This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3049911, IEEE Transactions on Fuzzy Systems

IFC-BD: An Interpretable Fuzzy Classifier for Boosting Explainable Artificial Intelligence in Big Data

Fatemeh Aghaeipoor, Mohammad Masoud Javidi, Alberto Fernández

***Abstract*—In current Data Science applications, the course of action has derived to adapt the system behavior for the human cognition, resulting in the emerging area of explainable artificial intelligence. Among different classification paradigms, those based on fuzzy rules are suitable solutions to stress the interpretability of the global systems. However, in case of addressing Big Data analytics, they may comprise an excessive number of rules and/or linguistic labels that not only may cause losing the system performance but also may affect the system semantic as well as the system interpretability. In this paper, we propose IFC-BD, an Interpretable Fuzzy Classifier for Big Data, aiming at boosting the horizons of explainability by learning a compact yet accurate fuzzy model. IFC-BD is developed in a cell-based distributed framework through the three working stages of initial rule learning, rule generalization, and heuristic rule selection. This whole procedure allows reaching from a high number of specific rules to less number of more general and confident rules. Additionally, in order to resolve possible rules conflict, a new estimated rule weight is proposed specifically for Big Data problems. IFC-BD was evaluated in comparison to the state-of-the-art approaches of the fuzzy classification paradigm, considering interpretability, accuracy, and running time. The findings of the experiments revealed that the proposed algorithm was able to improve the explainability of fuzzy rule-based classifiers as well as their predictive performance.**

***Index Terms*—Explainable Artificial Intelligence, Fuzzy Rule-Based Classification Systems, Big Data, Interpretability, Apache Spark Framework, Scalability**

I. Introduction

DATA Science and Big Data analytics have become a growing demand nowadays. These are complementary areas that aim at extracting knowledge from the vast amount of data, generated from a myriad of different sources [1]. In this regard, some recent innovative frameworks motivate data scientists to develop new algorithms intentionally for Big Data problems [2]. However, design of new scalable algorithms to handle large amounts of data is not a straightforward task and may bring out complex systems that affect understanding of the system behavior. This issue contradicts the objectives of the new resurgence of Artificial Intelligence known as eXplainable Artificial Intelligence (XAI) [3].

In the context of XAI, the focus is on providing more explainable, interpretable, and transparent systems. Although there is a little consensus on these terms in the literature [4], all have a common sense to clarify the inner functionality of the systems for direct interaction with the human users. Explainability mostly refers to understanding the internal functions of a model, so that we are able to determine why a certain output is given with relation to its variables or data representation. By contrast, interpretability is to understand the model itself, to be able to comprehend its components directly, in a similar way humans do the cognition process. Finally, transparency is a capability for models to ensure the former both features. All in all, these characteristics enable users to better understand, trust, modify, and manage behaviors of intelligent systems. In this regard, the use of rule-based systems in general [5], and Fuzzy Rule-based Classification Systems (FRBCSs) in particular is advisable [6], [7].

FRBCSs can be considered under two different perspectives of XAI. On the one hand, the use of linguistic fuzzy labels containing a semantic knowledge inspired by the human language is a straightforward transition to globally explain the phenomena to the practitioner [7]. On the other hand, simple linguistic rules with short antecedents are very manageable for the human user. More specifically, when the whole amount of rules be precisely the ones that explain the concept, the model would be similar to human cognition [8]. Furthermore, the success of FRBCSs is especially highlighted in those applications in which the decision-making process should be made clear to users [9]. Indeed, this classification paradigm allows for a total transparency when indicating how a given output is inferred from the input variables, e.g. if the inference is carried out through a single “winner” rule, the local explanation of the query instance is made straightforward.

Focusing on Big Data analytics, FRBCSs also show excellent properties for addressing complex applications [10]–[12]. Most of the proposed models in the specialized literature tried to adapt the sequential learning algorithms such as the Chi et al’s [13] through a Map-Reduce paradigm. Among these, Chi-BigData [14] was the first attempt, which was also extended in several different studies [15], [16]. More recently, an associative fuzzy classifier, named as CFM-BD [17], was developed showing a robust predictive performance with respect to more complex algorithms such as fuzzy decision trees [18]. Finally, the authors of this current paper presented a novel scalable

F. Aghaeipoor is with the Dept. of Mathematics and Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran, e-mail: ff.aghaei@eng.uk.ac.ir.

M. M. Javidi is with the Dept. of Mathematics and Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran, and also with the Mahani Mathematical Research Center, Shahid Bahonar University of Kerman, Box No. 76135-133, Kerman, Iran, e-mail: javidi@uk.ac.ir (Corresponding author: M. M. Javidi)

A. Fernández is with the DaSCI Andalusian Research Institute, University of Granada, Spain, e-mail: alberto@decsai.ugr.es.

fuzzy classifier for Big Data, known as Chi-BD-DRF, which complements the Chi-based fuzzy Big Data learning by adding what we noted as "dynamic rule filtering" approach for the sake of simplification of the rule set [19].

These previously developed Big Data FRBCSs are involving with some intrinsic challenges. They are mostly computationally intensive. Even if they can come up with scalable solutions, they may be complex in terms of their Data Base (DB) or Rule Base (RB). In the case of DB, employing high number of variables as well as high number of granularities not only increase the number of generated rules but also result in longer rules that are conceptually difficult for human beings to be handled. In the case of rule set, some designs lead to obtaining an excessive number of rules, whereas many of which may be so specific belonging to the non-dense areas or outliers. Another challenge is related to the rules' weights, where the adaptation of the classical approaches via several iterations over all data samples is not practical for the Big Data scenarios. On the other perspective to obtain a competitive predictive performance, some methods like CFM-BD [17], apply a pre-processing transformation that is not straightforwardly interpretable. In order to address these challenges, the fusion of association rule mining concepts and FRBCSs can be taken into account to provide efficient measures and develop robust yet scalable solutions specifically for Big Data scenarios.

In this study, we propose an Interpretable Fuzzy Classifier for Big Data, noted as IFC-BD, a novel fuzzy classifier that is designed under the umbrella of XAI and ensuring scalability constrains. IFC-BD aims at learning a compact yet accurate FRBCS containing less number of short rules. Additionally, it provides a confident and reliable RB based on a lower granularity, i.e., a fewer number of fuzzy sets per variable that are more comprehensible and manageable by the practitioner. It is also remarkable that IFC-BD preserves the original semantics of the linguistic fuzzy labels. All these objectives assist to provide explainable fuzzy models easing the translation from Machine Learning (ML) models to human cognition and guaranteeing practical artificial intelligence systems.

IFC-BD is developed under Apache Spark [20], and it is composed of three stages: initial rule learning, rule generalization, and rule selection. In the first stage, initial rules are learned by employing our former Chi-BD-DRF methodology [19]. In the second stage, the focus is on compacting the RB in terms of rule length. Finally, in the third stage, the number of final rules is controlled by a heuristic approach. Along the generalization process, a fast and accurate conflict resolution method is presented using an estimated rule weight. It avoids using crisp measures in fuzzy environments, capturing all the power of fuzzy modeling but taking into account the scalability issue that is mandatory for Big Data applications.

The main novelties of this current approach are listed below:

- A dynamic rule filtering scheme to focus on the high density areas of the problem, for the sake of simplifying the baseline fuzzy classification model.
- A rule learning enhancement of the fuzzy rule set for obtaining more general and interpretable rules.
- A heuristic rule selection mechanism to boost the global interpretability of the system by keeping the most influential rules.

The capabilities of the IFC-BD algorithm were validated using 9 big classification datasets with different numbers of variables and samples. IFC-BD was compared with the current state-of-the-art fuzzy classifier for Big Data, CFM-BD [17], and the baseline Chi-BD-DRF [19]. Several statistical tests were also conducted to provide stronger support for the findings extracted throughout the analysis. The obtained results revealed that IFC-BD could efficiently generate more compact RBs and meaningful DBs, improving the explainability of the FRBCSs without hindering accuracy or running time.

The remainder of this study is structured as follows. Section II presents the fundamental concepts and frameworks employed in this study, including some backgrounds of FRBCSs, XAI, and Big Data environments. Section III introduces the proposed IFC-BD algorithm, detailing the three stages from initial rule generation to final rule selection. Section IV includes the experimental analysis, contrasting IFC-BD to state-of-the-art Big Data fuzzy classifiers from different aspects of accuracy and interpretability. Finally, Section V summarizes and concludes this study.

II. Preliminaries: Fuzzy Rule-Based Classification Systems for eXplainable Artificial Intelligence and Big Data

Throughout this section, the fundamental concepts and frameworks employed in this contribution are described. At first, the basics of the FRBCSs besides the used symbols and notations are reviewed (Section II-A). Then, we discuss some details of XAI and its importance in nowadays researches (Section II-B). Finally, we briefly introduce the characteristics and technical solutions related to the Big Data environments (Section II-C).

A. Components and Structure of FRBCSs

Suppose that we have dataset D with n input variables, m class labels, and $|\mathrm{D}|$ data samples in the form of $X_i = \left(x_i^1, x_i^2, ..., x_i^n\right)$, where $i = 1, ..., |\mathrm{D}|$, and X_i belongs to class $c_i \in C = \{c_1, ..., c_m\}$. This dataset can be employed as the base of different rule learning algorithms to generate fuzzy rules in the following structure:

$$\begin{aligned}\mathrm{Rule}_j : \ &\textbf{If } x^1 \text{ is } A_j^1 \ and \ ... \ and \ x^n \text{ is } A_j^n, \\ &\textbf{Then} \text{ class is } c_j \ : \ \mathrm{RW}_j\end{aligned} \tag{1}$$

where A_j^k is the linguistic variable corresponding to the dimension $k = 1, 2, ..., n$, and c_j and RW_j are the class label and the weight of this rule (RW), respectively. This rule can also be represented as a fuzzy association rule as follows:

$$R_j : A_j \to c_j : \mathrm{RW}_j \quad ; \quad A_j = \left\{A_j^1, ..., A_j^n\right\} \tag{2}$$

where A_j is a set containing all the antecedents and c_j is the consequence part of this rule. In the area of Association Rule Mining (ARM), two criteria are commonly used to quantify the interestingness of the rules, namely confidence and support. Whereas the former measures the frequency of the occurrence

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3049911, IEEE Transactions on Fuzzy Systems

of a rule, the latter evaluates the strength or reliability of the rules [21]. Formally, confidence of rule R_j is computed as:

$$\mathrm{Conf}(R_j) = \frac{\sigma(A_j, c_j)}{\sigma(A_j)} \tag{3}$$

where $\sigma(A_j, c_j)$ counts the number of rules present in the RB, having the structure exactly composed of A_j and c_j, and $\sigma(A_j)$ counts the number of rules whose antecedents are equivalent to the j-th rule. Similarly, support of this rule is defined as follows:

$$\mathrm{Supp}(R_j) = \frac{\sigma(A_j, c_j)}{|\mathrm{D}|} \tag{4}$$

where $\sigma(A_j, c_j)$ counts how many rules with the structure equivalent of R_j are present in the RB, and $|\mathrm{D}|$ shows the number of available examples. These measures present crisp support and confidence that are differentiated from the original fuzzy values by considering matching degree of 1 for the covering examples [22]. This consideration results in computationally efficient measures, well-suited for Big Data applications. For the sake of simplicity, we look at the support values at the scale of dataset D, i.e., $\mathrm{Supp}(R_j) = \sigma(A_j, c_j)$.

After generating an FRBCS, an inference module is needed to get new predictions from the learned model. To this end, a fuzzy reasoning method is applied. Among different alternatives, the one that provides a higher degree of explainability is the winning rule scheme [23], for which the rule with the highest matching degree is the one that sets the final class label, which is determined as follows:

$$R_o = arg\ max_{R_j \in RB} \left\{ \mu_{A_j}(X_i) \cdot \mathrm{RW}_j \right\} \tag{5}$$

where $\mu_{A_j}(X_i)$ is the matching degree of the new example (X_i) with rule R_j and it is defined as:

$$\mu_{A_j}(X_i) = \prod_{k=1}^{n} \mu_{A_j^k}(x_i^k) \tag{6}$$

where $\mu_{A_j^k}(x_i^k)$ is the membership degree of the input value x_i^k in fuzzy set A_j^k. As was shown in (6), the product t-norm is selected as the default aggregation function in all the computations for the sake of simplicity.

B. Capabilities of Fuzzy Modeling for Promoting XAI Systems

Artificial Intelligence methods, especially ML models, are increasingly applied to solve complex and computational problems of human life. They are establishing intelligent systems perceiving, learning, deciding, and operating almost without human intervention. In such a situation where these intelligent systems are highly employed in critical aspects of our lives like medicine, law, finances, self-driving cars, robotic assistants, and so on. Understanding and explaining their internal logic finds a significant importance [3], as it helps human users to trust sincerely, manage effectively, avoid biases, evaluate decisions, and provide more robust ML models. All these objectives are nowadays following in the context of XAI [4].

Some ML models like deep neural networks are black boxes in which their inner mechanism is either basically unknown to the users or it is known but totally difficult to interpret by human recognition. Although, there are some approaches to alleviate this crucial weakness [24], [25], the inherently interpretable models like rule-based systems are more reliable options that must be considered to replace black box models if possible [9]. These systems are straightforward ways to fulfill two important perspectives of ML models, namely accuracy and interpretability. Specifically, rule-based systems provide a good representation of the phenomena under study in the form of simple interpretable rules, leading to a direct understanding of the prediction process [5].

Semantic knowledge can be boosted in rule-based systems via linguistic fuzzy systems [7], leading to a more human-compatible representation style of the model. As pointed out, the use of linguistic terms is a natural knowledge representation, facilitating the system interpretability of and human interactions as well as modeling in the imprecise domains. In this context, the RB compactness and the semantic comprehensibility of the DB must be further emphasized. The former can be taken into account regarding the coverage and the specificity of the rules. Indeed, high number of rules with a limited coverage are difficult to be interpreted. Similarly, too specific rules with a high number of antecedents increase the system complexity and are in direct contradiction to the interpretability criteria [26]. In the case of comprehensibility of the DB, a lower number of fuzzy sets and homogeneous fuzzy sets are desirable. These all make the whole DB more meaningful and convenient to be understood by the human cognition [27].

C. Big Data Environments, Tools, and Frameworks

Due to the special characteristics of Big Data environments, data scientists are nowadays putting their effort into providing scalable and fault-tolerant algorithms executing in tolerable times [28]. In this regard, distributed computing frameworks offer new opportunities. These frameworks are able to split input data into several partitions and spread them across a cluster of nodes to be processed in parallel [29].

Map-Reduce execution methodology, mainly provided by the Hadoop ecosystem, is one of the most commonly used distributed approaches [2]. A Map-Reduce job is comprised of two functions, `Map` and `Reduce`, applying on a set of distributed data partitions provided by Hadoop Distributed File System (HDFS). Whereas the `Map` function is devoted to execute the working algorithm on local partitions, the `Reduce` function is in charge of combining Maps results. These functions are highly iterative and interactive when performing in parallel mode, needing lots of data replication, serialization, and cost-extensive I/O operations. These all cause substantial running-time overhead. Apache Spark resolves these problems using memory-based operations, where the computations are carried out in memory over the local partitions.

Spark has also other favorable features, making it a popular and powerful framework in Big Data processes. Among others, Lazy evaluation using immutable Resilient Distributed Datasets (RDD), multiple programming languages, streaming, and real-time processing, compatibility with Hadoop can be remarked [20].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3049911, IEEE Transactions on Fuzzy Systems

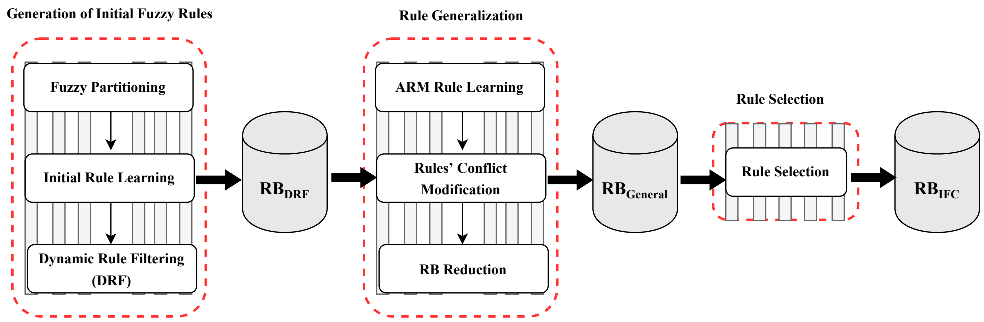

Fig. 1. The general schema of IFC-BD: an illustration of the RB evolution.

Spark has been also supporting practical ML tasks by providing MLlib library [30]. Particularly, pipeline API facilitates implementation of multi-stage algorithm so that each stage can be either a `Transformer` or an `Estimator` executing on the structured data as `DataFrames`. The use of pipeline tools helps to design complex ML algorithms in a simple and transparent yet robust way. It is a direct and straightforward solution to replicate and distribute the same operations across several executing nodes in Big Data environments [19].

III. IFC-BD Proposal: Interpretable Fuzzy Classifier for Big Data

This section describes the working procedure of the IFC-BD algorithm to learn a compact and efficient fuzzy classifier from big datasets. This algorithm is developed with respect to the priorities of XAI, especially in terms of interpretability at the level of DB and RB as previously discussed in Section II-B.

For the sake of completeness, the workflow of IFC-BD is illustrated in Fig. 1. As it can be observed, it starts by creating a baseline RB that is then optimized by considering three key design elements. These correspond to the three depicted stages that ensure obtaining both high confidence rules that promote the predictive performance, and a highly interpretable classification system, as described below:

1) In the first stage (Section III-A), all the possible rules of the input space are generated in a fast procedure.
2) Although these rules are learned from the dense areas, they are too specific that each covers a limited region. Hence, these rules are generalized in the second stage (Section III-B), leading to shorter antecedent rules that are easier to be handled by the human user.
3) Finally, in the third stage (Section III-C), the size of RB is reduced by selecting the best-performing rules.

A. Stage 1: Generation of Initial Fuzzy Rules

This stage aims at building an initial RB in a simple and scalable way. To this end, we use the idea of our original design in the Chi-BD-DRF algorithm [19], in which all possible rules covering an acceptable number of examples are generated through the three steps of Fuzzy Partitioning, Initial Rule Learning, and Dynamic Rule Filtering. In what follows, all these steps are described in detail.

1) Fuzzy Partitioning: All input values must be converted to fuzzy values to be usable in the fuzzy frameworks. To this end, we apply uniform fuzzy partitioning and triangular membership functions in which a set of homogeneous fuzzy labels is defined for each variable. By doing so, the input spaces are split into several cells like a grid environment, and each example falls into one of the formed cells. Depending on the support of the training examples within a given cell, that cell can be a dense or a sparse area. Our interest lies in the former because these might be considered to store the actual knowledge of a Big Data case study. These cells are the key elements to design the distributed operations of our method over a large amount of data examples. That is, each cell operates as an independent data-processing unit where the learning operations are carried in and aggregated from.

2) Initial Rule Learning: This step is meant to learn one single fuzzy rule from every cell containing at least one example. In addition, simultaneously to the learning process, three different measures, namely confidence, support, and prototype, are computed for each rule. These measures are intended to aggregate and represent the information captured from all the examples available in a cell. They are computed in a scalable and efficient way to be employed in the second and third stages of IFC-BD.

The whole rule learning process is developed through a `Map` and a `Reduce` function. Along with the `Map` operation, Chi et al's rule learning algorithm [13] is employed to generate all the initial rules. This function generates one rule for each data example, using the fuzzy labels having maximum membership degrees, in parallel and without computation overhead. On the other hand, the `Reduce` function is meant to aggregate the information, solve the possible conflicts, and determine the single final rule of a cell.

The examples contained in a certain cell may belong to several classes, creating the same-antecedent different-consequent rules in that cell. These rules are in a clear conflict that must be resolved along the `Reduce` process. For this purpose, we utilize the confidence metric to evaluate conflicting rules. This is a straightforward way to quantify the rules' strength, as it refers to "how sure" is a rule with respect to its consequent value (class label). In this way, that rule which has the highest confidence is considered to be the most reliable one and is chosen as the final rule of that cell in conflict. Due to the need for scalability in the Big Data context, the values of rules' confidence and support are obtained using the computationally efficient schemes shown in (3) and (4).

For each generated rule, the values of rule's weight, as well as rule's prototype, are similarly computed to supply the next stages of IFC-BD. Regarding the RWs, those procedures developed based on iterative processing of all the data examples will be completely inefficient in the face of large amounts of data, and therefore they should be avoided in the Big Data algorithms. For this reason, we propose to take advantage of the rule's confidence (3) as the RW too, i.e., the RW of the j-th rule is computed as follows:

$$\mathrm{RW}_j = \mathrm{Conf}(R_j) \tag{7}$$

This RW is an approximate value showing the reliability of a certain rule in a rule set. It does not directly deal with the data samples and consequently reduces the computation time than the fuzzy confidence or the other RW heuristics [31].

Finally, the prototype of each rule is computed using the arithmetic mean of its corresponding examples. The idea behind this is to fuse the supporting examples of each rule and providing a measure to represent those examples approximately. The prototype of the j-th rule is $\mathrm{Pr}_j = \left(p_j^1, ..., p_j^n\right)$, a vector with the same dimension as the input examples. Each component of this prototype is calculated as follows [32]:

$$p_j^f = \frac{\sum_{k=1}^{|\mathrm{D}_j|} x_k^f}{|\mathrm{D}_j|} : f = 1, ..., n \tag{8}$$

where $\mathrm{D}_j = \left\{X_1, ..., X_{|\mathrm{D}_j|}\right\}$,and it contains those examples from dataset D that generate the j-th rule. It is clear that these examples and this rule belong to the same cell.

These measures are computed alongside the `Map` operation over the initial RB, where there may be several repetitive and conflicting rules. Same structure rules might be produced by different executing nodes (over different chunks of data) and the values associated with the measures are partial, needing to be globally aggregated through the `Reduce` function.

The support and confidence values are globally aggregated by simple summations over all the same structure rules, where technically the counts for the number of covered examples and their class labels can be easily obtained. Referring to this information, the values of RWs and prototypes are straightforwardly computable. The former is directly equal to the confidence of each rule and the latter is obtained using the actual arithmetic mean of all the corresponding examples of each rule.

3) Dynamic Rule Filtering (DRF): In Big Data scenarios, where the number of cells may increase exponentially, the aforementioned rule learning method can lead to obtaining high number of fuzzy rules. As such, many of these rules are likely to be related to the outliers or non-dense areas. They may pose several challenges to the system behavior, e.g. the computation overheads, increasing running times, loss of the interpretability, or even hindering the predictive performance, among others. We consider several special arrangements to overcome these problems in this study. The first effort is applying a Dynamic Rule Filtering (DRF) approach [19] to get rid of low support rules.

DRF evaluates the interestingness of the rules with respect to the density of their covering examples. In this regard, the cell density and the measure of support, which calculates the number of examples leading to the creation of each rule, are focused as two interrelated concepts. That is, low support rules are related to the low dense areas and are not truly vital to model the problem under study. Therefore, by determining the minimum support requirements, we can discover these rules and remove them from the RB. To this end, DRF defines the threshold of $\mathrm{MinSupp}$.

$\mathrm{MinSupp}$ is indeed the minimum number of required examples to consider a cell as a dense area. This value is obtained based on the pigeon hole principle and determines that those rules supported by less than average of examples must be filtered and eliminated from the RB. $\mathrm{MinSupp}$ is defined for dataset D as follows:

$$\mathrm{MinSupp} = \lfloor \frac{|\mathrm{D}| - 1}{|\mathrm{RB}|} \rfloor \tag{9}$$

where $|\mathrm{D}|$ is the total number of examples, and $|\mathrm{RB}|$ is the size of the RB created through the initial rule learning step (Section III-A2). This RB is examined using $\mathrm{MinSupp}$ so that those rules do not satisfy $\mathrm{MinSupp}$ are eliminated and $\mathrm{RB_{DRF}}$ is obtained as follows:

$$\mathrm{RB_{DRF}} = \{R_i \in \mathrm{RB} \,|\, \mathrm{Supp}\,(R_i) > \mathrm{MinSupp}\} \tag{10}$$

where $\mathrm{Supp}\,(R_i)$ is the support of the i-th rule as (4). The premise of this threshold is to ensure that several rules will remain after applying the DRF procedure, even in the worst scenario where all the examples have been evenly divided within the cells. Furthermore, this threshold is a dynamic / adaptive one defined for each case study, avoiding fixed and user-defined values that usually found by trial and error and are not practical and efficient in the Big Data application.

B. Stage 2: Rule Generalization

The rules generated in the previous stage compose a valid and probably accurate FRBCS. However, we must acknowledge that due to the contribution of all the input dimensions in the structure of rules, the length of these rules is longer than ideal and they are quite specific. In other words, the cell corresponding to each rule is the intersection of all the dimensions, leading to covering and explaining a very small region of the input space. This happens while many of the current rules have a lot in common in their antecedent part. Focusing on such similarities, the specific rules can be horizontally summarized to provide rules covering more general (wider) areas of the input space. In this context, a generalization process is presented along this section, assisting in a fewer number of low-length rules and promoting the whole interpretability of the FRBCS.

The working procedure of the generalization stage is mainly inspired by the concepts of ARM and presented in the three steps. First, all the general rules are derived from the available specific rules through an ARM-learning scheme. Then, the obtained general rules are aggregated and the possible rules' conflicts are modified using a novel estimated fuzzy RW.

Finally, some of the low-confident rules are discarded for the sake of further interpretability improvement. In the following, these steps are detailed.

1) ARM-Learning: Every single specific rule can produce several general rules by mapping to a lower-dimensional space, i.e., if an n-dimensional rule is transformed into the l-dimensional space ($l <= n$), several general rules with the length of l will be obtained. Formally, given the specific rule R_j with the structure similar to (2), the general rules are made as follows:

$$\begin{aligned}\mathrm{RB}_{\mathrm{ARM}} = \big\{ R_i \,|\, \forall R_j \in \mathrm{RB}_{\mathrm{DRF}}\,,\, c_i = c_j\,, \\ A_i \subseteq A_j\,,\, |A_i| = l : l = 1, ..., \mathrm{MaxLen} \big\}\end{aligned} \tag{11}$$

where the antecedent set A_j is considered as the itemset and all of its subsets having cardinality l are extracted to form the general rules like R_i. These rules are made using the same class label of R_j, and create the obtained RB of this step as $\mathrm{RB}_{\mathrm{ARM}}$. Regarding the priorities of XAI in the fuzzy modeling (see Section II-B), which we are looking for the shorter rules, a user-defined threshold as $\mathrm{MaxLen} <= n$ is applied in this step. MaxLen specifies the maximum cardinality of the antecedent subset and is equivalent to the maximum length of the general rules. To accommodate some similar studies [17], [22] as well as the empirical trials, MaxLen has been set to 3 in this work. This limitation indirectly assists to obtain less number of rules, as well.

After mapping all the specific rules, the obtained general rules must be aggregated (Reduced) and their corresponding measures including support and prototype be updated for the subsequent refinements and/or improvements, if necessary. In the actual circumstances, those specific rules that include the same antecedent labels can produce equivalent general rules. Indeed, they are the producer origins of a certain general rule. In these cases, the measures must be updated considering the values of all the origins, e.g., the support value of an obtained general rule is re-computed by adding the support values of its origins. Additionally, since we have the number of corresponding examples of each specific rule (support) as well as their arithmetic mean (see (8)), the actual prototype of all the covering examples related to each general rule is calculable. According to these statements, no approximation is applied in updating the information of the general rules, helping to provide accurate measures and consequently the robustness of the proposed method.

To better understand the above descriptions, an illustrative example has been provided in Table I. Suppose that three specific rules, namely R_1, R_2, and R_3 are available in $\mathrm{RB}_{\mathrm{DRF}}$. All these rules have class label c_1. An ARM-learning process with $\mathrm{MaxLen} = 1$ is applied on these 3-dimensional rules. The top rows of Table I show the specific rules and the bottom ones indicate the final obtained general rules, each indexed by its origin(s). As can be seen, five new 1-dimensional general rules are extracted from the three 3-dimensional ones. The support values of these rules are computed using the summation of the origin values, shown in the last column. Since the other information of these rules depends on the distribution of data examples, we have not included them, here. This example clarifies that the ARM-learning process is indeed a kind of unification of those cells containing similar information.

TABLE I
A SIMPLE EXAMPLE OF THE GENERALIZATION PROCESS AND DIRECT SUPPORT COMPUTATION

Rule	X1	X2	X3	class	Supp
R_1	A_2	A_2	A_3	c_1	5
R_2	A_3	A_2	A_1	c_1	11
R_3	A_3	A_2	A_3	c_1	7
R_1	A_2	-	-	c_1	5
$R_{2,3}$	A_3	-	-	c_1	18
$R_{1,2,3}$	-	A_2	-	c_1	23
$R_{1,3}$	-	-	A_3	c_1	12
R_2	-	-	A_1	c_1	11

2) Rules' Conflict Modification: Imagine that in the above example, we have another specific rule as R_4 : $\{A_2, A_2, A_2\} \rightarrow c2$. If MaxLen is considered 2, this rule makes a general rule as $\{A_2, A_2, -\} \rightarrow c2$ that conflicts with the rule obtained by R_1. As Fig. 2 shows, both of these rules belong to the same cell while having different class labels.

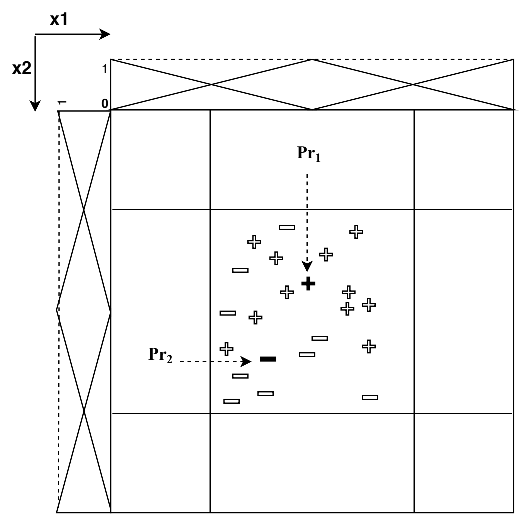

Fig. 2. Rules' conflict schema.

Such a situation may occur repeatedly in many cells of the input space, wherein a conflict resolution strategy must be applied to find out the best rule. However, since in this step, rules are so general and cover a very larger amount of examples, the crisp measures of Section II-A are not profitable here and the distribution of examples must be taken into account. Indeed, with the standard support, the actual power of the "incremental" / "variable" degree of coverage of fuzzy variables is lost, and therefore the intrinsic advantages especially related to the overlapping among rules are also lost.

In this context, some methods like Penalized Certainty Factor (PCF) [31] are available in the literature to exploit information of all the examples and present fuzzy measures such as RW. However, these methods are so computational to be employed in all the stages of a Big Data algorithm. As a solution to these problems, we come up with a procedure that is able to approximate the values in a very efficient and accurate way, i.e., an estimated fuzzy RW based on the idea of PCF is proposed by following the next formulation. This RW can be employed in the Big Data analytics frameworks to resolve the rules' conflict in a fast and reliable way.

Suppose two obtained general rules as follows:

$$\begin{cases} R_i : A_i \rightarrow c_i & : \mathrm{Pr}_i, \mathrm{RW}_i^e \\ R_j : A_j \rightarrow c_j & : \mathrm{Pr}_j, \mathrm{RW}_j^e \end{cases} \tag{12}$$

in which $A_i = A_j$ and $c_i \neq c_j$, indicating these rules are in conflict. According to these information, the weight of rule i is estimated as follows:

$$\mathrm{RW}_i^e = \frac{\mathrm{matchClass}_i^e - \mathrm{matchNotClass}_i^e}{\mathrm{matchClass}_i^e + \mathrm{matchNotClass}_i^e} \tag{13}$$

where $\mathrm{matchClass}_i^e$ is the estimated degree of matching between rule i and all the examples whose class labels are consistent with this particular rule. Due to the fact that the prototypes of the conflicting rules represent the examples of the conflicting classes, they are utilized to avoid iterating over all the examples and costly computations. That is, the prototype of rule i (Pr_i) is employed as the representative of the consistent examples to compute $\mathrm{matchClass}_i^e$ as follows:

$$\mathrm{matchClass}_i^e = \mu_{A_i}(\mathrm{Pr}_i) \tag{14}$$

Similarly, $\mathrm{matchNotClass}_i^e$ is the same measure but for inconsistent examples, namely those examples that are not in the class of rule i. This measure is estimated with the prototype of the other class[1] (Pr_j) as the representative of inconsistent examples as follows:

$$\mathrm{matchNotClass}_i^e = \mu_{A_i}(\mathrm{Pr}_j) \tag{15}$$

We must recall that the matching degree of an example/prototype with a typical rule is computed using (6). In the case of general rules, those antecedents that are not present in the rule's structure, are marked as don't care, having the membership value of 1.

Finally, the estimated RWs of the conflicting rules are considered to assign the class consequent to the one with the highest value. In the case of a tie, the rule related to the majority class is chosen.

3) Rule Base Reduction: Up to this step, the obtained RB contains many general rules which much probably overlap among each other. This implies the existence of redundant and/or ineffective rules, especially due to the smooth coverage of the fuzzy representation. Therefore, we must focus on the most reliable rules to be promoted in the RB, guaranteeing the predictive ability of the fuzzy classifier, while boosting the interpretability properties. For this matter and also following the efforts towards the RB refinement, a filtering process is carried out in this step, aiming at discarding less confident rules.

To do so, all the rules are first categorized with respect to their class labels. They are then sorted in descending order of their confidence values, and finally, the top $\alpha\%$ of each class are only kept in the RB and the others are discarded.

[1]In the case of multi-class problems, this value is estimated using a summation over the prototypes of all the remaining classes.

C. Stage 3: Heuristic Rule Selection

As the last stage of IFC-BD, a rule selection process is carried out to improve the compactness of the RB from the vertical view. This process is heuristically developed using a novel performance measure introducing for every single rule.

A given rule is considered $P\%$ well-performing if it can properly classify $P\%$ of its own covering examples. P is indeed a criterion measuring the rules' performance with respect to the available training examples. It is defined for rule i as follows:

$$P(R_i) = \frac{\sigma_{\mathrm{well-classified}}(i)}{\sigma_{\mathrm{covered}}(i)} \times 100 \tag{16}$$

where $\sigma_{\mathrm{well-classified}}(i)$ is the number of examples that are well classified by rule i (see (5)), and $\sigma_{\mathrm{covered}}(i)$ is the number of examples that are generally covered by this rule. The latter includes both well-classified and miss-classified examples (see (6)).

When the RB of the previous stage ($\mathrm{RB}_{\mathrm{General}}$) obtained, the rules are examined in a class-wise way and those satisfying a minimum performance threshold are selected to be in the final RB ($\mathrm{RB}_{\mathrm{IFC}}$):

$$\mathrm{RB}_{\mathrm{IFC}} = \big\{ R_i \in \mathrm{RB}_{\mathrm{General}} \,|\, P(R_i) > \beta, \\ \forall c_k \in C,\, c_i = c_k \big\} \tag{17}$$

in which β is a user-defined threshold to provide different trade-offs between model performance and system interpretability. In the beginning, it sets equally for all the classes. However, if it would be too high for a certain class so that all the rules of that class are removed, it is automatically decreased by the rate of 0.1, until the matter is resolved. In this way, it is ensured that the final RB is constructed with the top well-performing rules from every single class present in the initial dataset.

IV. Experimental Study

Along this section, the empirical experiments conducted to assess the performance of IFC-BD are detailed. At first, the experimental framework including the datasets used, the evaluation criteria, the cluster-server configuration, and the methods and parameter setup are described in Section IV-A. Next, the performance of IFC-BD is compared with several state-of-the-art Big Data fuzzy classifiers from different perspectives of accuracy and complexity in Section IV-B. Finally, the running times of the algorithms are provided in Section IV-C. We have also provided some additional evaluations with respect to the scalability issues in the supplementary material accompanying this paper.

A. Experimental Framework

In this study, we employed 9 Big Data classification problems with different numbers of examples and input variables as indicated in Table II. All the datasets were selected from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3049911, IEEE Transactions on Fuzzy Systems

the UCI[2] [33] and OpenML[3] [34] dataset repositories. Although these datasets are for binary classification problems, the proposed algorithm is a general solution and can be easily adapted to be applied in multi-class problems. The training and test data of the experiments were generated using the 5-fold cross-validation mechanism, and therefore, final results of each method were computed using the average of five trails performing on the five obtained folds.

TABLE II
PROPERTIES OF THE USED DATASETS

Dataset	Abbr.	#Samples	#Features	#Classes
susy	susy	5,000,000	18	2
BNG-heart	BNG-h	1,000,000	13	2
BNG-Australian	BNG-Au	1,000,000	14	2
covtype1	cov1	581,012	54	2
covtype2	cov2	581,012	54	2
covtype1-vs-2	cov1-vs-2	495,173	54	2
higgs	higgs	11,000,000	28	2
skin	skin	245,056	3	2
hyperplane	hyp	1,000,000	10	2

The proposed method was developed using Spark framework and its ML APIs (MLlib), written in Scala language. Technically, the working procedure of IFC-BD was implemented as an ML pipeline so that different stages of the algorithm were matched to the pipeline stages either a Transformer or an Estimator [19], [30]. Furthermore, the training and test data were structured as the Spark DataFrames and fed into the pipeline to learn or classify data examples distributively. On the other hand, in order to obtain a straightforward distributed system, IFC-BD considered the entire input space as a cell-based area and developed the working procedures around those cells. The cell-based operations were a direct and straightforward way to replicate and distribute the same functions across several executing nodes to be performed over the DataFrame's partitions. In this way, it was also linear to aggregate the partial results of each cell. For the sake of having an efficient learning algorithm and reducing the network communication costs, the RBs were broadcasted in the intermediate steps to be shared across the executing nodes, if necessary.

Regarding the execution platform, the models were run on a cluster of nodes, comprising 14 slaves and one master all having the following configuration:

- Processor: Intel Xeon CPU E5645 @ (2.40GHz) x2,
- Cores: 12 threads (6 cores),
- Main memory: 96GB,
- Cache: 12 MB,
- Network: 40 Gb/s Infiniband,
- Operative System: CentOS 6.9,
- HDFS: Version 2.6.0-CDH5.8.0,

In the case of evaluation criteria, two perspectives of interpretability and accuracy are considered in the following tables. The three measures of #Rules, $\overline{\mathrm{RL}}$, and $\overline{\mathrm{TRL}}$ are used to analyze the interpretability of the fuzzy systems. In a final generated RB, #Rules shows the number of available rules and $\overline{\mathrm{RL}}$ is the average length of these rules. Moreover, $\overline{\mathrm{TRL}}$ is the total length of all the rules that is calculated by multiplication of #Rules, $\overline{\mathrm{RL}}$, and $\overline{\mathrm{FS}}$, where $\overline{\mathrm{FS}}$ is the average number of fuzzy sets per variable. Regarding the accuracy perspective, the measure of ACC is calculated to assess the performance of different classifiers as follows:

$$\mathrm{ACC} = \frac{TP + TN}{TP + FP + TN + FN} \tag{18}$$

where $TP, TN, FP,$ and FN are the values of True Positive, True Negative, False Positive, and False Negative present in the confusion matrix, respectively [35].

Throughout the experiments of IFC-BD, we considered 3 fuzzy labels per variable to ensure generating as few as possible rules in the initial stage as well as preserving the semantic comprehensibility at the level of DB. Furthermore, the values of confidence threshold α (see Section III-B3) and performance threshold β (see (17)) were set at 25% and 70%, respectively. However, for the sake of completeness, we have provided some additional evaluations by varying these two parameters in the supplementary material.

In the case of comparing methods, Chi-BD-DRF [19] and CFM-BD [17] were employed in the experiments. Chi-BD-DRF is the baseline contribution of this study, focusing on generating a fast yet accurate FRBCS, as described in Section III-A. On the other hand, CFM-BD is the current state-of-the-art fuzzy classifier recently proposed for Big Data problems. Because of the training times and avoiding the CHC evolutionary algorithm that is computationally highly expensive in the case of big datasets, we considered the lightweight version of this algorithm to benefit the models in tolerable execution times and performing the comparisons of this section.

In all the versions of CFM-BD, while γ was kept at two[4], the other parameters were assigned as suggested in the original paper [17]. Regarding the number of fuzzy sets used per variable, it was set at 3 and 5, in which their corresponding models were denoted as CFM-BD^3 and CFM-BD^5, respectively. While the former was determined similar to the proposed method with the aim of providing a fairer framework (especially for the complexity comparisons), the latter was set following the original method.

In order to provide more comprehensive comparisons, several statistical tests including Friedman's and Wilcoxon's have been utilized in this study [36]. Friedman's test is employed to rank the algorithms considering a certain criterion and all the datasets. This test first evaluates the equality hypothesis (H_0) of all the algorithms, which can be accepted or rejected. On the other hand, Wilcoxon's test compares two specific algorithms versus each other. It calculates the measures of $\mathrm{p-value}$, R^+, and R^- and decides about the acceptance/rejection of the hypothesis as well as the differences between the intended algorithms. These assessments are performed using the significance level parameter (α) that set at 0.1 in all the following tests of this study.

[2]http://archive.ics.uci.edu/ml

[3]https://www.openml.org/search?type=data

[4]It provides the best configuration of CFM-BD with respect to the interpretability perspective, although, low values of γ might result in a decrease of the classification performance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3049911, IEEE Transactions on Fuzzy Systems

TABLE III
RESULTS OF BOTH PERSPECTIVES

Algorithm	CFM-BD^3				CFM-BD^5				Chi-BD-DRF				IFC-BD			
Dataset	#Rules	$\overline{\text{RL}}$	$\overline{\text{TRL}}$	ACC	#Rules	$\overline{\text{RL}}$	$\overline{\text{TRL}}$	ACC	#Rules	$\overline{\text{RL}}$	$\overline{\text{TRL}}$	ACC	#Rules	$\overline{\text{RL}}$	$\overline{\text{TRL}}$	ACC
susy	200.4	2.4	1444	**75.3**	330.4	2.4	3971	73.3	1002.6	18	54140	66.0	**55.6**	2.9	**490**	69.6
BNG-h	**144.2**	2.4	**1045**	84.0	234	2.4	2860	83.9	3230.8	13	126001	**86.4**	157.8	3.0	1398	86.3
BNG-Au	144	2.5	1089	82.0	239.4	2.5	3005	84.2	910.8	14	38254	84.5	**54.6**	2.8	**459**	**84.8**
cov1	443.4	2.5	3361	73.4	738.6	2.6	9477	73.6	1408.4	54	228161	**77.4**	**206.4**	3.0	**1850**	73.9
cov2	436	2.6	3402	72.7	740.8	2.6	9622	74.5	1393	54	225666	**75.6**	**248**	3.0	**2228**	70.3
cov1-vs-2	464.2	2.6	3646	75.0	740	2.6	9632	75.8	1216	54	196992	**76.1**	**188.6**	3.0	**1689**	75.7
higgs	**220.2**	2.7	**1780**	60.8	425.6	2.6	5556	**61.7**	77649.7	28	6522572	57.4	222.6	3.0	1975	56.3
skin	20.4	2.1	127	91.9	47.8	2.2	530	**94.4**	**5**	3	45	89.9	**5**	1.6	**24**	91.9
hyp	102	2.5	774	66.4	172	2.5	2170	66.1	10483.2	10	314496	**67.5**	**81.2**	3.0	**725**	67.3
Avg	241.64	2.5	1852	75.7	407.62	2.5	5203	**76.4**	10811.05	28	856259	75.6	**135.53**	2.8	**1204**	75.1

B. Evaluations: Interpretability-Accuracy Trade-offs

Throughout this section, the capabilities of different models in terms of system interpretability and discrimination performance are analyzed. To this end, the measures of the number of rules (#Rules), the average rule length ($\overline{\text{RL}}$), and the total rule length ($\overline{\text{TRL}}$) are considered from the interpretability perspective. Whereas the former is meant to evaluate the RB compactness, the latter assesses the overall complexity considering both RB and DB. On the other perspective, the performance of each model is estimated using ACC as (18). The values of these measures have been presented in Table III, where one column has been allocated to each algorithm and the best value of each measure has been highlighted in bold for every single dataset.

Starting from the complexity evaluation, it can be observed the values of #Rules and $\overline{\text{TRL}}$ in Chi-BD-DRF are considerably higher than the other methods. Indeed, this method sacrifices the system interpretability by using a very large number of long rules, leading to a complex system cognitively arduous to be interpreted. Thus, the main comparison of this perspective is between IFC-BD and CFM-BD.

According to Table III, IFC-BD achieved the lowest #Rules values in the majority of the datasets, namely 7 out of 9, and there are only two datasets (BNG-h and higgs) in which CFM-BD^3 obtained slightly better values, but apparently with no significant differences. The success of IFC-BD is highlighted when a similar functionality is observed for the measure of $\overline{\text{TRL}}$, as well. Given these results as well as the overall averages indicated in the last row, it is argued that IFC-BD most likely fulfills the terms of interpretability better than the other models. To pursue this finding, we performed Friedman's test for #Rules and $\overline{\text{TRL}}$.

Table IV provides the results of this test, thereby IFC-BD attained the top-ranking in both complexity criteria, followed by the CFM-BD^3, CFM-BD^5, and Chi-BD-DRF. The rank values of this table imply that the differences between IFC-BD and the third and the fourth-ranked methods are truly high. But, the level of difference between IFC-BD and the second method (CFM-BD^3) should be investigated. For this purpose, we performed Wilcoxon's test that compared IFC-BD against CFM-BD^3 on #Rules and $\overline{\text{TRL}}$. Results of these tests, provided in Table V, reveal that both null hypotheses are rejected in favor of IFC-BD, meaning that this method is considerably superior to CFM-BD^3 in both complexity criteria. According to all these results, IFC-BD could significantly improve the RB compactness and the DB comprehensibility, leading to more simple and explainable systems, among all.

TABLE IV
RESULTS OF RANKING BY FRIEDMAN'S TEST FOR THE COMPLEXITY MEASURES

Algorithm	#Rules	$\overline{\text{TRL}}$
IFC-BD	**1.3**	**1.2**
CFM-BD^3	1.9	1.9
CFM-BD^5	3.1	3.1
Chi-BD-DRF	3.7	3.8
p-value	1.00E-05	1.00E-05
Result	H_0 is rejected	H_0 is rejected

TABLE V
RESULTS OF WILCOXON'S TEST FOR THE COMPLEXITY MEASURES

Comparison	Measure	p-value	R^+	R^-	Result
IFC-BD vs. CFM-BD^3	#Rules	0.021	42	3	H_0 is rejected
IFC-BD vs. CFM-BD^3	$\overline{\text{TRL}}$	0.066	38	7	H_0 is rejected

In the case of accuracy, Table III indicates that the values are close to each other in the majority of the datasets, i.e., in some cases like BNG-Au and hyp, the ACC differences are less than 2%. However, the best values are mostly related to Chi-BD-DRF and the highest average belongs to CFM-BD^5, although with a very low variant diversity, around 1% in the worst case. Therefore, it seems that all the methods perform similarly well but Chi-BD-DRF and CFM-BD^5 operate slightly better than the others. To substantiate this finding, we carried out a Friedman's test as Table VI. Given the results of this test, the null hypothesis related to the equality of all the models is not rejected, implying that these models are not statistically different in the accuracy perspective. The rank values of Table VI implicitly confirm this issue, where the values are in close proximity to each other[5]. Nevertheless, we have to take this into account Chi-BD-DRF applies many numbers of long rules to yield the best ACC ranking and CFM-BD^5 requires a more complex system composed of more fuzzy sets to achieve this classification performance.

TABLE VI
RESULTS OF RANKING BY FRIEDMAN'S TEST FOR ACCURACY

Algorithm	ACC
Chi-BD-DRF	2.0
CFM-BD^5	2.4
IFC-BD	2.7
CFM-BD^3	2.9
p-value	0.53
Result	H_0 is not rejected

Despite the above results, it is noteworthy that CFM-BD has an accuracy-oriented pre-processing stage which transforms the input values and changes the shape of fuzzy sets. These transformations are not in the direction of XAI and influence the semantic of linguistic fuzzy labels and consequently linguistic fuzzy rules. On the other hand, as its authors stated [17], this stage has an important role in the efficiency and robustness of the model so that by omitting this stage, the accuracy values would probably decrease.

To summarize, Fig. 3 illustrates the interpretability-accuracy trade-off provided by the average results of the main comparing methods, in which we have shown $\overline{\mathrm{TRL}}$, as a sign of overall complexity, versus ACC. As depicted, IFC-BD stands out from the rest, obtaining by far the highest interpretability level with a very similar accuracy performance. Summing up, practitioners must generally regard to the models' trade-offs, and due to the competitive predictive performances of these models and the actual differences in the interpretability issues, they must base the selection of the final solutions on how easily understandable they are, as evaluated at the beginning of this section, allowing to provide a proper explanation for the phenomena under study.

C. Running Times

Table VII presents the execution times of all the algorithms. These times are related to the whole learning and classifying procedures. In the case of IFC-BD, the maximum times are consumed by the covtype problems, where the number of features is considerably high, and both more initial specific rules and general rules are generated. In general, the IFC-BD running times are highly dependent on the number of general rules that are passed to the rule selection stage. Indeed, by setting an appropriate α threshold and removing most of the less confident rules in advance (see Section III-B3), the running times would be perfectly tolerable, e.g. for higgs, a truly big dataset, the results were obtained in less than 10 minutes, in contrast to the remaining methods that need up to 4 times additional time.

[5]The system performance has also been evaluated using the Geometric Mean (GM) criterion in the supplementary material, which confirms the results are comparable and good enough considering all the class labels as well.

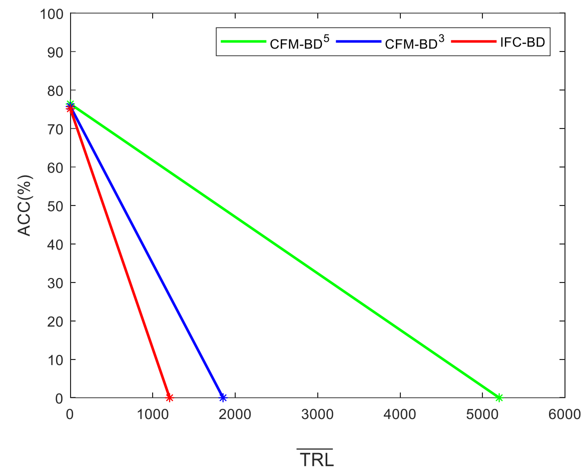

Fig. 3. The models' trade-off: $\overline{\mathrm{TRL}}$ vs. ACC.

The analysis of computational complexity includes two main calculation procedures, namely initial rule learning and RW computation. The time complexity associated with the former is $\mathcal{O}(n/N.d)$, where n is the number of data samples, d is the number of input dimensions, and N is the number of data partitions. For the RW computation, it is $\mathcal{O}(m/N.c.p.l)$, where l is the maximum length of the general rules, m is the number of cells for a given dimension l (See 11), c is the number of class labels, and p is the number of prototypes within a cell (Refer to Fig.4 in the supplementary material of this paper).

TABLE VII
RESULTS OF RUNNING TIME (HH:MM:SS)

Dataset	CFM-BD^3	CFM-BD^5	Chi-BD-DRF	IFC-BD
susy	0:01:21	0:07:52	0:01:37	0:01:03
BNG-h	0:00:46	0:00:43	0:04:03	0:00:50
BNG-Au	0:00:29	0:00:40	0:01:29	0:00:39
cov1	0:02:36	0:06:26	0:02:40	0:16:57
cov2	0:02:42	0:06:58	0:01:41	0:20:15
cov1-vs-2	0:02:42	0:07:39	0:01:38	0:16:45
higgs	0:37:59	0:32:46	1:07:42	0:08:54
skin	0:00:09	0:00:05	0:00:11	0:00:26
hyp	0:00:15	0:00:36	0:05:26	0:00:38
Avg	0:05:27	0:07:05	0:09:36	0:07:23

V. CONCLUSION

This study proposed, IFC-BD, an interpretable FRBCS for boosting the tenets of XAI in the Big Data scenarios. Basically, IFC-BD generated an initial RB over a cell-based area and attempted to compact it through the three working stages, not only by a vertical viewpoint to achieve a limited number of rules but also by a horizontal one to reduce the rules' length. For the sake of scalability, IFC-BD took advantage of an approximate fuzzy rule weight and a heuristic rule selection method.

The effectiveness of IFC-BD was evaluated using 9 different classification datasets in terms of classification performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2021.3049911, IEEE Transactions on Fuzzy Systems

and complexity measures. The experimental results and the conducted statistical tests revealed that IFC-BD could achieve a straightforward yet accurate fuzzy classifier composed of a compact RB and DB. In the case of RB, obtaining less number of short antecedent rules makes it more manageable and understandable. About DB, using less number of linguistic labels with a straightforward transition helps to provide a semantic knowledge similar to the human cognition. These characteristics allow for having explainable models, which can be conveniently interpreted by the non-expert human users.

As future work, we intend to further investigate the internal information provided by the fuzzy rules being discovered by IFC-BD. Among others, the focus should be on the data density, prototypes, and other additional complexity characteristics, which allows linking fuzzy rules with Smart Data [37], leading to a better understanding for each case study, and promoting the transformation of raw information into high quality knowledge.

ACKNOWLEDGMENT

This work has been partially supported by the Andalusian regional project P18-TP-5035.

REFERENCES

[1] F. Amalina, I. Targio Hashem, Z. Azizul, A. Fong, A. Firdaus, M. Imran, and N. Anuar, "Blending big data analytics: Review on challenges and a recent study," *IEEE Access*, vol. 8, pp. 3629–3645, 2020.

[2] S. Ramírez-Gallego, A. Fernández, S. García, M. Chen, and F. Herrera, "Big data: Tutorial and guidelines on information and process fusion for analytics algorithms with mapreduce," *Inform. Fusion*, vol. 42, pp. 51–61, 2018.

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

[4] 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," *Inform. Fusion*, vol. 58, pp. 82 – 115, 2020.

[5] H. Liu, A. Gegov, and M. Cocea, "Rule-based systems: a granular computing perspective," *Granular Computing*, vol. 1, no. 4, pp. 259–274, 2016.

[6] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining*. Springer-Verlag Berlin Heidelberg, 2004.

[7] 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 Comput. Intell. Mag.*, vol. 14, no. 1, pp. 69–81, 2019.

[8] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Inform. Sciences*, vol. 181, no. 20, pp. 4340–4360, 2011.

[9] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models inform. sciencestead," *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.

[10] A. Fernández, C. J. Carmona, M. J. del Jesus, and F. Herrera, "A view on fuzzy systems for big data: progress and opportunities," *Int. J. Comput. Intell. Syst.*, vol. 9, no. sup1, pp. 69–80, 2016.

[11] P. Ducange, M. Fazzolari, and F. Marcelloni, "An overview of recent distributed algorithms for learning fuzzy models in big data classification," *J. Big Data*, vol. 7, no. 1, pp. 1–29, 2020.

[12] Y. Cui, H. E, W. Pedrycz, and Z. Li, "Designing distributed fuzzy rule-based models," *IEEE Trans. Fuzzy Syst.*, vol. in press, 2020.

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

[14] S. del Rio, V. Lopez, J. M. Benítez, and F. Herrera, "A MapReduce approach to address big data classification problems based on the fusion of linguistic fuzzy rules," *Int. J. Comput. Intell. Syst.*, vol. 8, no. 3, pp. 422–437, 2015.

[15] M. Elkano, M. Galar, J. Sanz, and H. Bustince, "CHI-BD: a fuzzy rule-based classification system for big data classification problems," *Fuzzy Set. Syst.*, vol. 348, pp. 75–101, 2018.

[16] L. Íñiguez, M. Galar, and A. Fernández, "Improving fuzzy rule based classification systems in big data via support-based filtering," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2018, pp. 1–8.

[17] M. Elkano, J. A. Sanz, E. Barrenechea, H. Bustince, and M. Galar, "CFM-BD: A distributed rule induction algorithm for building compact fuzzy models in big data classification problems," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 1, pp. 163–177, 2020.

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

[19] F. Aghaeipoor, M. M. Javidi, I. Triguero, and A. Fernández, "Chi-BD-DRF: Design of scalable fuzzy classifiers for big data via a dynamic rule filtering approach," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2020, pp. 1–7.

[20] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin *et al.*, "Apache spark: a unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[21] N. Bhargava and M. Shukla, "Survey of interestingness measures for association rules mining: data mining, data science for business perspective," *IJCSITS*, vol. 6, no. 2, 2016.

[22] J. Alcala-Fdez, R. Alcala, and F. Herrera, "A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 5, pp. 857–872, 2011.

[23] O. Cordón, M. J. Del Jesus, and F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," *Int. J. Approx. Reason.*, vol. 20, no. 1, pp. 21–45, 1999.

[24] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, 2018.

[25] M. Islam, D. T. Anderson, A. J. Pinar, T. C. Havens, G. Scott, and J. M. Keller, "Enabling explainable fusion in deep learning with fuzzy integral neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1291–1300, 2020.

[26] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Inform. Sciences*, vol. 181, no. 20, pp. 4340–4360, 2011.

[27] J. Casillas, O. Cordón, F. H. Triguero, and L. Magdalena, *Interpretability issues in fuzzy modeling*. Springer, 2013, vol. 128.

[28] J. Maillo, S. García, J. Luengo, F. Herrera, and I. Triguero, "Fast and scalable approaches to accelerate the fuzzy k-nearest neighbors classifier for big data," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 5, pp. 874–886, 2020.

[29] K. Vassakis, E. Petrakis, and I. Kopanakis, "Big data analytics: Applications, prospects and challenges," in *Mobile big data*. Springer, 2018, pp. 3–20.

[30] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen *et al.*, "Mllib: Machine learning in apache spark," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1235–1241, 2016.

[31] H. Ishibuchi and T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 428–435, 2005.

[32] M. Elkano, M. Galar, J. Sanz, and H. Bustince, "CHI-PG: A fast prototype generation algorithm for big data classification problems," *Neurocomputing*, vol. 287, pp. 22–33, 2018.

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

[34] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "Openml: Networked science in machine learning," *SIGKDD Explorations*, vol. 15, no. 2, pp. 49–60, 2013. [Online]. Available: http://doi.acm.org/10.1145/2641190.2641198

[35] A. Fernández, A. Altalhi, S. Alshomrani, and F. Herrera, "Why linguistic fuzzy rule based classification systems perform well in big data applications?" *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, pp. 1211–1225, 2017.

[36] 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." *Inform. Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[37] J. Luengo, D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera, *Big Data Preprocessing: Enabling Smart Data*, 1st, Ed. Springer, 2020.