Human in the Loop Fuzzy Pattern Tree Evolution

Aidan Murphy¹ · Gráinne Murphy¹ · Douglas Mota Dias^{1,2} · Jorge Amaral² · Enrique Naredo¹ ·
Conor Ryan¹

Received: 17 July 2021 / Accepted: 21 January 2022 / Published online: 14 February 2022
© The Author(s) 2022

Abstract

Fuzzy pattern trees evolved using grammatical evolution, a system we call Fuzzy Grammatical Evolution, are shown to be a robust Explainable Artificial Intelligence technique. Experimental results show Fuzzy Grammatical Evolution achieves competitive results when compared against SVM, Random Forest and Logistic Regression on a set of real world benchmark problems. Fuzzy Grammatical Evolution allows for human input throughout the evolutionary process. Regularization methods and double tournament selection were investigated to determine what method was most successful at finding smaller, more interpretable models. A domain expert was recruited to investigate the interpretability of the models found and to give a confidence score for each model. This expert successfully identified overfit models, validating that Fuzzy Grammatical Evolution can be regarded a powerful Explainable Artificial Intelligence technique.

Keywords Grammatical evolution · Fuzzy logic · Explainable AI

Introduction

In the last 25 years there has been an explosion in the number of machine learning (ML) applications, with some form of ML deployed in most fields of research. The reasons for this are many. Massive amounts of data are available to train these algorithms which can run quickly and relatively inexpensively on massively parallel high-performance hardware. On certain benchmarks ML frameworks and systems can now accomplish near-perfect performance, often exceeding human performance. This has produced AI success stories in chess and Go [39]. It is predicted that global revenues for the AI market will surpass \$300 billion in 2024 [36].

Criticism of these results has also been widespread, though [20, 42]. Smaller and smaller errors often come with the sacrifice of transparency. These models are referred to as “black box” models as they do not allow their internal

workings to be investigated or understood directly by a human. Instead, they merely return input and output pairs. While systems exist which can offer an explanation as to why a particular decision was made by the black box, checking that the model exhibits “common sense” is not feasible and requires the user to make a leap of faith to trust the model. It also makes debugging and error checking an impossibility. These deficiency’s have lead to the creation and implementation of AI systems which exhibit racist and sexist behaviour [44].

While interpretability has been an important aspect since their inception, the interpretability of models has largely been sacrificed for improved performance on benchmarks.

To address and solve these issues a new area of ML research was spawned, XAI [1, 2]. XAI seeks to produce interpretable models and methods that can somehow explain themselves to humans. This must come with no, or minimal, impact on the models performance [4].

Fuzzy Pattern Trees, first introduced 15 years ago [13, 41], have recently been suggested as a technique which can potentially contribute in XAI [35]. Based on fuzzy set theory, a Fuzzy Pattern Tree is a hierarchical tree structure. This is in contrast to most other fuzzy-based systems which use rules as representations. As a fuzzy system, it uses linguistic labels and is more easily interpretable to humans. Clearly, the interpretability of the model is dependent on its

This article is part of the topical collection “Genetic Programming” guest edited by Aniko Ekart, Ting Hu and Nuno Lourenço.

✉ Aidan Murphy
aidan.murphy@ul.ie

¹ Lero & University of Limerick, Limerick, Ireland

² Rio de Janeiro State University, Rio de Janeiro, Brazil

complexity and size not being extremely large. Grammatical Evolution has been shown to be a very efficient and effective approach for evolving accurate, and, vitally, small Fuzzy Pattern Trees [22].

Our goal is to verify the assertion that Fuzzy Pattern Trees are intrinsically interpretable in their own right. It further aims to show this interpretability can aid in the evolutionary process by finding faults in the best performing individuals found by the search. It allows for the inclusion of human expertise and feedback and uses this to generate improved results. The system was tested on various real-live fairness benchmarks. The results show that Fuzzy Pattern Trees may allow the identification of data or algorithmic bias that may be present in final models.

This is an extended version of a paper published in the proceedings of EuroGP 2021 [23]. The previous work is built upon by including the human in more facets of the evolutionary run. The expert was asked to design the fuzzy partitions and to create modules which could be added to the grammar to help guide the search.

The next section reviews the main background concepts GE, Fuzzy GE and XAI. The following section explains the proposals and describes the paper's contributions in more detail. Next section presents the experimental set-up. It outlines all performance measures which were investigated. The following section presents the main results of the experiments described in the next section. Finally, the last section summarises the research and discusses future work suitable for investigation.

Background

Explainable Artificial Intelligence

Conferences, workshops and articles on the topic of XAI and ML model interpretability has grown and continues to grow rapidly [1, 12]. However, these talks, conferences, workshops and papers have existed in a space where the term 'interpretability' has not been agreed upon or even well-defined [6, 7, 10]. There is not yet a consensus as to when a model has been 'explained' fully or indeed what would constitute an 'explanation' [6]. The terms *interpretable*, *understandable* and *intelligible* are often used interchangeably or without distinction [18]. Depending on the context, interpretability may have a vastly different meaning; for example, a loan approving system may simply need to show that it is not discriminatory against any group and does not break any laws, whereas a safety-critical system may need to describe every step in its internal logic to a user [7].

According to Samek et al. [33], interpretation is the translation of abstract concepts into a domain humans can understand. In contrast, an explanation is the collection of aspects

of the interpretable domain that have led to the development of choice in a given case. In our work, the concept of interpretation is connected with this definition.

The main techniques used to "open" the black box can generally be categorized into three approaches; 1. Create intrinsically interpretable proxy model, 2. Give important features, 3. Create visualisation. These can further be divided into either global and local explanations. Creating an intrinsically interpretable model proxy, the most popular technique, attempts to replicate the logic of a black box in another form. That is to say, learn an intrinsically interpretable model (e.g. decision tree, linear model etc.) which mimics the predictions of the black box. This appears to give the user the best of both worlds; a highly accurate model but with an interpretable and familiar presentation. No extra expertise is necessarily needed to engage and interpret the outputs. This suite of techniques aim to create a global approximation to the original black box, and then use this approximation for inference. A very popular example of this is LIME (Local Interpretable Model-agnostic explanations) [28]. They derive an interpretation that is "locally faithful" by supplying a perturbed sample of points to a model black box and fitting an interpretable linear model to the outputs. Their system finds a local, interpretable representation in the vicinity of some point in their model space. By inspecting the coefficients of the linear model, it is possible to infer which are the most important attributes. However, depending on the decision boundary it is possible to have very different linear models for objects which are very close in the attribute space. In contrast, by inspecting the Fuzzy Pattern Tree structure and computing the membership values, it is possible to see the attributes' contribution to the classification for different objects on the same model, which can provide a more consistent explanation.

Many of the explanations put forward in papers require some ML expertise [17]. It is possible these explanations will be incomprehensible or appear incomplete to a user without a ML background, while other explanations might require some domain knowledge to fully interpret the results [21]. While a user of the model will undoubtedly have some domain knowledge, it is unclear if defining a model as interpretable implies that the user does have this knowledge.

There has been some work aiming to develop a rigid framework to define and evaluate interpretability and explainability [7]. The authors stress that a human centered approach, either human evaluation or human abstraction, is vital to any idea of judging interpretability.

Others argue that it is not sufficient for ML models to be interpretable and comprehensible but that they must also have logic and put forward some form of rationale to the user as to how a particular decision was arrived at [6].

Whether a model or automated system can be trusted is an important quality which is often neglected [28]. It is often

insufficient for a model to show high accuracy for a user to accept its outputs. To ensure the decisions of the model are trustworthy or ethical or legal a user may require evidence. If the model is a black box or its logic is presented in a way that makes it difficult or impossible for a human to abstract knowledge from its results, they may simply refuse to use it, or may not be allowed to under the General Data Protection Regulation, GDPR, [11].

If a domain expert is working in collaboration with a model, knowledge of its logic or internal workings is key to enable them to know when the model will predict something sub-optimally. More importantly, perhaps, it may also allow them to know the bounds of knowledge of the model, in what areas the model will fail and when its outputs can be discarded or ignored.

For a model to be useful, usable and fully ‘explained’ we propose it should have the following properties: it must be transparent in its internal workings; have equivalent or superior accuracy to any other model; be cogent in its statements (particularly for finding complex relationships); be able to incorporate domain knowledge in its learning process; and be deterministic.

Grammatical Evolution

Grammatical Evolution (GE) [30], is a well known and popular evolutionary computation technique, often thought of as a variant of genetic programming (GP) [16]. As with many evolutionary algorithms its primary influence comes from nature, with GE drawing inspiration from genetics. GE takes binary strings and, using a grammar, builds computer programs from them. A key selling point of GE is its ability to produce programs in any arbitrary language, usually specified using a Backus-Naur Form (BNF) grammar [26, 31] or Attribute Grammar (AG) [27]. The traditional evolutionary operators of crossover and mutation occur on the string, not the final computer program which is executed. The string and the grammar combine to create the program. Therefore any representation the user wishes to use for the solution to aid interpretability is possible. The traditional and by far the most popular way to represent the solutions is to use tree structures [32]. These structures have been shown to be the most easily interpretable representations by humans [14] and makes them quite transparent.

Therefore GE, or GP in general, may represent a better ML algorithm to leverage a humans ability to generalize/abstract with the strengths of computers.

GE searches for solutions which best minimise (or maximise depending on the goal) a given target function. GE, however, allows for much more knowledge and nuance to be integrated into its objective function than the usual validation or test case accuracy. Multi-objective [5] and many objective [15] optimisation are common in EAs. It would be

straightforward to find and present to a user what trade-off, if any, had to be made to accommodate this interpretability by creating a Pareto front, an attractive feature pointed out by [7], and illustrated in Fig. 1. The user has their pick of solutions. This allows the final programs to have the highest possible precision all the while keeping their valuable interpretability. GE can be trained to be highly accurate on particular cases if they are of importance to the user or critical in the domain. The objective function has the flexibility to allow the user to personalize their goals more than traditional ML models by including ethics, fairness, legality, profit etc. as an integral part of the search.

It is difficult to exactly define concepts like fairness. The user has the freedom to define and implement these ideas in a way which is important to them in GE. This system also enables human factors to be added into the solution [8]. The interactive GA is a technique in which the user is directly responsible for giving a fitness score to each individual [40]. Integrating an interactive GA would mean solutions could take the form which maximises the utility of the person working with them. That is to say, there would be pressure on the search to look for solutions which have the structure that would be most suitable to the user interacting with it. All of this helps to build a personalized explanation, seen as a key facet in XAI [37].

This would normalise the concept of interpretability to the user and allow accuracy to be improved instead of trying to improve the vague idea of interpretability while keeping accuracy high. GE may involve the user at every stage of its learning process. The user may create the grammar and specify the target before the search begins. The user may change the meta-parameters, augment the grammar, impart

Fig. 1 Graph of Pareto front of error and size for the recidivism dataset. The exact decrease in error can be seen with increasing complexity. This allows the expert user to explicitly observe the trade-off between error and size

domain knowledge in the form of subtrees or modules during run time[24]. The user would be the main architect of the structure and appearance of the solutions. To the user, all this would build and solidify ‘trust’ in the model, if it is needed.

Fuzzy GE

GE was recently introduced as a method to evolve Fuzzy Pattern Trees [22]. This approach, named Fuzzy Grammatical Evolution, was shown to produce competitive performance compared with state of the art black box methods and exceeded the performance of another GP variant, Cartesian GP, on a set of benchmark classification problems [34].

A Fuzzy Pattern Tree is a hierarchical, tree-like structure. The internal nodes are fuzzy logical and fuzzy arithmetic operators and the leaf nodes are the fuzzified input variables and constants. Like traditional GP or GE trees, the information passes from the bottom of the tree to the top. An operator node takes the value or values of its descendants as inputs, performs the required operation, and conveys the result to its preceding node. Thus, a Fuzzy Pattern Tree implements a recursive mapping producing outputs in the $[0,1]$ interval. In a Fuzzy Pattern Tree, it is possible to infer the contribution of each attribute.

Fuzzy Pattern Trees are a Fuzzy Rule Classification technique, with the most most popular of these classifiers *fuzzy rule-based systems* (FRBS) and *fuzzy decision trees* (FDTs).

FDTs are an extension of standard decision trees. Despite their similar hierarchical structure, they are quite different that Fuzzy Pattern Trees. They follow a top down approach and work by continually partitioning the domain to build their single classifier.

FRBS are rule based classifiers and are flat structures which use a fuzzy rule base to model the relationships in the data. Despite their obvious differences in representation (flat vs hierarchical) it has been shown that Fuzzy Pattern Trees are a generalization of rule-based systems [38].

Unlike conventional GE classification techniques which require one expression to be evolved and a boundary or threshold to be picked, Fuzzy Grammatical Evolution requires numerous Fuzzy Pattern Trees to be created. To classify an individual in traditional GE a boundary or boundaries are decided upon. The output of the single tree evolved is then compared against this boundary and a decision is made about its classification. There are many downsides to this approach, much time, effort and expertise is required to optimise these boundaries [9]. In classic GE classification one expression is found regardless of the number of classes present, Fuzzy Grammatical Evolution needs one Fuzzy Pattern Tree for each class that exists in the dataset. These Fuzzy Pattern Trees serve as the logical description of the class.

Fuzzy Grammatical Evolution evolves one, large solution and treats the subtrees of this solution as it's Fuzzy Pattern Trees, as seen in Fig. 2. The Fuzzy Pattern Tree which returns the largest score for an individual is declared the winner and that individual is designated as belonging to that class. This is illustrated in Fig. 3, with the largest score belonging to FT_c and therefore that instance being designated as class c . The root node of the tree, the *WinnerTakesAll* function, is responsible for this process. This function receives the score from each Fuzzy Pattern Tree and labels the individual corresponding to the highest scoring tree. Representing each Fuzzy Pattern Tree as subtrees of one large solution combined with GE's inbuilt separation between search space and program space leads to another major advantage Fuzzy Grammatical Evolution experiences. We use standard crossover and mutation operators and do not require the use of any protected operators. To tackle different problem specifications all that is required is to make a small grammar augmentation.

Explainable GE

Reducing Size of Trees

Each Fuzzy Pattern Tree serves as the class descriptor for each class in the benchmark problem. The interpretability of the Fuzzy Pattern Tree is therefore of upmost importance. Interpretability is only achieved if the sizes of the Fuzzy Pattern Trees are small and are kept as compact as possible. Previous experiments have shown that conventional

Fig. 2 Pictorial representation of a multi-classifier evolved by fuzzy grammatical evolution. This example has c classes so necessitates the creation of c fuzzy pattern trees, FT_1 to FT_c , with each fuzzy pattern tree given a different colour. FT_1 is the fuzzy pattern tree concerning class 1, FT_2 is the Fuzzy Pattern Tree concerning class 2 and so on. The root node is the winner take all (WTA) function, which assigns the individual to the class corresponding to the highest output of FT_1 to FT_c . For instance if a problem had 3 classes we need to create 3 Fuzzy Pattern Trees, FT_1 , FT_2 and FT_3 . If FT_1 has an output of 0.2, FT_2 an output of 0.6 and FT_3 an output of 0.3 for a particular instance, then the WTA function will assign this instance as class 2, as it has the highest score

Fig. 3 Graphical depiction of the mapping process from the feature space to a 1-dimensional space $[0,1]$ using a set of fuzzy trees FT_1 to FT_c

GE may be bloating individuals, yielding trees which are excessively large and contain worthless material [22]. The addition of a simple parsimony pressure was seen to greatly reduce the size of individuals without having a noticeable effect on their fitness. However, it was not investigated if this simple procedure is the most effective method for producing smaller, accurate Fuzzy Pattern Trees. We investigate various approaches for producing smaller trees and compares them against conventional GE and against each other. This comparison considers both their size and accuracy. Each method is outlined below.

Intron Removal

There often exist fragments of GP individuals which exert no effect on that individual's final output. That is to say, it is a redundant piece of the individual. These are called introns. Despite their lack of involvement they can play an important part in evolution, by insulating important parts of an individual from destruction by crossover or mutation [25]. However, it is necessary to remove them from individuals when trying to interpret them as they may cause unnecessary confusion.

In the case of Fuzzy Pattern Trees, the internal nodes are fuzzy operators. Therefore, the process of removing introns consisted of finding nested complementary operators and removing them. An example would be $\text{complement}(\text{complement}(x))$ which would simply be converted x .

Strict/Easy Regularization

Regularization is a popular technique which adds a penalty term in the fitness function to counteract overfitting [43]. Two types of regularisation were used in our experiments. The first was an easy regularization where a small penalty to the fitness of an individual was applied based on its maximum depth. The second was a strict regularization. This procedure set the fitness to the minimum (0 in this case) if the max depth of the individual exceeded a specified threshold. Fitness was calculated as usual if it was equal to or below this threshold.

Double Tournament

The last method for bloat control which was investigated was double tournament [19]. This strategy conducts two tournaments, one of which chooses the individual with the highest fitness while the other chooses the one with the smallest size. Both potential orders of series were investigated, that is, first fitness and secondly size, and vice versa.

Fuzzy Pattern Tree Representation

A Fuzzy Pattern Tree uses a hierarchical tree structure, differentiating it from other fuzzy based classifiers. The fuzzified problem variables make up the leaf nodes of these trees and the inner nodes are fuzzy logical and arithmetic operators. As with regular GP classifiers, the information is propagated from the bottom to the top. The output of each Fuzzy Pattern Tree is confined to the $[0,1]$ interval. More formally, a Fuzzy Pattern Tree maps $f_i(\mathbf{x}) : \mathbb{R}^n \rightarrow [0, 1]$, where x are the input variables.

The following operators are used, where a and b are the inputs to the operator:

$$WTA = IF\{\}\dots ELSE() \quad (1)$$

$$MAX = \max(a, b) \quad (2)$$

$$MIN = \min(a, b) \quad (3)$$

$$WA(k) = ka + (1 - k)b \quad (4)$$

$$OWA(k) = k \cdot \max(a, b) + (1 - k)\min(a, b) \quad (5)$$

$$CONCENTRATE = a^2 \quad (6)$$

$$DILATE = a^{\frac{1}{2}} \quad (7)$$

$$COMPLEMENT = 1 - a \quad (8)$$

where WTA , WA & OWA denote **Winner Takes All**, **Weighted Average** and **Ordered Weighted Average**, respectively, and k is a randomly created value in $(0,1)$.

Figure 4 shows an example of an Fuzzy Pattern Tree which was trained on the **Heart** benchmark dataset. It represents the fuzzy concept - a fuzzy criterion for - the presence of heart disease. This tree was picked as it was considered as very interpretable by a domain expert, who was also very confident in the logic of this model.

An interpretation of this tree could be:

The presence of heart disease is strongly determined by three criteria. The first criteria is that a reversible defect was found while conducting a Thallium Stress Test. The second is that the number of major vessels coloured by fluoroscopy was moderate. The third criteria is the slope of the peak exercise ST segment was flat. Criteria I and II are the major contributors to the decision roughly contributing equally, with criterion III has a small but not insignificant effect.

Grading Individuals

It is unsatisfactory to simply state that Fuzzy Pattern Trees are interpretable *a priori*. Therefore, an expert was sought to empirically validate the interpretability of the Fuzzy Pattern Trees. A domain expert, a Registrar in a local hospital with previous experience working in a cardiology department of a university hospital, was found to examine the results from the Heart experiments. After each run, the fittest individual was stored, parsed into a tree and saved as an image. This gave 150 graphs of trees, 30 for each of the five selection

Fig. 4 Tree representing the interpretable class “Presence of Heart Disease”. Gray boxes show the actual operators, while the coloured boxes show each variable

Fig. 5 Fuzzy pattern tree classifier with *Interpretability score* of 5

methods, which were then presented to the expert. The evaluation was a two step process. First, the expert was asked to evaluate the trees in terms of *interpretability*. Afterwards, the expert was asked to grade the *logic* of the model. That is to say, do the variables and operators present in the tree make medical sense and if it would help diagnostically. The expert was told to score these from 1, lowest, to 5, highest. This novel two-step approach aimed to confirm interpretability by allowing the expert to identify overfit models, those with incorrect logic. This identification was confirmed by comparing the test accuracy of models with correct and incorrect logic.

Interpretability

The expert was first asked to evaluate the *interpretability* of the best models found. That is to say, could they follow the flow of the tree and understand what input variables it was aggregating and how it was doing it. This was heavily influenced by the depth of the trees and the operators used. A score of 5 would mean the expert could confidently predict what the output of the tree would be for a given input. A score of 1 would mean the opposite. A score of 4 would mean that a tree was mostly straightforward to follow but may have one or two difficult aggregations that may complicate some inputs. A score of 3 would mean that the trees were mostly interpretable but had some branches which are large or have complicated aggregations. Finally, a score of 2 would mean the tree was mostly uninterpretable but had some branches which were easy to follow. Some example are shown below.

Figure 5 shows an Fuzzy Pattern Tree classifier which was deemed as having an *Interpretability score* of 5. It considers *Thal*, *STdepression* and *MajVesselsColFluoroscopy* when creating a score for *no presence of heart disease*, the left subtree.

To create the Fuzzy Pattern Tree for *presence of heart disease*, the right subtree, the variables *Thal*, *STdepression* and *ChestPain* are used. Only two operators are used across both Fuzzy Pattern Trees, *OWA* is used in both while the left Fuzzy Pattern Tree uses the *min* operator.

In contrast, Fig. 6 shows an Fuzzy Pattern Tree classifier which was judged as having an *Interpretability score* of 1. While being difficult to grasp even pictorially and

Fig. 7 Fuzzy pattern tree classifier with *Interpretability score* of 5 and *Logic score* of 5

containing far more variables than Fig. 5, it also contains many *complement* operators at various points in the tree. This “logic reversal” operator means it is almost impossible for an expert to follow the logic of the model as they traverse the tree.

Logic

The expert was next asked to give each model a score on their *logic*. That is to say, does the model use the same variables and exhibit the same logic the expert would use when diagnosing the patient. This is a very important characteristic, as if the user is unsure or dubious of the model’s logic it will mean that the user will never be able to fully trust, or trust at all, the predictions that the model makes. Obviously, all models which have an *interpretability score* of 1 were not given a *logic score* which would deem them as having correct logic, as the expert was unable to establish what the logic of the model actually was. If a tree did exhibit the same logic the expert would use it was given a score of 5. If a tree contained all the correct variables but was a bit large or contained a difficult aggregation it was given a score of 4. If it contained most, but not all, important variables it was given a score of 3. A score of 2 was awarded if some key variables were absent and complicated aggregations existed in the tree.

An example of a classifier with an *interpretability score* of 5 and a *logic score* of 5 can be seen in Fig. 7. This classifier considers the results of the two diagnostic tests, *Thal* and *MajVesselsColFluoroscropy*, in both Fuzzy Pattern Trees. Figure 8 shows a classifier with *interpretability score* score of 5 and a *logic score* of 3. Although it correctly identifies

risk factors associated with *heart disease*, it only considers one diagnostic test, *Thal*, in the right Fuzzy Pattern Tree.

Human in the Loop

Once the interpretability of Fuzzy Pattern Tree has been validated the next step is to include the expert in more areas of the system. This begins before the run has commenced in the preprocessing of the data. The expert was first asked to design the partitions used to fuzzify the data. The expert was also invited to suggest modules, or subtrees, that they felt would be helpful or necessary in solving the problem.

Designing Fuzzy Partitions

In the previous experiments [22], the continuous data for each benchmark was partitioned into three fuzzy sets; Low, Medium and High. Each set had either a linear (Low and High) or triangular (Medium) membership function and the values for the limits of the membership functions were based on the distribution of the input data. There was no insight from a user, expert or otherwise, in these values. The expert, therefore was asked to design the membership functions and their values which best reflects real world medical categorization. As an example, the old and new partitions for *bloodpressure* are seen in Fig. 9. The Low and High sets have gone from linear membership functions to triangular memberships functions while the Medium set has gone from a triangular to a trapezoidal membership function.

Fig. 8 Fuzzy pattern tree classifier with *Interpretability score* of 5 and *Logic score* of 3

Constructing Modules

The expert was next asked to design modules which they believed would help Fuzzy Grammatical Evolution build better models. These were added to the grammar. The aim of these additions was to firstly build greater trust in the expert of the model's decisions and secondly to boost the interpretability of the trees by allowing information, which would usually be represented by a subtree of depth 2 or 3, to be replaced by a single node.

Experimental Setup

The claim that Fuzzy Grammatical Evolution creates interpretable models was investigated. That is to say, Fuzzy Grammatical Evolution provides a transparent model which can be understood by a user and reaches accuracy similar to other, black box, ML approaches. Various ways of adding the human in the evolutionary loop was also tested. Five selection methods were tested on six real world benchmark problems. These benchmarks were chosen as they have been identified as benchmark problems which often produce models which contain bias or some form of unwanted discrimination. Therefore, large value would be placed on gaining some insight into the logic of any model which was trained on these data.

Figure 10 shows the grammar used in the experimentation. Figure 11 shows the grammar with the addition of the expert produced modules. In all, six modules were added to

the grammar. One concerning the results of diagnostic tests and five referencing risk factors. The Fuzzy Pattern Trees for each class are contained within the *WTA* node, the $\langle exp \rangle$ non-terminals. Adding more $\langle exp \rangle$ symbols to the *WTA* node is all that is needed to extend the grammar for use in multi-class classification. For example, three classes would need the addition of one more $\langle exp \rangle$ symbol, four classes would need a further two symbols and so on. Digit concatenation was employed to generate constants [3].

The expert changed the fuzzy partitions of 4 variables, *Age*, *Resting Blood Pressure*, *Maximum Heart Rate* and *Cholesterol*. *Age* and *Resting Blood Pressure* remained with 3 fuzzy classes (Low, Medium and High) while *Maximum Heart Rate* and *Cholesterol* were reduced to 2 classes (Normal or High).

GE Parameters

The values for the experimental parameters can be seen in Table 1. A population size of 500 and for 50 generations were chosen. For each run, Sensible Initialisation was used to create the initial population and effective crossover was also employed [29]. The data was divided randomly at the beginning of each run, with 75% for training and the remaining 25% for test. This was repeated at the beginning of each run so that a different, randomized training and test data was used in each experiment. The exception was Census Income, which came with partitioned training and test data and was used as such. There was a total of 30 runs per experiment.

Two selection methods were employed, tournament selection and double tournament selection. Double tournament selection involved randomly selecting 4 individuals from the population and performing performing two, nested tournaments. For experiments FGE_{DT1} , the first tournament winners were decided by fitness with the second tournament

Fig. 9 Comparison of the standard fuzzy partitions, top image, against the expert designed fuzzy partitions

```

< start > ::= WTA(< exp >, < exp >)
< exp > ::= max(< exp >, < exp >) |
           min(< exp >, < exp >) |
           WA(< const >, < exp >, < exp >) |
           OWA(< const >, < exp >, < exp >) |
           concentrate(< exp >) |
           dilation(< exp >) |
           complement(< exp >) |
           x1 | x2 | x3 | ...
< const > ::= 0. < digit > < digit > < digit >
< digit > ::= 0 | 1 | 2 | ...

```

Fig. 10 Grammar used to evolve a fuzzy pattern tree. Extra $\langle exp \rangle$, as needed, can be added in the *WTA* node to make it a multi-class grammar

```

< start >::=WTA(< exp >,< exp >)
< exp >::=max(< exp >,< exp >) |
    min(< exp >,< exp >) |
    WA(< const >,< exp >,< exp >) |
    OWA(< const >,< exp >,< exp >) |
    concentrate(< exp >) |
    dilation(< exp >) |
    complement(< exp >) |
    x1 | x2 | x3 | ...
    module1 | module2 | ...
< const >::=0.< digit >< digit >< digit >
< digit >::=0 | 1 | 2 | ...

```

Fig. 11 Grammar used to evolve a fuzzy pattern tree with the experts modules included

Table 1 List of parameters used for fuzzy grammatical evolution

Parameter	Value
Runs	30
Total generations	50
Population	500
Elitism	Best individual
Selection	Tournament, Capitalise D & T
Crossover	0.9 (effective)
Mutation	0.01
Initialisation	Sensible

winner being the individual with the smaller size. FGE_{DT2} was the reverse of this approach. The first tournament compared the size of the individuals while the second tournament winner was chosen by whoever had the highest fitness.

Fitness Function

The fitness function used for each experiment, except FGE_{L1} & FGE_{L2} , was $1 - RMSE$. That is to say, FGE , FGE_{DT1} and FGE_{DT2} use the fitness function shown in Eq. 9.

$$F = 1 - RMSE \quad (9)$$

The fitness function for FGE_{L1} , is calculated to penalise solutions with a large size. It is computed as follows;

$$F_{L1} = 1 - RMSE - MaxDepth \times 0.001 \quad (10)$$

Finally, the fitness function for FGE_{L2} , is calculated to allow solutions attain a max depth of 2. It is computed as follows;

$$F_{L2} = \begin{cases} 1 - RMSE, & \text{if } MaxDepth < 3 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

The max depth of an individual is the largest path which exists in any Fuzzy Pattern Tree of that individual. While restricting the max depth of any Fuzzy Pattern Tree to just 2 may seem to lead to the creation of simple, perhaps trivial, models it must be noted that each classifier needs as many trees as there is classes. For example, binary classification requires the creation of 2 trees which means a classifier could have up to 8 factors, which may interact non-linearly.

Fairness Benchmarks

Six binary classification benchmark datasets were used for the experiments, all of which are available online in the UCI repository. An overview of the datasets can be seen in Table 2. These datasets are often used as benchmarks in AI fairness experiments, an area XAI could prove fruitful in.

Expert Validation

It is unsatisfactory to simply state that Fuzzy Pattern Trees are interpretable a priori. Therefore, an expert was sought out to empirically validate the interpretability of the Fuzzy Pattern Trees. A domain expert, a doctor working in a local hospital, was sought out to examine the results from the Heart experiments. After each run, the fittest individual was stored, parsed into a tree and saved as an image. This gave 150 graphs of trees, 30 for each of the 5 selection methods. These trees were then presented to the domain expert over Zoom. The evaluation was a two step process. First, the expert was asked to evaluate the trees in terms of interpretability. Afterwards, the expert was asked to grade the logic of the model. That is to say, do the variables and operators present in the tree make medical sense and would help diagnostically. Both of these were scored from 1, lowest, to 5, highest.

Results

The experimental results are summarized in Table 3 showing the mean best performance from 30 runs of Fuzzy Grammatical Evolution and the various selection methods as well as other ML approaches.

The first five columns show the results for Fuzzy Grammatical Evolution, Fuzzy Grammatical Evolution with fitness function in equation 10 applied, Fuzzy Grammatical Evolution with fitness function in equation 11 applied, Fuzzy Grammatical Evolution with double tournament selection, first considering size then fitness, and finally Fuzzy

Table 2 Benchmark datasets for the classification problems, taken from the UCI repository

Datasets	Short	Class	Vars	Instances
Bank marketing	Bank	2	20	41,188
Census income	Census	2	14	45,222
German credit	Credit	2	20	1000
ProRepublica recidivism	Recid	2	52	7214
ProRepublica violent recidivism	V/Recid	2	52	4743
Heart disease	Heart	2	13	303

Grammatical Evolution with double tournament selection, first fitness then size. The sixth column shows the results for Support Vector Machine (SVM) and seventh Random Forest classifiers. Finally, column eight shows the result of a logistic regression classifier.

No one selection method for Fuzzy Grammatical Evolution was seen to outperform any other selection method with respect to accuracy. FGE_{L2} performed best among all Fuzzy Grammatical Evolution variants, finding the mean best solutions in 4/6 benchmark problems. However, it also achieved the worst performance on the Recid problem. No selection method was seen to statistically significantly outperform any other.

A Friedman test was carried out on the data to compare the performance of all the classifiers. This test showed evidence that the Random Forest classifier was statistically significantly better than all others, achieving or matching the best performance on each problem. As a black box model, however, it does not enable any further extraction of knowledge. Fuzzy Grammatical Evolution was able to achieve very competitive results against the black box approaches. Fuzzy Grammatical Evolution was outperformed by 5% on the Credit dataset, achieving 71% in both Fuzzy Grammatical Evolution and FGE_{L2} compared to the best performing technique Random Forest which found 76%. Fuzzy Grammatical Evolution accomplished 81% accuracy on the Census problem, 4% worse than both SVM and Random Forest which obtained 85% accuracy. For the Bank, Recid and V/Recid problems, Fuzzy Grammatical Evolution evolved solutions which were within 2% of those found by either SVM or Random Forest.

On all but one problem Fuzzy Grammatical Evolution was seen to outperform the interpretable ML algorithm considered, logistic regression. Fuzzy Grammatical Evolution significantly exceed the performance of logistic regression on the Bank, Recid and V/Recid problems. Fuzzy Grammatical Evolution found better solutions on the Census dataset by 2%, 81% vs 79%, while both achieved parity on the Credit dataset, attaining 71% accuracy. The Heart problem was the sole exception to this, it was seen to favour LR by 2%.

The best of run individual underwent an intron removal process, outlined above, at the conclusion of each experiment. The aim of this process is to remove any bloat which may exist in the program. The mean size of the Fuzzy Pattern Trees in the best individual found by Fuzzy Grammatical Evolution in each of the 30 runs are shown in Table 4. The best results (smallest trees) are highlighted in bold. Unsurprisingly, as the most rigid selection technique, FGE_{L2} finds by far the smallest individuals. FGE_{L1} and FGE_{DT2} are next best at finding small individuals. They are, however, more than double the size of the solutions found by FGE_{L2} on average even after the removal of their introns.

The grades of the human expert's examinations of the trees can be seen in Table 5. The top performing technique for finding interpretable solutions was FGE_{L2} , scoring perfectly with all 30 runs finding trees attaining scores of 4 or 5. The next best performing method was FGE_{L1} , with 8 solutions being scored 4 or 5, followed by FGE_{DT1} , having 6 interpretable solutions. FGE_{DT2} and Fuzzy Grammatical Evolution were the poorest performing methods, both only finding 4 interpretable solutions in their 30 runs. A plot showing the decrease in interpretability as depth increases is seen in Fig. 12. The plot suggests that any reasonable expectation of interpretability disappears after trees have exceeded depth 5 or 6.

The next goal was to validate that the Fuzzy Pattern Trees were indeed transparent and clear in their statements. To accomplish this, the logic of the models which were adjudged to be interpretable (those scoring 4 or 5) were examined by the domain expert. 52 of the original 150 models met this criteria. Any models identified by the expert as having 'incorrect' logic, that is to say the confidence score was 1 or 2, were separated from the population. Similarly,

Table 3 Classification performance comparison of each selection method used with fuzzy grammatical evolution, showing the classification accuracy and standard deviation in brackets on the test data for the best solution found averaged across the 30 runs

Dataset	FGE	FGE_{L1}	FGE_{L2}	FGE_{DT1}	FGE_{DT2}	SVM	RandF	LogReg
Bank	0.89 (0.01)	0.89 (0.00)	0.89 (0.00)	0.89 (0.00)	0.89 (0.00)	0.91	0.91	0.84
Census	0.79 (0.03)	0.78 (0.03)	0.81 (0.01)	0.80 (0.02)	0.78 (0.02)	0.85	0.85	0.79
Credit	0.71 (0.01)	0.70 (0.02)	0.71 (0.01)	0.70 (0.02)	0.70 (0.02)	0.73	0.76	0.71
Recid	0.71 (0.02)	0.72 (0.02)	0.69 (0.02)	0.72 (0.01)	0.70 (0.02)	0.74	0.74	0.56
V/Recid	0.83 (0.01)	0.83 (0.01)	0.83 (0.01)	0.83 (0.01)	0.83 (0.01)	0.84	0.84	0.54
Heart	0.79 (0.04)	0.77 (0.05)	0.77 (0.05)	0.77 (0.05)	0.75 (0.06)	0.82	0.82	0.81

Table 4 Maximum depth size of the best solution found compared with each approach. Averaged over 30 runs, best results are in bold

Dataset	FGE	FGE _{L1}	FGE _{L2}	FGE _{DT1}	FGE _{DT2}
Bank	7.90	2.82	1.83	8.33	4.70
Census	10.13	6.70	1.96	9.80	8.03
Credit	10.63	7.53	2.00	12.90	10.67
Recid	10.90	5.33	2.00	10.13	10.33
V/Recid	8.17	3.97	2.00	9.40	8.57
Heart	10.63	5.85	2.00	9.40	8.57

Table 5 Count of interpretability scores for the best individual in each run for each selection type. There are 30 individuals for each selection type

Selection type	Interpretability score				
	1	2	3	4	5
FGE	19	2	5	4	0
FGE _{L1}	14	3	5	5	3
FGE _{L2}	0	0	0	2	28
FGE _{DT1}	20	1	3	4	2
FGE _{DT2}	19	1	6	3	1

models with borderline trust, those with a confidence score of 3, were separated.

This left 24 models remaining, described in Table 6, which were deemed interpretable and inspired confidence. The mean accuracy of those models is 77.1%, shown in Table 7. Including models with borderline confidence, those with confidence score of 3, the number of models jumps to 39, as shown in Table 6, and the mean accuracy marginally increased to 77.3%, as seen in Table 7. Models which have been deemed to have ‘correct’ logic perform ~2% better than those adjudged to have ‘incorrect’ logic. This is very notable as both groups contain almost identical fitness scores on the training data. By inspecting the models and judging their logic, an expert is able to improve the overall performance of the population by identifying models which are likely to be over-trained. As an added benefit, this process is an effective way for an expert to build trust in the models which are being evolved.

Now that the interpretability of Fuzzy Pattern Trees was established, the next step was to add the expert’s input in more stages of the evolution. The performance of Fuzzy Grammatical Evolution using the standard fuzzy partitions was compared with the performance of Fuzzy Grammatical Evolution using the expert designed fuzzy partitions. The performance of Fuzzy Grammatical Evolution when expert chosen modules are added to the grammar are also compared. For these experiments, the number of runs was increased from 30 to 100. The results are shown in Table 8.

Fig. 12 Decrease in the mean human interpretability determined by the expert as the maximum depth of the model increases. Error bars show the standard deviation of interpretability score for models at that depth**Table 6** Selection methods of fuzzy grammatical evolution models with interpretability score ≥ 4

Confidence score	FGE	FGE _{L1}	FGE _{L2}	FGE _{DT1}	FGE _{DT2}
≤ 2 (Incorrect logic)	2	2	5	2	2
≥ 4 (Correct logic)	0	3	19	1	1
≥ 3 (Correct & Marginal logic)	2	6	25	4	2

Table 7 Accuracy of fuzzy grammatical evolution models with interpretability score ≥ 4

Confidence score	Number	Accuracy
≤ 2 (Incorrect logic)	13	0.75
≥ 4 (Correct logic)	24	0.77
≥ 3 (Correct logic & Marginal logic)	39	0.77

A two sample Kolmogorov-Smirnov test was performed on all pairs of results and found they were not statistically significantly different from one another. Both standard dataset and expert curated dataset found best performance of 86% accuracy, but the addition of subtrees boosted this to 91%. Indeed this 91% was the best performance found by any one single classifier found by any technique examined in this study.

Table 8 Accuracy of fuzzy grammatical evolution with standard fuzzy partitions and expert designed fuzzy partitions

Partitions	Mean accuracy	Best
Standard	0.77	0.86
Expert	0.78	0.86
Expert & Subtree	0.76	0.91

Conclusion

We investigated the suitability of Fuzzy Grammatical Evolution as an explainable artificial intelligence approach by empirically analysing the interpretability of the Fuzzy Pattern Trees it produced. It also investigated methods of including a human in the evolutionary loop.

Experiments show that Fuzzy Grammatical Evolution has competitive performance on a selection of real world classification tasks. The best models found were then presented in comprehensible terms to a domain expert, a medical doctor. This expert was able to verify the interpretability of the models and to extract the knowledge obtained in the learning process of the model. Contrasting the performance of models which the domain expert labeled as having ‘incorrect’ logic vs those the domain expert labeled as ‘correct’ logic validated the interpretability of the Fuzzy Pattern Trees. Models with ‘incorrect’ logic were outperformed by those models deemed as having correct ‘correct’ logic, thus demonstrating Fuzzy Grammatical Evolution as an explainable artificial intelligence method. Techniques to include the human in the loop, namely designing fuzzy partitions and creating modules, were also explored. The effect these had on improving the search was inconclusive, although the addition of the expert partitions and modules did lead to the discovery of the best performing classifier.

The next major step in this work is the inclusion of the human expert in more stages of the evolutionary process. Further pre-processing of the data, highlighting important cases in the training set, encapsulating more information into modules and incorporating more domain knowledge in the grammar, setting and adapting maximum depth size of the individuals during the run, being involved in the selection process are some of the many possibilities going forward. This would enable Grammatical Evolution to tailor its search to the expertise and capabilities of the user.

Funding Open Access funding provided by the IReL Consortium. Funding was provided by Lero (Grant numbers: 13/RC/2094, 16/IA/4605), and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Finance Code 001).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Adadi A, Berrada M. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*. 2018;6:52138–60.
2. Arrieta AB, Díaz-Rodríguez N, Ser JD, Bennetot A, Tabik S, Barbado A, Garcia S, Gil-Lopez S, Molina D, Benjamins R, Chatila R, Herrera F. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf Fusion*. 2020;58:82–115.
3. Azad R.M.A, Ryan C. The best things don't always come in small packages: Constant creation in grammatical evolution. In: M. Nicolau, K. Krawiec, M.I. Heywood, M. Castelli, P. García-Sánchez, J.J. Merelo, V.M. Rivas Santos, K. Sim (eds.) *Genetic Programming*, pp. 186–197. Springer Berlin Heidelberg, Berlin, Heidelberg 2014.
4. Carvalho DV, Pereira EM, Cardoso JS. Machine learning interpretability: a survey on methods and metrics. *Electronics*. 2019;8(8):832.
5. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput*. 2002;6(2):182–97.
6. Doran D, Schulz S, Besold T.R. What does explainable AI really mean? A new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794* 2017.
7. Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* 2017.
8. Dou R, Zong C, Li M. Application of an interactive genetic algorithm in the conceptual design of car console. Tianjin: Tianjin University; 2014.
9. Fitzgerald J, Ryan C. Exploring boundaries: optimising individual class boundaries for binary classification problem. In: *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, 2012;pp. 743–750.
10. Gilpin L.H, Bau D, Yuan B.Z, Bajwa A, Specter M, Kagal L. Explaining explanations: An overview of interpretability of machine learning. In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89. IEEE 2018.
11. Goodman B, Flaxman S. European union regulations on algorithmic decision-making and a “right to explanation.” *AI Magazine*. 2017;38(3):50–7.
12. Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. *ACM Comput Surv (CSUR)*. 2018;51(5):93.

13. Huang Z, Gedeon TD, Nikraves M. Pattern trees induction: A new machine learning method. *Trans Fuz Sys*. 2008;16(4):958–70. <https://doi.org/10.1109/TFUZZ.2008.924348>.
14. Huysmans J, Dejaeger K, Mues C, Vanthienen J, Baesens B. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decis Support Syst*. 2011;51(1):141–54.
15. Ishibuchi H, Tsukamoto N, Nojima Y. Evolutionary many-objective optimization: A short review. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 2419–2426. IEEE 2008.
16. Koza J.R, Koza J.R. Genetic programming: on the programming of computers by means of natural selection, vol. 1. Cambridge: MIT press; 1992.
17. Krakovna V, Doshi-Velez F. Increasing the interpretability of recurrent neural networks using hidden markov models. *arXiv preprint arXiv:1606.05320* 2016.
18. Lipton ZC. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*. 2018;16(3):31–57.
19. Luke S, Panait L. Fighting bloat with nonparametric parsimony pressure. In: International Conference on Parallel Problem Solving from Nature, pp. 411–421. Springer 2002.
20. Marcus G. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631* 2018.
21. Moore A, Murdock V, Cai Y, Jones K. Transparent tree ensembles. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, pp. 1241–1244. ACM, New York, NY, USA 2018. <https://doi.org/10.1145/3209978.3210151>.
22. Murphy A, Ali M.S, Dias D.M, Amaral J, Naredo E, Ryan C. Grammar-based fuzzy pattern trees for classification problems. In: Proceedings of the 12th International Joint Conference on Computational Intelligence - Volume 1: ECTA., pp. 71–80. INSTICC, SciTePress 2020. <https://doi.org/10.5220/0010111900710080>
23. Murphy A, Murphy G, Amaral J.L.M, Dias D.M, Naredo E, Ryan C. Towards incorporating human knowledge in fuzzy pattern tree evolution. In: T. Hu, N. Lourenço, E. Medvet (eds.) Genetic Programming - 24th European Conference, EuroGP 2021, Held as Part of EvoStar 2021, Virtual Event, April 7-9, 2021, Proceedings, *Lecture Notes in Computer Science*, vol. 12691, pp. 66–81. Springer 2021. https://doi.org/10.1007/978-3-030-72812-0_5.
24. Murphy A, Ryan C. Improving module identification and use in grammatical evolution. In: Jin Y, editor. 2020 IEEE Congress on Evolutionary Computation, CEC 2020. IEEE Press: IEEE Computational Intelligence Society; 2020.
25. Nordin P, Francone F, Banzhaf W. Explicitly defined introns and destructive crossover in genetic programming. *Adv Genet Progr*. 1995;2:111–34.
26. O'Neill M, Ryan C. Grammatical evolution. *IEEE Trans. Evolutionary Computation*. 2001;5(4):349–58.
27. Patten J.V, Ryan C. Attributed grammatical evolution using shared memory spaces and dynamically typed semantic function specification. In: Genetic Programming - 18th European Conference, EuroGP 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings, 2015; pp. 105–112. https://doi.org/10.1007/978-3-319-16501-1_9.
28. Ribeiro M.T, Singh S, Guestrin C. Why should I trust you?: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135–1144. ACM 2016.
29. Ryan C, Azad R.M.A. Sensible initialisation in grammatical evolution. In: GECCO, pp. 142–145. AAAI 2003.
30. Ryan C, Collins J.J, Neill M.O. Grammatical evolution: Evolving programs for an arbitrary language. In: European Conference on Genetic Programming, pp. 83–96. Springer 1998.
31. Ryan C, Collins J.J, O'Neill M. Grammatical evolution: Evolving programs for an arbitrary language. In: W. Banzhaf, R. Poli, M. Schoenauer, T.C. Fogarty (eds.) EuroGP, *Lecture Notes in Computer Science*, vol. 1391, pp. 83–96. Springer 1998.
32. Ryan C, O'Neill M, Collins J. Handbook of grammatical evolution. Berlin: Springer; 2018.
33. Samek W, Montavon G, Vedaldi A, Hansen L.K, Müller K.R. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer Nature 2019. Google-Books-ID: j5yuDwAAQBAJ
34. dos Santos A.R, do Amaral J.L.M. Synthesis of fuzzy pattern trees by cartesian genetic programming. *Mathw Soft comput*. 2015;22(1):52–6.
35. dos Santos A.R, do Amaral J.L.M, Soares C.A.R, de Barros A.V. Multi-objective fuzzy pattern trees. In: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6. IEEE 2018.
36. Savage N. The race to the top among the world's leaders in artificial intelligence. *Nature*. 2020;588(7837):S102–4.
37. Schneider J, Handali J. Personalized explanation in machine learning: A conceptualization. *arXiv preprint arXiv:1901.00770* 2019
38. Senge R. Machine learning methods for fuzzy pattern tree induction. Ph.D. thesis, Philipps-Universität Marburg 2014.
39. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*. 2018;362(6419):1140–4.
40. Takagi H. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proc IEEE*. 2001;89(9):1275–96.
41. Yi Y, Fober T, Hüllermeier E. Fuzzy operator trees for modeling rating functions. *Int J Comput Intell Appl*. 2009;8:413–28.
42. Yuille AL, Liu C. Deep nets: What have they ever done for vision? *Int J Comput Vision*. 2021;129(3):781–802.
43. Zhang BT, Mühlenbein H. Balancing accuracy and parsimony in genetic programming. *Evol Comput*. 1995;3(1):17–38.
44. Zou J, Schiebinger L. AI can be sexist and racist—it's time to make it fair 2018.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.