# Evolutionary algorithms and decision trees for predicting poor outcome after endovascular treatment for acute ischemic stroke

Kappelhof, N.; Ramos, L.A.; Kappelhof, M.; Os, H.J.A. van; Chalos, V.; Kranendonk, K.R. van; ... ; Marquering, H.A.

**Note:** To cite this publication please use the final published version (if applicable).

# Evolutionary algorithms and decision trees for predicting poor outcome after endovascular treatment for acute ischemic stroke

N. Kappelhof [a], L.A. Ramos [a,b,*], M. Kappelhof [c], H.J.A. van Os [d], V. Chalos [e,f,g], K. R. van Kranendonk [c], N.D. Kruyt [d], Y.B.W.E.M. Roos [h], W.H. van Zwam [i], I.C. van der Schaaf [j], M. A.A. van Walderveen [k], M.J.H. Wermer [d], R.J. van Oostenbrugge [l], Hester Lingsma [f], Diederik Dippel [e], C.B.L.M. Majoie [c], H.A. Marquering [a,c]

[a] *Department of Biomedical Engineering and Physics, Amsterdam UMC, University of Amsterdam, Amsterdam, the Netherlands*
[b] *Department of Clinical Epidemiology and Biostatistics, Amsterdam UMC, University of Amsterdam, Amsterdam, the Netherlands*
[c] *Department of Radiology and Nuclear Medicine, Amsterdam UMC, University of Amsterdam, Amsterdam, the Netherlands*
[d] *Department of Neurology, Leiden University Medical Center, Leiden, the Netherlands*
[e] *Department of Neurology, Erasmus MC - University Medical Center, Rotterdam, the Netherlands*
[f] *Department of Public Health, Erasmus MC - University Medical Center, Rotterdam, the Netherlands*
[g] *Department of Radiology and Nuclear Medicine, Erasmus MC - University Medical Center, Rotterdam, the Netherlands*
[h] *Department of Neurology, Amsterdam UMC, University of Amsterdam, Amsterdam, the Netherlands*
[i] *Department of Radiology, Cardiovascular Research Institute Maastricht, Maastricht University Medical Center, Maastricht, the Netherlands*
[j] *Department of Radiology, University Medical Centre, Utrecht, the Netherlands*
[k] *Department of Radiology, Leiden University Medical Center, Leiden, the Netherlands*
[l] *Department of Neurology, Maastricht University Medical Center, Maastricht, the Netherlands*

## ARTICLE INFO

## ABSTRACT

Despite the large overall beneficial effects of endovascular treatment in patients with acute ischemic stroke, severe disability or death still occurs in almost one-third of patients. These patients, who might not benefit from treatment, have been previously identified with traditional logistic regression models, which may oversimplify relations between characteristics and outcome, or machine learning techniques, which may be difficult to interpret. We developed and evaluated a novel evolutionary algorithm for fuzzy decision trees to accurately identify patients with poor outcome after endovascular treatment, which was defined as having a modified Rankin Scale score (*mRS*) higher or equal to 5. The created decision trees have the benefit of being comprehensible, easily interpretable models, making its predictions easy to explain to patients and practitioners. Insights in the reason for the predicted outcome can encourage acceptance and adaptation in practice and help manage expectations after treatment. We compared our proposed method to CART, the benchmark decision tree algorithm, on classification accuracy and interpretability. The fuzzy decision tree significantly outperformed CART: using 5-fold cross-validation with on average 1090 patients in the training set and 273 patients in the test set, the fuzzy decision tree misclassified on average 77 (standard deviation of 7) patients compared to 83 ($\pm$7) using CART. The mean number of nodes (decision and leaf nodes) in the fuzzy decision tree was 11 ($\pm$2) compared to 26 ($\pm$1) for CART decision trees. With an average accuracy of 72% and much fewer nodes than CART, the developed evolutionary algorithm for fuzzy decision trees might be used to gain insights into the predictive value of patient characteristics and can contribute to the development of more accurate medical outcome prediction methods with improved clarity for practitioners and patients.

## 1. Introduction

Stroke remains among the leading causes of death and disability worldwide, with over 80% of all stroke cases being ischemic [1]. In patients with acute ischemic stroke, endovascular treatment (EVT) was proven effective and safe in 2015. For every five patients treated with

EVT, one additional patient can live independently after their stroke [2]. However, despite high rates of procedural success, almost one-third (50–60%) of EVT patients do not fully recover after their stroke, but end up severely disabled and needing support in their daily activities or do not survive [3].

Since EVT is a costly intervention which has its inherent complications, identifying patients that will not benefit from EVT before initiation of treatment is desirable, as to enable a more efficient, targeted use of resources [4]. Machine learning models are increasingly used to predict patient outcomes after EVT [5–7]. Although these models show promising results, factors like non-transparency can limit their practical application.

A decision tree can provide this desired clarity by intuitively expressing which set of attributes cause a difference in predicted outcome. Algorithms to generate decision trees can be used as a stand-alone method or incorporated in other algorithms to assist in outcome prediction as has, for instance, been done to improve the diagnostic process of Mitral Valve Prolapse Syndrome [8], to generate diagnostic rules for cardiac diagnoses [9] and in many other applications [10,11]. However, most decision tree algorithms make greedy (locally optimal) choices, are deterministic, and have no inherent mechanism to escape local optima. Moreover, numerous studies have proven that a mathematically efficient (polynomial) algorithm to create a decision tree of minimum size which correctly classifies all subjects does not yet exist: the problem is *NP*-hard [12–14]. Hence, traditional decision tree algorithms might not find the optimal solution.

Evolutionary methods use an explorative and stochastic approach to perform a wider, non-greedy search for different and better classifying decision trees. The probability of continuing with a new solution is dependent on how well the solution performs on the problem at hand, which can be quantified by an objective function or fitness function. This stochastic nature can prevent the algorithm from getting stuck in a local optimum, improving chances of finding the *global* optimal solution where deterministic algorithms cannot. Additionally, evolutionary methods are flexible with respect to the objective function and can work under many different parameter settings. Hence, with these methods, we will be able to tailor the method to our specific problem.

Multiple evolutionary algorithms to generate decision trees have previously been developed. Llorà and Garell introduce an evolutionary algorithm, named GALE, designed to build a generic type of orthogonal (split on one variable), oblique (linear combination of variables), or multivariate decision tree (function of multiple variables, not necessarily linear) [15]. A more specialized evolutionary algorithm for decision trees is introduced with the LEGAL tree [16] which uses a multi-objective fitness function with lexicographical ordering to generate orthogonal decision trees. The related E-MOTION algorithm [17] uses the same kind of fitness function to generate oblique decision trees. More examples include the GATree which creates orthogonal trees with a multiplicative fitness function [18], the MOGP algorithm which creates pareto optimal orthogonal trees [19], or can be found in Refs. [20–22]. A survey containing a methodological comparison between evolutionary algorithms for decision trees can be found in Refs. [11,23]. To our knowledge, no study compared these evolutionary algorithms with respect to accuracy, size, or any other performance measure.

We propose a novel evolutionary algorithm that searches for *fuzzy* binary decision trees, which originates from the Soft Decision Trees (SDTs) by Olaru and Wehenkel [24]. These SDTs handle numerical data differently and might give a more accurate reflection of the underlying continuous nature of numerical data. We hypothesize that the designed evolutionary algorithm can find comprehensible fuzzy decision trees that are smaller and more accurate than the decision trees generated by the traditional deterministic algorithms. To test the hypothesis, the novel algorithm was used as a proof of concept to construct fuzzy decision trees to identify acute ischemic stroke patients who will die or remain severely disabled after their stroke ($mRS \geq 5$). We investigated our hypothesis by comparing the proposed approach to the commonly used Classification and Regression Tree (CART) algorithm [25] by analyzing the accuracy and interpretability, quantified by size, of both models.

## 2. Methods

### 2.1. Clinical data and pre-processing

The MR CLEAN Registry is a nationwide, prospective, observational, multicenter study including EVT-treated acute ischemic stroke patients from 16 intervention hospitals in the Netherlands [3]. Inclusion criteria and patient demographics have been described previously [3]. For the current study, we included patients treated between March 16, 2014 and June 15, 2016 (N = 1,488, with 1363 recorded outcomes and 125 missing outcomes).

The dependent variable used to measure functional outcome is the modified Rankin Scale score (*mRS*) determined 90 days post-stroke and indicating the degree of dependence of the patient on their caretakers [26]. The scale ranges from 0 (no symptoms) to 6 (death). We defined poor outcome as *mRS* 5 (severe dependence on nursing care, bedridden) or 6 (death) and moderate to good outcome as *mRS* 0–4 (ranging from no symptoms to walking with assistance).

Of the 1,488 patients, also referred to as *subjects*, 893 (60%) had moderate to good functional outcome, 470 (32%) had poor functional outcome, and 125 (8%) outcomes were missing. The outcome values are slightly skewed, which is a well known characteristic of *mRS* outcome in patients with acute ischemic stroke. Although a skewed distribution of the dependent variable can negatively affect the performance of an estimator, it is expected that this skewness will not cause poor results since the number of samples is quite large. Fig. 1 depicts the distribution of the response variable in the training set, which is the subset of subjects used to build and train the fuzzy decision tree. The method of constructing the training and test set is presented in section 2.5.

To predict *mRS*, we included a total of 86 variables in our models out of which 42 were numerical, 33 were binary, and 11 categorical (excluding *mRS*). Of these categorical variables, 4 were ordinal. The remaining 7 categorical variables had no ordering and were one-hot-encoded. We did not stratify the patients based on clinical or other factors. The population on which the model is trained should be similar to the population for which the model would be used. Any imbalances that are present in our population will also be found in populations in clinical practice. For some variables, there were many missing values (>50%). Removing all patients with missing values can introduce bias, while removing variables with many missing values might remove
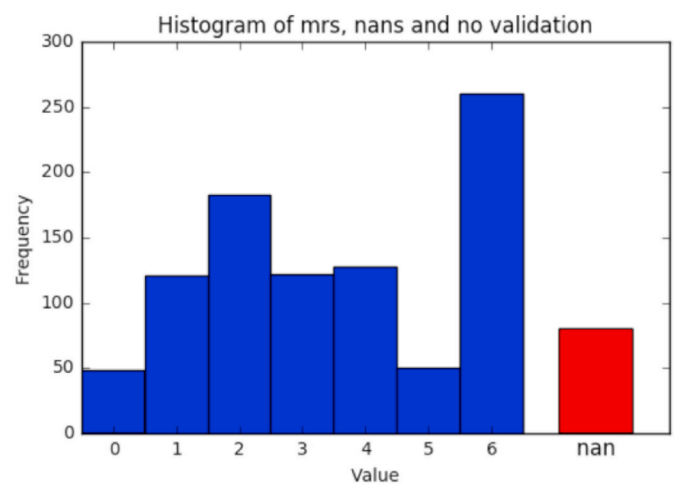


**Fig. 1.** Histogram of mRS values, training set. The blue bars represent the frequency (y) or the dependent variable mRS (x), ranging from 0 to 6. The red bar indicated the number of missing outcomes.

valuable predictors. The treatment of missing values is described in section 2.3.7. An elaborate description of the available variables is supplied in Appendix A. Standard imaging biomarkers available before treatment were included in the data, but full imaging data was omitted since this data is of a different dimensionality and would require substantial modifications to the model to include. Additionally, the high dimensionality of radiological data could increase the risk of overfitting.

## 2.2. CART

In this study, we focus on binary decision trees. A short explanation of binary decision trees and decision trees in general is given in Appendix D. The CART algorithm [25] is an established algorithm that is designed to create binary decision trees. It was ranked in the top 10 algorithms for data mining [27] and hence we regarded CART and its performance as a satisfactory reflection of the expected performance of the decision tree algorithms that are currently often used in practice. An implementation of CART can be found in the Scikit-learn Python package [28], which we used as benchmark. The splitting conditions in CART are created with a greedy approach, where in each decision node the variable and value are chosen according to a minimization criterion over a function expressing the (dis)similarity in response among the two created groups. CART uses Gini impurity or entropy to calculate this (dis)similarity. These measures are comparable but have different preferences with respect to the distribution of the impurity across the downward branches, and are more elaborately described in Appendix E. The prediction given in the leaves is set to the most frequently occurring category of all test subjects ending in the leaf.

In the used implementation of the CART algorithm, all missing values, including explanatory variables, need to be removed or imputed. To impute missing values, three methods (*mean, median* and *most occurring*) are considered for the explanatory variables (non-*mRS*), and five methods (*no imputation, mean, median, most occurring, extended dataset*) are considered for *mRS*. The extended dataset method uses the original dataset expanded with variables observed after treatment, with imputations generated by building a CART decision tree for *mRS* on this extended dataset and replacing missing *mRS* values with the score predicted by the tree. Additionally, since CART has no inherent stopping criterion, it needs an external measure to prevent or prune extensive growth. In this research, we limit tree growth by defining a minimal required number of subjects per leaf, denoted by $n_{min}$.

The most suitable imputing methods and value of $n_{min}$ for the splitting criteria Gini impurity and entropy were identified by running preliminary tests. The setup of the preliminary tests is explained in Appendix B. The values of $n_{min} = 0.07 \cdot N$ (Gini) and $n_{min} = 0.05 \cdot N$ (entropy) resulted in the lowest average amount of misclassifications, with $N$ the number of patients in the training set. Imputation of the dependent variables by *mean* and *median* for the Gini and entropy splitting criterion respectively showed most promising results and were implemented. For the dependent variable, the test indicated that CART with Gini should have no imputation, while the trees split by entropy benefit from imputation by *extended dataset*.

## 2.3. Fuzzy decision trees

Nodes in the decision trees created by CART consist of a variable name (the *split-attribute*) and a value (the *split-value*). In *fuzzy* decision trees, as proposed by Olaru and Wehenkel [24]; decision nodes contain, instead of the split-value, a *split-range*. If the value of the split-attribute is below the left bound of the range, the left branch is to be followed, while if the value is above the right bound of the interval the right branch should be followed. However, if the value falls inside the split-range interval, *both* branches are followed. Each of these branches receives a certain weight, depending on the place of the value in the interval. This weight is similarly distributed further down the tree, until the leaves are reached.

To define this mathematically, we use the concept of *membership* and a *piecewise linear membership function*, which was introduced by Ref. [24]. The top node (*root*) has the highest possible membership: 1. When a fuzzy split is encountered, the current membership is divided over the child nodes. These fuzzy splits are defined by two parameters, $\alpha$ and $\beta$, which are both real numbers and express the interval $[\alpha - \beta, \alpha + \beta]$. Let the current membership be $m$ and the attribute-value of the split-attribute be $x$, the membership of the left branch, defined as $m_L(x, m)$, is equal to $\min\left(\max\left(m \cdot \frac{\alpha + \beta - x}{2 \cdot \beta}, 0\right), 1\right)$ and the membership of the right branch, $m_R(x, m)$ is equal to $m - m_L$. The functions defining membership are explicitly stated below:

$$m_L(x, m) = \begin{cases} m \cdot \dfrac{\alpha + \beta - x}{2\beta} & \text{for } \alpha - \beta \leq x \leq \alpha + \beta \\ m & \text{if } x < \alpha - \beta \\ 0 & \text{else} \end{cases}$$

$$m_R(x, m) = \begin{cases} m \cdot \dfrac{x - \alpha + \beta}{2\beta} & \text{for } \alpha - \beta \leq x \leq \alpha + \beta \\ m & \text{if } x > \alpha + \beta \\ 0 & \text{else} \end{cases}$$

Equation 1 membership functions.

When using the model for predicting, the final predicted outcome for a subject is the outcome which has the highest aggregated membership over all leaves with the same outcome (weighted majority). For an example, see Appendix C. If both outcomes are tied, a good outcome is predicted since this is the most common outcome.

The split-range together with its concept of membership is the fuzzy part of the decision tree: it allows the tree to define an interval in which it admits to be *unsure* or *in transition*. Gradual transition of membership from one branch into another might more accurately express the gradual change in health and outcome when values of numerical variables, such as age or weight, are slowly increasing or decreasing. Combining the outcome prediction of both branches by using the calculated membership allows for the combination of multiple hypotheses, taking into account more of the available information and utilizing more of the predictive power of the tree, possibly leading to better and more stable predictions. Additionally, the resulting membership distribution across dependent variables provides insights into the confidence in prediction for a specific instance, which can be useful for users. Appendix C contains a mathematical proof of the predictive advantage of fuzzy algorithms over orthogonal (non-fuzzy) trees for a given size restriction. Similar to the CART algorithm, when training the model, the prediction of a specific leaf is set to the highest weighted outcome category, where the weight of an outcome is determined by the sum of membership over all training subjects in the leaf with the corresponding outcome. If the aggregated weight for both outcomes is equal, a good outcome is assigned to the leaf. Predictions of leaves are reassigned during the training phase of the algorithm if the highest weighted category changes.

To search for the fuzzy decision tree with the highest prediction accuracy on unseen data under certain size restrictions, a specialized evolutionary algorithm is developed. An explanation of this algorithm is given in the following section.

## 2.4. Evolutionary algorithms for decision trees

Evolutionary algorithms are iterative algorithms, which can be used as a meta-algorithm to improve existing models or predictors, such as decision trees. For a description of the general evolutionary method and the common terminology, we refer to Appendix F. In this study, we construct and use an evolutionary algorithm that searches for the most *suitable* fuzzy algorithm, where *suitability* is defined as the desirable trade-off between accuracy and size. An appropriate trade-off helps to

avoid overfitting and enhances interpretability by improving compactness of the generated model. The *fitness function*, which quantifies this objective, is discussed in the next section together with the procedures for *initialization, selection, crossover, mutation, intermediate pruning, and imputation*.

### 2.4.1. Fitness function

The main objective of the algorithm is to find a fuzzy decision tree that maximizes the prediction accuracy on unseen data, while limiting the size of this tree to assure interpretability. As common with decision trees, fuzzy decision trees are prone to overfitting. Since a leaf always predicts its most occurring category, converting a leaf into a decision node (and consequently additional downward leaves) will never decrease the number of correctly classified elements in the training set. To restrict growth and thereby avoid overfitting, while also promoting interpretability by favoring concise trees, a second objective to minimize the size of the tree should be included into the fitness function. This is achieved by defining the objective function as a weighted linear multi-objective function. The measure for the second objective is defined as the number of nodes (leaf and decision nodes) in the tree, which needs to be minimized. The number of nodes will be incorporated into the fitnes function as a squared term, which expresses the reasoning that an extra decision node will be more likely to cause overfitting when the tree is already large. The weights for size and accuracy are set with upper and lower bounds and are allowed to be altered during the run of the algorithm. This allows the algorithm to dynamically switch between the *explorative phase* (accuracy receives a high weight) and the *exploitative phase* (size measure receives a high weight) [29]. We define the parameter T as total time the evolutionary algorithm is allowed to run, and the parameter g as the current number of generations simulated respectively. The fitness for a tree in the population (referred to as an *individual*) I, which has to be maximized, will be determined with the following formula:

$$f(I, g, T) = -e(I) - w_a(g, T) \cdot s(I)^2$$

Equation 2 Fitness function evolutionary algorithm

Where $e(I)$ is the amount of misclassifications of the tree (which is equivalent to using accuracy), $s(I)$ is the total number of nodes (decision + leaf, or *length*) and $w_a(g, T)$ as the penalty for size. The value of this penalty is set to a predefined value in the first generation and is re-evaluated every next 25 generations. To promote exploring new solutions, the penalty is decreased by 10% if a considerable fraction of the last 25 generations shows little variation. We define a single generation to have little variation if over 50% of its individuals have the same number of nodes and errors. The fraction of generation for which this should hold, is set to $20 \cdot \frac{t(g)}{T}$, with $t(g)$ the running time of the algorithm until generation g, which expresses the reasoning that convergence of solutions should not be actively discouraged when the algorithm is almost terminated. Additionally, to promote accuracy and convergence, the value of $w_a(g, T)$ is increased by 10%, if less than $20 \cdot \frac{t(g)}{T}$ of these last 25 generations have less than 10% of individuals with the same amount of nodes and errors.

Preliminary tests described in Appendix B show that without re-evaluating the penalty, a value of 0.01 gave the lowest number of misclassifications. Therefore, the starting value of $w_a(g, T)$ is set to 0.01. Moreover, the number of misclassifications sharply rose when the penalty fell below 0.01, which could indicate that the algorithm overfits the training sample. Hence, we restricted the penalty value to only values above and including 0.01, avoiding bias against decision trees that overfit. Additionally, the above method together with its hyper-parameters (25 generations, 10% and 50%, and the defined fraction)

was constructed after a series of pre-runs indicating that the method obtains superior results to a static penalty value.

### 2.4.2. Initialization

Our algorithm creates a first generation of decision trees using the *grow* method [15,23]. This method creates a diverse initial population with respect to size and decision nodes, which might circumvent introducing a bias from the very beginning. Additionally, to limit uncontrollable growth, we defined an upper bound on the number of decision nodes to be created in a newly initialized tree, referred to as hyperparameter *l*. The upper bound was found by fine-tuning this hyper-parameter together with the number of individual trees (in short, individuals) to initialize, the *population size*, and the probability of a randomly generated node in the *grow* method to be a leaf ($p_{leaf}$) (Appendix B). The best maximum intilization depth was found to be 1 and the number of individuals to initialize was 650. The population size remained unchanged for the entire algorithm. A more elaborate description of the implemented *grow* procedure can be found in Appendix F.

The values of α and β, which define the fuzzy interval as explained in section 3.3, are randomly set in the initialization phase. The variable β is chosen uniformly between [0, 0.5·*range(r)*], where *range(r)* is the difference between the maximum and the minimum value of the earlier chosen splitting-attribute *r* (set by the grow method). The value of β is set to zero for binary and non-ordinal categorical variables: they do not possess any form of gradual change, which makes it inappropriate to incorporate fuzziness. Furthermore, the value of α is randomly and uniformly drawn out of all values in the dataset of attribute *r* (each value has equal probability of being drawn).

### 2.4.3. Selection

*Tournament selection* [30] is used to select which individuals continue to the crossover phase and thereby the next generation. This procedure, described in Appendix F, needs a predefined parameter *k*, indicating the size of the tournament. Large values of *k* lead to a high *selection pressure*: the probability of a weak individual surviving until the next generation is low. A high selection pressure might cause the algorithm to converge before having explored the entire search space, leading to local, but not global, optima. If selection pressure is too low, the algorithm might never converge at all (due to *drift* [31]). It might be favorable to vary this pressure for different stages of the run. Jebari et al. (2011) gave a concise review about tournament selection and proposed a method to change the parameter *k* dynamically, dependent upon the diversity in the population. This is implemented in our algorithm, where *k* is increased or decreased in the same way as $w_a$. The start value of *k* is set to 8, which was found by the fine-tuning procedure described in Appendix B.

Additional to the *tournament selection* method, our algorithm uses an *elitist strategy* [23,32] and an *immigration strategy* [20]. An *elitist strategy* ensures that a predefined number of fittest individuals always advance unchanged to the next generation. Consequently, building blocks composing the fittest individual are never lost and will always be available for selection and crossover. Fine-tuning this hyper-parameter (Appendix B) showed an elite of 1 provides the fittest results, which is the number used in our proposed algorithm. An *immigration strategy* introduces a predefined amount of newly created individuals into the population when there is an indication that the algorithm is stuck in a (local) optimum. We define the algorithm to be stuck when the lower bound on misclassifications over all individuals has not been 4 misclassifications lower than the upper bound on misclassifications for 100 sequential generations. When this happens, 10% of the least fit

individuals are replaced with newly initialized individuals. The hyper-parameters are defined after a series of pre-runs indicating it obtains superior results to static values.

### 2.4.4. Crossover

Our algorithm uses the common one-point crossover [23], which is explained in Appendix F. Crossover probability ($p_{cross}$) is set at 0.151 and was determined by fine-tuning this hyperparameter (Appendix B). Individuals from the selected population that are not marked for crossover advance unchanged to the mutation phase.

### 2.4.5. Mutation

Individuals entering the mutation phase continue unchanged to the next generation with a predefined probability of $(1 - p_{mut})$. All remaining individuals reaching the mutation phase are mutated by a random alteration of one node. The node is selected randomly and uniformly over all nodes in the tree. The chosen node is converted into (or stays) a leaf with probability $p_{leaf}$, which is the same probability as used during initialization (3.4.2). If the node becomes a leaf, its entire subtree is removed. If not, a new split-attribute is chosen randomly and uniformly from all possible attributes. The variables $\alpha$ and $\beta$ are set in the same way as during initialization (3.4.2). The child-nodes stay unchanged. If the chosen node was a leaf, new leaves are added as children.

The value $p_{leaf}$ controls the growth of the tree. A large $p_{leaf}$ not only causes mutated trees to be smaller on average, but also removes possible valuable structures. The values for $p_{leaf}$ and $p_{mut}$ were fine-tuned to 0.15 and 1.0 respectively (Appendix B). The mutation probability $p_{mut}$ is altered during the run with the same procedure as $w_a$ is altered, except that $p_{mut}$ is increased when $w_a$ is decreased and vice versa. Since the values $\alpha$ and $\beta$ are chosen randomly, it might take many iterations for the values to converge to their optimal values after changes higher in the tree. We implemented a procedure to accelerate the convergence. When the maximum fitness value has not improved in 400 generations, both $\alpha$ and $\beta$ are adjusted up (+10%) or down (−10%) for every node in every individual tree. Adjustments that improve the fitness value are kept. This method together with its hyperparameters (400 generations and the percentages +10% and −10%) was incorporated into the algorithm after a series of pre-runs indicated it reduced the number of misclassifications in the final tree.

### 2.4.6. Incorrect trees and pruning

Even though the fitness function incorporates a penalty for size, it does not provide the necessary subtlety we desire: insignificant splits on small sets of data can still occur. Although selection pressure will eventually remove individuals with insignificant splits, it can take many iterations. It might be beneficial to guide the algorithm and prohibit splits which create near-empty branches. For this reason, we introduced a hard restriction: leaf nodes may never hold less than a predefined fraction of test subjects, $f_{leaf}$, which is set to 0.05 (fine-tuned by the procedures described in Appendix B). If, after executing crossover and mutation, less test subjects than this predefined number end up in this leaf, the decision node that causes this violation is *pruned*: it is replaced with either the first downward decision node that does not violate the restriction, or with a leaf. Note that a high fraction $f_{leaf}$ could cause aggressive pruning and removal of decision nodes that might have had predictive value after being relocated in a subsequent mutation or crossover phase.

Additionally, to limit growth, all trees are pruned more aggressively when the average length has been increasing for a considerable amount of generations. This is measured by defining a counter which starts at zero and increases by one if the average length over all individuals in a generation has increased with more than 50% compared to the previous generation and the average number of misclassifications over all individuals has not changed with more than 30% compared to the previous generation. If the counter reaches 250, then the minimum amount of

allowed test subjects per leaf is increased with a factor 2 and all trees are pruned. The counter resets when the average length decreases and the difference in average amount of misclassifications increases by 50% in one generation. This method and its hyperparameters (250, 2, and 50%) were defined after a series of pre-runs indicated it improved the results of the algorithms by reducing the final amount of misclassifications.

### 2.4.7. Imputation

The dataset contains missing values for both the independent variables and response variables: in the entire dataset 21.6% of the values are missing. The fuzzy and CART algorithms differ on their treatment of missing independent variables. If a decision node is reached in the fuzzy algorithm for which the current subject misses its corresponding attribute-value, the algorithm continues as if the value is equal to the split-value $\alpha$. This means that the algorithm travels down both branches, with equal membership. For the CART algorithm the treatment of missing independent variables is described in section 3.2.

Subjects with a missing *mRS* can not be removed without the risk of adding bias to the data. The same methods were considered to impute these response variables as in the CART algorithm (*no imputation, mean, median, most occurring and extended dataset*), as described in section 3.2. The method *median* would always result in the same imputation as the method *mean*. Of these methods, preliminary tests showed that *no imputation* resulted in the highest average fitness. The setup of the preliminary tests is explained in Appendix B.

## 3. Experimental setup

Three algorithms were compared: the *fuzzy* algorithm and two benchmarks, CART with the Gini impurity as splitting criterion and CART with entropy as splitting criterion. They were compared using *stratified* 5-fold cross-validations on the entire dataset (each fold contains about the same distribution of poor outcomes to good outcomes as in the entire dataset). Although subjects with missing outcomes are included, predictions for these subjects are not counted as either correct or incorrect, since the actual outcome is unknown. The 5-fold cross-validation was computed 6 times, using differently assigned folds each of the 6 times, generating 30 results per algorithm. This results in an average of 1090 patients with known outcome in the training set and 273 patients with known outcome in the test set. This setup was chosen as an adequate tradeoff between running time, sample size for the statistical test (30) and the size of the test set. Under the defined fitness function and with the chosen amount of test subjects it was considered unlikely that leaves in the trained tree would overfit and end up empty or near-empty. The obtained 30 results were used to test the following $H_0$-hypothesis:

**H0**. *the evolutionary algorithm does not make less classification errors than either of the two CART methods*

Since we only have 30 results per algorithm, and the results of the same algorithm might be correlated, we cannot assume normality of the errors. Therefore, we used the Wilcoxon signed-rank test with a 5% significance level ($\alpha = 0.05$). Additionally, the average length and number of false negatives and false positives are reported to give some additional insight in the behavior and possible bias of the algorithm. For the same purpose, a histogram was created with the number of occurrences per variable. It was constructed by running a simplified version of the evolutionary algorithm for 200 generations, where only the occurrences in generation 100–200 are counted.

Finally, after testing our hypothesis, all three algorithms were used to create a final decision tree. No statistical results originate from this and the obtained lengths or misclassifications are not incorporated in the previously explained experiment: these trees are only created for interpretation. For this, the dataset was split into a *test* and *training* set. The training set contains about two-thirds of the entire dataset (992 subjects, 602 moderate to good outcome, 310 poor outcome, 80 missing

outcome), while the test set contains the remaining one-third (496 subjects, 291 moderate to good outcome, 160 poor outcome, 45 missing outcome). Which subjects belong to the training set and which to the test set, is decided once, at random. All three algorithms are trained on the same training set and tested on the same test set. Predictions for subjects with missing outcomes are neglected.

## 4. Results

The *fuzzy* algorithm found its final solution on average within the first hour of running. In the sets that were used to evaluate the tree *(test folds)*, there were on average 179 moderate to good outcomes (positives) and 94 poor outcomes (negatives). Table 1 shows the average number of misclassifications and the standard deviation (SD) of the number of misclassifications, together with the Wilcoxon signed-rank p-value. Additionally, the average length (number of nodes), false positives (FP), and false negatives (FN) in absolute numbers are given in Table 2.

The decision trees generated with the training set are displayed in Fig. 2, Fig. 3 and Fig. 4. Colored nodes represent leaves. Green leaves predict an outcome of 1 (moderate to good outcome); red leaves predict an outcome of 0 (poor outcome). Each leaf also contains information about the number of subjects in the training set which end up in the leaf, as well as the median membership and amount of misclassifications (errors) of these subjects in the leaf. The number of errors is calculated as the sum of membership of all wrongly classified test-subjects in the leaf and might therefore not sum to the grand total. Additionally, each decision node contains the variable name of the split-variable and the boundaries of the split-range. An elaborate description of the variables is given in Appendix A. The number of misclassifications in the entire tree for the test set is displayed at the bottom of the figure.

The number of times a variable occurred as a split variable in all generated trees is depicted in Fig. 5. Since trees with a low fitness value are less likely to survive selection, split variables that decrease fitness will have a low probability to occur in many trees. Hence, occurrence of a variable indicates a beneficial impact on the fitness. The variable age was by far the most occurring variable, but it is omitted since displaying it would decrease visibility of the smaller bars considerably. It is important to realize that due to the design of the evolutionary algorithm, it is intrinsically biased towards variables with a large range of values.

## 5. Discussion

Our study results suggest that the proposed evolutionary algorithm for fuzzy decision trees can outperform the conventional CART algorithm in the MR CLEAN Registry dataset with respect to classification errors when predicting moderate to good *mRS* outcome ($\leq 4$) versus poor outcome ($mRS > 4$). Additionally, the trees created by the evolutionary algorithm are significantly smaller than those created by the CART algorithm. For these reasons, fuzzy evolutionary methods for decision trees can be a good alternative to the CART algorithm for predicting poor outcome after acute ischemic stroke. This approach is general and could, therefore, be applied to other applications and other datasets. However, other applications may have different boundary conditions and a comparison of model performance to other outcome prediction models and validation with other datasets is necessary to evaluate the general performance of the proposed algorithm.

**Table 1**
Number of misclassifications in test fold. Abbreviations: SD, Standard deviation; NA, Not applicable.

| Algorithm | Misclassifications (mean and SD) | 1-sided p-value Gini | 1-sided p-value Entropy |
|---|---|---|---|
| Fuzzy | 77.2 (6.94) | $9.31 \cdot 10^{-5}$ | 0.00127 |
| B. Gini | 83.5 (6.50) | NA | NA |
| B. Entropy | 82.2 (8.13) | NA | NA |

**Table 2**
Length as number of decision nodes and leaves in test fold, absolute numbers. Abbreviations: FP, false positive; FN, false negative; SD, standard deviation; NA, not applicable.

| Algorithm | Length (mean and SD) | FP (mean and SD) | FN (mean and SD) |
|---|---|---|---|
| Fuzzy | 11.4 (2.19) | 50.4 (8.78) | 26.8 (5.96) |
| B. Gini | 20.8 (1.21) | 59.9 (9.97) | 23.6 (9.70) |
| B. Entropy | 30.3 (1.69) | 52.0 (9.13) | 30.2 (8.71) |

Most of the variables identified as being important, by being included in the decision trees (Figs. 2–4) or showing a high frequency in Fig. 5, have already been shown in the literature to have predictive value. The variable *age* shows considerable predictive power (Figs. 2–5) and can be considered one of the most important predictors, with higher values above 60 and 70 being the most common split for poor outcome. *National Institutes of Health Stroke Scale at baseline (NIHSS), collaterals,* and *leukoaraïosis on baseline NCCT* have also already been identified as important predictors of functional outcome, which is shown by their presence in three, two and two of the created decision trees respectively, and the high frequency of occurrence during the run of the evolutionary algorithm (Fig. 5). This is consistent with earlier results from other models predicting endovascular treatment outcome for acute stroke [6, 33–35], although the variables *age* and *duration from onset to hospital or groin puncture* had no considerable predictive value in the results in Brugnara et al. (concerning age) and Grech et al. (concerning duration). Other predictors that appear in the final decision trees are: *C-reactive protein (CRP), pre-stroke mRS* and *glucose levels*. Despite less common, most of them have already been identified as predictive in other studies that investigated the prediction of poor functional outcome using the same dataset with different methods [7]. Additionally, the overall performance is similar to the performance of other machine learning algorithms on the same dataset [6]. The combination with algorithms predicting the same outcome variable with different datasets and approaches might give additional insights in the predictive value of the dataset compared to other datasets, such as the work of [36]. However, the error rate is still quite high ($> 20\%$), leading to a significant risk of incorrectly classifying a patient. Although the algorithm outperforms CART, more understanding is needed about potential benefits and costs of correctly or incorrectly classifying a patient. As long as there is no clear guideline on the desirable tradeoff between sensitivity and specificity and the required accuracy of the prediction model [7], the created decision trees with their relatively high error rate should not be used in clinical practice, but should be seen as a proof of concept illustrating its potential use.

The improved performance of fuzzy trees over CART might have been caused by the way the trees were generated (evolutionary over a greedy search) or the way missing variables were treated (traveling down both branches over imputation). It might also indicate that the fuzzy splits from the *fuzzy* decision trees might be a more accurate display of reality over the conventional orthogonal splits used by CART. Note that, contrary to the results in Table 1, the final decision tree shown in Fig. 2 obtains a *higher* error on the test set than CART with entropy as splitting criterion (Fig. 3). Since the final decision trees are generated from one run, they are susceptible to the variance in the results, which is shown to be high (Tables 1 and 2). It seems that small changes in the dataset can cause large differences in classification errors. This variance does not only occur in the results of the fuzzy algorithm, but also in CART. Variance in outcome and created prediction models can cause the method to be somewhat unstable: the prediction model obtaining the least amount of misclassifications might differ significantly depending on the subset data used, which is not desirable from a practical point of view. Ensemble learners, such as random forests, can benefit from this variance, and can return a more stable prediction model, which does not change when the training set is slightly altered [37,38]. Implementing the developed method in an ensemble learner, such as a random forest,
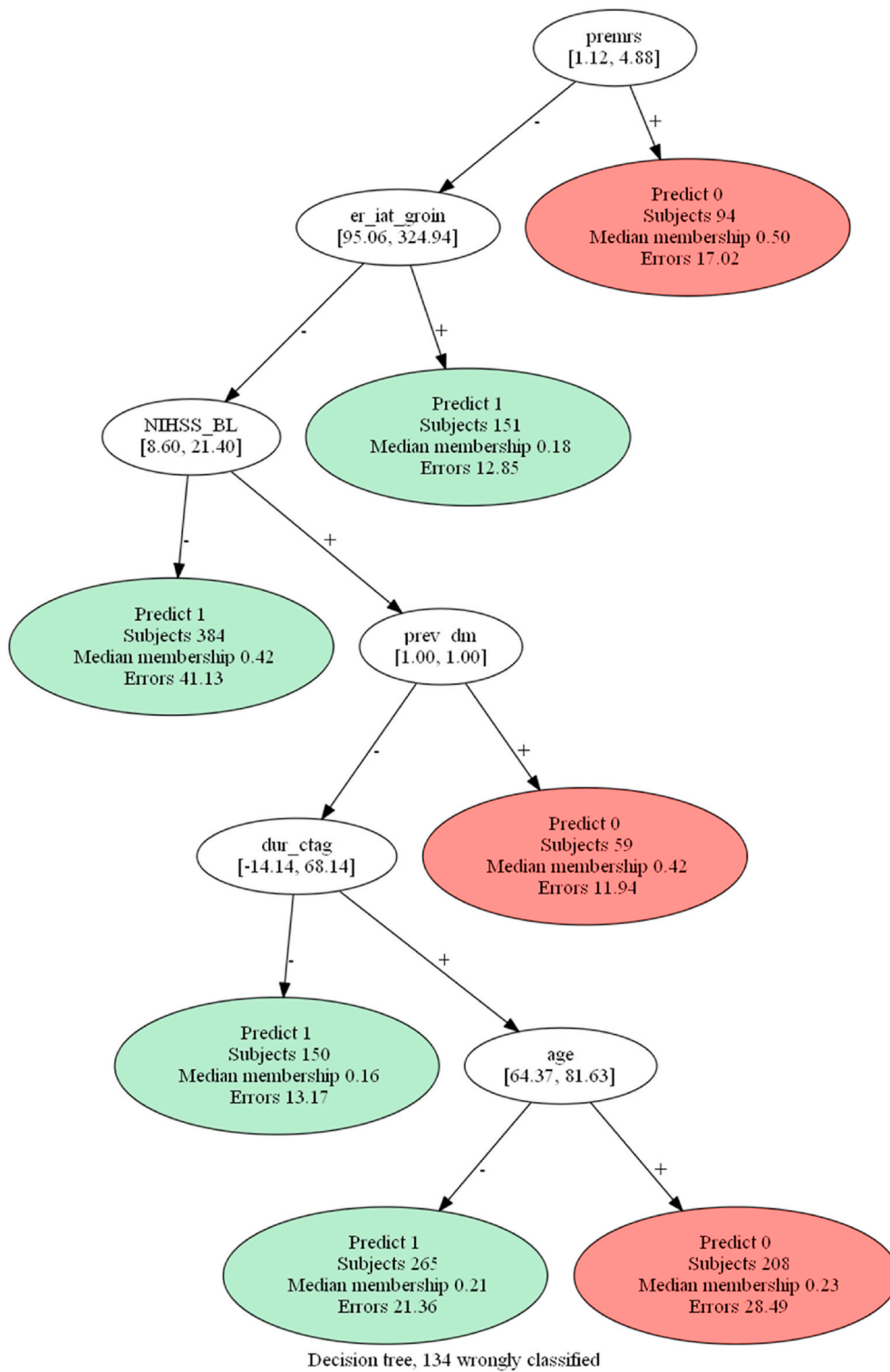
**Fig. 2.** Generated fuzzy decision tree: 134 wrongly classified in the test dataset. Details on the split variables can be found in Appendix A. Abbreviations: premrs, pre-stroke mRS; er_iat_groin, duration in minutes from presentation at the emergency room to groin puncture (start of endovascular thrombectomy); NIHSS_BL, National Institutes of Health Stroke Scale as a measure for stroke severity at baseline; prev_dm, history of diabetes mellitus.
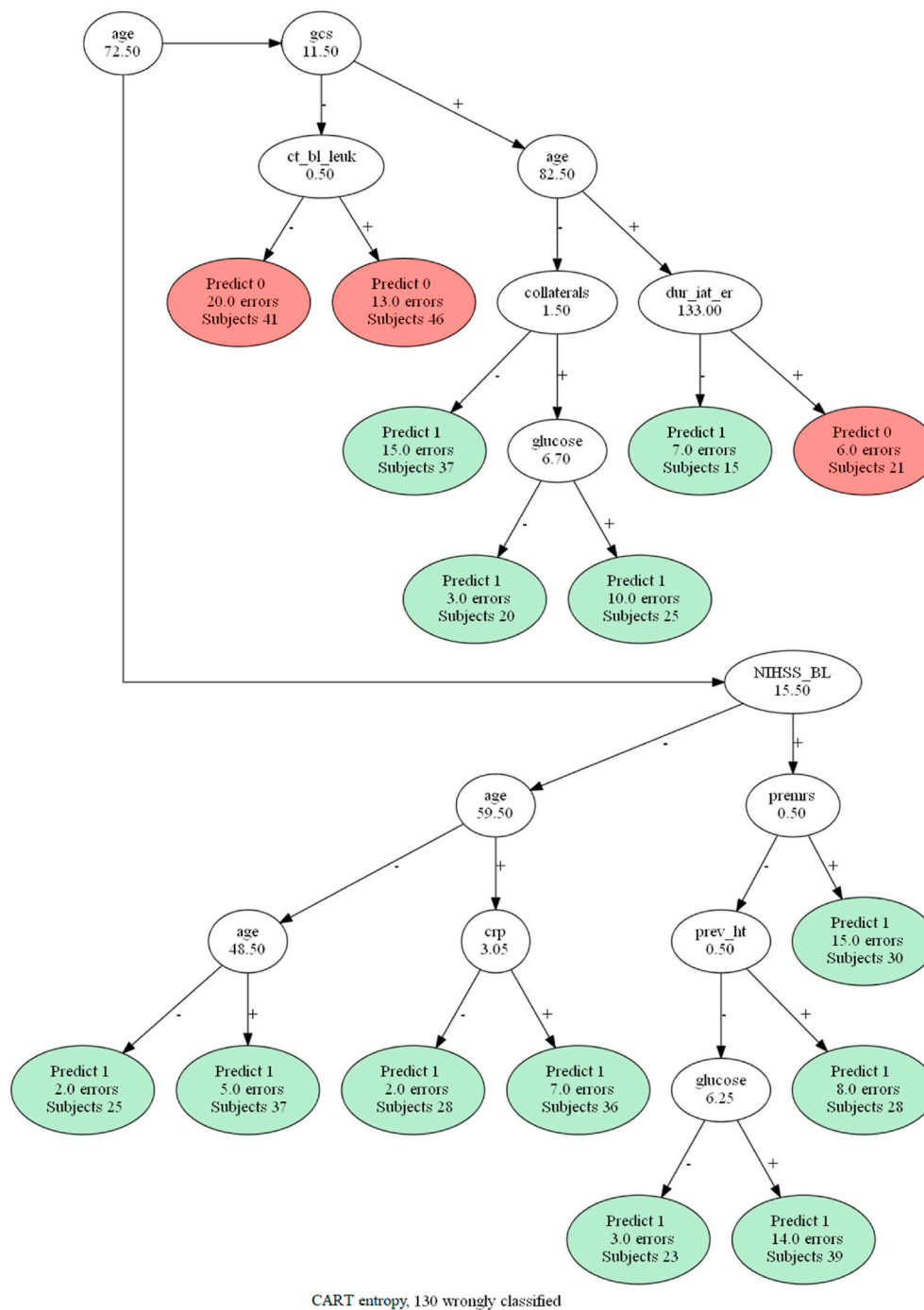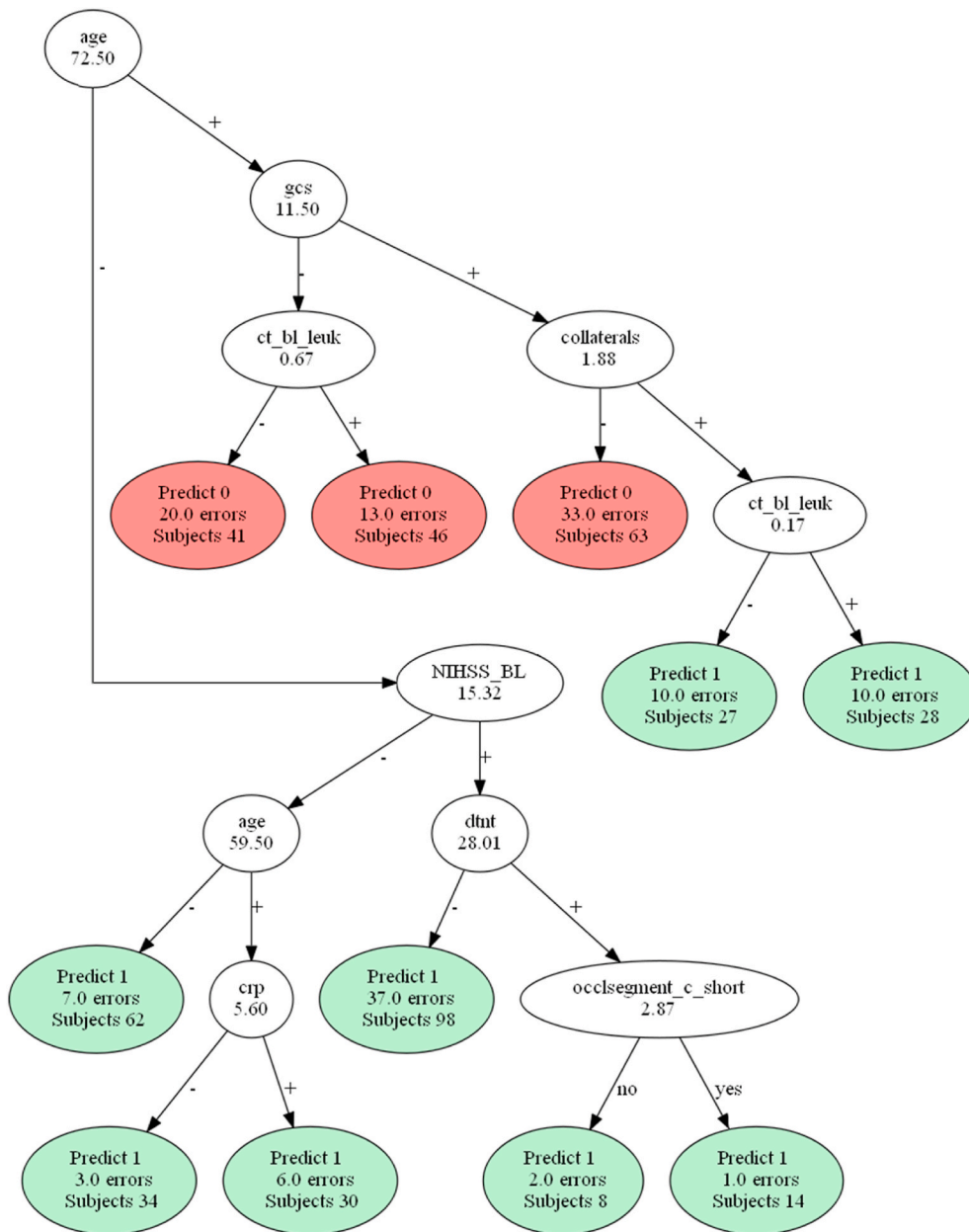
**Fig. 3.** Generated CART decision tree (splitting criterion: entropy): 130 wrongly classified in the test dataset. This tree obtains less wrongly classified subjects then the fuzzy decision tree, but is far larger due to the lack of an appropriate pruning method. Details on the split variables can be found in Appendix A. Abbreviations: age, age in years; gcs, Glasgow Coma Scale for patient level of consciousness at baseline; ct_bl_leuk, brain white matter lesions on baseline imaging; collaterals, degree of collateral blood flow to the infarcted brain region; dur_iat_er, duration in minutes from stroke onset to emergency department of intervention centre; glucose, blood glucose level at baseline; NIHSS_BL, National Institutes of Health Stroke Scale for stroke severity at baseline; premrs, pre-stroke mRS; crp, blood C-reactive protein level at baseline; prev_ht, history of hypertension.

might give a prediction model which is better transmittable to other datasets, and will obtain similar results in new datasets as in the data with which they were trained. However, ensemble learners are not as easy to interpret as decision trees. The trade-off between interpretability and amount of correct classifications should play a role in the decision on when ensemble learners should be used. Additionally, note that the CART with the Gini impurity as splitting criteria obtains on average less false negatives (non-true poor outcome) than our proposed method. This does not affect our conclusion, since we did not define a predilection for either false positives or false negatives over the other in this study. However, if desired, the fitness function of the *fuzzy* algorithm can be altered to reflect a difference in preference between the two types of misclassifications.

Besides the severe variance, results from the current study are characterized by several other limitations. The parameter values (size of population, size of tournament, etc.) or hyperparameters (used to update the fitness weight, mutation probability, fuzzy parameters, prune or introduce new individuals) were found by running preliminary tests, but there is no guarantee that the obtained values are optimal with respect to running time or classification performance. Even without optimal parameters, the stochastic nature of evolutionary algorithms would theoretically guarantee the optimal decision model is found if the algorithm runs long enough, but there is no bound on how long the algorithm needs to run to reach this optimal solution. More extensive research on the optimal values of the parameters might result in better parameters, improving both the classification results and running time of the fuzzy algorithm and the CART algorithm. Additionally, the limited runtime of each preliminary test might have introduced a bias towards

**Fig. 4.** Generated CART decision tree (splitting criterion: Gini): 142 wrongly classified in the test dataset. Notice the tree contains many unnecessary branches due to an inappropriate pruning method. Details on the split variables can be found in Appendix A. Abbreviations: age, age in years; gcs, Glasgow Coma Scale for level of consciousness at baseline; ct_bl_leuk, brain white matter lesions on baseline imaging; collaterals, degree of collateral blood flow to the ischemic brain region; NIHSS_BL, National Institutes of Health Stroke Scale for stroke severity at baseline; dtnt, duration in minutes from hospital arrival to start of intravenous thrombolysis; crp, blood C-reactive protein level at baseline; occlsegment_c_short, location of intracranial occlusion causing the stroke.

parameters which cause the algorithm to converge towards a local optimum too quickly. Parameters exploring more of the search space and generating many different kinds of trees but not reaching a local optimum within the limited preliminary test might have been more beneficial in the final experiment. However, running preliminary tests longer would also mean fewer parameter values could be explored within the same time. A better understanding of this tradeoff could improve preliminary testing and thereby the results in the final experiment. Third, practitioners indicated that they suspect missing outcomes of the dependent variable are correlated to a poor outcome. Imputing data missing not at random with a method that uses the non-missing outcomes as input will by definition be based on a biased sample. The imputation method of the dependent variable used in the evolutionary algorithm (omitting patients with missing *mRS* score) could cause the outcome prediction for these patients to contain a bias to the prevalent outcome (good or moderate). An imputing method without this

potential bias might improve not only the classifications on the test set, but also the representation of reality. This might require obtaining additional data to mitigate the problem of the bias in the original data. Fourth, although we have worked with one of the most extensive databases of endovascularly treated patients with an acute ischemic stroke, the decision variables and therefore the created decision trees are impacted by the variables included in the dataset. Variables in the dataset might be associated with each other and with other (unavailable) characteristics that can be extracted from stroke patients. Moreover, it may be that other (strongly related) variables may be chosen in decision models were it to be run on a different or extended dataset. This also holds true for a dataset extended with full images obtained from radiological data, which was omitted in the study in favor of manually-derived imaging biomarkers as an alternative, since the high-dimensionality of the images would strongly complicate the analyses. Lastly, optimizing CART was not in scope of this article. Adding a
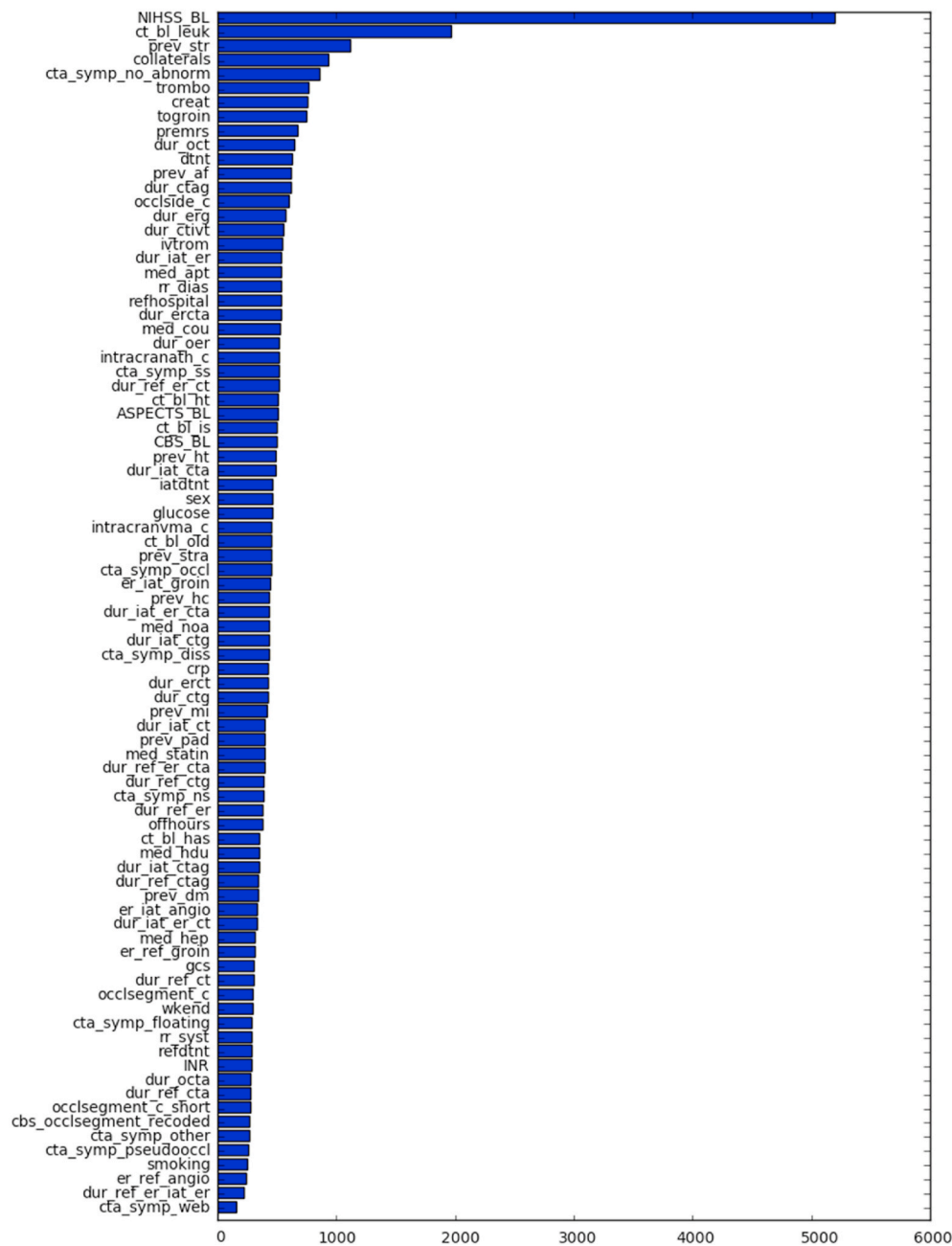
**Fig. 5.** Histogram variable occurrence in fuzzy decision tree. A description of the variable encoding is available in Appendix A (Table 4). The variables NIHSS_BL, ct_bl_leuk, prev_str, collaterals and cta_symp_no_abnorm occur most often, indicating the benefit of a split on these variables to the fitness value. All these variables are binary, except NIHSS_BL (numerical, 32 different values) and collaterals (categorical, 4 different values). Notably, age was by far the most occurring variable, but is omitted from this graph to enable interpretability of the other variables.

pruning phase to CART to remove the many unnecessary decision nodes (Figs. 3 and 4), or using a different imputation method, might influence the results. Comparisons to different implementations of CART, together with comparisons to other prediction models, can be the subject of successive studies.

## 6. Conclusions

Today's widespread use of EVT is the result of successful patient selection for the positive trials of 2015, in addition to improved time metrics and procedural results of EVT [2,39]. Further improvements in patient selection will facilitate future advances in patient outcomes after acute ischemic stroke. The presented evolutionary algorithm provides insights in the outcome prediction in acute ischemic stroke with a good accuracy and reduced complexity. The algorithm proved increased performance in accuracy over the compared algorithm CART, while

generating on average smaller trees. However, improvements are necessary before this approach can be used in clinical practice.

ITEA3—Medolution: Project number 14003.

## Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2021.104414.

## References

[1] E.S. Donkor, Stroke in the Century: A Snapshot of the Burden, Epidemiology, and Quality of Life, Stroke research and treatment, 2018, https://doi.org/10.1155/2018/3238165, 2018.

[2] M. Goyal, B. Menon, Z. v, D. Dippel, P. Mitchell, A. Demchuk, G. Donnan, Endovascular thrombectomy after large-vessel ischaemic stroke: a meta-analysis of individual patient data from five randomised trials, Lancet 387 (10029) (2016) 1723–1731, https://doi.org/10.1016/S0140-6736(16)00163-X.

[3] I.G. Jansen, M.J. Mulder, R.J. Goldhoorn, Endovascular treatment for acute ischemic stroke in routine clinical practice: prospective, observational cohort study (MR CLEAN Registry), BMJ (2018) 50–70, https://doi.org/10.1136/bmj.k949.

[4] M. Goyal, M.A. Almekhlafi, C. Cognard, R. McTaggart, K. Blackham, A. Biondi, M.D. Hill, Which patients with acute stroke due to proximal occlusion should not be treated with endovascular thrombectomy? Neuroradiology 61 (1) (2019) 3–8, https://doi.org/10.1007/s00234-018-2117-y.

[5] E. Venema, M.J. Mulder, B. Roozenbeek, J.P. Broderick, S.D. Yeatts, P. Khatri, H.F. Lingsma, Selection of patients for intra-arterial treatment for acute ischaemic stroke: development and validation of a clinical decision tool in two randomised trials, BMJ 357 (2017) j1710, https://doi.org/10.1136/bmj.j1710.

[6] H. van Os, L.A. Ramos, A. Hilbert, M. van Leeuwen, M. van Walderveen, N.D. Kruyt, M.C. Investigators, Predicting outcome of endovascular treatment for acute ischemic stroke: potential value of machine learning algorithms, Front. Neurol. 9 (2018) 784, https://doi.org/10.3389/fneur.2018.00784.

[7] L.A. Ramos, M. Kappelhof, H.J. van Os, V. Chalos, K. van Kranendonk, N.D. Kruyt, H.A. Marquering, Predicting poor outcome before endovascular treatment in patients with acute ischemic stroke, Front. Neurol. 11 (2020) 1215, https://doi.org/10.3389/fneur.2020.580957.

[8] V. Podgorelec, P. Kokol, Towards more optimal medical diagnosing with evolutionary algorithms, J. Med. Syst. 25 (3) (2001) 195–219, https://doi.org/10.1023/A:1010316906.

[9] C. To, T.D. Pham, Analysis of cardiac imaging data using decision tree based parallel genetic programming, in: Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis, IEEE, Salzburg, 2009, pp. 317–320, https://doi.org/10.1109/ISPA.2009.5297730.

[10] M. Zorman, H.P. Eich, B. Stiglic, C. Ohmann, M. Lenic, Does size really matter—using a decision tree approach for comparison of three different databases from the medical field of acute appendicitis, J. Med. Syst. 26 (5) (2002) 456–477, https://doi.org/10.1023/A:1016461301710.

[11] V. Podgorelec, P. Kokol, B. Stiglic, I. Rozman, Decision trees: an overview and their use in medicine, J. Med. Syst. 26 (5) (2002) 445–463, https://doi.org/10.1023/A:1016409317640.

[12] L. Hyafil, R.L. Rivest, Constructing optimal binary decision trees is NP-complete, Inf. Process. Lett. 5 (1) (1976) 15–17, https://doi.org/10.1016/0020-0190(76)90095-8.

[13] T. Hancock, T. Jiang, M. Li, J. Tromp, Lower bounds on learning decision lists and trees, Inf. Comput. 126 (2) (1996) 114–122, https://doi.org/10.1006/inco.1996.0040.

[14] X. Wang, B. Chen, G. Qian, F. Ye, On the optimization of fuzzy decision trees, Fuzzy Set Syst. 112 (1) (2000) 117–125, https://doi.org/10.1016/S0165-0114(97)00386-2.

[15] X. Llorà, J.M. Garrell, Evolution of decision trees, in: Proceedings of the Fourth Catalan Conference on Artificial Intelligence, ACIA Press, 2001, pp. 115–122.

[16] M.P. Basgalupp, A.C. De Carvalho, R.C. Barros, D.D. Duiz, A.A. Freitas, Lexicographic multi-objective evolutionary induction of decision trees, Int. J. Bio-Inspired Comput. 1 (1–2) (2009) 105–117, https://doi.org/10.1504/IJBIC.2009.022779.

[17] R.C. Barros, M.P. Basgalupp, D.D. Ruiz, A.C. de Carvalho, A.A. Freitas, Evolutionary model tree induction, in: Proceedings of the 2010 ACM Symposium on Applied Computing, 2011, pp. 1131–1137, https://doi.org/10.1145/1774088.1774327.

[18] V. Papagelis, D. Kalles, Breeding decision trees using evolutionary techniques, ICML 1 (2001) 393–400, 10.1.1.85.3752.

[19] H. Zhao, A multi-objective genetic programming approach to developing Pareto optimal decision trees, Decis. Support Syst. 43 (3) (2007) 809–826, https://doi.org/10.1016/j.dss.2006.12.011.

[20] J.B. Gray, G. Fan, Classification tree analysis using TARGET, Comput. Stat. Data Anal. 52 (3) (2008) 1362–1372, https://doi.org/10.1016/j.csda.2007.03.014.

[21] K. Sörensen, G.K. Janssens, Data mining with genetic algorithms on binary trees, Eur. J. Oper. Res. 151 (2) (2003) 253–264, https://doi.org/10.1016/S0377-2217(02)00824-X.

[22] V. Podgorelec, P. Kokol, Evolutionary decision forests – decision making with multiple evolutionary constructed decision trees, Problems in Applied Mathematics and Computational Intelligence (2001) 97–103, 10.1.1.559.9756.

[23] R.C. Barros, M.P. Basgalupp, A.C. De Carvalho, A.A. Freitas, A survey of evolutionary algorithms for decision-tree induction, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (3) (2012) 291–312, https://doi.org/10.1109/TSMCC.2011.2157494.

[24] C. Olaru, L. Wehenkel, A complete fuzzy decision tree technique, Fuzzy Set Syst. 140 (3) (2003) 221–254, https://doi.org/10.1016/S0165-0114(03)00089-7.

[25] L. Breiman, J. Friedman, R.A. Olshen, C.J. Stone, Classification and Decision Trees, Wadsworth, Belmont, 1984.

[26] B. Farell, J. Godwin, S. Richards, C. Warlow, The United Kingdom transient ischaemic attack (UK-TIA) aspirin trial: final results, J. Neurol. Neurosurg. Psychiatr. 52 (12) (1991) 1044–1054, https://doi.org/10.1136/jnnp.54.12.1044.

[27] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, Z.H. Zhou, Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (1) (2008) 1–37, https://doi.org/10.1007/s10115-007-0114-2.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, J. Vanderplas, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[29] G.T. Pulido, C.A. Coello, The micro genetic algorithm 2: towards online adaptation in evolutionary multiobjective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, Berlin, Heidelberg, 2003, pp. 252–266.

[30] T. Blickle, Tournament selection, Evol. Comput. 1 (2000) 181–186.

[31] G.R. Harik, F.G. Lobo, A parameter-less genetic algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 1, Morgan Kaufmann Publishers Inc, Orlando, Florida, 1999, pp. 258–265, https://doi.org/10.5555/2933923.2933949.

[32] K.A. De Jong, Analysis of the Behavior of a Class of Genetic Adaptive Systems, vol. 266, 1975, https://doi.org/10.5555/907087.

[33] R. Grech, P.L. Galvin, S. Power, A. O'Hare, S. Looby, P. Brennan, J. Thornton, Outcome prediction in acute stroke patients considered for endovascular treatment: a novel tool, Intervent Neuroradiol. 20 (3) (2014) 312–324, https://doi.org/10.15274/INR-2014-10029.

[34] G. Brugnara, U. Neuberger, M.A. Mahmutoglu, M. Foltyn, C. Herweh, S. Nagel, M. Bendszus, Multimodal predictive modeling of endovascular treatment outcome for acute ischemic stroke using machine-learning, Stroke 51 (12) (2020) 3541–3551, https://doi.org/10.1161/STROKEAHA.120.030287.

[35] A. Sarraj, K. Albright, A.D. Barreto, A.K. Boehme, C.W. Sitton, J. Choi, O. Mir, Optimizing prediction scores for poor outcome after intra-arterial therapy in anterior circulation acute ischemic stroke, Stroke 44 (12) (2013) 3324–3330, https://doi.org/10.1161/STROKEAHA.113.001050.

[36] A. Hilbert, L.A. Ramos, H.J. van Os, S.D. Olabarriaga, M.L. Tolhuisen, M.J. Wermer, W.H. van Zwam, Data-efficient deep learning of radiological image data for outcome prediction after endovascular treatment of patients with acute ischemic stroke, Comput. Biol. Med. 115 (2019), https://doi.org/10.1016/j.compbiomed.2019.103516.

[37] P. Cunningham, J. Carney, Diversity versus quality in classification ensembles based on feature selection, in: Machine Learning: ECML 2000, Springer, Berlin, Heidelberg, 2000, pp. 109–116, https://doi.org/10.1007/3-540-45164-1_12.

[38] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844, https://doi.org/10.1109/34.709601.

[39] M. Mokin, A.A. Khalessi, J. Mocco, G. Lanzino, T.M. Dumont, R.A. Hanel, E. Veznedaroglu, Endovascular treatment of acute ischemic stroke: the end or just the beginning? Neurosurg. Focus 36 (1) (2014) E5, https://doi.org/10.3171/2013.10.FOCUS13374.

[40] T. Blickle, L. Thiele, A comparison of selection schemes used in evolutionary algorithms, Evol. Comput. 4 (4) (1996) 361–394, https://doi.org/10.1162/evco.1996.4.4.361.

[41] J.R. Koza, Genetic programming as a means for programming computers by natural selection, Stat. Comput. 4 (2) (1994) 87–112, https://doi.org/10.1007/BF00175355.

[42] R. Poli, W.B. Langdon, N.F. McPhee, A Field Guide to Genetic Programming, 2008. Retrieved from, http://www.gp-fieldguide.org.uk.

[43] Q. Zhao, M. Shirasaka, A study on evolutionary design of binary decision trees, Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 3 (1999) 1988–1993, https://doi.org/10.1109/CEC.1999.785518.