Towards incorporating Human Knowledge in Fuzzy Pattern Tree Evolution*

Aidan Murphy[1 [0000−0002−6209−4642]], Gráinne Murphy[[0000−0001−8166−8499]], Jorge Amaral[2 [0000−0001−6580−5668]], Douglas Mota Dias[1,2 [0000−0002−1783−6352]], Enrique Naredo[1 [0000−0001−9818−911X]], and Conor Ryan[1 [0000−0002−7002−5815]]

[1] Lero & University of Limerick, Limerick, Ireland
http://bds.ul.ie/
[2] Rio de Janeiro State University, Rio de Janeiro, Brazil
aidan.murphy@ul.ie

Abstract. This paper shows empirically that Fuzzy Pattern Trees (FPT) evolved using Grammatical Evolution (GE), a system we call FGE, meet the criteria to be considered a robust Explainable Artificial Intelligence (XAI) system. Experimental results show FGE achieves competitive results against state of the art black box methods on a set of real world benchmark problems. Various selection methods were investigated to see which was best for finding smaller, more interpretable models and a human expert was recruited to test the interpretability of the models found and to give a **confidence** score for each model. Models which were deemed interpretable but not trustworthy by the expert were seen to be outperformed in classification accuracy by interpretable models which were judge trustworthy, validating that FGE can be a powerful XAI technique.

Keywords: Grammatical Evolution · Fuzzy Logic · Explainable AI.

1 Introduction

The number of machine learning (ML) applications has expanded massively since the turn of the millennium. ML algorithms these days have access to massive amounts of data and can run on massively parallel high-performance hardware. ML frameworks and systems now achieve near-perfect performance, which can outperform human agents. This has led to headline-grabbing AI success stories in chess and Go [34]. Global spending on artificial intelligence is estimated to hit $50 billion a year.

These results are not without their critics, though [20]. There often exists a trade-off between high accuracy and transparency. These models are referred

* The authors are supported by Research Grants 13/RC/2094 and 16/IA/4605 from the Science Foundation Ireland and by Lero, the Irish Software Engineering Research Centre (www.lero.ie). The third and fourth authors are partially financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

to as 'black box' (BB) models. They do not allow their internal workings to be understood. They simply return input and output pairs. No knowledge of how a decision is made can be obtained from the system. A user, expert or otherwise, has no means to understand how a model arrived at a conclusion. This inability to interpret and check that the model has 'common sense' makes trusting the model difficult as well as making debugging and error checking an impossibility. These shortcomings have, too, been headline-grabbing and shown AI systems can exhibit racist and sexist behaviour [37].

To tackle these issues, a new area of research was spawned, XAI [1,2]. XAI aims to create interpretable models and methods that can somehow explain themselves without, or with minimal, impacting performance [4].

A recent addition to XAI has been Fuzzy Pattern Trees (FPTs) [14,36]. Based on fuzzy set theory, a FPT is a hierarchical tree structure. This is in contrast to most other fuzzy-based systems which use rules as representations. As a fuzzy model, it is easily interpretable due to its usage of linguistic labels. This interpretability, obviously, depends on the tree size not being excessively large. Grammatical Evolution has shown it can be a very effective approach for evolving accurate, and, crucially, small FPTs [22].

This paper sets out to validate the claim that these FPTs are intrinsically interpretable in their own right. It further wishes to show this interpretability can aid in the evolutionary process by finding faults in the best performing individuals found by the search. The system was tested on various real-live fairness benchmarks. The results show that FPTs may allow the identification of data or algorithmic bias that may be present in final models.

The remainder of this paper is organized as follows: Section 2 reviews the main background concepts GE, Fuzzy GE and XAI. Section 3 explains the proposals and describes the paper's contributions in more detail. Next, Section 4 presents the experimental set-up. It outlines all performance measures which were investigated. Section 5 presents the main results of the experiments described in 4. Finally, Section 6 summarises the research and discusses future work suitable for investigation.

2 Background

2.1 XAI

Papers and conference talks in the area of XAI and computer model interpretability has grown and is continuing to grow rapidly [1,13]. However, these conferences talks and papers have existed in a space where the term 'interpretability' has not been agreed upon or even well-defined [1,6,7,10]. There is not yet a consensus when exactly as to when a model has been 'explained' fully or indeed what an 'explanation' even is [6]. The terms *interpretable*, *understandable* and *intelligible* are often used interchangeably or without distinction [19]. In different contexts, interpretability may have differences; for example, a loan approving system may simply need to show that it is not discriminatory against any group, whereas a safety-critical system may need to describe every step in its internal logic [7].

Many of the explanations put forward in papers require some machine learning expertise [18]. They may look incomplete to somebody with no machine learning experience, while others may require some domain knowledge to 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.

Some work has been done to develop a rigid framework to define and evaluate interpretability [7]. They stress that human evaluation or abstraction is essential to any idea of judging interpretability.

Others argue that it is not enough for 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].

Trust in a model or system is an often overlooked and important feature [27]. It may not be sufficient for a model to show high accuracy for a user to accept its outputs. A user may require sufficient evidence that the decisions are fair or ethical or legal. If the model is a BB or its logic is presented in a way that makes it difficult for a human to abstract knowledge from its results, they may refuse to use it, or may not be allowed to under the General Data Protection Regulation, GDPR, [12].

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

For a model to be useful, usable and fully 'explained' this paper proposes it should have the following properties: it must be transparent in its workings; have similar or better accuracy to any other model; be cogent in its statements (particularly for finding complex relationships); be able to incorporate domain knowledge; and deterministic.

2.2 Grammatical Evolution

Grammatical Evolution (GE) [30], often thought of as a variant of genetic programming (GP) [17], is a popular evolutionary computation technique. As with many evolutionary algorithms, GE's inspiration comes from nature and genetics. GE creates computer programs by mapping a binary string using a grammar. A key point to note is GE can produce programs in any arbitrary language, usually specified using a Backus-Naur Form (BNF) grammar [29,25] or Attribute Grammar (AG) [26]. The evolutionary operators of crossover and mutation do not occur on the actual computer program, but on the string which, combined with the grammar, creates the program. Therefore any representation the user wishes to use for the solution to aid interpretability is possible. The most popular way to represent the solutions is to use tree structures [31], which have been shown to be the most easily interpretable representations [15] 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 finds solutions which best minimise (or maximise depending on the goal) an objective function. GE, however, allows much more knowledge and nuance to be built into its objective function than typical test set accuracy. Multi-objective [5] and many objective [16] optimisation are common in EAs. It would be easy to find explicitly 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]. The user has their pick of solutions. This allows the programs to have the highest possible fidelity while at the same time being interpretable to whomever is using them. GE can be trained to be highly accurate on particular cases if they are of important to the user. The objective function allows the user to personalize their goals more than traditional ML models by including ethics, fairness, legality, profit etc. as a component of the search.

Ideas like fairness are vague and hard to define. This allows the user to define these concepts in a way which is important to them. GE also allows for the human factors to be built into the solution [8]. If combined with an interactive GA [35], in which the person is directly responsible for giving a fitness score, solutions could take the form which maximises the utility of the person using them. That is to say, the search would look for a solution in the form which would be most suitable to the user interacting with it. This creates a personalized explanation, seen as a key facet in Machine Learning going forward [33].

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 can have the user involved at every stage of the machine learning process. Before the search begins the user may set up the grammar and specify the objective. During run time they may impart domain knowledge in the form of subtrees or modules [23]. The user would be the main architect of the form of the solutions. All this would help build 'trust' in the model to the user, if it is needed.

2.3 Fuzzy GE

FPTs which use GE as their search technique were recently introduced [22]. This approach, FGE, showed competitive performance against black box methods and was shown to outperform another GP variant, Cartesian GP, on a set of benchmark classification problems [32].

In order to perform classification using FGE a set of FPTs are needed, one for each class that exists in the problem. These FPTs serve as the logical description of the class. This sets FGE apart from traditional classification approaches in GE which only require one expression to be evolved, regardless of the number of classes in the problem. To classify an individual a boundary or boundaries are decided upon. The output of the tree 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].

FGE evolves one, large solution and treats the subtrees of this solution as it's FPTs, as seen in Figure 1. The FPT which yields the largest output for an

individual is declared the winner and that individual is designated as belonging to that class. This is illustrated in Figure 2. The root node of the tree is responsible for this process. Representing each FPT as subtrees of one large solution combined with GE's inbuilt separation between search space and program space leads to another major advantage FGE experiences. No special or protected operators are needed for crossover or mutation. A simple grammar augmentation is all that is needed to tackle different problem specifications.

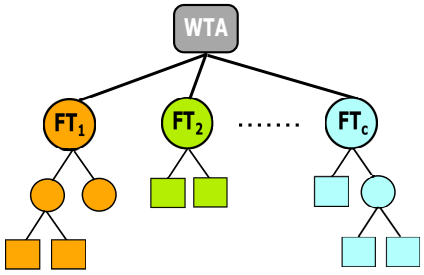

Fig. 1. Pictorial representation of a multi-classifier evolved by FGE, where FT_c is the fuzzy tree for each available class, and at the root the winner take all (WTA).

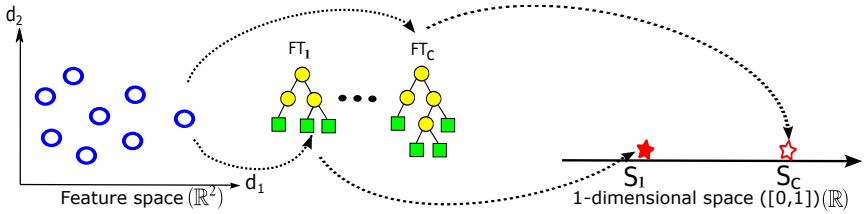

Fig. 2. Graphical depiction of the mapping process from the feature space to a 1-dimensional space [0,1] using a set of fuzzy trees FT_1 to FT_c.

3 Explainable GE

3.1 Reducing size of Trees

A FPT serves as a class descriptor in each benchmark problem. It is therefore paramount for FPTs to be interpretable. This interpretability is only achieved if their size is kept as small as possible. It was seen previously that standard GE may be bloating individuals, leading to trees which are excessively large and contain worthless material. The addition of a simple parsimony pressure was seen to greatly reduce the size of individuals without having a noticeable

effect on their fitness. It was not established if this simple procedure was the most effective at producing smaller, accurate FPTs. In this paper we investigate various methods for producing smaller trees were investigated and compared against standard GE and against each other, and consider both their size and accuracy. Each method is outlined below.

Intron Removal In the context of GP, an intron is a section of an individual which does not have an effect on that individual's output. That is to say, it is a redundant piece of the individual. Despite their lack of involvement they can play an important part in evolution [24]. However, when trying to interpret an individual it is necessary to remove them as they may lead to confusion.

Strict/Easy Regularization Two types of regularisation were used. The first was an easy regularization where a small penalty to fitness was applied based on the maximum depth of the solutions found. The second was a strict regularization. This procedure set the fitness to 0 if the max depth exceeded a certain threshold. Fitness was set as usual if it was below this threshold.

Double Tournament The final procedure for bloat control implemented was double tournament. This strategy conducts 2 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.

3.2 FPT Representation

A FPT differs from other fuzzy based classifiers by adopting a hierarchical, tree structure. The leaf nodes of these trees are the fuzzified problem variables and the inner nodes are fuzzy logical and arithmetic operators. The information is propagated from the bottom to the top, similar to a regular GP classifier. The output of the tree is in the [0,1] interval. More formally, a FPT maps $f_i(\mathbf{x}) : \Re^n \rightarrow [0,1]$, where x are the input variables.

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

$$WTA = IF\{\}()..ELSE() \tag{1}$$
$$MAX = max(a, b) \tag{2}$$
$$MIN = min(a, b) \tag{3}$$
$$WA(k) = ka + (1 - k)b \tag{4}$$
$$OWA(k) = k \cdot max(a,b) + (1 - k)min(a,b) \tag{5}$$
$$CONCENTRATE = a^2 \tag{6}$$
$$DILATE = a^{\frac{1}{2}} \tag{7}$$
$$COMPLEMENT = 1 - a \tag{8}$$

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 3 shows an example of an FPT 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.

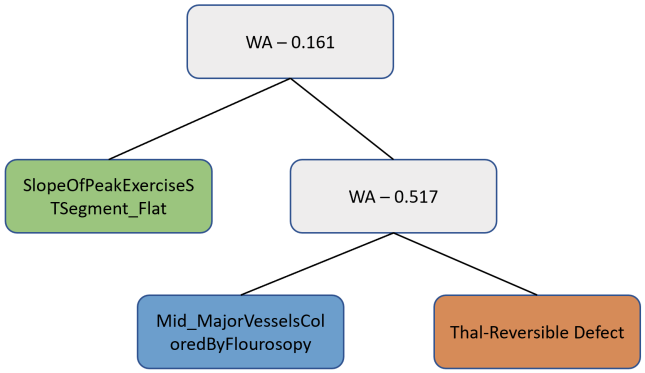

Fig. 3. Tree representing the interpretable class "Presence of Heart Disease", showing each variable with different color.

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.

3.3 Human in the loop

The goal of many ML processes is to replace a human agent by a model which makes as good as, or better, decisions than a human would. The main obstacle is a machines inability to reason or abstract. The goal of so-called strong AI is to develop systems that possess this 'common sense' [11]. However, until such time as that is available it is reasonable to add this common sense into a model through human interaction. Therefore, a solution to the short comings of modern ML models and interpreting them lies in creating models which are transparent,

unambiguous and which place humans prominently in their design, a human in the loop ML algorithm.

In order to create a fully Explainable GE system, meeting all the criteria highlighted above, it is essential a human is incorporated in the evolutionary cycle as much as possible. However, it is first necessary to empirically validate that FPTs are, in fact, interpretable. The depth at which FPTs cease to become interpretable also requires investigation. This required an extra operation to be performed. This involved giving each model to a domain expert who ranked each model's interpretability.

4 Experimental Setup

The hypothesis that FGE meets all, or most, of the criteria of an interpretable model was tested. That is to say, FGE provides a transparent model which can be understood and attains accuracy comparable to other, black box ML approaches. Five selection methods were tested on six benchmark problems. These benchmark datasets were chosen as they have been identified as problems which often produce models containing bias or discrimination. Insight into the logic of any model trained on this data would, therefore, be very useful.

The grammar used can be seen in Figure 4. The FPTs for each class are contained within the WTA node, the $< exp >$ non-terminals. To extend the grammar for multi-class classification, the simple addition of more $< exp >$ symbols in the expression are needed. For example, three classes would need the addition of one more $< exp >$ symbol and so on. Constants were created using digit concatenation [3].

$$
\begin{aligned}
&< start > ::= WTA(< exp >, < exp >)\\
&< exp > ::= max(< exp > , < exp >) \mid\\
&\qquad\quad min(< exp > , < exp >) \mid\\
&\qquad\quad WA(< const > , < exp > , < exp >) \mid\\
&\qquad\quad OWA(< const > , < exp > , < exp >) \mid\\
&\qquad\quad concentrate(< exp >) \mid\\
&\qquad\quad dilation(< exp >) \mid\\
&\qquad\quad complement(< exp >) \mid\\
&\qquad\quad x_1 \mid x_2 \mid x_3 \mid ...\\
&< const > ::= 0. < digit >< digit >< digit >\\
&< digit > ::= 0 \mid 1 \mid 2 \mid ....
\end{aligned}
$$

Fig. 4. Grammar used to evolve a Fuzzy Pattern Tree. Extra $< exp >$, as needed, can be added in the WTA node to make it a multi-class grammar.

4.1 GE Parameters

The experiments were run with a population size of 500 and for 50 generations. For each run, Sensible Initialisation was used to create the initial population and effective crossover was also employed [28]. At the beginning of each run, the data was split randomly 75% for training and the remaining 25% for test. This was repeated so each run had a different, randomized training and test data. The exception was Census Income, which came with the data already partitioned 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 performing two, nested tournaments. For experiments FGE_{DT1}, the first tournament winner was decided by fitness with the second tournament winner being the individual with the smaller size. FGE_{DT2} was the inverse of this, the first tournament considered size while the second was determined by an individual's fitness.

Table 1. List of parameters used for FGE

Parameter	**Value**
Runs	30
Total Generations	50
Population	500
Elitism	Best Individual
Selection	Tournament, Double Tournament
Crossover	0.9 (Effective)
Mutation	0.01
Initialisation	Sensible

4.2 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.

$$F = 1 - RMSE \tag{9}$$

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

$$F_{L1} = 1 - RMSE - MaxDepth \times 0.001 \tag{10}$$

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

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

The max depth of a solution is the largest path which exists in any FPT of an individual.

4.3 Fairness Benchmarks

The experiments are run on six binary classification benchmark datasets, all of which can be found online in the UCI repository. A summary of all 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.

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	1,000
ProRepublica Recidivism	Recid	2	52	7,214
ProRepublica Violent Recidivism	V/Recid	2	52	4,743
Heart Disease	Heart	2	13	303

4.4 Expert Validation

An expert was sought out to empirically validate the interpretability of the FPTs. A domain expert, a doctor working in a local hospital, was sought out to examine the results from the Heart experiments. The best individual from each run was saved and parsed into a tree. This gave 150 graphs of the trees, 30 for each of the 5 selection methods. These trees were then presented to the domain expert over Zoom. The evaluation consisted of two steps. The expert was first asked to evaluate the trees in terms of interpretability. Afterwards, the expert was asked to score the logic of the model. That is to say, do the variables and operators in the make sense medically. Both of these were scored from 1, lowest, to 5, highest.

5 Results

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

The first five columns show the results for FGE, FGE with fitness function in equation 10 applied, FGE with fitness function in equation 11 applied, FGE with double tournament selection, first considering size then fitness, and finally FGE with double tournament selection, first fitness then size. The sixth column shows

Table 3. Classification performance comparison of each selection method used with FGE, showing the classification accuracy 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	RF	LR
Bank	0.89	0.89	0.89	0.89	0.89	0.91	0.91	0.84
Census	0.79	0.78	0.81	0.80	0.78	0.85	0.85	0.79
Credit	0.71	0.70	0.71	0.70	0.70	0.73	0.76	0.71
Recid	0.71	0.72	0.69	0.72	0.70	0.74	0.74	0.56
V/Recid	0.83	0.83	0.83	0.83	0.83	0.84	0.84	0.54
Heart	0.79	0.77	0.77	0.77	0.75	0.82	0.82	0.81

the results for Support Vector Machine (SVM) and seventh Random Forest (RF). Finally, column eight shows the result of a Logistic Regression (LR).

No one selection method for FGE was seen to outperform any other selection method with respect to accuracy. Among all FGE experiments, FGE_{DT1} found the best solutions on 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 RF classifier was statistically significantly better than all others, achieving or matching the best performance on each problem. As a BB model, though, it does not allow any further knowledge to be extracted. FGE was able to achieve very competitive results against the BB approaches. FGE was outperformed by 5% on the Credit dataset, achieving 71% in both FGE and FGE_{L2} compared to the best performing technique RF which found 76%. FGE accomplished 81% accuracy on the Census problem, 4% worse than both SVM and RF which obtained 85% accuracy. For the Bank, Recid and V/Recid problems, FGE evolved solutions which were within 2% of those found by either SVM or RF.

On all but one problem FGE was seen to outperform the interpretable ML algorithm considered, LR. FGE significantly exceed the performance of LR on the Bank, Recid and V/Recid problems. FGE 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%.

At the end of each run, the best of run individual underwent an intron removal process, outlined above, to remove any bloat which may exist in the program. The mean size of the FPTs in the final individual found 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.

Table 4. Size comparison between each approach. 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

The results of the human expert's analysis can be seen in Table 5. FGE_{L2} was the best performing method for finding interpretable solutions, with all 30 runs finding trees attaining scores of 4 or 5. The next best performing method was FGE_{L1}, with 8/30 being scored 4 or 5, followed by FGE_{DT1}, having 6 interpretable solutions. The worst performing methods were FGE_{DT2} and FGE, both only finding 4 interpretable solutions in their 30 runs. A plot showing the decrease in interpretability as depth increases is seen in Figure 5. The plot suggests that any reasonable indication of interpretability disappears after trees have exceeded depth 5 or 6.

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.

	Interpretability Score				
Selection Type	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

To validate that the FPTs were indeed transparent and clear in their statements, the logic of the models deemed interpretable (those scoring 4 or 5) was examined by the domain expert. This gave 52 of the original 150 models. Any models flagged as having 'incorrect' logic, that is to say the confidence score was 1 or 2, were separated from the population. Similarly, models with marginal trust, those with a confidence score of 3, were separated.

This left 24 models, described in Table 6, which were deemed interpretable and inspired confidence. The mean accuracy of those models is 77.1%, shown in Table 7. When marginal models were included, 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

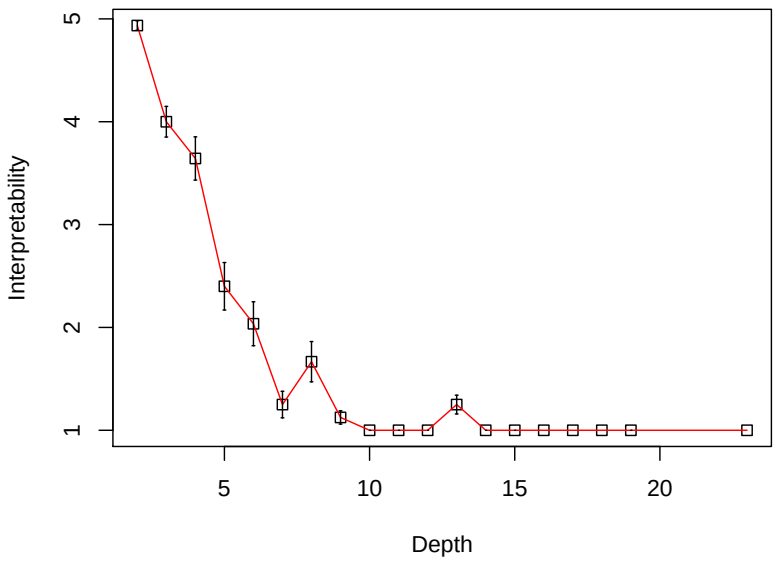

Fig. 5. Decrease in Human Interpretability as the Maximum Depth of the Model Increases

deemed to have 'correct' logic perform ∼2% better than those adjudged to have 'incorrect' logic. This is despite both groups containing almost identical fitness on the training data. By investigating 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. This process is also an effective way for an expert to build trust in the models which are being evolved.

Table 6. Selection Methods of FGE 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 FGE Models with Interpretability Score ≥ 4.

Confidence Score	Number	Accuracy
≤ 2 (Incorrect Logic)	13	75.2%
≥ 4 (Correct Logic)	24	77.1%
≥ 3 (Correct Logic & Marginal Logic)	39	77.3%

6 Conclusion

This paper empirically evaluates the suitability of FGE as an XAI approach by analysing the Fuzzy Pattern Trees it produced.

The experimental results show that FGE has a competitive performance on real world classification tasks. These models were then presented in comprehensible terms to a human domain expert, a medical doctor. This expert was able to validate the interpretablity of the models and to extract the knowledge obtained in the learning process of the model. This was validated by comparing the performance of models which the domain expert labeled their logic as 'incorrect' vs those the domain expert labeled as 'correct'. Models with 'incorrect' logic were seen to perform worse than those deemed as 'correct'.

The next major step in this work is the inclusion of the human expert in more stages of the evolutionary process. Pre-processing by picking membership function values, encapsulating information into modules and incorporating domain knowledge in the grammar, setting maximum depth size of the individuals, being involved in the selection process are some of the many possibilities going forward. This would enable GE to tailor its search to the expertise and capabilities of the user.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Arrieta, A.B., Díaz-Rodríguez, N., Ser, J.D., 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. Information Fusion **58**, 82 – 115 (2020)
3. Azad, R.M.A., Ryan, C.: The best things don't always come in small packages: Constant creation in grammatical evolution. In: Nicolau, M., Krawiec, K., Heywood, M.I., Castelli, M., García-Sánchez, P., Merelo, J.J., Rivas Santos, V.M., Sim, K. (eds.) Genetic Programming. pp. 186–197. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
4. Carvalho, D.V., Pereira, E.M., Cardoso, J.S.: Machine Learning Interpretability: A Survey on Methods and Metrics. Electronics **8**(8), 832 (2019)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)

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 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. pp. 743–750 (2012)
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. Goertzel, T.: The path to more general artificial intelligence. Journal of Experimental & Theoretical Artificial Intelligence **26**(3), 343–354 (2014)
12. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI Magazine **38**(3), 50–57 (2017)
13. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys (CSUR) **51**(5), 93 (2018)
14. Huang, Z., Gedeon, T.D., Nikravesh, M.: Pattern trees induction: A new machine learning method. Trans. Fuz Sys. **16**(4), 958–970 (Aug 2008). https://doi.org/10.1109/TFUZZ.2008.924348
15. 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. Decision Support Systems **51**(1), 141–154 (2011)
16. 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)
17. Koza, J.R., Koza, J.R.: Genetic programming: on the programming of computers by means of natural selection, vol. 1. MIT press (1992)
18. Krakovna, V., Doshi-Velez, F.: Increasing the interpretability of recurrent neural networks using hidden markov models. arXiv preprint arXiv:1606.05320 (2016)
19. Lipton, Z.C.: The mythos of model interpretability. arXiv preprint arXiv:1606.03490 (2016)
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. pp. 1241–1244. SIGIR '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3209978.3210151, http://doi.acm.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., Ryan, C.: Improving module identification and use in grammatical evolution. In: Jin, Y. (ed.) 2020 IEEE Congress on Evolutionary Computation, CEC 2020. IEEE Computational Intelligence Society, IEEE Press (2020)

24. Nordin, P., Francone, F., Banzhaf, W.: Explicitly defined introns and destructive crossover in genetic programming. Advances in genetic programming **2**, 111–134 (1995)
25. O'Neill, M., Ryan, C.: Grammatical evolution. IEEE Trans. Evolutionary Computation **5**(4), 349–358 (2001)
26. 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. pp. 105–112 (2015). https://doi.org/10.1007/978-3-319-16501-1_9, `https://doi.org/10.1007/978-3-319-16501-1_9`
27. 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)
28. Ryan, C., Azad, R.M.A.: Sensible initialisation in grammatical evolution. In: GECCO. pp. 142–145. AAAI (2003)
29. Ryan, C., Collins, J.J., O'Neill, M.: Grammatical evolution: Evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP. Lecture Notes in Computer Science, vol. 1391, pp. 83–96. Springer (1998)
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., O'Neill, M., Collins, J.: Handbook of Grammatical Evolution. Springer (2018)
32. dos Santos, A.R., do Amaral, J.L.M.: Synthesis of Fuzzy Pattern Trees by Cartesian Genetic Programming. Mathware & soft computing **22**(1), 52–56 (2015)
33. Schneider, J., Handali, J.: Personalized explanation in machine learning: A conceptualization. arXiv preprint arXiv:1901.00770 (2019)
34. 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 **362**(6419), 1140–1144 (2018)
35. Takagi, H.: Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. Proceedings of the IEEE **89**(9), 1275–1296 (2001)
36. Yi, Y., Fober, T., Hüllermeier, E.: Fuzzy operator trees for modeling rating functions. International Journal of Computational Intelligence and Applications **8**, 413–428 (2009)
37. Zou, J., Schiebinger, L.: AI can be sexist and racist—it's time to make it fair (2018)