradeoff. They are the most compact models in terms of NR, TRL and NC among all models with local semantics

(i.e., J48, RandomTree, REPTree, HDT and FURIA), but they also have an accuracy that is not far from FURIA in all the three problems under study.

It is worthy to note that the linguistic approximation applied on top of these models is the only way to verbalize their behavior in natural language. For example, *flavonoids* $<$ *1.23 AND color intensity* $\geq$ *3.42 THEN Wine3* is a branch of a REPTree for the WINE classification problem, and it is translated into *Wine is Wine3 because flavonoids is low and color intensity is high* with LS-2 equals 0.494. In practice, the reported values for LS are always under 0.7 (they are actually under 0.5 in many cases) what may jeopardize explainability and yield to misunderstanding in some cases. Someone may argue that in case LS is smaller than 0.5 it may be better just to keep the numbers as part of the explanation (e.g., *Wine is Wine3 because flavonoids is smaller than 1.23 and color intensity is greater or equal than 3.42*) instead of using vague linguistic approximations. However, the validation of this hypothesis remains out of the scope of this work.

C. Analysis of multi-modal explanations associated with models with global semantics

Models built with FHDT (the only algorithm endowed with global semantics in our experiments) represent a Pareto of solutions with different balance between accuracy and interpretability. The smaller the number of fuzzy sets the better the interpretability but the worse the accuracy. It seems that at least 5 fuzzy sets are needed to achieve accuracy comparable to models with local semantics. In addition, allowing repetition of features seems to produce more accurate models while they still exhibit good interpretability in terms of NR, TRL and NC. Notice that, as expected, FHDT clearly overwhelms the rest of the algorithms regarding LS. Models built with FHDT and no repetition of features (F) always produce some LSs equal to 1. However, models built with FHDT and with repetition of features produce usually high values of LS as well.

In the case of the BEER classification problem, we can analyze results in comparison with fuzzy partitions defined by humans. Contrary to intuition, FHDT-EXP does not produce the best LS-EXP. Due to the fact that FHDT does not implement yet fuzzy sets with trapezoidal shape, so the generated partitions differ from the expert ones. As a result, we observe that the linguistic approximation (LS-EXP) is better when considering uniform SFP with 5 fuzzy sets per feature (FHDT-5) than when approximating the expert partitions with triangular fuzzy sets (FHDT-544); but at the cost of lower accuracy. We can conclude that FHDT-EXP-F produces the best model regarding both accuracy and explainability.

As an illustrative example, Fig. 2 shows the FINGRAM generated for FHDT-EXP-F and one given data instance. Each node represents one rule and the colored area corresponds to the degree of activation for the majority class associated with such a rule. Only rules fired for the given data instance are displayed with the aim of facilitating the interpretation of the fuzzy inference process. We observe that rules R9 and R10 are partially redundant (connected by green edge) since both vote for Belgian Strong Ale as main class, while they are in competition with rule R6 (connected by red edge) which is for Barleywine. It is worthy to remind that FHDT can assign several classes to the same rule with different weights. Thus, the final class is chosen after analyzing the interpolation of weights among all fired rules. In the RB, the average number of co-fired rules is 2.618 which is small enough to allow a human-friendly fingram-based analysis. The interested reader is kindly referred to [26] for further details about how to generate and interpret FINGRAMS. Moreover, illustrative examples about how FINGRAMS facilitate understanding of the FURIA inference process are also provided in [26].

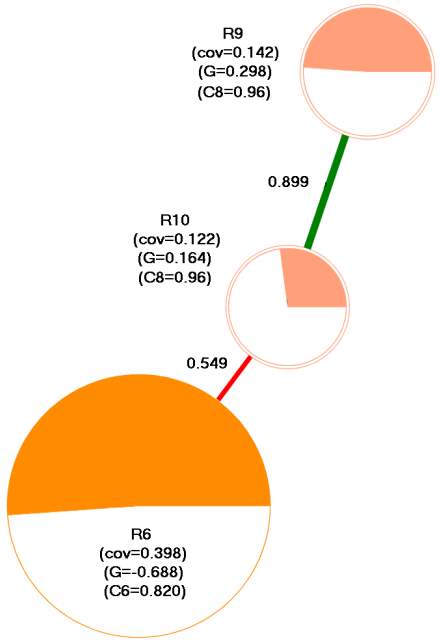

Fig. 2. Instance-based FINGRAM for FHDT-EXP-F

V. Conclusions and Future Work

In this paper, we have presented the new version of ExpliClas which is enhanced with the explainable fuzzy decision trees called FHDTs. They are endowed with global semantics thanks to the use of SFPs. As we have shown in the experimental section, FHDTs exhibit a good interpretability-accuracy tradeoff. Moreover, they yield more natural textual explanations than those based on linguistic approximations of numerical intervals coming out of transparent models with local semantics (but without inherent linguistic interpretability). In addition, we have added a new service for generation and visualization of FINGRAMs, i.e., for facilitating the visual analysis and the comprehension of fuzzy rules at inference level. As we have shown with an illustrative example, FINGRAMs assist users to identify the most relevant rules but also to identify potential redundancies and inconsistencies that should be fixed in order to produce a rule-based classifier with better interpretability-accuracy tradeoff.

As future work, we plan to add more Weka algorithms (even black-box algorithms) to ExpliClas. In addition, we will

generate textual explanations associated with FINGRAMs. Moreover, we will combine numbers and vague linguistic terms to enhance the explanations currently generated by ExpliClas. The effectiveness of such explanations will be measured through intrinsic and extrinsic evaluation with humans.

ACKNOWLEDGMENT

Jose M. Alonso is a *Ramón y Cajal* Researcher (RYC-2016-19802). This research is supported by the Spanish Ministry of Science, Innovation and Universities (grants RTI2018-099646-B-I00, TIN2017-84796-C2-1-R, TIN2017-90773-REDT, RED2018-102641-T), the Galician Ministry of Education, University and Professional Training (grants ED431F 2018/02, ED431C 2018/29, ED431G/08, ED431G2019/04), and the Italian Ministry of Education and Research (MIUR), in the framework of the CrossLab project (Departments of Excellence) and of the PON R&I 2014-2020 "AIM: Attraction and International Mobility" project. Some of the previous grants were co-funded by the European Social and Regional Development Fund.

REFERENCES

[1] European Commission, "Artificial Intelligence for Europe," European Commission, Brussels, Belgium, Tech. Rep., 2018, Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions (SWD(2018) 137 final), https://ec.europa.eu/digital-single-market/en/news/communication-artificial-intelligence-europe.

[2] C. Molnar, *Interpretable Machine Learning*. Leanpub, 2019.

[3] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 93:1–93:42, 2019.

[4] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, pp. 206—215, 2019.

[5] J. M. Alonso, C. Castiello, and C. Mencar, "Interpretability of Fuzzy Systems: Current Research Trends and Prospects," in *Springer Handbook of Computational Intelligence*, J. Kacprzyk and W. Pedrycz, Eds. Springer Berlin / Heidelberg, 2015, pp. 219–237.

[6] D. Dubois, "Forty years of fuzzy sets," *Fuzzy Sets and Systems*, vol. 156, no. 3, pp. 331–333, 2005.

[7] J. M. Alonso, C. Castiello, and C. Mencar, "A bibliometric analysis of the explainable artificial intelligence research field," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU)*, 2018, pp. 3–15.

[8] J. M. Alonso, "From Zadeh's computing with words towards explainable artificial intelligence," in *International Workshop on Fuzzy Logic and Applications*. Springer, 2018, pp. 244–248.

[9] O. Cordón, "A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems," *International Journal of Approximate Reasoning*, vol. 52, pp. 894–913, 2011.

[10] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[11] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[12] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. Springer, 2017.

[13] R. Layton, *Learning Data Mining with Python*. Packt Publishing, 2015.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1135–1144.

[15] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini, "Factual and counterfactual explanations for black box decision making," *IEEE Intelligent Systems*, 2019.

[16] J. Alcala-Fdez and J. M. Alonso, "A Survey of Fuzzy Systems Software: Taxonomy, Current Research Trends, and Prospects," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 40–56, 2016.

[17] S. Guillaume and B. Charnomordic, "Learning interpretable fuzzy inference systems with FisPro," *Information Sciences*, vol. 181, no. 20, pp. 4409–4427, 2011.

[18] D. P. Pancho, J. M. Alonso, and L. Magdalena, "Quest for interpretability-accuracy trade-off supported by fingrams into the fuzzy modeling tool GUAJE," *International Journal of Computational Intelligence Systems*, vol. 6, pp. 46–60, 2013.

[19] J. M. Alonso and A. Bugarín, "ExpliClas: Automatic generation of explanations in natural language for weka classifiers," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–6.

[20] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[21] J. Hühn and E. Hüllermeier, "FURIA: an algorithm for unordered fuzzy rule induction," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[22] R. Pecori, P. Ducange, and F. Marcelloni, "Incremental learning of fuzzy decision trees for streaming data classification," in *11th Conf. European Society for Fuzzy Logic and Technology (EUSFLAT)*. Atlantis Press, 2019.

[23] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD'00, 2000, pp. 71–80.

[24] D. Pancho, J. M. Alonso, O. Cordón, A. Quirin, and L. Magdalena, "Fingrams: Visual representations of fuzzy rule-based inference for expert analysis of comprehensibility," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 6, pp. 1133–1149, 2013.

[25] A. Segatori, F. Marcelloni, and W. Pedrycz, "On distributed fuzzy decision trees for big data," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 174–192, 2018.

[26] D. Pancho, J. M. Alonso, and L. Magdalena, "Enhancing fingrams to deal with precise fuzzy systems," *Fuzzy Sets and Systems*, pp. 1–25, 2016.

[27] B. Vargas-Quesada and F. Moya-Anegón, *Visualizing the structure of science*. Springer-Verlag, 2007.

[28] A. Quirin, O. Cordón, J. Santamaría, B. Vargas-Quesada, and F. Moya-Anegón, "A new variant of the pathfinder algorithm to generate large visual science maps in cubic time," *Information Processing and Management*, vol. 44, no. 4, pp. 1611–1623, 2008.

[29] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, pp. 7–15, 1989.

[30] D. P. Pancho, J. M. Alonso, and J. Alcala-Fdez, "A new fingram-based software tool for visual representation and analysis of fuzzy association rules," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2013, pp. 1–7.

[31] D. P. Pancho, J. M. Alonso, J. Alcalá-Fdez, and L. Magdalena, "Analyzing fuzzy association rules with Fingrams in KEEL," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2014.

[32] A. Segatori, A. Bechini, P. Ducange, and F. Marcelloni, "A distributed fuzzy associative classifier for big data," *IEEE Transactions on Cybernetics*, vol. 48, no. 9, pp. 2656–2669, 2018.

[33] E. Ruspini, "A new approach to clustering," *Information and Control*, vol. 15, no. 1, pp. 22–32, 1969.

[34] C. Mencar and A. M. Fanelli, "Interpretability constraints for fuzzy information granulation," *Information Sciences*, vol. 178, no. 24, pp. 4585–4618, 2008.

[35] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[36] G. Castellano, C. Castiello, and A. M. Fanelli, "The FISDeT software: Application to beer style classification," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1–6.

[37] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[38] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.

[39] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *The Psychological Review*, vol. 63(2), pp. 81–97, 1956.

[40] T. L. Saaty and M. S. Ozdemir, "Why the magic number seven plus or minus two," *Mathematical and Computing Modelling*, vol. 38(3), pp. 233–244, 2003.