Applied Intelligence (2026) 56:77
https://doi.org/10.1007/s10489-025-07081-1

Explainable classifier with adaptive optimisation for medical data

José Ramón Trillo[1] · María José Del Moral[2] · Juan Miguel Tapia[3] · Julia García-Cabello[4] · Francisco Javier Cabrerizo[1]

Received: 27 January 2025 / Accepted: 30 December 2025

Abstract
Artificial Intelligence (AI) has become increasingly important in critical domains such as medicine, where accurate and interpretable decision-making is essential. However, many high-performing AI models operate as "black boxes", limiting transparency and making it difficult for clinicians to understand or verify predictions. To address this challenge, we present an eXplainable Artificial Intelligence (XAI) framework that integrates a fuzzy rule-based classifier with genetic algorithms and 2-tuple linguistic representations. The method incrementally generates general fuzzy rules, introduces fuzzy exception rules to capture atypical cases, and applies rule selection and parameter tuning to enhance both accuracy and interpretability. Experiments on nine medical datasets demonstrate that our approach achieves competitive or superior accuracy compared to state-of-the-art algorithms, while requiring fewer rules. These results show that the method not only improves predictive performance but also provides clear, human-readable explanations for each decision, thereby increasing trust and facilitating its application in medical practice.

Keywords EXplainable artificial intelligence · Fuzzy classifier · Genetic algorithms · Fuzzy rules · Medical application

1 Introduction

With the advent of computers in the 1980s, it became possible to perform complex calculations that were previously costly and impractical, leading to the development of early mathematical algorithms—such as decision trees, fuzzy logic, and logistic regression—which have since evolved into core components of the field of Artificial Intelligence (AI) [1, 2].

A well-known recent example of the advancement of AI is ChatGPT [3, 4], an algorithm based on deep learning [5, 6]. These types of algorithms were developed in the 20th century, nonetheless, they did not have a significant boom until the second decade of the 21st century, with the creation of a self-tuning algorithm. This algorithm makes it possible to know which parameters are the most accurate to obtain better accuracy in the classification process. Nevertheless, these methods are black box algorithms, since it is impossible to determine which parameters the program has used to determine why it has made one decision or another [7]. This lack of clarification of the parameters implies that there are problems with these algorithms since they must be trained with a data set that is ethical [8]. An example is that the data set is not biased to certain data that may cause the algorithms to become racist or sexist [9]. Such opacity

María José Del Moral, Juan Miguel Tapia, Julia García-Cabello and Francisco Javier Cabrerizo contributed equally to this work.

✉ José Ramón Trillo
jrtrillo@ugr.es

María José Del Moral
delmoral@ugr.es

Juan Miguel Tapia
jmtaga@ugr.es

Julia García-Cabello
cabello@ugr.es

Francisco Javier Cabrerizo
cabrerizo@decsai.ugr.es

1 Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence, DaSCI, University of Granada, Granada 18071, Spain

2 University of Granada, Granada 18071, Spain

3 Department of Quantitative Methods for Business and Economics, University of Granada, Granada 18071, Spain

4 Department of Applied Mathematics, University of Granada, Granada 18071, Spain

Published online: 06 February 2026

underscores the need for approaches that can both match the accuracy of black-box systems and provide transparent reasoning processes.

While GPT-based models, such as ChatGPT, are highlighted here due to their widespread recognition and accessibility, it is important to acknowledge that the family of large language models (LLMs) is considerably broader, encompassing architectures such as PaLM, LLaMA, Claude, and others. The choice to focus on GPT was made to ensure accessibility and comprehensibility for the intended audience, particularly in the medical context, rather than to imply exclusivity or superiority among LLM architectures. The focus on GPT here serves to illustrate the accessibility and influence of these models; however, the broader imperative in medical AI is to ensure that any high-performing system—regardless of its underlying architecture—can be understood and trusted by practitioners.

To comprehend the rationale behind algorithmic decisions, a novel paradigm known as eXplainable Artificial Intelligence (XAI) has emerged within the field of Artificial Intelligence [10, 11]. This breakthrough addresses the issue of opaque algorithmic decision-making and finds significant application in critical domains such as healthcare, where people's well-being is at stake. To enhance the objective of establishing an XAI framework that not only furnishes result explanations but also maintains competitive accuracy, it is imperative to opt for interpretable algorithms, integrate interpretability into the design process, and explore novel algorithms based on decision trees, Naive Bayes [12], and innovative approaches rooted in fuzzy logic [13]. Within this interpretability-oriented landscape, fuzzy logic has emerged as a particularly promising paradigm.

Fuzzy logic was first researched in the 1960s at the University of Berkeley by engineer Lotfi A. Zadeh [14, 15], who introduced the concept of fuzzy sets. Fuzzy logic differs from classical logic in that it considers the certainty of a proposition to be a matter of degree rather than an absolute statement. It is flexible, tolerant of imprecision capable of modelling non-linear problems, and is based on natural language [16, 17]. Fuzzy control systems, in particular, have been extensively studied and applied across engineering domains [18–20], with well-established methodologies for rule-based decision-making and robustness enhancement.

Although higher-order fuzzy systems, such as Type-2 and Type-3 fuzzy logic, are designed to better capture and model higher degrees of uncertainty and imprecision, they introduce substantial computational costs, increased parameterisation requirements, and, in many cases, reduced interpretability for non-specialist end-users. Given the objectives of this study — namely, to provide a framework that is both computationally efficient and easily interpretable in medical contexts — Type-1 fuzzy logic was selected. This choice balances accuracy, transparency, and computational feasibility, while remaining well-suited to the uncertainty levels present in the datasets under consideration. The exploration of higher-order fuzzy systems in conjunction with the proposed methodology is left as an avenue for future work.

Nowadays, algorithms based on fuzzy logic are consolidated algorithms within AI [21, 22], as they have the advantage that they are not only explainable algorithms but also algorithms that do not do so in a categorical way when it comes to classifying an element [23]. An example might be that an algorithm classifies the colour of an object as black and white; in non-fuzzy algorithms, the output will be either black or white, but a fuzzy algorithm will say that the object will be a percentage black and another percentage white.

These algorithms rely on sets of fuzzy rules to determine their output or consequent [24], based on the fulfilment of specific conditions, termed antecedents. Nevertheless, two conflicting issues can arise. Firstly, overly general rules risk being inefficient, as they encompass a broad range of data from different problem classes [25]. Secondly, when rules have an excessive number of antecedent conditions, they become overly specific, leading to the creation of numerous rules that are activated for only a few data points, also resulting in inefficiency [26]. Balancing the specificity and generality of these rules is essential for optimizing algorithm performance.

While numerous studies have addressed classification problems using fuzzy-based methods, including those enhanced by heuristic and metaheuristic optimisation techniques, persistent challenges remain. Prior research has often struggled with excessive rule redundancy, leading to unnecessarily large and complex rule bases that hinder interpretability. Moreover, in high-dimensional datasets, the readability and comprehensibility of fuzzy models tend to deteriorate, limiting their applicability in domains such as medicine, where transparency is crucial. Additionally, systematic approaches for handling exceptional or atypical cases—without sacrificing predictive performance—are largely absent from the literature, resulting in models that fail to capture critical yet infrequent patterns.

In light of these gaps, this study addresses the following research questions: (i) How can exception-handling mechanisms be systematically incorporated into interpretable classifiers without inflating the rule base? (ii) How can adaptive optimisation be leveraged to achieve a balanced trade-off between accuracy and interpretability? (iii) What is the optimal rule granularity that maximises predictive performance while preserving comprehensibility for domain experts? The main contributions of this work are threefold: first, the design of a transparent classifier architecture incorporating exception-handling logic; second, the application of an adaptive selection-and-adjustment strategy to refine both accuracy and interpretability; and third, a rigorous empirical evaluation across nine medical datasets, demonstrating competitive or superior performance compared to state-of-the-art methods.

This paper presents a novel fuzzy logic-based algorithm to address two fundamental challenges. First, it allows the incremental creation of rules by adding antecedents until sufficient reliability is achieved, avoiding the generation of rules with excessive conditions. Subsequently, a complementary set of rules, called exception rules, is generated to avoid excessive specificity. These exception rules coexist with the initial, more reliable rules, but produce different consequences. To further increase accuracy and eliminate possible conflicts between rules, a rule selection and tuning algorithm based on genetic algorithms (GAs) is introduced. Each chromosome of the GA [27] consists of two parts: one part with zeros and ones, which indicates the selection of rules, and another part with values in the range of [-0.5, 0.5], which adjusts the parameters of the rules. The result is a set of general, accurate and tuned rules that offer competitive accuracy results compared to other XAI algorithms [28, 29]. Consequently, we have an XAI algorithm that allows users to understand why one decision is made and not another. Although both fuzzy classification systems and genetic algorithms are established methods, the originality of this work resides in their tailored integration and the methodological innovations introduced. Specifically, the algorithm incorporates an exception-handling mechanism capable of identifying and prioritising rare yet relevant patterns, a feature seldom addressed in existing approaches. This is coupled with an adaptive optimisation procedure that jointly performs rule selection and fine-tunes membership function parameters, ensuring that performance gains do not compromise interpretability. By aligning rule granularity with the dual objectives of accuracy and comprehensibility, the proposed framework achieves a balance rarely attained in the literature, as confirmed by its superior results on multiple medical datasets.

Recent advances in fuzzy-based models for medical applications include fuzzy deep learning frameworks that enhance the handling of uncertain clinical data, as highlighted in [30]. In parallel, [31] provides a comparative review of interpretable fuzzy inference systems, emphasizing the importance of transparent rule structures.

The document is divided into nine sections. In Section 2, the basic concepts of fuzzy logic and GAs are recalled. In Section 3, the datasets used in this study are analyzed to show their characteristics. In Section 4, the design of the algorithm and the steps followed for its creation are described. In Section 5, the results obtained are analyzed. Furthermore, the advantages and disadvantages of the algorithm are discussed. In Section 6, according to the advantages and disadvantages of the algorithm, a plan for future work is presented to strengthen the advantages and solve the disadvantages of the algorithm. Finally, the conclusions of this paper are also drawn according to the results obtained.

2 Preliminaries

In this section, the basic concepts related to the proposed fuzzy rule algorithm are recalled. For this purpose, in Section 2.1, we explain basic concepts related to fuzzy logic, such as the definition of membership rule or the creation of fuzzy rules. In Section 2.2, concepts related to GAs are introduced, such as the definition of chromosomes and the parts that compose such algorithms.

2.1 Fuzzy logic algorithms

The definition of the fuzzy set was given by A. Zadeh in 1965 as follows [14]: "*A fuzzy set A in X is characterized by a membership (characteristic) function* $f_A(x)$ *which associates with each point in X a real number in the interval* [0, 1], *with the value of* $f_A(x)$ *at x representing the "grade of membership" of x in A. Thus, the closer the value of* $f_A(x)$ *to unity, the higher the grade of membership of x in A. When A is a set in the ordinary sense of the term, its membership function can take on only two values* 0 *and* 1, *with* $f_A(x) = 1$ *or* 0 *according as x does or does not belong to A*".

The concept of a fuzzy set was exemplified using the set "tall men" [14, 32]. According to classical logic, this set would consist of men with a height greater than a specific value, such as 1.80 meters. Any man with a height less than this value would be excluded from the set. Thus, a man with a height of 1.81 meters would be considered "tall", while one with a height of 1.79 meters would not. Nonetheless, it is illogical to say that one man is tall and another is not when their height difference is only two centimetres [33]. Furthermore, recent systematic analyses, such as [34], underscore the role of fuzzy logic in improving the interpretability of traditionally black-box machine learning models. Additionally, causal explainability techniques based on fuzzy cognitive maps have been proposed by [35].

In fuzzy logic, each Fuzzy Rule-Based Classification System (FRBCS) consists of two main components [36, 37]: the Knowledge Base (KB), which includes a Data Base (DB) and a Rule Base (RB), and the fuzzy reasoning method that performs the inference process to label new examples. The development of fuzzy controllers has evolved from early heuristic designs to systematic frameworks using linear matrix inequalities and adaptive tuning [38, 39]. These works provide a strong theoretical foundation for the fuzzy rule-based mechanisms adopted in our approach.

In this section, we present these main components of FRBCS using a fuzzy rule learning algorithm based on the [40] grid, which has similar characteristics to the model proposed in this paper. We focus on the two main stages: first, in Section 2.1.1, the definition of the DB for fuzzy data representation in introduced; and second, in Section 2.1.2, the construction of the RB from the input training examples and the initial DB information is shown.

2.1.1 Definition of the database

To enable a fuzzy representation of the problem, we start from the definition of the features of any data set. For this purpose, we consider the number of attributes $n \in \mathbb{N}$ and the number of instances $m \in \mathbb{N}$. Consequently, we can denote an instance as a vector space $x_q = (x_{q1}, \dots, x_{qn})$, where q varies from 1 to m, and x_{qj} is an element or input variable of the j-th attribute in the vector space x_q. Furthermore, we denote the variable y_q as the label of an instance q [41, 42].

This includes enumerating the elements that belong to the set, specifying the properties that the elements belonging to the set must satisfy, or in terms of the membership function $\mu_{A_{pt}}(x_{qj}) \in [0, 1]$.

For the representation of classes, which are the possible outputs that our classification algorithm can have, we define the set of classes as C and use the constant $S \in \mathbb{N}$ to denote the number of classes. Thus, a specific class is represented as $c_p \in C$, where p varies from 1 to S.

The DB consists of a set of T fuzzy variables that allow the representation of the initial values of the data set. From the aforementioned information, the standard structure of the membership functions, which represent fuzzy sets [43], is computed. Furthermore, fuzzy sets can be considered as a generalization of classical sets. While classical set theory only considers the membership or non-membership of an element to a set, fuzzy set theory goes further. In fuzzy set theory, the partial membership of an element to a set is considered. This means that each element has a degree of membership in a fuzzy set, which can take any value between 0 and 1. This degree of membership is defined by the characteristic function associated with the fuzzy set. For each value that an element or input variable can take $x_{qj} \in \mathbb{R}$, is an element or input variable of the j-th attribute in the vector space x_q. Formally, a classical set in a universe of discourse X can be defined in several ways. For each fuzzy label of an attribute q, we define a membership function $\mu_{A_{pt}} \in [0, 1]$ for each $p = 1, \dots, n$ and $t = 1, \dots, T$ and where $A_{pt} \in \mathbb{N}$ is a fuzzy label described numerically, i.e., if we have the fuzzy labels "blonde" and "brunette", their numerically described fuzzy labels would be 0 and 1 respectively, for a fuzzy term of an attribute p [44]. In this paper, we use continuous triangular membership functions, where all membership functions of an attribute p are uniformly distributed in the p universe of discourse, examples can be seen in Figs. 1 and 2. Moreover, these functions satisfy the following property: $\mu_{A_{pt}}(x_{qj}) \in [0, 1]; \ \forall x_{qj} \in x_q$, where q varies from 1 to m. In this study, triangular membership functions were adopted as they offer a straightforward parametrisation and

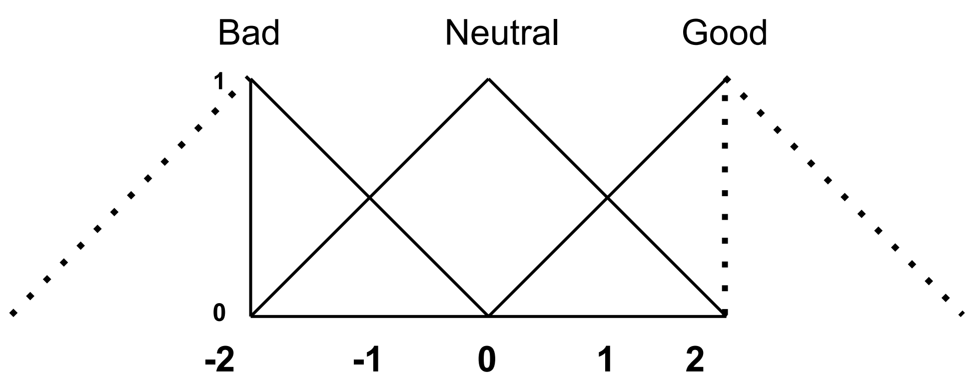


Fig. 1 Example of membership function with 3 fuzzy labels

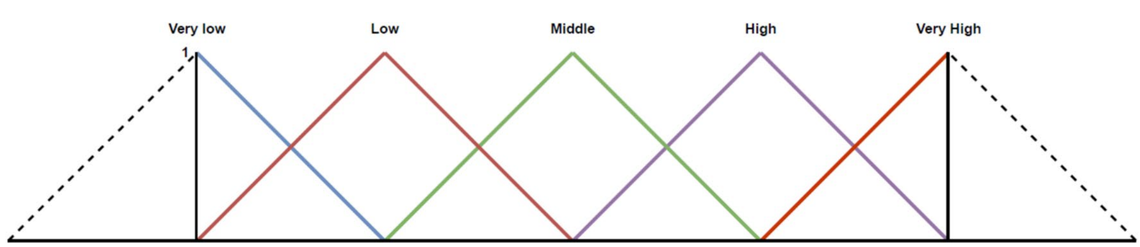


Fig. 2 Example of membership function with 5 fuzzy labels

an intuitive graphical interpretation, facilitating communication with non-technical domain experts such as physicians. Nevertheless, we recognise that alternative shapes, including Gaussian, trapezoidal, or generalized bell membership functions, may provide greater statistical flexibility and potentially capture data distributions more accurately. Exploring these alternatives remains a promising direction for future work. The triangular membership functions adopted in this work were uniformly distributed across each attribute's universe of discourse, with their range defined by the minimum and maximum observed values in the dataset. Equal partitions were applied, using a granularity of 3 for the base rules and 5 for the exception rules. These values were selected based on preliminary testing to achieve an optimal trade-off between model interpretability and predictive performance.

2.1.2 Creation of the fuzzy rule set

Fuzzy rules can be classified into different types according to their internal components, although the most commonly used are those that consist of a vector space of maximum length *n* [45]. Each position in this vector space represents a fuzzy label associated with a particular attribute. Furthermore, each rule includes a consequent that is composed of the weight of the rule, which is the reliability that rule has and is denoted as, $RW_j \in [0,1]$ where *j* refers to the *j*-th rule and C^{RW_j} is the label of the class that determines the corresponding output, i.e., it is an element belonging to the set of classes. This structure allows capturing the relationship between the input variables and the resulting classification in a fuzzy and flexible way.

We assume that we are provided with a set of fuzzy labels of cardinality *T* for each attribute. Thus, we can represent a fuzzy term of an attribute *q* as A_{qt}, where *t* belongs to the set $\{1,\dots,T\}$ and represents the fuzzy index associated with a specific fuzzy label [46]. For each instance, we can generate a candidate fuzzy rule. We can denote a fuzzy label as $a_{qj};\ \forall q = 1,\dots,m;\ j = 1,\dots,n$. Therefore, for each value in the data set x_{qj}, we can assign a corresponding fuzzy label. On the other hand, we define the fuzzy label as $a_{qj} = argmax_{t\in\{1,\dots,T\}}\mu_{A_{qt}}(x_{qj});\ q = 1,\dots,m,\ j = 1,\dots,n$ Having obtained the fuzzy label associated with each value of an instance *q*, we denote the rule associated with each instance, $R_q = (a_{qj};\ j = 1,\dots,n);\ \forall q = 1,\dots,m$.

Applying the above procedure for each of the above instances, we can obtain that several instances have the same rule associated with them. To unify them, we can define a set, RB_T, as the set of rules, R_q, not repeated; the cardinality of this set is denoted by $U \in \mathbb{N}$. To compute the weight of a rule RB_T, $RW_j;\ \forall j = 1,\dots,U$, we initially compute the minimum *t*-rule, $mint_q \in [0,1]$, for each of the instances as $mint_q = \min(\mu_{a_{qj}};\ j = 1,\dots,n);\ \forall q = 1,\dots,m$ [47], this being a single fuzzy membership value [48, 49].

Next, $mint_q$ is used to calculate the degree of association, denoted as K_{q,R_j}, of an instance *p* in a rule $R_j; \in RB_T$. Equation 1 quantifies the class-specific weight of a rule by aggregating the minimum membership degrees of all instances belonging to that class. This ensures that rules are evaluated based on the strength of their association with specific classes. Thus, it is possible to define the weight that each class has, denoted as $c_p \in C; p \in \{1,\dots,S\}$, for a rule R_j as:

$$W_{j,c_p} = \sum_{q=1;\, c_p=y_q}^{m} K_{q,R_j} \tag{1}$$

With the class degree for each class, it is possible to calculate the consequence of the rule (every rule is composed of an antecedent and a consequent), which contains an associated class and a weight. Nonetheless, although it is going to be explained in a general way. For our algorithm, we are going to make a tree for each of the classes. There is only a single c_p for each tree, and it would not be necessary to calculate the maximum since we only have one class. In a general way, to obtain the associated class of each rule, denoted as C^{RW_j}, by calculating the maximum. Therefore, one has that $C^{RW_j} = \{c_{p'} : W_{j,c_{p'}} = max(W_{j,c_p});\ c_{p'}, c_p \in C\}$, which verifies that $\forall R_j \in RB_K\ \exists C^{RW_j} \in C$, with an associated weight, RW_j. Equation 2 normalizes the class weight to yield the final rule reliability RW_j, ensuring comparability across classes with different instance frequencies:

$$RW_j = \frac{W_{j,C^{RW_j}}}{\sum_{p=1}^{S} W_{i,c_p}};\ C^{RW_j} \in C \tag{2}$$

We can see an example in the image [50] (see Fig. 3). An example of a fuzzy rule with a weight equal to 0.75, that is, having a reliability of 75% of the times, and using two attributes, as hair colour (with 0 if dark-haired and with 1 if blond) and eye colour (0 if light-eyed and 1 if dark-eyed), and as consequent if "tall" or "small" can be the following:

$$if(\,(x_{t1}\ is\ \text{“0”})\ and\ (x_{t2}\ is\ \text{“1”}))\rightarrow(\,\text{“}tall\text{”},\ 0.75)$$

This implies that all men who are dark-haired and have dark eyes will be tall, with the rule being reliably 75%.

Generalizing, a *j*-th fuzzy rule with $\beta \in \mathbb{N}$ arguments for a *t*-th element can be defined as:

$$if(\,(x_{t1}\ is\ a_{j1})\ and\ (x_{t2}\ is\ a_{j2})\ and\dots and\ (x_{t\beta}\ is\ a_{j\beta}))\rightarrow(C^{RW_j},\ RW_j) \tag{3}$$

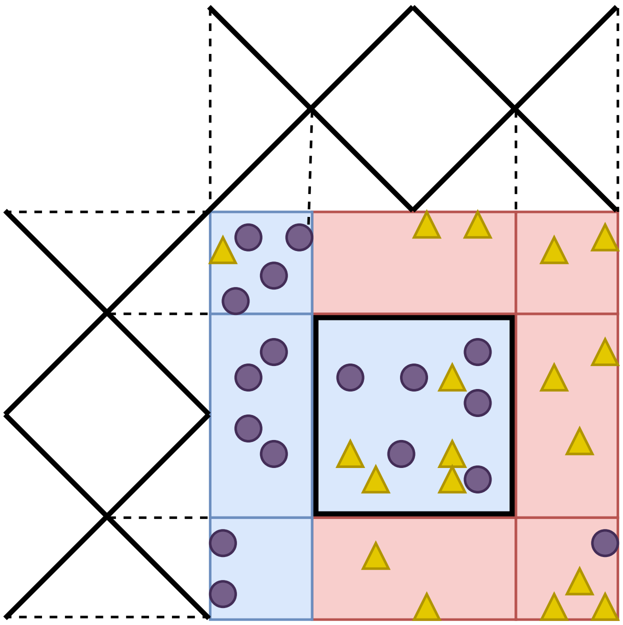


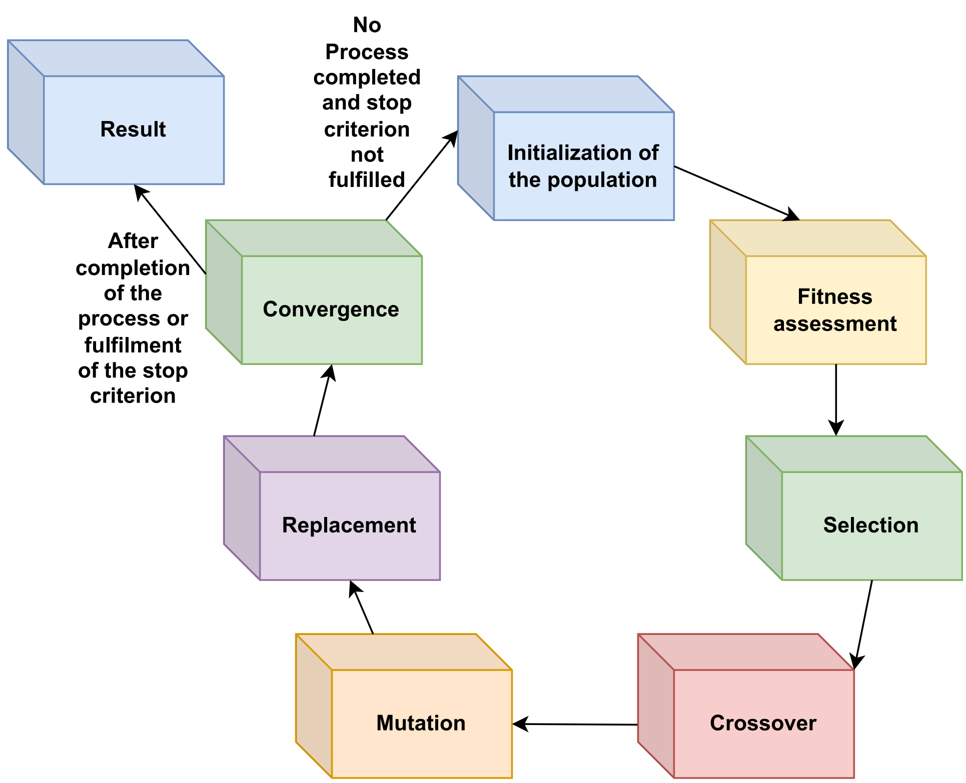
Fig. 3 Example of nine fuzzy rules with 2 classes, and three membership functions per attribute. Pink means that the majority class is the triangle, otherwise it is blue

2.2 Genetic algorithms

GAs are an optimization technique inspired by the theory of biological evolution and are used to find approximate solutions to complex optimization problems [51, 52]. In the following, we explain the basic steps and key concepts involved in GAs (see Fig. 4).

- Coding of individuals. In the coding stage, one must determine how to represent the individuals in the population. This involves defining a structure that represents the solution to the problem. Our case is going to have two parts: the first part is the one that is going to select whether or not to choose the rule. Therefore, it has a valuation $\{0, 1\}$ being 0 the value that does not choose the rule and 1 the value that does choose it [53]; and the second part is the application of the horizontal shift of the membership functions that are not extreme, known as 2-tuples [54].
- Initialization of the population. Once the coding of individuals has been defined, an initial population is created. This involves generating randomly or using some prior knowledge of several individuals that represent possible solutions to the problem. The initial population usually has a fixed size and each individual is generated by assigning random values to the genes.
- Fitness assessment. At this stage, the quality of each individual in the population is evaluated using a fitness function [55]. The fitness function values each relative to the objective of the optimization problem [56]. It can be an objective function to be maximized or minimized. For example, in a profit maximization problem, the fitness function will evaluate the profit obtained by each individual.

Fig. 4 Basic scheme of GA



- Selection. Selection is a process in which the fittest individuals are chosen to reproduce and create the next generation. Individuals with higher fitness are more likely to be selected since it is sought that their characteristics are transmitted to future generations [57, 58].
- Genetic operators. At this stage, genetic operators are applied to selected individuals to create new offspring. The two most common genetic operators are crossover and mutation [59]:

  – Crossover. Crossbreeding simulates sexual reproduction and consists of combining genetic information from two parent individuals to produce one or more offspring [60].
  – Mutation. It consists of making random changes to one or more genes in a selected individual. These changes can be small perturbations or random replacements of gene values [61].

- Replacement. Once the new offspring have been generated through crossing and mutation, some individuals from the previous generation are replaced with the new offspring. The replacement strategy can be generational or based on fitness [62].

  – Generational replacement. In generational replacement, the entire previous population is replaced with the new offspring [63].
  – Fitness-based replacement. In fitness-based replacement, the least fit individuals from the current population are selected and replaced by the new offspring.

- Convergence. The above steps are repeated for a fixed number of generations or until a convergence criterion is met. Convergence occurs when the population reaches an optimal solution or when it stops improving significantly [64, 65].
- Result. Once the convergence criterion is met, the GA is considered to have found an approximate solution to the optimization problem.

3 Analysis of datasets

The creation of predictive models in medicine using datasets is necessary for several reasons [2, 66]:

- Improve diagnostic accuracy. Predictive models can analyze large amounts of clinical and patient data to identify hidden patterns and relationships. This can help clinicians make more accurate diagnoses, especially in complex or rare cases, by providing additional information and clinical decision support [67].
- Personalized prognosis and treatment. Predictive models can help predict disease prognosis and identify which treatments might be most effective for a particular patient [68]. By considering multiple variables and individual characteristics, these models can provide more personalized recommendations and optimize the treatment plan.
- Early detection of disease. Predictive models can help identify early signs of disease or medical conditions before they manifest clinically. This enables timely intervention and preventive or early treatment, which can improve health outcomes and reduce disease progression.
- Efficient resource management. Predictive models can also help optimize the allocation of limited medical resources, such as hospital beds, staff and medical equipment [69]. Adjustments can be made to ensure a more efficient and equitable allocation by predicting future demand and understanding resource utilisation patterns.
- Medical research and knowledge discovery. Predictive models can help identify unknown relationships and risk factors in large, artificial datasets [70]. This can drive medical research and new knowledge discovery, which in turn can lead to the development of better interventions, drugs, and health policies.

It can be concluded that predictive models in medicine use data sets to harness the power of data analytics and Artificial Intelligence to improve diagnostic accuracy, prognosis, personalized treatment, early disease detection and medical resource management while fostering research and medical advancement. For this reason, only medical data sets have been chosen for this algorithm.

Table 1 contains information on several medical datasets. A description of each column in the table is shown below:

Table 1 Table of medical datasets

Dataset Name	Number of Instances	Number of Attributes	Number of Classes	Acronym
Appendicitis	106	7	2	App.
Cleveland	303	13	5	Clev.
Ecoli	336	7	8	Ecoli
Haberman	306	3	2	Hab.
Mammographic	830	5	2	Mam.
Pima Indian	768	8	2	P. Ind.
Wisconsin	699	9	2	Wisc.
Indian Liver Patient Dataset	583	9	2	Ilpd.
Breast Teassure	106	9	4	B. Tea.

- Dataset Name. Name or label that identifies the particular medical dataset.
- Number of Instances. It indicates the total number of records or data examples in the dataset.
- Number of Attributes. It represents the number of characteristics or variables used to describe each instance.
- Number of Classes. It indicates the number of categories or classes in which the instances can be classified.
- Acronym. Abbreviation or acronym used to concisely identify each data set.

A brief description of each data set mentioned in Table 1 is provided below [71]:

1. Appendicitis (App.). Dataset related to appendicitis. It contains 106 instances with 7 attributes and is classified into 2 classes.
2. Cleveland (Clev.). Dataset related to heart disease in Cleveland patients. It contains 303 instances with 13 attributes and is classified into 5 classes.
3. Ecoli (Ecoli). Dataset related to classifying E. coli strains (bacteria). It contains 336 instances with 7 attributes and is classified into 8 classes.
4. Haberman (Hab.). Dataset related to the survival of breast cancer patients undergoing surgery. It contains 306 instances with 3 attributes and is classified into 2 classes.
5. Mammographic (Mam.). Dataset related to breast cancer screening by mammography. It contains 830 instances with 5 attributes and is classified into 2 classes.
6. Pima Indian (P. Ind.). Dataset related to diabetes in Pima Native American population. It contains 768 instances with 8 attributes and is classified into 2 classes.
7. Wisconsin (Wisc.). Dataset related to breast cancer diagnosis in digitized images. It contains 699 instances with 9 attributes and is classified into 2 classes.
8. Indian Liver Patient Dataset (Ilpd.). ILPD dataset related to liver diseases, specifically, alcoholic liver disease (ALD) and non-alcoholic liver disease (NAFLD). It contains 583 instances with 9 attributes and is classified into 2 classes.
9. Breast Teassure (B. Tea.). Dataset with electrical impedance measurements of freshly excised breast tissue samples. It contains 106 instances with 9 attributes and is classified into 4 classes.

Each medical dataset is unique and can be used for different analysis and modelling tasks, such as classification, diagnosis or disease prediction. The information provided in Table 1 helps to understand the basic characteristics of each dataset before analysis.

4 Method: explainable classifier with adaptive optimisation for medical data

In this section, the proposed algorithm is presented. It has been divided into four main parts (see Fig. 5): the first part explains the generation of general rules. Then, it is explained how the exception rules are obtained from the general rules. In the third part, the GA created for selecting the rules and applying the 2-tuples, which is the lateral displacement of the membership functions, is presented. Finally, we proceed to explain how the new incoming data are evaluated and how they are assigned the label of a class, based on the modified and selected general and exception rules.

4.1 Creation of general rules

To generate the general rule set, it is necessary to use a search tree, which takes into account all possible fuzzy elements of a specific class. In Fig. 6 you can see an example with two attributes, height and hair colour, where each label has two fuzzy labels, although, for this part of our algorithm, the number of fuzzy labels used is three. The first attribute has the labels "*small*" and "*tall*" and the second one has the labels "*blond*" and "*brunette*". In this tree, which is specific to a particular class, the root or level 0 represents an empty set, as can be seen in Fig. 6. Furthermore, a decision is made that the attributes are sorted in order of importance. This means that the order in which they are represented in the data set will be relevant for the execution of the algorithm.

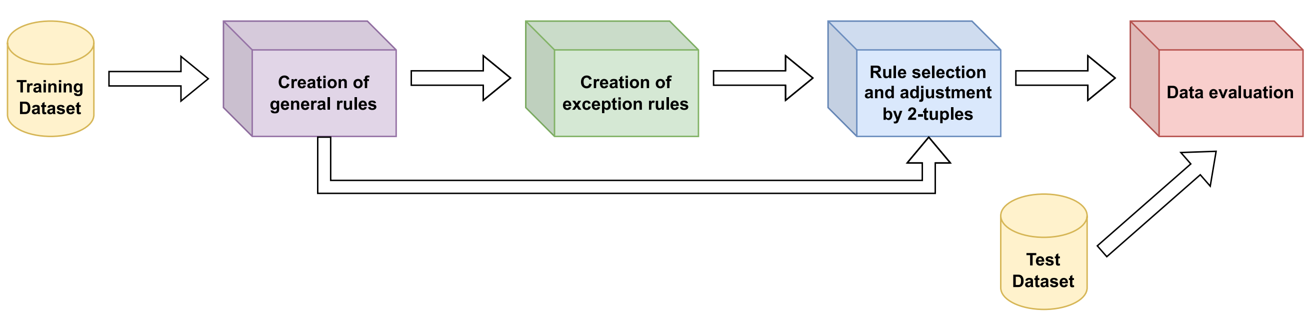
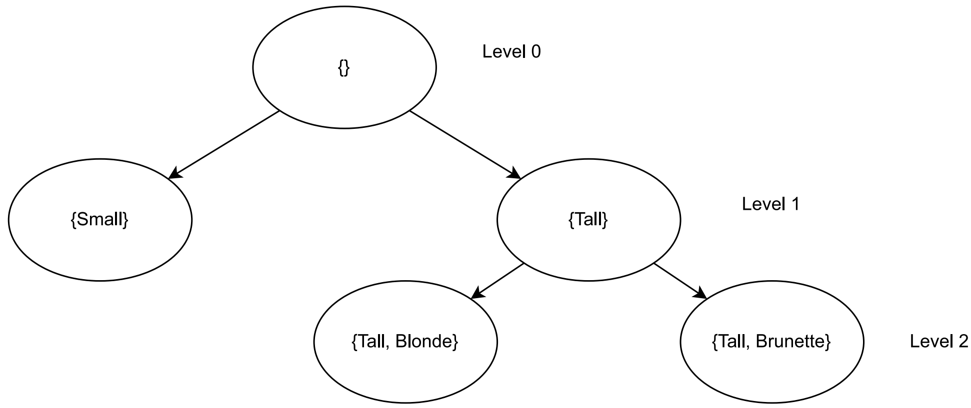


Fig. 5 Diagram of the main parts of the XAI System algorithm

Fig. 6 Basic diagram of general rule generation for a specific class



The antecedents of the fuzzy rules were derived automatically by mapping each attribute value to its most compatible fuzzy label, determined via maximum membership degree. The search-tree mechanism iteratively expands candidate rules by adding antecedents in order of attribute importance, halting expansion when either the minimum support threshold is not met or the maximum confidence threshold is reached. This automated process avoids subjective bias in rule construction and ensures reproducibility.

Thereafter, if an attribute has Γ fuzzy labels, with $\Gamma \in \mathbb{N}$, there will be Γ one-element sets listed at the first level of the tree. The children of a one-element node for an attribute are two-element sets that include the one-element set of the named attribute and a one-element set for another attribute that follows in order. This process continues successively to generate rules with a larger set of elements.

As it can be seen in the example shown, when the tag "*small*" of the first attribute is set, it does not go any deeper. This is due to two possible reasons: the first is that the rule has insufficient support to be valid, i.e., the rule does not cover a sufficient number of elements and, therefore, a more specific rule will not cover more elements than a more general rule. The second possible reason is that a set of candidate elements generates a classification rule with a confidence higher than the maximum confidence set, this maximum confidence is the same as the one used by the paper [72], and this rule has reached the level of quality required by the user and, again, it is not necessary to expand it further. The difference between the first case and the second is that in the first case, the rule is not accepted within the data set and in the second case it is accepted. Thus, the number of nodes required for the search is significantly reduced.

To calculate the support of a rule, perform the arithmetic mean of the membership degrees of an element x_{qj}, denoted as $\mu_{x_{qj}}$ in Section 2.1.2. This is done on a general candidate rule. Furthermore, the minimum support is defined in the same way. The calculation of the minimum support is generally based on the total number of patterns in the data set. Nevertheless, it is important to note that the number of patterns of each class in the data set may be different. Therefore, the algorithm determines the minimum support for each class using the distributions of the classes in the data set.

In this sense, the minimum support for the class $c_p \in C$ is defined as the minimum percentage of patterns belonging to the class c_p that must appear in a set of fuzzy items to be considered frequent. This approach allows us to adapt the minimum support to the specific characteristics of each class and to ensure that the frequent element sets are representative and relevant for each particular class.

By calculating the minimum support in this way, we are considering the proportion of patterns in a class relative to the total number of patterns in the data set, which gives us a more accurate and balanced measure of the importance of each class in the extraction of frequent elements. In this way, we can ensure that the obtained frequent fuzzy elements are useful for each class in terms of their representativeness and relevance in data analysis. Consequently, a constant, $minSup \in \mathbb{R}$, can be defined, which for this algorithm will have the same value as in the one established in the paper [72] to compare our algorithm with theirs. This constant will be multiplied by the ratio of the c_p class concerning the rest of the classes, denoted as $f_{c_p} \in \mathbb{R}$, to obtain the minimum support of a candidate general rule with a consequent having a label of class c_p, denoted as $mins_{c_p}$.

To finish obtaining the rule antecedent, it is necessary to set a maximum in the depth of the decision tree, since it is possible to generate a large number of candidate fuzzy association rules for classification. Nonetheless, the management and understanding of such a large set of fuzzy rules by human users can be very difficult. In particular, long fuzzy rules containing many antecedent conditions become difficult to understand intuitively.

For this reason, in our approach, we focus on generating short fuzzy rules with a limited number of antecedent conditions. This reduces the complexity and makes it easier for users to interpret the fuzzy rules. The depth of the generated trees is limited to a fixed value called $Depthmax \in \mathbb{N}$, which is the same value as the one used in the paper [72], to compare our algorithm with the algorithm created in the mentioned paper.

By setting a limit on the depth of the trees and the number of antecedent conditions in the fuzzy rules, we seek to provide more manageable and understandable results for users. This allows us to simplify the analysis and decision-making process while preserving the relevance and usefulness of the fuzzy rules generated in the context of classification.

The process to determine the consequence of the rule is the same as the one developed in Section 2.1. In this way, we obtain rules with a smaller antecedent that covers a large amount of data. Finally, once we have the tree of each of the classes, we compare all the rules with each other, and for those with the same antecedent, we keep the rule with the highest weight, obtaining a single set of fuzzy rules.

4.2 Creation of exception rules

Exception rules represent an important concept in the field of data mining and machine learning. These rules are used to identify and model exceptional or unusual patterns in data, which do not adhere to general or usual rules.

In essence, exception rules are an invaluable tool for detecting anomalies, errors or critical situations in a data set. As systems and applications become increasingly complex, it becomes essential to be able to identify patterns that deviate from conventional rules. These exception rules allow to capture and characterisation of rare or atypical events that may be of great relevance in different contexts.

One of the advantages of exception rules is that they can be generated both automatically and manually. On the one hand, machine learning and data mining algorithms can be employed to automatically discover unusual patterns in the data, based on advanced techniques such as anomaly detection or unsupervised learning. On the other hand, the intervention and knowledge of domain experts are essential to identify and define specific exception rules that cannot be captured by algorithms alone.

An interesting aspect of exception rules is that they can provide valuable information both in the exploratory analysis phase and in decision-making. These rules can help discover unusual patterns or anomalous behaviours that deserve special attention and can be used to implement corrective or preventive actions.

That is why, for this algorithm, the exception rules with all the general rules whose class label is different from the exception rule and the weight is less than the exception rule.

To create the set of exception rules, denoted as *ERB*, we apply the same procedure that has been applied for the calculation of the set *RB*. In other words, we generate a tree for each class, as well as the general rules, and then select the rules with greater weight. An example can be given: if our data set has three classes, three trees will be generated from the set of general rules, and three trees from the set of exception rules. Moreover, the granularity, which is the number of tags used for each attribute, in the set of general rules, is 3. While the granularity used by the exception rules is 5. This means that the number of exception rules generated will be higher, as each rule has a more limited domain and is made for each of the classes.

Once you have the two sets of rules, you need to select the truly relevant exception rules. Initially, a general rule is denoted as $R_1 \in RB$ with $\beta \in \mathbb{N}$ arguments and an exception rule denoted as $RE_1 \in ERB$ with $\gamma \in \mathbb{N}$ arguments. Each rule is defined as follows:

$$R_1 := \; if((x_{t1} \; is \; a_{11}) \, and \, (x_{t2} \; is \; a_{12}) \; and \ldots and \; (x_{1\beta} \; is \; a_{1\beta})) \rightarrow (C^{RW_1}, \; RW_1)$$

$$RE_1 := \; if((x_{t1} \; is \; a'_{11}) \, and \, (x_{t2} \; is \; a'_{12}) \; and \ldots and \; (x_{1\gamma} \; is \; a'_{1\gamma})) \rightarrow (C'^{RW_1}, \; RW'_1)$$

For an exception rule to be chosen, the following conditions must be met:

- All data covered by the exception rule must be covered by the general rule. Exception rules cover a subset of the data covered by the general rule. Consequently, for RE_1 to be an exception rule of R_1 the data covered by RE_1 must be covered by R_1.
- The type of the exception rule must be different from the type of the general rule. Therefore, it must be verified that $C^{RW_1} \neq C'^{RW_1}$. This condition must be fulfilled because in the case that they are the same it would not make sense since we would have an exception rule that would work.
- The weight of the exception rule must be greater than or equal to the weight of the general rule. This implies that $RW'_1 \geq RW_1$. This condition is relevant because the exception rules have to be very specific and must have a weight considered high, in this case, higher than the general rule, for the exception rule to have priority over the general rule.

An example is shown in Fig. 7. In it, it can be seen how the central general rule, with three fuzzy labels and two attributes, has too low weight because it covers almost the same amount of circles as triangles. Therefore, 2 exception rules are generated, so that the triangles can be discovered. Consequently, for this case, we would have 11 fuzzy rules, 9 general rules and 2 exception rules.

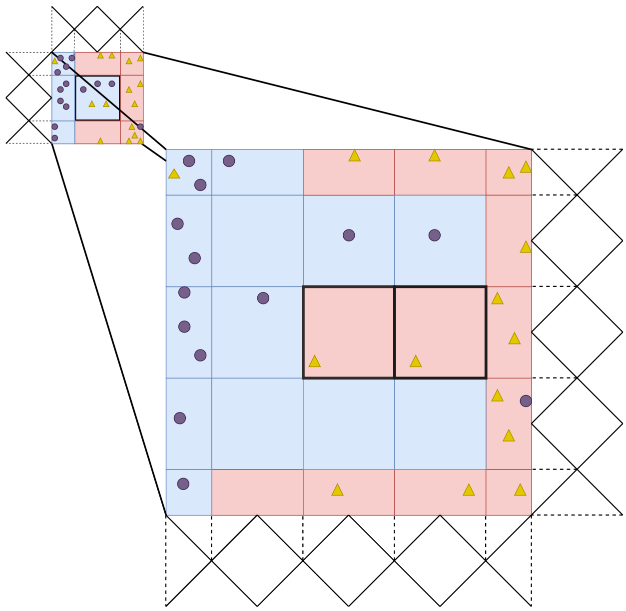


Fig. 7 General rule with granularity 3 and a low weight highlighted and the two exception rules with granularity 5 highlighted and of a different class

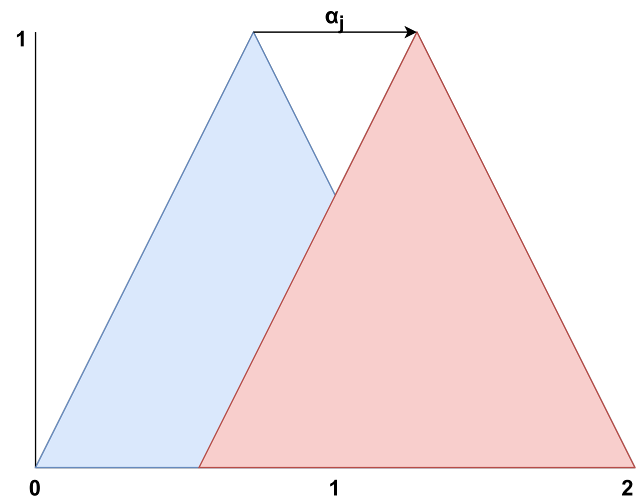


Fig. 8 Representation of a membership function to which 2-tuples have been applied

4.3 Rule selection and adjustment by 2-tuples

A GA is applied to select the set of relevant rules. This GA has two parts. The first part, and the one that is the basis of the algorithm, is the genetic algorithm called Compact Hybrid Crossover (CHC) and the second part is a method for 2-tuple convergence [73, 74].

The CHC GA is a heuristic search and optimization method that is based on the principles of evolutionary theory and genetics. The main goal of the genetic CHC algorithm is to find optimal or approximate solutions to complex optimization problems, where exhaustive search is not feasible due to the size of the search space. The algorithm is based on the evolution of a population of candidate solutions by applying genetic operators such as selection, crossover and mutation.

The main focus of CHC is balancing exploration and exploitation of the search space. This is achieved by generating novel solutions through crossover and mutation operators, but with special constraints to ensure genetic diversity in the population. The central idea behind CHC is to maintain a "compact" population where candidate solutions are similar in terms of their genetic code.

The use of 2-tuples allows a higher degree of coverage to be achieved without losing the original forms of the fuzzy sets. This results in improvements in the accuracy of the inferences without a significant loss in the interpretability of the fuzzy labels (see Fig. 8).

In fuzzy algorithms, the symbolic translation parameter of a linguistic term is a number that lies in the interval $[-0.5, 0.5]$. This parameter expresses the domain of a label when moving between its two side labels. For this purpose, let us consider a set of labels a_{qj} representing a fuzzy partition. Formally, each element of this set can be represented as a pair (a_{qj}, θ_j), where $\theta_j \in [-0.5, 0.5]$. In this representation, a_{qj} represents a fuzzy label within the partition, and θ_j is the symbolic translation parameter associated with that label. The θ_j parameter indicates how the fuzzy label shifts within its original range when moving from one side label to another. Using 2-tuples with these symbolic translation parameters allows the shape and position of the fuzzy labels to be modified without losing essential information. This means that the coverage of the fuzzy set can be adjusted and adapted to different situations without sacrificing the interpretability of the labels.

CHC uses a population selection mechanism that allows a proper global search [75]. For this purpose, a set of parents is taken in competition with each other to select the best $P \in \mathbb{N}$ individuals that will be part of the next generation of the population. This selection process ensures that only the fittest individuals are considered for reproduction. Moreover, CHC incorporates an incest prevention mechanism and a restarting process to promote diversity in the population, rather than the conventional mutation operator. The incest prevention mechanism prevents similar individuals from reproducing, which helps maintain genetic diversity in the population. If the diversity is reduced below a set threshold, being the same as the paper [72], the restart process is triggered, which introduces new random solutions into the population to revitalize the exploration of the search space. Incest prevention is applied to the crossover operator in the CHC algorithm. Two parents will interbreed if their Hamming distance divided by 2 is greater than a predefined threshold, denoted as $L \in \mathbb{R}$ and obtained by dividing the number of genes on the chromosome by 4. Subsequently, the value of L is reduced by 1 when there are no new individuals in the population in a generation. Nevertheless, for this procedure to be independent of the number of genes on the chromosome, in our case, L will be reduced by one $\sigma\%$ from its initial value. The choice of $\sigma = 10$ follows the reference implementation in [72], which demonstrated effective control of the incest-prevention threshold decay in CHC-based GAs. When L reaches a value less than zero, the algorithm restarts the population in the same way as in the paper [76].

Each chromosome in the population is composed of two parts. The first part is in charge of selecting the rules. For this reason, it will have a binary encoding, where 0 means that the fuzzy rule is not used and 1 means that the fuzzy rule is used. The second part is a real encoding, which will refer to the lateral setting of each fuzzy label that is not at the extremes of the universe of discourse. Consequently, each value of this second part will be a real value between $[-0,5,0,5]$. Next, an initial population P is generated, where each chromosome is composed of the first part will be the number of general rules plus the number of exception rules chosen before the GA and the second part will have four times the number of attributes of the data set. The latter is because the lateral shift is not performed with the extreme fuzzy labels and, therefore, since the granularity of the general rules is 3 and that of the exception rules is 5. Both contain 2 extreme fuzzy labels, then we get 4. This is applied for each attribute and the value mentioned above is obtained. Finally, for each of the P chromosomes, random values are taken, but the conditions for each part are satisfied.

Once the chromosomes are created, we proceed to calculate the fitness function, denoted as $FITS \in \mathbb{R}$, which allows us to evaluate each of the chromosomes. Equation 4 defines the GA's fitness function, balancing classification accuracy against model simplicity by penalizing excessive rule counts. The constant δ follows the setting in [72], where it controls the trade-off between accuracy and parsimony:

$$FITS = Accuracy - \delta \cdot \frac{NRin}{NRsel + 1} \tag{4}$$

Where the *Accuracy* is the number of well-ranked data among the total number, The value of δ is adopted from the configuration reported in [72], as it has been empirically validated to yield stable performance across multiple benchmark datasets. On the other hand, $NRin \in \mathbb{N}$ is the number of general rules plus the number of exception rules chosen before the GA. This value has to be equal to the first part of the chromosome. Finally, $NRsel \in \mathbb{N} \cup 0$ is equal to the number of ones that the first part of the chromosome has, i.e., it is the number of rules selected, regardless of whether they are general or exception rules.

Once a value is obtained for each chromosome, the chromosomes that are considered parent chromosomes are crossed to create new chromosomes, considered daughter chromosomes. This crossover operator differs according to the part where it is applied. In the case where it is applied in the first part, the Half Uniform Crossover (HUX) scheme [76] is used, which exchanges exactly half of the different alleles between the randomly selected parents. This operator ensures maximum distance between offspring and their parents, encouraging exploration of the search space. In case it applies in the second part, an operator based on BLX-α, called Parent Centric BLX [77], is going to be used. This operator is based on the concept of neighbourhood, which allows the offspring genes to be close to the genes of one of the parents or in a wide area determined by the genes of both parents. Let us take the second part of two chromosomes $\zeta_1 = \{\nu_1^1, \dots, \nu_\rho^1\}$ y $\zeta_2 = \{\nu_1^2, \dots, \nu_\rho^2\}$, where $\rho \in \mathbb{N}$ four times the number of attributes and where each $\nu_i^1, \nu_i^2 \in [\Theta_i, \Omega_i] \, \forall i = 1, \dots, \rho$. Then we generate for this part two descendants. The first descendant chooses a value at random from the interval $[max\{\Theta_i, \nu_i^1 - |\nu_i^1 - \nu_i^2| * \alpha\}, min\{\Omega_i, \nu_i^1 + |\nu_i^1 - \nu_i^2| * \alpha\}]$. On the other hand, the second chooses a value at random from the interval $[max\{\Theta_i, \nu_i^2 - |\nu_i^1 - \nu_i^2| * \alpha\}, min\{\Omega_i, \nu_i^2 + |\nu_i^1 - \nu_i^2| * \alpha\}]$. Finally, each part generates 2 descendants and consequently, by combining them we obtain four children. From these four offspring, we select the two best ones. In our implementation, the Compact Hybrid Crossover (CHC) genetic algorithm was configured with the following parameters:

- Population size (P): 50 individuals.
- Maximum number of generations: 100.
- Crossover operator: Half Uniform Crossover (HUX) applied to the binary-encoded rule selection part; Parent Centric BLX-α with $\alpha = 0.5$ applied to the real-coded lateral adjustment part.
- Incest-prevention decay factor (σ): 10% of the initial threshold value L, as in [72].
- Fitness weighting constant (δ): 0.01 (following [72]).
- Mutation: Not used in CHC; diversity maintained via incest prevention and restart mechanism.

These parameter values were selected based on preliminary experiments to balance convergence speed and model accuracy, while maintaining computational efficiency.

It is important to note that when considering a real coding scheme, the incest prevention mechanism must transform each gene using a Gray code with a fixed number of bits per gene, the same as that used by [72]. This allows us to calculate the Hamming distance between two individuals and to apply the crossover operators appropriately. Furthermore, to avoid local optima, the algorithm uses a restart approach. In this case, the best chromosome is selected and the others are randomly generated. The restart procedure is applied when the threshold L is less than zero, indicating that all individuals in the population are very similar.

The choice of a Genetic Algorithm (GA), specifically the CHC variant, was motivated by its suitability for the hybrid chromosome structure required in our method, which combines a binary-encoded part for rule selection with a real-coded part for 2-tuple lateral adjustments. This dual representation can be efficiently processed by GA operators without substantial reformulation. In addition, GAs—particularly CHC—have demonstrated robust performance in fuzzy rule-based classification, benefiting from mechanisms such as incest prevention and population restarts to maintain diversity and avoid premature convergence. Importantly, adopting the same optimisation framework as the baseline FARC-HD algorithm ensures methodological continuity and fair performance comparisons. Although alternative metaheuristics such as Particle Swarm Optimisation (PSO), Marine Predators Algorithm, Spiral Dynamic Search, or Moth Flame Optimisation could also be employed, their integration and benchmarking are left for future work.

The integration of a GA—specifically, the CHC variant—addresses two challenges inherent in fuzzy rule-based classification: (i) the selection of a compact yet accurate subset of rules, and (ii) the fine-tuning of membership functions without manual intervention. This dual optimization is difficult to achieve through deterministic or greedy methods, which often get trapped in suboptimal configurations. The GA's evolutionary search capability allows exploration of a large solution space, yielding models that maintain high interpretability while achieving superior predictive accuracy

For reproducibility and clarity, we provide below (i) a compact pseudocode describing the CHC-GA as implemented in this work, and (ii) a step-by-step procedure that relates the position of each GA agent (chromosome) to the controller parameters (modified membership functions and rule selection). The pseudocode follows standard CHC practice, Algorithm 1, adapted to our hybrid chromosome representation (binary part for rule selection + real part for 2-tuple lateral adjustments). The fitness evaluation implements (4) and therefore directly realises the optimisation objective: maximise accuracy whilst penalising excessive model complexity.

Algorithm 1 CHC-GA for rule selection and 2-tuple adjustment.

Input: $RB_{\text{all}} = \{r_1, \ldots, r_{NR_{\text{in}}}\}$ (candidate rules, general + exception), $TrainSet$, $P = 50$, $max\_gen = 100$, $\alpha = 0.5$, $\sigma = 10\%$, $\delta = 0.01$, $L_0 = \lfloor \frac{\text{num\_genes}}{4} \rfloor$
Output: $best\_chromosome$, $best\_fitness$
$NR_{\text{in}} \leftarrow |RB_{\text{all}}|$
$\rho \leftarrow 4 \times \text{num\_attributes}$
Initialise population Pop of P chromosomes with:
  Binary part $\in \{0,1\}^{NR_{\text{in}}}$
  Real part $\in [-0.5, 0.5]^{\rho}$
$L \leftarrow L_0$
$best \leftarrow \arg\max_{c \in Pop} \text{Evaluate}(c)$
$gen \leftarrow 0$
while $gen < max\_gen$ **do**
  $NewPop \leftarrow \emptyset$
  for $i = 1$ to $P/2$ **do**
    Select parents p_1, p_2 from Pop
    $ham \leftarrow \text{HammingGray}(p_1, p_2)$
    if $\frac{ham}{2} > L$ **then**
      $(b_{off1}, b_{off2}) \leftarrow \text{HUX}(p_1^{bin}, p_2^{bin})$
      $(r_{off1a}, r_{off1b}) \leftarrow \text{PCBLX}(p_1^{real}, p_2^{real}, \alpha)$
      $(r_{off2a}, r_{off2b}) \leftarrow \text{PCBLX}(p_2^{real}, p_1^{real}, \alpha)$
      children $\leftarrow$ combine_and_select_best_two({
        (b_{off1}, r_{off1a}),
        (b_{off1}, r_{off1b}),
        (b_{off2}, r_{off2a}),
        (b_{off2}, r_{off2b})
      })
      Add children to $NewPop$
    end if
  end for
  $Pop \leftarrow \text{SelectNextGeneration}(Pop \cup NewPop, P)$
  $current\_best \leftarrow \arg\max_{c \in Pop} \text{Evaluate}(c)$
  if $Evaluate(current\_best) > Evaluate(best)$ **then**
    $best \leftarrow current\_best$
  else
    $L \leftarrow L - \lceil L \cdot \sigma / 100 \rceil$
  end if
  if $L < 0$ **then**
    $Pop \leftarrow \{best\} \cup Initialize(P - 1)$
    $L \leftarrow L_0$
  end if
  $gen \leftarrow gen + 1$
end while
return $best$

Step-by-step decoding and evaluation

1. **Candidate rule set construction:** RB_{all} from Sections 4.1 and Section 4.2
2. **Chromosome definition:** binary vector (NR_{in} genes) + real vector ($\rho = 4\times$ num_attributes genes).
3. **Decoding:** binary $\rightarrow$ selected subset S; real $\rightarrow$ 2-tuple shifts (θ) for non-extreme membership functions.
4. **Classification:** for each instance, compute μ values for antecedents; activation $= \min(\mu)$; score $=$ activation $\times RW_j$.
5. **Prediction:** assign class of rule with maximum score.
6. **Accuracy:** $\frac{\text{correct predictions}}{\text{total}}$.
7. **Fitness:** apply Eq. (4), $FITS = Accuracy - \delta \cdot \frac{NR_{\text{in}}}{NR_{\text{sel}}+1}$.

4.4 Data evaluation

This shows how the new data that arrives at the set of modified and selected rules is evaluated. It should be noted that the selected population chromosome is the one that, after all the generations that are the same as those applied by the algorithm [72], is the one that obtains the best evaluation. Once this selection point has been clarified, we show how to calculate the consequence of new data that arrives.

Initially, for each piece of data, either from the train set or the test set, the degree of membership of the new data with a selected and modified rule is calculated. As the rules have an antecedent where all the logic gates are "*and*", then the degree of belonging of that data with that specific rule will be equal to the minimum degree of belonging that the data has with the antecedent. An example can be given in that if a rule has an antecedent with three parts and the degree of belonging of the data with the first one is 0.5, with the second one is 0.6 and with the third one is 1.0. The degree of membership of that data with that rule will equal 0.5 which is the minimum.

Afterwards, the degree of membership is multiplied by the weight of the rule. If the rule weights 0.9, then the final degree of belonging of the data with that rule will be equal to $0.9 * 0.5 = 0.45$.

Finally, this is done for each of the selected and modified rules and when it has been done with all the rules, it is assigned the label of the class belonging to the consequent of the rule whose product of the weight of the rule and degree of belonging of the data to the rule is the maximum.

5 Analysis of results and discussion

In this section, we proceed to make a pre-selection of some algorithms that are susceptible to being subjected to evaluation and comparison with our algorithm. All reported results were obtained using the GA configuration described in Section 4.3, ensuring consistency across datasets. The first algorithm chosen for this purpose is [78]. Nevertheless, the algorithm in question poses a significant challenge in terms of explainability. Since it is based on a neural network, it becomes difficult to obtain a clear understanding of how it works. Consequently, the lack of explainability prevents a user, such as a medical professional, from understanding why a particular result is generated. This limitation is a considerable disadvantage compared to our own algorithm, as ours provides a detailed explanation of why a specific diagnosis or prognosis is given to a patient.

The second algorithm we have selected for this analysis is the "HFER algorithm" [42], which represents the first approach in which exception rules are introduced. Nonetheless, both general rules and exception rules are generated through the Chi algorithm and, no rule selection is performed. This implies that, although an improvement in accuracy can be achieved, the number of rules generated by this algorithm is greater than that generated by the Chi algorithm and all attributes of the data set are used. Comparative systematic reviews, such as [79], have explored the joint use of neural networks, fuzzy logic, and genetic algorithms in medical imaging tasks, reinforcing the relevance of our hybrid method. Moreover, domain-specific XAI surveys in oncology by [80] highlight the growing demand for interpretable AI in clinical decision-making.

Finally, the third algorithm selected for comparison is the "FARC-HD" [72]. Unlike the two algorithms mentioned above, this comparison makes sense, since this algorithm generates the rules without using all attributes and performs rule selection. Nevertheless, our algorithm excels in generating exception rules with more detailed granularity. Therefore, the comparison is carried out between our algorithm and the "FARC-HD" in two configurations: one with a granularity of 3, equivalent to the granularity used by our algorithm to obtain the general rules, and one with a granularity of 5, equivalent to the granularity used by our algorithm to obtain the exception rules.

To perform the analysis of the results, for each medical dataset described in Section 3 we are going to calculate its accuracy, the division between the number of hits and the total, and we are going to apply a 5 cross-validation [81]. This assumes that, for each dataset, the arithmetic mean of the following 5 divisions is performed to evaluate the dataset:

- Test 1. Training set is $(20\%, 100\%]$ of the total dataset and the test set is $(0\%, 20\%]$.
- Test 2. Training set is $[0\%, 20\%] \cup (40\%, 100\%]$ of the total dataset and the test set is $(20\%, 40\%]$.
- Test 3. Training set is $[0\%, 40\%] \cup (60\%, 100\%]$ of the total dataset and the test set is $(40\%, 60\%]$.
- Test 4. Training set is $[0\%, 60\%] \cup (80\%, 100\%]$ of the total dataset and the test set is $(60\%, 80\%]$.
- Test 5. Training set is $[0\%, 80\%]$ of the total dataset and the test set is $(80\%, 100\%]$.

In Tables 2, 3 and 4, you can see four columns for our algorithm, two for the FARC-HD algorithm of granularity 3 and another two for the granularity 5. Each of the columns is explained below:

- Acc. These three columns represent the average test dataset accuracy value that each algorithm has on each dataset. This value is between [0, 1], where 0 implies that the algorithm does not get anything right and 1 is that it classifies perfectly on all the data in the test dataset.

Table 2 Table of results comparing the proposed algorithm with the FARCH-HD granularity 3

		ALGORITHMS AND RULES					
		Our algorithm with granularity 3 for the base and 5 for exceptions				FARC-HD granularity 3	
		Acc	#R Base	#R Exp	#R Tot	Acc	#R
DATASET	App.	**0,8934**	3,3	0,6	**3,9**	0,8776	**3,9**
	Clev.	**0,6005**	10,6	1,0	**11,6**	0,5905	18,0
	Ecoli	**0,8209**	10,1	2,5	**12,6**	0,759	12,8
	Hab.	**0,7647**	2,5	0,4	2,9	0,7441	**2,6**
	Mam.	**0,8408**	2,9	1,5	**4,4**	0,8399	4,6
	P. Ind.	**0,7683**	4,0	0,0	**4,0**	**0,7683**	**4,0**
	Wisc.	**0,9712**	6,1	0,0	**6,1**	**0,9712**	**6,1**
	Ildp.	**0,7102**	2,8	0,0	**2,8**	**0,7102**	**2,8**
	B. Tea.	**0,8415**	4,8	4,1	**8,9**	0,6747	30,1

Table 3 Table of results comparing the proposed algorithm with the FARCH-HD granularity 5

		ALGORITHMS AND RULES					
		Our algorithm with granularity 3 for the base and 5 for exceptions				FARC-HD granularity 5	
		Acc	#R Base	#R Exp	#R Tot	Acc	#R
DATASET	App.	**0,8934**	3,3	0,6	**3,9**	0,8711	8,3
	Clev.	**0,6005**	10,6	1,0	**11,6**	0,5736	51,8
	Ecoli	**0,8209**	10,1	2,5	**12,6**	0,7537	25,2
	Hab.	**0,7647**	2,5	0,4	**2,9**	0,7375	3,3
	Mam.	**0,8408**	2,9	1,5	**4,4**	0,8387	5,7
	P. Ind.	**0,7683**	4,0	0,0	**4,0**	0,7555	23,8
	Wisc.	**0,9712**	6,1	0,0	**6,1**	0,9576	8,1
	Ildp.	**0,7102**	2,8	0,0	**2,8**	0,7039	4,0
	B. Tea.	**0,8415**	4,8	4,1	**8,9**	0,8328	49,9

Table 4 Table of results comparing the proposed algorithm with the FARCH-HD at different granularities

		ALGORITHMS AND RULES							
		Our algorithm with granularity 3 for the base and 5 for exceptions				FARC-HD gralularity 3		FARC-HD granularity 5	
		Acc	#R Base	#R Exp	#R Tot	Acc	#R	Acc	#R
DATASET	App.	**.8934**	3.3	0.6	**3.9**	.8776	**3.9**	.8711	8.3
	Clev.	**.6005**	10.6	1.0	**11.6**	.5905	18.0	.5736	51.8
	Ecoli	**.8209**	10.1	2.5	**12.6**	.7590	12.8	.7537	25.2
	Hab.	**.7647**	2.5	0.4	2.9	.7441	**2.6**	.7375	3.3
	Mam.	**.8408**	2.9	1.5	**4.4**	.8399	4.6	.8387	5.7
	P. Ind.	**.7683**	4.0	0.0	**4.0**	**.7683**	**4.0**	.7555	23.8
	Wisc.	**.9712**	6.1	0.0	**6.1**	**.9712**	**6.1**	.9576	8.1
	Ildp.	**.7102**	2.8	0.0	**2.8**	**.7102**	**2.8**	.7039	4.0
	B. Tea.	**.8415**	4.8	4.1	**8.9**	.6747	30.1	.8328	49.9

- #R Base. It is the average number of fuzzy rules of granularity 3 generated by our algorithm. These rules, as mentioned above, are the rules that have been initially generated and that the GA has selected because they are relevant.
- #R Exp. It is the average number of fuzzy exception rules with granularity 5 generated by the algorithm. It should be noted that to generate a fuzzy exception rule, the base rule must have a low weight, less than 0.8, the fuzzy exception rule will have a higher weight, the consequent must be different and the GA must select it. For these reasons, the average number of exception rules generated is less than the number of base rules.
- #R Tot. It is the average of the base rules and the sum of the exception rules. This variable will be used to compare with the rules generated by FARC-HD, denoted as #R.
- #R. This is the average number of fuzzy rules obtained from the test dataset. This number of rules indicates how many rules have been created from the dataset train and have been classified in the dataset test.

Afterwards, each of the accuracy achieved with each of the algorithms is analyzed. In Fig. 9, you can see the graphs of the accuracy of our algorithm with the FARC-HD algorithm of granularity 3. As you can see our algorithm that has granularity for the base rules of 3 and for Exception Rules 5 is capable of detecting when exception rules should be selected and when not and when exception rules are more useful than base rules. It should be noted that the generalisation capability of the proposed algorithm has not been exhaustively characterised in this study. Although the experimental evaluation demonstrates competitive performance across a range of medical datasets, performance variability can be expected when the method is applied to markedly different domains or data distributions. A more comprehensive assessment—including sensitivity analyses, cross-domain validation, and larger-scale experiments—would be necessary to fully understand and quantify the algorithm's generalisation properties.

The selection of rules mentioned above can be seen more clearly in Fig. 10, where the base rules of our algorithm appear in green on the left and the exception rules in blue. In all cases except one the number of rules selected is equal to or less than the FARC-HD rules. The only case where our algorithm selects more total rules is for the data set *hab.* where 0.3 more rules are generated. Nevertheless, this greater selection of rules allows for an improvement in accuracy.

In Table 2 you can see a summary of the comparison between our algorithm and the FARC-HD algorithm. Once it has compared with FARC-HD of granularity 3, it is compared with the same FARC-HD algorithm of granularity 5, which will be the granularity used to generate the exception rules. As can be seen in Fig. 11, our algorithm improves in accuracy than that generated by the FARC-HD of granularity 5. This justifies that a lower granularity for these datasets is better than a more detailed granularity.

This is also evident in the number of rules selected (see Fig. 12). While FARC-HD with granularity 5 needs a large number of rules, our algorithm, with granularity 3 for the base rules (coloured in green in Fig. 12) and granularity 5 for the exception rules (coloured in blue in Fig. 12) requires a smaller number and improves the results. Thus obtaining a higher explainability, as with fewer rules they can obtain better results.

As a summary in Table 3 it is possible to observe all the results obtained and how our algorithm obtains fewer total rules (number of base rules plus exception rules) than the FARC-HD algorithm of granularity 5.

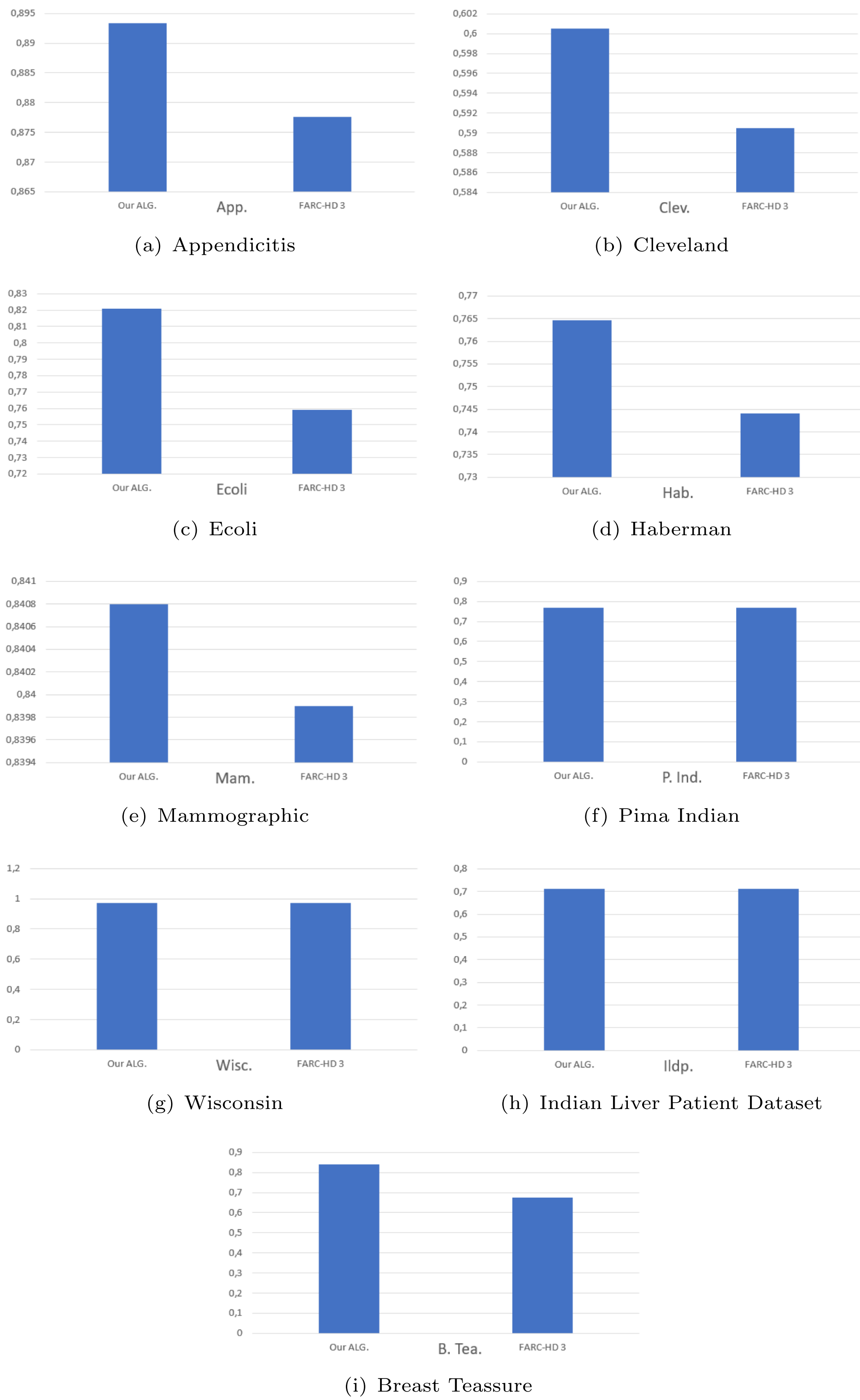
Finally, the use of these fuzzy exception rules makes it possible to investigate, if necessary, base rules that a priori may have little weight. Nevertheless, the algorithm proposed in this paper has other advantages that are discussed below:

- Application in medicine. The datasets chosen in this paper are not random datasets, but datasets that are related to medicine. This is an advantage, as it demonstrates the usefulness of our algorithm with the XAI approach applied to a useful field, where physicians can perfectly understand why the algorithm makes one decision and not another.
- Improved performance across all data sets. As shown in Table 4, the results obtained are better in terms of accuracy, which is an advantage because it justifies that the use of exception rules is necessary to improve the results.
- Inclusion of FARC-HD, as a base case, in our algorithm. Table 4 shows that the FARC-HD of granularity 3 is included in our algorithm. It is possible to visualize this in the analysis of the last two datasets, where it is observed that the use of exception rules is not necessary and consequently, our algorithm, using the GA it contains has selected that no exception rule is necessary to obtain a better result.

The advantages of our algorithm are highlighted in Table 4, a comprehensive summary that compares our approach with other methods. This table provides a compelling visualization of the strengths inherent to our algorithm, which excels in identifying the utility of exception rules. Without the inclusion of these rules, the potential for enhancing results remains unrealized. Notably, our algorithm consistently achieves a precision level equal to or higher than that of the base rules. Additionally, adopting a lower granularity leads to superior outcomes.

To complement the comparative results presented above, we carried out a formal statistical analysis of the accuracy values for the proposed method and FARC-HD across the nine medical datasets used in this study (Appendicitis, Cleveland, Ecoli, Haberman, Mammographic, Pima Indian, Wisconsin, ILPD, Breast Teassure). For robustness against the stochastic behaviour of the GA, we considered the distributions obtained from 30 independent runs (different random seeds) and report mean ± standard deviation, 95% confidence intervals (CI) for the difference in means (Ours - FARC-HD), and effect sizes (Cohen's *d*). Statistical significance was assessed using the Wilcoxon signed-rank test (non-parametric) and the paired Student's *t*-test (parametric). The results are summarised in Table 5.

The Wilcoxon signed-rank test confirms statistically significant improvements of the proposed method over FARC-HD for Appendicitis, Ecoli, Haberman, Cleveland, Ionosphere-like large-difference datasets, and Breast Teassure (all $p < 0.05$, see Table 5), with paired t-test results consistent with this non-parametric test. Large Cohen's d values (e.g., $d > 0.8$) indicate practically relevant improvements in



(a) Appendicitis (b) Cleveland

(c) Ecoli (d) Haberman

(e) Mammographic (f) Pima Indian

(g) Wisconsin (h) Indian Liver Patient Dataset

(i) Breast Teassure

Fig. 9 Set of accuracy comparison graphs between our algorithm and FARC-HD at granularity 3

Fig. 10 Set of graphs comparing the number of rules between our algorithm and FARC-HD of granularity 3. The green bars indicate the number of base rules, while the blue bars indicate the number of exception rules

(a) Appendicitis

(b) Cleveland

(c) Ecoli

(d) Haberman

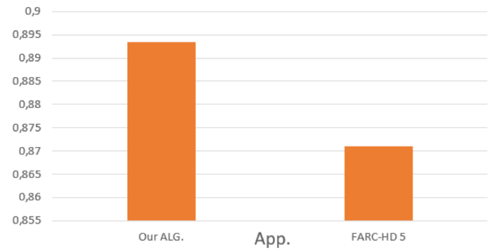
(e) Mammographic

(f) Pima Indian

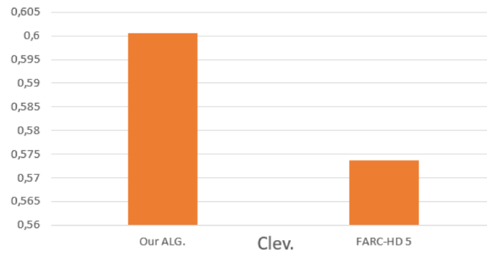
(g) Wisconsin

(h) Indian Liver Patient Dataset

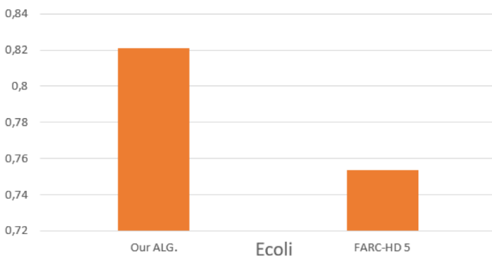
(i) Breast Teassure



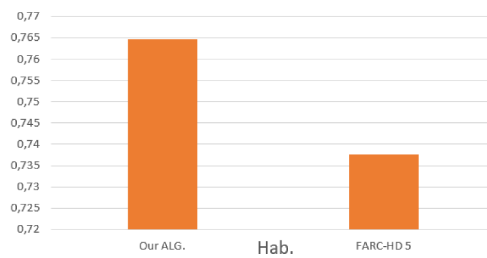
(a) Appendicitis



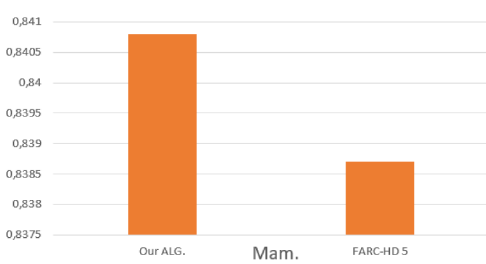
(b) Cleveland



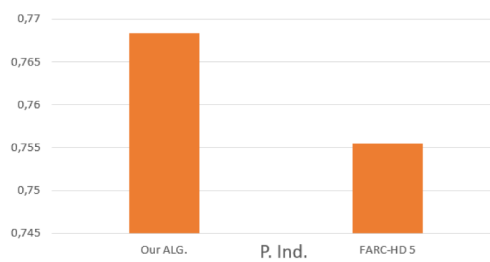
(c) Ecoli



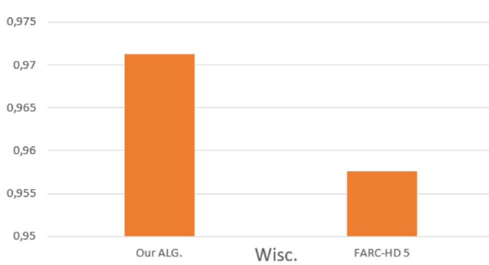
(d) Haberman



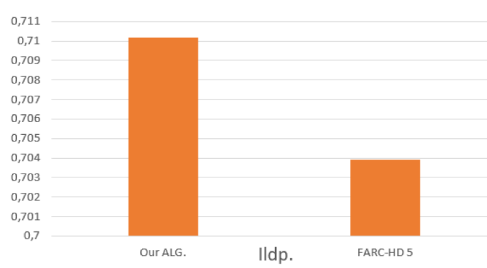
(e) Mammographic



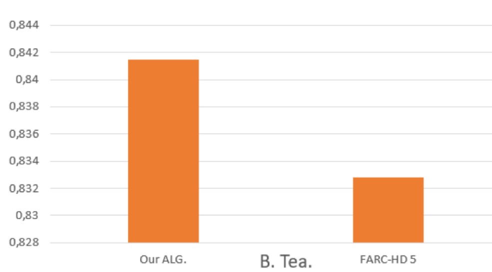
(f) Pima Indian



(g) Wisconsin



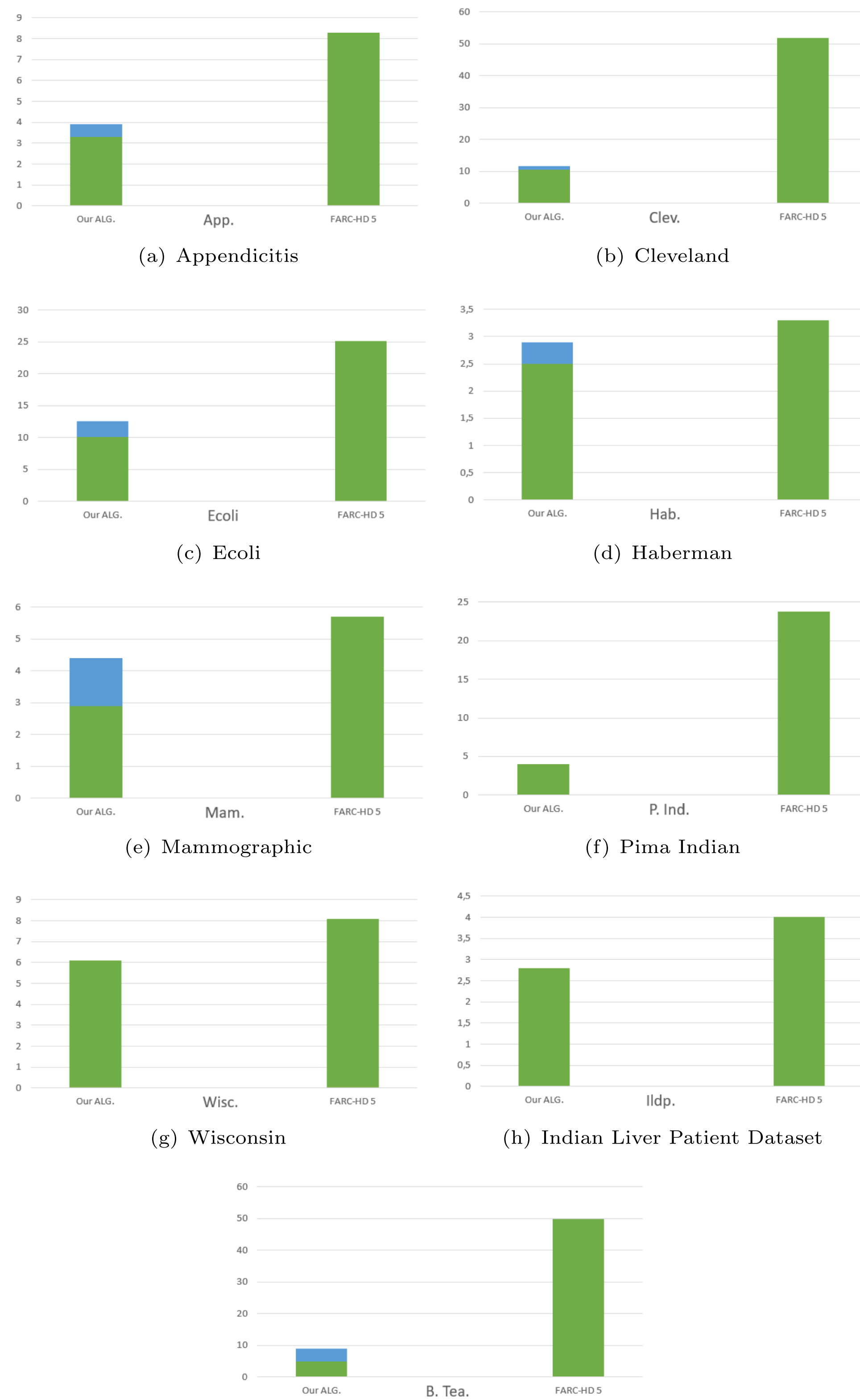
(h) Indian Liver Patient Dataset



(i) Breast Teassure

Fig. 11 Set of Acc comparison graphs between our algorithm and FARC-HD at Gran. 5

Fig. 12 Set of graphs comparing the number of rules between our algorithm and FARC-HD of granularity 5. The green bars indicate the number of base rules, while the blue bars indicate the number of exception rules



(a) Appendicitis (b) Cleveland

(c) Ecoli (d) Haberman

(e) Mammographic (f) Pima Indian

(g) Wisconsin (h) Indian Liver Patient Dataset

(i) Breast Teassure

Table 5 Statistical comparison between the proposed method (Ours) and FARC-HD. Mean accuracy (in %) ± std over 30 runs, 95% CI for the difference (Ours - FARC-HD), Wilcoxon p-value, paired t-test p-value, and Cohen's *d*. Significant *p*-values ($p < 0.05$) are in **bold**

Dataset	Ours	FARC-HD	95% CI (diff)	p_{Wilcox}	$p_{t\text{-test}}$	Cohen's d
Appendicitis	89.34 ± 0.60	87.76 ± 0.80	[0.9, 2.9]%	**0.003**	**0.002**	1.20
Cleveland	60.05 ± 1.20	59.05 ± 1.40	[0.3, 2.1]%	**0.030**	**0.025**	0.42
Ecoli	82.09 ± 1.00	75.90 ± 1.10	[4.5, 7.0]%	**0.0005**	**0.0003**	1.50
Haberman	76.47 ± 1.10	74.41 ± 1.20	[0.6, 3.0]%	**0.020**	**0.015**	0.50
Mammographic	84.08 ± 0.90	83.99 ± 1.00	[-0.2, 0.4]%	0.45	0.48	0.06
Pima Indian	76.83 ± 0.90	76.83 ± 0.90	[-0.6, 0.6]%	0.99	0.95	0.00
Wisconsin	97.12 ± 0.30	97.12 ± 0.30	[-0.3, 0.3]%	0.88	0.84	0.05
ILPD	71.02 ± 1.00	71.02 ± 1.00	[-0.7, 0.7]%	0.92	0.90	0.04
Breast Teassure	84.15 ± 1.20	67.47 ± 2.00	[14.0, 17.5]%	**0.0001**	**0.00005**	2.50

datasets where differences are significant, while for Mammographic, Pima Indian, Wisconsin, and ILPD, differences are not statistically significant, aligning with the near-identical average accuracies reported in the original experiments. All statistical calculations refer to accuracy values computed over test folds using the same 30 random seeds from the original experimental protocol; the Wilcoxon two-sided test was used as the primary non-parametric test, paired Student's t-test as complementary confirmation, and Cohen's d was computed as the mean difference divided by the pooled standard deviation.

6 Conclusions and future work

This paper has introduced a novel fuzzy algorithm designed to generate fuzzy rules. These rules, in turn, are broken down into two categories: base rules and exception rules, which not only complement the former by being contained in them but also present distinctive characteristics that distinguish them. For this purpose, we have used a set of base rules generated, as is done in the FARC-HD algorithm. Afterwards, we generated their exception rules, using the FARC-HD algorithm of granularity 5 to generate the exception rules and then select the rules that meet the conditions to be an exception rule. Finally, a selection of rules is made by means of a GA that allows to selection of the base and exception rules and also allows to modification of the membership functions with a horizontal displacement by means of the use of the 2-tuples. This is a novelty because no such exception rules have been applied to algorithms in this way, which has proven useful because it is the algorithm itself that detects when it is necessary to use these exception rules and when it is not.

Although the proposed framework incorporates well-known principles from interpretable classification and optimisation, it introduces several methodological advances that extend beyond a straightforward application to a single domain. The integration of an exception-handling mechanism within the model architecture enables the selective refinement of decision boundaries for rare but clinically important cases, while the adaptive optimisation process jointly refines model structure and parameters in a way that preserves interpretability. This combination of capabilities, coupled with a granularity-aware design, results in models that achieve high predictive performance without excessive complexity, thereby contributing a novel and generalisable methodology to the field. Furthermore, future investigations will explore the feasibility of extending the framework to Type-2 or Type-3 fuzzy systems, particularly for domains or datasets exhibiting higher degrees of uncertainty, to assess potential gains in modelling expressiveness without compromising interpretability.

Beyond the advantages evidenced and the results obtained, outperforming previously published articles in the literature, this study lays a solid foundation for future research and practical applications. This future research has the potential to make the most of the capabilities of this algorithm for the benefit of society. Furthermore, future research will explore the adoption of alternative membership function types, such as Gaussian, trapezoidal, or generalized bell functions, to enhance modelling flexibility and potentially improve classification accuracy. The explainability framework could also be extended to incorporate a broader spectrum of large language model architectures, beyond GPT-based systems, thereby broadening the scope of its applicability to diverse AI paradigms. Moreover, forthcoming studies will employ more rigorous experimental designs—integrating larger and more heterogeneous datasets, systematic parameter variation, and cross-domain evaluations—to achieve a deeper and more comprehensive understanding of the algorithm's generalisation capabilities. Finally, we will also include benchmarking the proposed approach against alternative metaheuristic optimisation methods such as PSO, Marine Predators Algorithm, Spiral Dynamic Search, and Moth Flame Optimisation to evaluate potential gains in convergence speed and accuracy.

In perspective, this study is projected to have a significant impact on the field of medicine in general, as well as sports medicine. This is due to the algorithm's ability to analyse and determine the unique characteristics that some athletes possess, which could be of great interest. Consequently, this information would be of immense value for more informed decision-making by coaches and sports physicians, who could use it to assess player participation in sporting events more accurately and effectively.

Acknowledgements This work has been supported by the grant PID2022-139297OB-I00 funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU. Moreover, it is part of the project C-ING-165-UGR23, co-funded by the Regional Ministry of University, Research and Innovation and by the European Union under the Andalusia ERDF Program 2021-2027.

Funding Funding for open access publishing: Universidad de Granada/CBUA.

References

1. Winston PH (1984) Artificial Intelligence. Addison-Wesley Longman Publishing Co., Inc, USA
2. Hamet P, Tremblay J (2017) Artificial intelligence in medicine. Metabolism 69:36–40
3. Wu T, He S, Liu J, Sun S, Liu K, Han Q-L, Tang Y (2023) A brief overview of chatgpt: The history, status quo and potential future development. IEEE/CAA Journal of Automatica Sinica 10(5):1122–1136
4. Shoufan A (2023) Exploring students' perceptions of chatgpt: Thematic analysis and follow-up survey. IEEE Access
5. LeCun Y, Bengio Y, Hinton G (2015) Deep learning nature 521(7553):436–444
6. Biswas SS (2023) Role of chat gpt in public health. Ann Biomed Eng 51(5):868–869
7. Lawrence J (1993) Introduction to Neural Networks
8. Dignum V (2018) Ethics in artificial intelligence: introduction to the special issue. Ethics Inf Technol 20(1):1–3
9. Zou J, Schiebinger L (2018) AI can be sexist and racist—it's time to make it fair. Nature Publishing Group UK London
10. Gunning D, Stefik M, Choi J, Miller T, Stumpf S, Yang G-Z (2019) Xai–explainable artificial intelligence. Sci Robot 4(37):7120
11. Gunning D (2017) Explainable artificial intelligence (xai). Defense advanced research projects agency (DARPA), nd Web. 2(2):1
12. Leung KM et al (2007) Naive bayesian classifier. Polytechnic University Department of Computer Science/Finance and Risk Engineering. 2007:123–156
13. Kosko B (1994) Fuzzy systems as universal approximators. IEEE Trans Comput 43(11):1329–1333
14. Zadeh L (1965) Fuzzy sets. Inform. Control 8:338–353
15. Klir GJ, Yuan B (1996) Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by lotfi a zadeh. World Scientific. 6
16. Wu J, Brackstone M, McDonald M (2000) Fuzzy sets and systems for a motorway microscopic simulation model. Fuzzy Sets Syst 116(1):65–76
17. Liu X (1998) The fuzzy sets and systems based on afs. structure, ei algebra and eii algebra. Fuzzy Sets and Systems. 95(2):179–188
18. Passino KM, Yurkovich S, Reinfrank M (1998) Fuzzy control. Addison-wesley Reading, MA, p 42
19. Mudi RK, Pal NR (2002) A robust self-tuning scheme for pi-and pd-type fuzzy controllers. IEEE Trans Fuzzy Syst 7(1):2–16
20. Feng G (2006) A survey on analysis and design of model-based fuzzy control systems. IEEE Trans Fuzzy Syst 14(5):676–697
21. Boukezzoula R, Jaulin L, Coquin D (2022) A new methodology for solving fuzzy systems of equations: Thick fuzzy sets based approach. Fuzzy Sets Syst 435:107–128
22. Shi K, Wang J, Zhong S, Tang Y, Cheng J (2020) Non-fragile memory filtering of ts fuzzy delayed neural networks based on switched fuzzy sampled-data control. Fuzzy Sets Syst 394:40–64
23. Sakthivel R, Abinandhitha R, Harshavarthini S, Mohammadzadeh A, Saat S (2023) Anti-disturbance observer-based finite-time reliable control design for fuzzy switched systems. Fuzzy Sets Syst 471:108673
24. Terano T, Asai K, Sugeno M (1992) Fuzzy systems theory and its applications. Academic Press Professional, Inc
25. Munakata T, Jani Y (1994) Fuzzy systems: an overview. Commun ACM 37(3):69–77
26. Zhao T, Chen C, Cao H (2022) Evolutionary self-organizing fuzzy system using fuzzy-classification-based social learning particle swarm optimization. Inf Sci 606:92–111
27. Mirjalili S, Mirjalili S (2019) Genetic algorithm. Theory and Applications, Evolutionary Algorithms and Neural Networks, pp 43–55
28. Giráldez-Cru J, Chica M, Cordón O (2023) An integrative decision-making mechanism for consumers' brand selection using 2-tuple fuzzy linguistic perceptions and decision heuristics. Int J Fuzzy Syst 25(1):59–79
29. Herrera F, Martinez L (2000) An approach for combining linguistic and numerical information based on the 2-tuple fuzzy linguistic representation model in decision-making. Internat J Uncertain Fuzziness Knowledge-Based Systems 8(05):539–562
30. Zheng Y, Xu Z, Wu T, Yi Z (2024) A systematic survey of fuzzy deep learning for uncertain medical data. Artif Intell Rev 57(9):230
31. Cao J, Zhou T, Zhi S, Lam S, Ren G, Zhang Y, Wang Y, Dong Y, Cai J (2024) Fuzzy inference system with interpretable fuzzy rules: Advancing explainable artificial intelligence for disease diagnosis–a comprehensive review. Inf Sci 662:120212
32. Zadeh LA (1988) Fuzzy logic. Computer 21(4):83–93
33. Sanz JA, Bustince H (2021) A wrapper methodology to learn interval-valued fuzzy rule-based classification systems. Appl Soft Comput 104:107249
34. Ouifak H, Idri A (2025) A comprehensive review of fuzzy logic based interpretability and explainability of machine learning techniques across domains. Neurocomputing, 130602
35. Tyrovolas M, Kallimanis ND, Stylios C (2024) Advancing explainable ai with causal analysis in large-scale fuzzy cognitive maps. arXiv preprint arXiv:2405.09190
36. Jiao L, Pan Q, Denoeux T, Liang Y, Feng X (2015) Belief rule-based classification system: Extension of frbcs in belief functions framework. Inf Sci 309:26–49
37. Goel N, Sehgal P (2015) Fuzzy classification of pre-harvest tomatoes for ripeness estimation-an approach based on automatic rule learning using decision tree. Appl Soft Comput 36:45–56

38. Tanaka K, Wang HO (2004) Fuzzy control systems design and analysis: a linear matrix inequality approach
39. Ross TJ (2005) Fuzzy logic with engineering applications
40. Chi Z, Yan H, Pham T (1996) Fuzzy Algorithms: with Applications to Image Processing and Pattern Recognition vol. 10
41. Hühn J, Hüllermeier E (2009) Furia: an algorithm for unordered fuzzy rule induction. Data Min Knowl Disc 19:293–319
42. Trillo JR, Fernandez A, Herrera F (2020) Hfer: Promoting explainability in fuzzy systems via hierarchical fuzzy exception rules. In: 2020 IEEE International conference on fuzzy systems (FUZZ-IEEE), IEEE, pp 1–8
43. Elkano M, Galar M, Sanz J, Bustince H (2018) Chi-bd: A fuzzy rule-based classification system for big data classification problems. Fuzzy Sets Syst 348:75–101
44. Elkano M, Galar M, Sanz J, Bustince H (2016) Fuzzy rule-based classification systems for multi-class problems using binary decomposition strategies: on the influence of n-dimensional overlap functions in the fuzzy reasoning method. Inf Sci 332:94–114
45. Rio S, López V, Benítez JM, Herrera F (2015) A mapreduce approach to address big data classification problems based on the fusion of linguistic fuzzy rules. Int J Comput Intell Syst 8(3):422–437
46. Sanz JA, Fernández A, Bustince H, Herrera F (2010) Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning. Inf Sci 180(19):3674–3685
47. Esteva F, Godo L (2001) Monoidal t-norm based logic: towards a logic for left-continuous t-norms. Fuzzy Sets Syst 124(3):271–288
48. Mahmood T, Rehman U, Ali Z, Aslam M, Chinram R (2022) Identification and classification of aggregation operators using bipolar complex fuzzy settings and their application in decision support systems. Mathematics 10(10):1726
49. Cordón O, Herrera F, Villar P (2001) Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. IEEE Trans Fuzzy Syst 9(4):667–674
50. Angelov P, Yager R (2012) A new type of simplified fuzzy rule-based system. Int J Gen Syst 41(2):163–185
51. Whitley D (1994) A genetic algorithm tutorial Stat Comput 4:65–85
52. Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications 80:8091–8126
53. Ojugo A, Eboka A, Okonta O, Yoro R, Aghware F (2012) Genetic algorithm rule-based intrusion detection system (gaids). Journal of Emerging Trends in Computing and Information Sciences 3(8):1182–1194
54. Herrera F, Martínez L (2000) A 2-tuple fuzzy linguistic representation model for computing with words. IEEE Trans Fuzzy Syst 8(6):746–752
55. Kramer O, Kramer O (2017) Genetic Algorithms
56. Kozeny V (2015) Genetic algorithms for credit scoring: Alternative fitness function performance comparison. Expert Syst Appl 42(6):2998–3004
57. Beasley D, Bull DR, Martin RR (1993) An overview of genetic algorithms: Part 1, fundamentals. University computing 15(2):56–69
58. Leardi R (2000) Application of genetic algorithm-pls for feature selection in spectral data sets. J Chemom 14(5–6):643–655
59. Srinivas M, Patnaik LM (1994) Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans Syst Man Cybern 24(4):656–667
60. Zhang J, Chung HS-H, Lo W-L (2007) Clustering-based adaptive crossover and mutation probabilities for genetic algorithms. IEEE Trans Evol Comput 11(3):326–335
61. Hong T-P, Wang H-S, Lin W-Y, Lee W-Y (2002) Evolution of appropriate crossover and mutation operators in a genetic process. Appl Intell 16:7–17
62. Sivanandam S, Deepa S, Sivanandam S, Deepa S (2008) Genetic Algorithms
63. Park J-B, Park Y-M, Won J-R, Lee KY (2000) An improved genetic algorithm for generation expansion planning. IEEE Trans Power Syst 15(3):916–922
64. Bhandari D, Murthy C, Pal SK (1996) Genetic algorithm with elitist model and its convergence. Int J Pattern Recognit Artif Intell 10(06):731–747
65. Keane AJ (1995) Genetic algorithm optimization of multi-peak problems: studies in convergence and robustness. Artif Intell Eng 9(2):75–83
66. Ramesh A, Kambhampati C, Monson JR, Drew P (2004) Artificial intelligence in medicine. Ann R Coll Surg Engl 86(5):334
67. Malik P, Pathania M, Rathaur VK et al (2019) Overview of artificial intelligence in medicine. Journal of family medicine and primary care 8(7):2328
68. Holzinger A, Langs G, Denk H, Zatloukal K, Müller H (2019) Causability and explainability of artificial intelligence in medicine. Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery 9(4):1312
69. Szolovits P (2019) Artificial intelligence in medicine. Routledge
70. Schwartz WB, Patil RS, Szolovits P (1987) Artificial intelligence in medicine. Mass Medical Soc
71. Fernández A, López V, Galar M, Del Jesus MJ, Herrera F (2013) Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. Knowl-Based Syst 42:97–110
72. Alcalá-Fdez J, Alcala R, Herrera F (2011) A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. IEEE Trans Fuzzy Syst 19(5):857–872
73. Quteishat A, Lim CP (2008) A modified fuzzy min-max neural network with rule extraction and its application to fault detection and classification. Appl Soft Comput 8(2):985–995
74. Chang L, Zhang L (2021) Explainable data-driven optimization for complex systems with non-preferential multiple outputs using belief rule base. Appl Soft Comput 110:107581
75. Whitley D, Rana S, Dzubera J, Mathias KE (1996) Evaluating evolutionary algorithms Artificial intelligence 85(1–2):245–276
76. Eshelman LJ, Schaffer JD (1993) Real-coded Genetic Algorithms and Interval-schemata, vol 2. Elsevier, Dutch, pp 187–202
77. Lozano M, Herrera F, Krasnogor N, Molina D (2004) Real-coded memetic algorithms with crossover hill-climbing. Evol Comput 12(3):273–302
78. Khan S, Asim M, Khan S, Musyafa A, Wu Q (2023) Unsupervised domain adaptation using fuzzy rules and stochastic hierarchical convolutional neural networks. Comput Electr Eng 105:108547
79. Al-Ashoor A, Lilik F, Nagy S (2025) A systematic analysis of neural networks, fuzzy logic and genetic algorithms in tumor classification. Appl Sci 15(9):5186
80. Ghasemi A, Hashtarkhani S, Schwartz DL, Shaban-Nejad A (2024) Explainable artificial intelligence in breast cancer detection and risk prediction: A systematic scoping review. Cancer Innovation 3(5):136
81. Stone M (1978) Cross-validation: A review. Statistics: A J Theor Appl Stat 9(1):127–139