Master's Thesis in Business and Economics
Major in Quantitative Methods

Ensemble Creation with Grammatical Evolution Decision Trees

An application on real world data sets

Written by: Dominik Gaiato
Matriculation-Number: 2017-050-337
Supervisor: Prof. Dr. Dietmar Maringer

Faculty of Business and Economics
University of Basel
Spring Term 2022
Submitted on: 13.06.2022

1 Introduction

Grammatical Evolution was pioneered in a paper by Conor Ryan, J.J. Collins and Michael O'Neill in 1998. The authors presented a method to evolve complete programs in a flexible application context. Grammatical Evolution takes inspiration from nature in the sense that it creates populations of individuals that compete with one another to enter the next generation or pass on their genes to their children. Conor Ryan, J. J. Collins, and Michael O'Neill (1998)
In the following years the method became popular and different fields of applications were explored. One of them is the creation of Decision Tree classifiers using Grammatical Evolution. This method was introduced in a medical context to predict gene-gene interactions in a paper by Sushamna Deodhar and Alison Motsinger-Reif in 2010. Deodhar and Motsinger-Reif (2010)
From there, it was only a short step to apply the paradigm of ensemble creation found in Random Forest, to the trees generated using Grammatical Evolution. The idea of ensemble creation is to attain a higher generalization of the ensemble predictions when compared to individual classifiers.
A paper mentioning this topic in the context of Grammatical Evolution was written by Jeannie M. Fitzgerald et al. in 2017. The paper introduced a structured approach to classification and clustering using Grammatical Evolution Decision Trees. For more complex problems with a higher number of classes, the authors used the aggregation of individual classifiers into ensembles to improve the predictive power. The paper took an applied approach to the calibration of ensemble creation and stated the avenue for a more structured look into ensemble creation. Fitzgerald, Azad, and Conor Ryan (2017)
This thesis applies the creation of classification ensembles to four real-world data sets. They vary from a two-class problem up to a five-class problem. Three approaches to the creation of populations of individual Grammatical Evolution Decision Trees are inspected, with a focus on their diversity. From the resulting populations of classifiers, ensembles are created. Then the impact of different cut-offs for exclusion based on the diversity of the population and effects of choosing different initial ensemble sizes on the median test accuracy is inspected.
The creation of ensembles follows the two strategies proposed in Fitzgerald, Azad, and Conor Ryan (2017). The best-ensemble approach used the best classifiers of fifty evolved runs. The population-ensemble approach is constrained to the individuals within a run for selection. These approaches are compared to conventional Decision Trees and their ensemble method, Random Forest.
The next chapters are structured as follows: Chapter 2 provides a short overview of the methods used in this thesis. It introduces Decision Trees and Random Forest as comparison methods. The chapter provides the basics of Grammatical Evolution and the extension to Grammatical Evolution Decision Trees, as well as ensemble creation using the latter. Chapter 3 introduces the data sets and the calibrations used in the creation process of the different classifiers and their ensemble aggregation. Chapter 4 presents the results from the fitting process and discusses some findings and problems encountered. Chapter 5 recaps the previous sections and provides a short outlook.

2 Theoretical Foundations

This chapter introduces the classification methods used in the experiments in the next chapters. In the context of a classification task, the underlying data contains labels for each observation. The goal is to classify new observations for which the features are observed, but the class they belong to is unknown. The first section introduces Decision Trees and their ensemble method Random Forest. The second section provides an overview of Grammatical Evolution and its modifications for use in classification tasks through Grammatical Evolution Decision Trees. This is extended to the creation of ensembles with the evolved trees, similar to the aggregation of Decision Trees to Random Forest classifiers. Lastly, the Q-statistic as a measure of the diversity of classifiers in an ensemble is introduced.

2.1 Decision Trees

A Decision Tree (DT) is a hierarchical model using a tree-like structure to represent a decision-making process. A tree is composed of three parts: decision nodes, directed edges and leaf (or terminal) nodes. The decision nodes test conditions for features in the data. Multiple outgoing edges connect to further decision nodes or leaf nodes based on the outcome of the test in the previous decision node. The nodes without further outgoing edges are called leaf nodes. The leaf nodes contain the final decisions of the tree. The connection structure of nodes and edges forms the tree. In a classification context, the leaf nodes contain the class labels, to which the observations are assigned based on the divisions of the data in the decision nodes. A popular subtype of Decision Trees are binary Decision Trees, which always have two outgoing edges at the internal decision nodes. The use of continuous subdivision of the observations represents a filtering and refining process to increase the accuracy of the assigned class labels. Fitzgerald, Azad, and Conor Ryan (2017)

Decision Trees belong to the so-called glass box methods. The resulting tree can be interpreted by a human and used in the decision-making process. The classifier resulting from a Decision Tree can be visualized as a flow-chart. The reasoning behind the assignment of a class can be understood by following the decision nodes from the root of the tree to its leaves. Rokach and Maimon (2014)

The important parameters that need to be chosen when working with Decision Trees are the maximum depth of the tree, the size of groups in the leaf nodes, as well as the method for creating the conditions in the internal decision nodes. The depth of the tree and the size of the groups in the terminal nodes control to a degree the possibility of over-fitting of the Decision Tree. A tree with a smaller depth contains less decision nodes, and a tree with a minimum number of observations ending up in a leaf node refrains from spreading out the observations until only one is left in each leaf node. Both encourage the division of data into larger groups, resulting in a higher degree of generalization of the predictions of the classifier. In the case of an over-fitting model, it near-perfectly predicts the observations in the training set containing the labels, but has a much worse performance on new observations. The method for creating the decision rules in the decision nodes is important to have an increased information gain in each decision node. Most versions of Decision Trees build the tree in a greedy fashion, making the next decision based on the outcome so far. This can be suboptimal if there is a more complex structure in the data that is missed by the order in which the decisions are formed. Pedregosa et al. (2011)

2.1.1 Random Forest

Random Forest (RF) is an ensemble method based on Decision Trees. The idea is to combine multiple different Decision Trees into a single classifier. The advantage of consulting multiple models is to reduce the variance by having a more broadly supported labelling decision for new observations. The ensemble reduces the individual errors of a Decision Tree by enriching the final decision with additional information gained from other Decision Trees. An important feature of the Decision Trees included in the ensemble is that they should have a low correlation in their predictions, or else the advantage of having multiple base classifiers is lost. If all the classifiers in the ensemble make the exact same predictions, there is no generalization of the ensemble prediction, so the diversity of the classifiers in the ensemble is an important attribute. Random Forest achieves this by assigning each of the underlying Decision Trees a sub-sample of available features in the data. With this, the different classifiers can pick up different patterns in the data that would be lost for others. Yiu (2019)
A disadvantage of Random Forest is that the method becomes a black box method as the final decision can't be followed by a human user as this would necessitate following and interpreting each individual Decision Tree in the ensemble to understand the classification of a new observation and forming the ensemble vote from the individual predictions. The important parameters that need to be chosen for a Random Forest classifier are the number of estimators to be aggregated into the ensemble, as well as the calibration of the underlying Decision Tree classifiers. Pedregosa et al. (2011)

2.2 Grammatical Evolution

Grammatical Evolution (GE) has been around since the late 1990s and is part of the evolutionary algorithms. Starting from a randomly generated population, the goal is to evolve it to a point, where the individuals adapt to the objective given to the method. The method provides a clear distinction between the genotype and the phenotype of an individual. The genotype represents the underlying genetic structure and is often implemented using an array of numbers. The phenotype describes characteristics translated from the underlying genotype according to the mapping process specified in the grammar, hence the name Grammatical Evolution. The evaluation of the phenotype is performed using a fitness function to facilitate comparisons of individuals. In a classification context, this is often the maximization of the predictive power of the model. However, the method itself can be adapted to solve many applications, with classification being one of them. C. Ryan, M. O'Neill, and J. Collins (2018)
The grammar is the essential part driving the flexibility of the method, as this part defines the applications the method can be used in. The genotype is translated to the characteristics of the individual through a context-free grammar. The grammar contains the production rules of the phenotype according to the underlying genetic material. The context-free grammar is often represented using the Backus-Naur Form. It represents the tuple (N, T, P, S), where N is the set of non-terminal symbols, which are in turn comprised of terminal symbols contained in set T. The mapping process is done through the set of creation rules P. The symbol S represents the starting symbol, from which all derivation sequences begin. C. Ryan, M. O'Neill, and J. Collins (2018)
An example grammar is presented in Figure 1, showing the creation rules P to create the decision nodes to form a Decision Tree. The starting symbol S of this grammar is <expr>. The terminal symbols are <digit>, <attr> and <label>. These rules decide which digit, attribute, or label is chosen when their respective production rule is reached in the mapping process. All other symbols belong to the non-terminals.
The grammar results in a sequence of decision rules and corresponding actions depending on the outcome to build a binary Decision Tree. With the constant being bounded

```
<expr> ::=  <label> if <cond> else <label>                |
            <label> if <cond> else <expr>                 |
            <label> if <cond> and <cond> else <expr>      |
            <label> if <cond> or <cond> else <expr>       |
            <expr> if <cond> else <expr>                  |
            <expr> | <label>

<label> ::= 0 | ... | g

<cond> :: = <attr> <= <const>    |
            <attr> <= <attr>

<const> ::= 0.<digit> | −0.<digit>

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<attr> ::= x[0] | x[1] | x[2] | ... | x[k]
```

Figure 1: *Context-free grammar for a multi-class classification problem with g classes and k attributes following the grammar presented in Fitzgerald, Azad, and Conor Ryan (2017). The labels and attributes can be chosen according to the underlying data.*

between -0.9 and 0.9, this grammar is ideally used on standardized data to facilitate the division of the data into sub-samples.
The array of numbers in the genotype is translated using modulo operations, so that the method can determine which production rule should be used. When the translation process terminates with a terminal symbol, there may be left over genetic material that is not needed in the translation process. The part used in mapping is called the active part of the genotype. The non-controlled translation process implies that the given genetic material is either sufficient to translate it to the phenotype of the individual or not. In such a case, there is the option to wrap around the array and continue at the start of the genetic material until completion or until a given maximum number of wrapping is reached, in case the translation process gets stuck in an infinite loop. If the translation fails, it is common to assign a poor fitness to the individual after the translation from the genotype to the phenotype. The performance of the phenotype of the individual is then compared to the phenotype of other individuals in the population by assigning a fitness value through a fitness function. C. Ryan, M. O'Neill, and J. Collins (2018)
The creation of the next generation involves a selection, cross-over and mutation step. In the selection step, individuals are chosen to continue in the evolutionary process. In tournament selection, a number of individuals is chosen with replacement from the population, and the fittest of them enter the next steps. In roulette-wheel selection, the individuals are assigned a probability of being chosen proportional to their fitness value. This allows for the selection of individuals with a lower fitness.
In the cross-over step, the individuals pass on part of their own genotype to the next generation by mixing it with the genotype of other individuals. A number of parents is selected to serve as donors of genetic material for their children. With a given cross-over probability, the parent's genotype is cut into pieces and recombined into new ones. With the complementary probability, the parents enter the new generation. The points for the cuts in the genetic material can be chosen completely randomly or semi-randomly through restriction to the active parts of the genome. The latter improves the chances of

beneficial cross-over as it excludes the possibility of the children from cross-over having the same phenotypes as their parents when the cross-over happens in the inactive parts of the genotypes and therefore leads to no changes when translating to the phenotype. Conor Ryan, J. J. Collins, and Michael O'Neill (1998)
The mutation step introduces additional randomness to the evolutionary process. With a given mutation probability, changes to the genotype of an individual are made. This should result in changes to the phenotype and allows breaking the dominance of a certain genotype with a suboptimal outcome. This is useful in the presence of an at the time, unknown better solution. The cross-over and mutation steps should strike a balance, as only cross-over may get the evolutionary process stuck in a local optimum, whereas only mutation does not give a direction to the evolutionary process. C. Ryan, M. O'Neill, and J. Collins (2018)
There are many parameters that can be adjusted in applications of Grammatical Evolution. The number of individuals and the number of generations until the halting of the evolutionary process are basic parameters which should strike a balance. The smaller the number of individuals, the lower the diversity in the parents used in cross-over and therefore the diversity in the next generations. A too low number of generations restricts the process in the search space it is allowed to reach. Two other important parameters are the cross-over probability and the mutation probability. A high value for the mutation probability leads to volatile individuals and a slow improvement of the population, as mutation can undo the progress done through cross-over. The cross-over probability drives the strength of the movement towards improvement. C. Ryan, M. O'Neill, and J. Collins (2018)

2.2.1 Grammatical Evolution Decision Trees

Grammatical Evolution has been adopted to the creation of Decision Trees to detect gene-gene interactions in Deodhar and Motsinger-Reif (2010). This leads to so-called Grammatical Evolution Decision Trees (GEDT). The idea behind them is to use the stochastic but guided nature of the creation process of the trees to find patterns that may not be found by greedy algorithms used in conventional Decision Trees. Depending on the grammar, there is also the possibility to include more complex conditions in the decision nodes, notably interactions between the attributes or higher powers of them. Deodhar and Motsinger-Reif (2010)
An example tree derived from the grammar in Figure 1 is shown in Figure 2. This tree shows the flexibility of the method as well as its shortcomings. This Grammatical Evolution Decision Tree does not consider the number of observations left after each split. There is also the possibility that the assignment of a class label does not differ depending on whether the condition in the node is true or not. This can be seen in the deepest leaf nodes of the tree, where the observations are assigned label 1 no matter the outcome of the above decision node. There could also be the case where the condition leads to a split in the observations with one outcome containing all observations entering the decision, leading to a higher depth of the tree without contributing towards a higher predictive power.
The creation of constants in the decision nodes of the resulting Decision Tree can be a difficult part to improve. As with the general nature of Grammatical Evolution, constants are created during the translation of the genotype. This method to create constants is called digit concatenation, as the constant is created by simply concatenating digits when mapping the genotype to the phenotype. This method of constant creation uses parts of the genotype and is sensitive to its structure. Cross-over and mutation influence the mapping process. Therefore, digit concatenation is a volatile method. Other methods have been proposed, however their implementation has proven

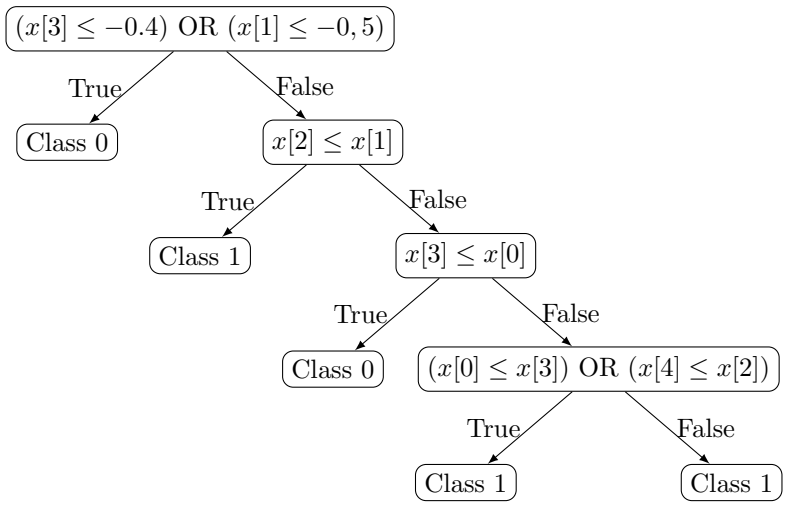

Figure 2: *Example Decision Tree derived from the above grammar for a multi-class classification problem with two classes and five attributes.*

difficult and has not achieved a considerate improvement to the method. The variability in the translation process has been shown to be a strength of Grammatical Evolution. C. Ryan, M. O'Neill, and J. Collins (2018)

Another integral part of the workings of Grammatical Evolution is the fitness function used to assess the performance of the phenotype of an individual. A popular measure in the context of classification is the accuracy of the predictions. The accuracy represents the percentage of predictions that were correctly classified. This measure weighs all classes equally, with every false prediction being valued the same. In the case of more than two possible classes, this can be problematic, as there may be differences in the severity of a wrong classification. For applications where the severity of a miss-classification is unequal for the different classes, measures such as the precision or the sensitivity are better suited.

The calibration of the evolutionary parameters leaves much design freedom. There has been some study into the optimal choice of parameters for Grammatical Evolution Decision Trees. Hoover et al. (2011) presented a parameter sweep for the principal evolutionary parameters for the evolutionary process used in Deodhar and Motsinger-Reif (2010). The authors surveyed the effect of different calibrations of the model done on a similar synthetic data set as in the latter paper on gene-gene interactions.

Their results showed significant improvement in predictive power when choosing tournament selection over roulette-wheel selection. For the mutation rate, a value of 5% lead to better results than a mutation rate of 1%. There was no significant difference in the predictive power between cross-over probabilities of 80% and 90%. The result that an increase in both population size and number of generations improves the predictive power is to be expected, as both parameters increase the search space for a possible best solution. According to the results in the paper, there seems to be a decrease in the additional predictive power gained with an increase in the population size and the number of generations. Hoover et al. (2011)

2.2.2 Ensemble Methods for Grammatical Evolution Decision Trees

The use of Grammatical Evolution to create Decision Trees opens the avenue to adopt the workings of Random Forest to the resulting Decision Trees created by Grammatical Evolution.
There has been previous work done on the creation of ensembles for Grammatical Evolution Decision Trees in Fitzgerald, Azad, and Conor Ryan (2017). The authors proposed a framework for multi-class classification and clustering. The authors conducted their experiments on synthetic data to showcase their framework. Their results were based on fifty runs of the evolutionary process for the different data sets. To improve the performance on more demanding four and five class classification tasks, they proposed the creation of an ensemble classifier. The predictions of the ensembles were based on a simple majority vote. Fitzgerald, Azad, and Conor Ryan (2017)
The paper investigated two approaches to forming the ensembles used in classification. The first version picked the best performing Grammatical Evolution Decision Trees from the fifty runs and used them in the creation of an ensemble. They called this version the best-ensemble EnsBest. The second version was restricted to the Grammatical Evolution Decision Trees within each run. The ones that exceeded a given threshold for the training accuracy were included in the ensemble. The authors called this second version the population-ensemble EnsPop. They noted that setting a fixed threshold of 60% yielded discouraging results, so the authors switched the criterion to be set relative to the average training accuracy across all runs on a synthetic data set. Fitzgerald, Azad, and Conor Ryan (2017)
After ensemble creation, the correlation in the predictions of the included classifiers was inspected and Grammatical Evolution Decision Trees that had a correlation coefficient higher than 90% with more than two thirds of the other Trees were removed. This step was done to improve the diversity of the classifiers in the ensemble to improve classifier generalization. Fitzgerald, Azad, and Conor Ryan (2017)
The two ensemble methods improved on the test accuracy when compared to the whole population of Grammatical Evolution Decision Trees that were available after evolution. The authors noted the possibility to improve on the creation of ensembles from the whole population by determining which individuals perform best on the training data and assigning a higher weight to the predictions on the test data, as well as a more refined inclusion criterion for the population ensemble. Fitzgerald, Azad, and Conor Ryan (2017)

2.3 Diversity measures for ensembles

In Fitzgerald, Azad, and Conor Ryan (2017), the correlation in the predictions of the classifiers was used as a criterion for inclusion in the ensemble. The correlation, however, can be problematic as there exists the possibility of having a classifier in the ensemble which makes the same prediction for every observation in the training set, resulting in a division by zero when using the Pearson correlation coefficient.
Grasping the concept of diversity is difficult, and so is constructing a measure for it. The Q-statistic was proposed in Yule (1900). It is a pairwise diversity-measure comparing two classifiers. This measure does not suffer from the problem described for the correlation coefficient.
This Q-statistic uses the numbers of coinciding correct and false classifications and the numbers of unilateral correct and false classifications as shown in Table 1. The Q-statistic is defined as

$$Q_{i,k} = \frac{N_{11}N_{00} - N_{01}N_{10}}{N_{11}N_{00} + N_{01}N_{10}}$$

This definition leads to values for the Q-statistic bounded in [-1, 1]. A positive value of the Q-statistic signifies that two classifiers give many coinciding predictions. A negative value of the Q-statistic signifies that two classifiers make errors for different observations. A value of zero is equivalent to two statistically independent classifiers. In the case of an ensemble of many classifiers, the average Q-statistic for all pairs of classifiers is a measure of the diversity in the ensemble, with a lower value corresponding to a higher value of diversity. Kuncheva and Whitaker (2003)

Table 1: *Confusion matrix for two classifiers*

	C_i is correct	C_i is wrong
C_k is correct	N_{00}	N_{01}
C_k is wrong	N_{10}	N_{11}

Simple confusion matrix of correct classifications for two classifiers C_i and C_k on the basis of Table 1 found in Kuncheva and Whitaker (2003). The coinciding indices represent cases in which both classifiers make the same correct or wrong decision. The differing indices show cases in which one classifier makes a correct and the the other a wrong prediction.

3 Experiments

This chapter goes over the specifics of the experiments conducted for this thesis. The first section provides an overview of the data sets used in the experiments. The second section introduces the calibration for the creation of the candidate Grammatical Evolution Decision Trees and their aggregation into ensembles. The last section explains the calibration of the methods used to compare the created ensembles to more established approaches.
The experiments were conducted using Python as the programming language of choice. The Scikit-learn library was used for the creation of individual Decision Tree classifiers and Random Forest ensembles (Pedregosa et al. (2011)). The Grammatical Evolution side of the experiments was carried out using the PonyGE2 library (Fenton et al. (2017)).

3.1 Data sets

The experiments in this section were performed on a selection of real world data sets containing numerical features accessed on the UCI Machine Learning Repository (Dua and Graff (2017)). This facilitates the replication of the results and reduces the issue of data mining of artificial data sets that deliver overly promising results.
The selected data sets cover multi-class problems from two to five classes. Observations with missing values in any feature were removed. The data sets were standardized to facilitate generalization of the grammar used for Grammatical Evolution with respect to the creation of constants.
Other than removing observations with missing values, no other pre-processing of the data sets was conducted to serve as a show-case of the out-of-the-box capabilities of the method. The data was randomly split into a training and a test set. The training set contained 70% of observations and the test set contained the remaining 30%.

3.1.1 Banknotes

The Banknote data set contains data on measurements done on images taken of genuine and forged banknotes. The original images had a size of 400 pixels in width and 400 pixels in length. They were processed using the Wavelet transformation tool. This transformation yielded a data set with four features and a total of 1'372 observations. The data set contains 762 observations of one class and 610 observations belonging to the other class. From the description of the data set, it is not discernible which of the two classes represents the genuine banknotes. Lohweg (2013)

3.1.2 Iris

The Iris data set is widely known in the pattern-recognition literature. It contains measurements on three types of iris plants. One class is linearly separable from the other two. The data set contains in total 150 observations and no missing values in any of the four features. The three groups are of equal size. The data has been rounded to two decimal places, leading to the number of unique values in the data set being far below the number of observations. Fisher (1988)

3.1.3 Vehicle

The Vehicle data set contains data extracted from pictures of four types of vehicles. The vehicles portrayed on the images were either a double-decker bus, a Chevrolet van, a Saab 9000 or an Opel Manta 400. The images were processed to yield numeric information resulting in 18 features for 846 observations. The data set contains values for 218 double-decker busses, 199 Chevrolet vans, 217 Saab 9000, and 212 Opel Manta 400. The features contain no missing values. Mowforth and Shepherd (n.d.)

3.1.4 Cleveland

The Cleveland data set contains medical data for 303 observations. The data set used is a subset with 13 features of a larger heart-disease database containing a total of 74 features. The original data set can be found in the UCI repository. The subset stems from the KEEL repository Alcalá-Fdez et al. (2011). The classes refer to the absence of a heart-disease or its severity in the case of a presence. For the experiments, the data set was cleaned of six observations with missing values, resulting in a final size of 297 observations. The class sizes are heavily skewed towards observations without the presence of a heart-disease. This group consists of a total of 160 observations. The other groups were of sizes 54, 35, 35 and 13, with more severe cases of a heart-disease becoming rarer. Janosi et al. (1988)

3.2 GEDT - Population Creation

For the creation of individuals, three different calibrations were used. The first calibration serves as a baseline to compare the other two against. The second and third calibrations use a different approach to increase diversity in the individuals. The second calibration uses smaller runs that are later combined to an equal size total population of individuals in hopes that the evolutionary process does not follow the same patterns in the different runs. These runs are independent of the first calibration and are not comprised of the first halves of these runs. The third calibration uses an additional objective function to promote a higher diversity in the resulting population. The addition of a second objective lead to a much higher time demand for the evolutionary process to terminate and was only applied to the Vehicle and Cleveland data sets. Each calibration was used in the creation of fifty runs. An overview of the calibrations can be found in Table 2.

Table 2: *Overview of the three calibrations used in the experiments*

	Calibration 1	Calibration 2	Calibration 3
Number of Generations	500	250	500
Optimization Process	Single-objective		Multi-objective
Fitness Functions	% of false classifications		% of false classifications Q-statistic
Data sets	Banknote, Iris, Vehicle, Cleveland		Vehicle, Cleveland

Calibrations 1 and 2 use single-objective evolution. Calibration 3 uses an evaluation of individuals based on two objective functions to promote diversity in the individuals of the generations. The third calibration saw an increase in the runtime and was only applied to the more complex data sets

3.2.1 Evolutionary parameters

The grammar was created according to the one shown in Figure 1. Adjustments have been made to accommodate for the evaluation of the resulting phenotypes in Python. This allows for fast generation of label predictions on the training and the test set. The grammar used in the translation from genotype to phenotype for the Banknote data set can be found in Figure 3 in the appendix. For the other data sets, only the production rule for the labels needs to be adjusted. The number of attributes is derived from the data by the PonyGE2 library. The grammar was expanded to allow for sub-expressions in the conditions. This enables shifting and scaling of features and constants, as well as interaction between features. The choice was made to exclude the division operator from the grammar, as its inclusion lead to a large number of individuals with non-valid phenotype due to a division by zero. The exclusion of the division operator also leads to a decrease in the complexity of sub-expressions and a reduction in the search space due to the combinatorial complexity of choosing the denominator and the numerator.
For all calibrations, the individuals were evaluated on their predictions, with the portion of falsely classified observations serving as the fitness measure to be minimized. This is equal to the maximization of the accuracy in the predictions of the individuals. The third calibration uses the Q-statistic as a measure of the diversity of the individuals in each generation, in addition to the portion of falsely classified observations.
The general calibration for the evolution of the population of Decision Trees using Grammatical Evolution follows the findings presented in Hoover et al. (2011). All of the calibrations use the standard search method implemented in PonyGE2. This method iterates through the evolutionary steps for each generation. Starting from an initial generated population, the methods steps through the selection, variation, replacement and evaluation steps, repeating them until the specified number of generations is reached.
During the selection step, a number of tournaments equal to the number of individuals in a generation is conducted. For each tournament, two individuals are randomly drawn with replacement from the underlying population and compared on their fitness value. Only the one with the higher fitness value is chosen to enter the next step in the evolutionary process.
For cross-over, two parent individuals are chosen at random and a variable one-point cross-over is performed. The cross-over is implemented in such a way that the cutting points for the genetic material are chosen randomly on the active parts of the genotypes of the parents, but not necessarily on the same index of the genotype. With the given cross-over probability of 80%, cross-over is conducted, and the genotypes are recombined into new ones. With the complementary probability, the parents enter the new population unchanged.
The evolutionary process then continues with the mutation step where, with the given mutation probability of 5%, a randomly chosen index of the active part of the genotype is exchanged by a randomly created number in the range of the other numbers in the genotype. The change to an index in the active part of the genotype can lead to it changing. There is also the possibility that the resulting phenotype only changes by a different constant, attribute or operator, in a decision node.
After the genotype of the individuals is created, it is translated into the phenotype. The individual is afterwards evaluated using the fitness functions. The choice was made to assign invalid individuals a missing value as their fitness. This leads to the individual losing to a valid individual in the selection step in the next generation.
Lastly, during the replacement step, the best individual from the old population is inserted into the newly generated population and the worst individual is excluded. The step method then starts the next iteration of at the selection step if the total number of generations has not yet been reached.

The PonyGE2 library offers additional parameters that can be adjusted. The individuals in the starting generation were initialized to have a minimum depth of three. This parameter influences the depth of the translation of the grammar and does not represent the depth of the resulting decision tree. No wrapping of genotype was used, and the maximum depth of the translation was set to be fifteen. An individual that runs out of genetic material in the translation process or crosses the maximum depth is flagged as invalid and is assigned a missing value as its fitness. The restriction of the trees to a certain maximum depth is useful for reducing the redundancy in the phenotypes, as they can grow to an immense size without an increase in predictive power, as the process does not consider the decisions made in the nodes. Just as in the example in Figure 2, the tree could have the same decision outcomes for multiple sub-nodes or could grow in size, with the subtree making decisions on an empty set of observations. This would lead to an increased number of trees with the exact same fitness. It is advised to pick a smaller model in the presence of no significant increase in the predictive power of a larger model. In addition, the length of the genotype was limited to 500 indices.

3.2.2 Calibration 1 – Single Objective with 500 generations

The number of evolved individuals for the first experiment follows Fitzgerald, Azad, and Conor Ryan (2017). Keeping the total number of individuals equal, the number of individuals per generation and the number of generations were flipped in contrast to Fitzgerald, Azad, and Conor Ryan (2017). This results in 100 individuals per generation and 500 generations. The choice to work with an increased number of generations and a reduced number of individuals per generation was made to allow for a longer evolutionary process in hopes of obtaining individuals with a higher fitness. The number of evolved individuals is the same.

3.2.3 Calibration 2 – Single Objective with 250 generations

The second experiment used a reduction in the total number of evolved individuals per run by halving the number of generations. This reduction was chosen due to a high similarity in the evolved trees in the 500 generation version and the observation of diminishing improvement of the average population fitness, as well as the fitness of the best individual(s) in a generation. This diminishing return to a higher number of generations was hinted at in Hoover et al. (2011). The high similarity between the evolved individuals leads to a low degree of generalization for the predictions of the ensembles.
To enable a comparison with the first calibration, two runs following the second calibration were combined at random. This leads to the same number of evolved individuals as in the first calibration. Through combination, the resulting total population of individuals should attain a sufficient performance with an increased diversity in the individuals. The merging of runs was done in a fashion such that each run is chosen at least once and that the same runs would not be chosen for combination, as this would defeat the idea of diversifying through the combination of independent runs. There was no control over the choice of the second run, so the possibility of one run being chosen more times than others exists. The evolutionary parameters and the grammar were chosen, analogous to the first experiment.

3.2.4 Calibration 3 – Multi Objective with 500 generations

The third calibration follows the same number of individuals per generation and number of generations as the first calibration. The difference is the inclusion of a second objective function in the evolutionary process.

The Q-statistic was chosen as a measure of diversity within a generation. With this, individuals making differing predictions from others with a worse accuracy are still retained from one generation to the next on the merit of them being different. This should hinder the movement towards many similar individuals and create a total run population that is more diverse. The possible lower accuracy of the individuals should not be problematic, as the creation of an ensemble leads to individuals correcting errors of one another in the aggregated majority vote.
The addition of another objective leads to changes in the replacement as well as in the selection steps. The implementation in the PonyGE2 infrastructure uses the Non-Dominated Sorting Genetic Algorithm II (NSGA II) for the optimization of multi-objective problems. The algorithm was proposed in Deb et al. (2002).
The selection process is adjusted to accommodate for the second objective function. This is done by creating a Pareto-front from the individuals and calculating the crowding distance as a measure of how spread along the Pareto-front the individuals are. The tournaments take place similar to the method used in the single-objective case, with two individuals in the original population being chosen at random and being compared. An individual with a lower non-domination rank (on a better front) is preferred to an individual with a higher non-domination rank (on a worse front). If the two individuals lie on the same front, the one in a less crowded region is chosen as the winner of the tournament. Deb et al. (2002)
The cross-over and mutation steps take place without additional adjustments. The individuals are then evaluated sequentially on the objective functions. The process is implemented in such a way that if one objective function assigns the default fitness value, then the default fitness is used for all objective functions and the individual is flagged as invalid. Deb et al. (2002)
The replacement step aggregates the new and old populations and calculates the Pareto-fronts and crowding distances on the combined population. The next generation is then populated with the best individuals of the aggregated population according to their position in the two-dimensional space. Deb et al. (2002)

3.3 Ensemble Creation

Given the total population of individuals for a run, duplicate trees were removed. This was done comparing the expressions obtained as phenotypes. This process is stricter than exclusion based on identical genotypes, as different genotypes can lead to the same phenotype expression.
The ensemble creation follows the two strategies proposed in Fitzgerald, Azad, and Conor Ryan (2017). The first strategy was the creation of an ensemble from the best individuals of the fifty runs conducted on the various data sets. As there was the possibility to have multiple individuals with the same training accuracy within a run, the decision was made to replicate the ensemble creation fifty times. For the runs with several equally performing individuals, one of them was chosen at random and added to the ensemble. Runs that resulted in only one best individual always contributed the same individual to the ensemble. The resulting ensembles were named best-ensemble (EnsBest) analogously to Fitzgerald, Azad, and Conor Ryan (2017).
The second strategy was restricted to the population of a run for ensemble creation, called the population-ensemble (EnsPop). For the population-ensemble approach, a restriction to ensembles of a fixed initial size was chosen. To keep the ensembles small, initial sizes were chosen to range from 50 individuals to 500 individuals. The smallest population-ensemble could be compared to the best-ensemble with respect to the performance of fifty individuals. A relative threshold for entry into the ensemble, as in Fitzgerald, Azad, and Conor Ryan (2017), lead to very bloated and under-performing

ensembles due to an underlying population of similar trees using the first calibration for testing. The high ensemble size was in part due to the high variability of fitness values due to the mutation process influencing the average fitness in the last generation. A lower average fitness in the last generation leads to a high number of individuals retroactively included in the ensemble.
The population-ensembles were created by sorting the total population of individuals by their performance and taking the best individuals according to the initial ensemble size. For the mixed runs with a lower number of generations, the individuals were shuffled in a way, guaranteeing that half of the ensemble consists of the best individuals of one run and the other half of the best individuals of the second run.
The Q-statistic was chosen to serve as a measure of diversity of the ensemble and also as a threshold for trimming too similar individuals from the ensemble. For the exclusion in the ensemble based on the diversity of the individuals, five cut-offs were chosen to inspect possible changes in the ensemble vote. If the average Q-statistic of an individual with every other individual in the ensemble was higher than the cut-off, the votes of the individual were not considered in the prediction of the ensemble. The Q-statistic was calculated based on the initial candidate individuals and not recalculated after the exclusion of the most similar individuals. The cut-offs were set to values -0.25, 0.0, 0.25, 0.5 and 0.75. The lower the threshold, the stricter the restriction on the diversity of the candidate individuals. If the restriction would eliminate all individuals from the ensemble, a random individual with the highest training accuracy was kept. In such a case, there would be no difference between the ensemble vote and the vote of a single randomly chosen individual with the highest training accuracy. The choice of the individual was seeded for reproduction of the ensembles.
The evaluation of the ensembles was conducted using two variants of a majority vote: a weighted majority vote and an unweighted majority vote. For the unweighted majority vote, each individual entered the vote for the ensemble prediction with an equal weight. For the weighted majority vote, each individual in the ensemble was assigned a weight equal to its contribution to the overall number of correct classifications on the training set. The vote of the ensemble was then based on the class that gained the most weights. This method favors the predictions of individuals with a higher training accuracy. The same weights from the training set were reused for the creation of the ensemble vote on the test set.

3.4 Comparison with other classification methods

To compare the ensemble creation with Grammatical Evolution, a Random Forest ensemble is created. Most of the default values of the implementation in the Scikit-learn library were used. A change was made to the number of classifiers that were trained. This number was changed to be fifty to facilitate comparison with the best-ensemble approach. The individual trees considered a sub-sample of features equal to $\sqrt{n_{features}}$ in the data sets. For creating the splits in the decision nodes, the Gini-coefficient was used. The trees were given a maximum depth of 10. The method uses a bootstrapped sub-sample of observations for fitting each of the individual trees.
Another comparison was done with the use of conventional Decision Trees. This was done to assess the quality of the trees generated by Grammatical Evolution. To create the Decision Trees, the implementation in the Scikit-learn library was used. The parameters were chosen equal to the individuals in the Random Forest classifier. The Random Forest and Decision Tree classifiers were seeded, resulting in the same predictions and values for all calibrations.

4 Results

This chapter presents the results from the experiments described in the previous chapter. The results are grouped according to the data sets. The sections report the median test accuracy for the classification methods. The same tables for the median training accuracy can be found in the corresponding sections in the appendix. The appendix also holds plots showing the test and training accuracy of the different combinations between initial ensemble size and cut-off for the population-ensembles.

4.1 Banknote data set

Table 3: *Comparison of the median test accuracy for the Banknote data set*

	EnsBest	EnsBest weighted	EnsPop	EnsPop weighted	GEDT	RF	DT
Calibration 1	99.7 (0.0)	99.7 (0.0)	97.3 (2.8)	97.3 (2.8)	99.2 (3.5)	99.6 (0.3)	97.3 (0.0)
Calibration 2	99.7 (0.0)	99.7 (0.0)	96.2 (3.4)	96.2 (3.4)	96.0 (1.8)		

The five columns on the left show the median test accuracy for the Grammatical Evolution approaches. The last two columns show the performance of the comparison methods. The top numbers represent the median test accuracy in percent. The numbers in parentheses stand for the standard deviation of the accuracy.

The highest median accuracy on the test set was achieved by the best-ensembles (EnsBest) with a median test accuracy of 99.7%, which beats Random Forest (RF) by 0.1%. Between the weighted and unweighted versions, no difference in the median could be observed. There was also no difference between the first and second calibration. Additionally, there was no spread around this value for the best-ensembles, as shown in Table 3.
The performance of the population-ensembles (EnsPop) is lower, with a median accuracy of 97.3% for the calibration using 500 generations and a value of 96.2% for the smaller calibration. There were no differences between the median values for the test accuracy between the population-ensembles.
A similar difference between the calibrations can be made for the best individuals of the runs (GEDT). The individual Grammatical Evolution Decision Trees from the first calibration achieved a test accuracy of 99.2% and outperformed the median test accuracy of the best individuals of the smaller calibration, which achieved a value of only 96.0%. This is the lowest median test accuracy of all methods.
The median test accuracy of the population-ensemble of the second calibration is higher than the one for the best individuals of the runs. This is not the case for the first calibration. The median test accuracy of the best trees of the first calibration is higher than the one from the conventional Decision Trees (DT). The spread in the accuracy is high, so there is some overlap between the achieved test accuracy, as shown in Figures 4 and 7 in the appendix.
As expected from the observation of a high number of similar individuals in the case of the runs with 500 generations, there were no large changes between the ensembles of fixed initial size with different cut-offs for exclusion from the ensembles. All the population-ensembles for this calibration achieved a median test accuracy of 97.3% except for the ones with initial ensembles sizes 250 and 500 and the loosest cut-off of 0.75. Those two combinations achieved slightly lower median test accuracies of 97.2% and 97.0%.

The high similarity in the performance of the first calibration can be explained in part by the fact that all individuals of the ensemble would be flagged as too similar and removed from the ensemble, so that only a randomly chosen tree remains for the evaluation of the test set. This is also the reason that there was no difference between the weighted and unweighted versions of the population-ensembles.
For the second calibration, there were more ensembles that consisted of more than a single individual. There were also more differences in the median test accuracy of the different combinations, as well as between the weighted and unweighted ensembles.
The unweighted ensembles with an initial ensemble size of 50 individuals achieved a lower median test accuracy for all cut-offs than the ensembles with an initial size of 100 with the exception for the cut-off value of 0.25 where the ensembles were tied in their median test accuracy. For the other combinations, there was no initial ensemble size that achieved a better median test accuracy for all cut-offs. Looking at the cut-offs, there was no value that had the highest median test accuracy for all initial ensemble sizes. However, the ensembles with a cut-off of 0.0 achieved the highest median training accuracy for the four largest initial ensemble sizes. The smallest cut-off -0.25 achieved this value for the largest ensembles, the cut-off 0.5 for the two smallest ensembles and the highest cut-off 0.75 for the initial ensemble size of 100 individuals.
The weighted combinations had the same median accuracy values as the ones of the unweighted ones except for the combinations with ensemble size 50 and cut-offs 0.0, 0.25 and 0.75 as well as the ensemble with an initial size of 100 and cut-off 0.75. These combinations achieved slightly higher median test accuracies. This difference is lost to rounding in Table 3.

4.2 Iris data set

Table 4: *Comparison of the median test accuracy for the Iris data set*

	EnsBest	EnsBest weighted	EnsPop	EnsPop weighted	GEDT	RF	DT
Calibration 1	95.6 (0.0)	95.6 (0.0)	95.6 (3.3)	95.6 (3.4)	95.6 (2.0)	95.6 (0.0)	95.6 (0.0)
Calibration 2	95.6 (0.0)	95.6 (0.0)	95.6 (3.8)	95.6 (3.7)	95.6 (2.5)		

The five columns on the left show the median test accuracy for the Grammatical Evolution approaches. The last two columns show the performance of the comparison methods. The top numbers represent the median test accuracy in percent. The numbers in parentheses stand for the standard deviation of the accuracy.

For the Iris data set, the median test accuracy is the same for all methods and calibrations used. They achieved a value of 95.6% correctly classified observations, which can be seen in Table 4. Based on the median test accuracy, there is no method that outperforms the others. The best-ensemble, Random Forest and conventional Decision Tree approaches had no spread in the accuracy. There is spread in the values for the population-ensembles and the best Grammatical Evolution Decision Trees of both calibrations.
For the first calibration, the weighted population-ensembles had a higher spread in the test accuracy than the unweighted population-ensembles. The opposite can be observed for the population-ensembles of the second calibration. The second calibration had a higher overall spread in the accuracy of the population-ensembles than the first calibration.

The median test accuracy for all combinations of cut-off and initial ensemble size is the same, with a value of 95.6%. This is the case for the weighted and unweighted combinations for the first calibration. For the second calibration the combination with a cut-off of 0.75 and an initial ensemble size of 100, the unweighted and weighted versions achieved lower median test accuracies of 94.4% and 93.3% as shown in Figures 14 and 15 in the appendix. The other combinations all achieved a median test accuracy of 95.6%.

4.3 Vehicle data set

Table 5: *Comparison of the median test accuracy for the Vehicle data set*

	EnsBest	EnsBest weighted	EnsPop	EnsPop weighted	GEDT	RF	DT
Calibration 1	68.1 (0.2)	68.1 (0.2)	53.9 (6.4)	53.9 (6.4)	48.0 (5.2)	74.4 (1.2)	68.3 (0.9)
Calibration 2	68.9 (0.2)	68.9 (0.2)	54.7 (5.2)	54.7 (5.2)	51.2 (4.4)		
Calibration 3	67.7 (0.5)	67.3 (0.5)	48.0 (9.6)	48.6 (9.3)	55.1 (6.2)		

The five columns on the left show the median test accuracy for the Grammatical Evolution approaches. The last two columns show the performance of the comparison methods. The top numbers represent the median test accuracy in percent. The numbers in parentheses stand for the standard deviation of the accuracy.

The highest median test accuracy was achieved by Random Forest, with a value of 74.4%. Conventional Decision Trees had the third-highest test accuracy of 68.5% being beaten by the best-ensembles of the second calibration with a value of 68.9%. Within the Grammatical Evolution Decision Tree methods, the best-ensembles achieved the highest median test accuracy for all calibrations, as shown in Table 5.
The 500 generation calibration had a value of 68.1% and the multi-objective calibration a value of 67.7%. The population-ensembles had a comparably low value, with the highest median test accuracy being achieved by the 250 generation calibration, with a value of 54.7%. The 500 generation calibration has a lower median test accuracy of 53.9%. Regarding the first two calibrations, there were no differences between the weighted and unweighted versions of the ensembles
The multi-objective calibration achieved considerably lower test accuracies, with values of 48.0% for the unweighted version and 48.6% for the weighted version. Also, the weighted version of the population-ensembles had a lower spread in the test accuracy than the unweighted version. The individual GEDT had differing values, with the third calibration having the highest median of 55.1%, followed by the second and first calibrations with values 52.2% and 48.0% each.
The lower values for the performance of the individual GEDT can partly be explained by the large number of features compared to the other data sets. This leads to a large search space for the creation of conditions in the decision nodes. For such a situation, the more informed decision-making of conventional Decision Trees and Random Forest presents a clear advantage. The performance of the Grammatical Evolution approaches could be improved by reducing the search space in features, through dimensionality reduction, before running the evolutionary process.
Additionally, it is interesting to observe that the individual Grammatical Evolution decision trees used in the creation of the best-ensemble achieved a higher test accuracy when using the multi-objective calibration. This can be due to the changes in the

individuals that are kept from one generation to the other, as well as the change in the selection of the parents.
It is also surprising how much better the best-ensembles performed when comparing to the best individual Grammatical Evolution Decision Trees, even though the former are comprised of the latter. This could be an indicator that the individual Trees struggle to find the global optimum and get stuck in local optima during the evolutionary process. However, they attain a high degree of generalization through the aggregation into ensembles.
For the first calibration, there was a difference in the median test accuracy between the weighted and unweighted ensemble votes for the combination of initial ensemble size 250 and cut-off 0.25. For all others, the two versions yielded the same values. The highest value of 54.1% was achieved by the combination of 200 and 250 individuals, with a cut-off of 0.25. The lowest value was a median test accuracy of 53.0% by the ensembles with an initial ensemble size of 200 individuals and a cut-off of 0.75.
The second calibration has many more differences in the median test accuracy between the weighted and unweighted versions of the combinations. There was no change for the ensembles using a cut-off of 0.0. For the others, at least one combination was different. For the cut-off value -0.25, the largest ensemble with an initial size of 500 individuals saw an increase from 53.7% to 53.9%. An increase of the same magnitude is present for the smallest ensemble, with a cut-off of 0.25. For the higher cut-off values 0.5 and 0.75, there were multiple differences for the various ensemble sizes. For the lower cut-off value 0.5, there was an increase in the median test accuracy of the four larger ensembles. For the cut-off value 0.75, there was an increase for the smallest ensemble and a decrease for the largest four initial ensemble sizes.
For the multi-objective calibration, only six of the twenty-five combination showed no difference in the median test accuracy between the weighted and unweighted versions. The highest value of 52.8% was achieved by the largest weighted ensemble with the loosest cut-off. The same ensemble without weighting had a median test accuracy of 50.4%. The lowest value of 26.0% was achieved by the largest initial ensemble size and the strictest cut-off. There was no difference between the weighted and unweighted versions.
The rather low values for the median test accuracy for the strictest cut-offs of the ensembles with a larger initial size can be explained by the presence of very diverse individuals with a low test accuracy. In such a case, the ensemble consists of individuals that make mistakes on different observations but do not reach an acceptable predictive power. With the addition of more individuals in the ensemble, the low performance of the diverse individuals is in part compensated by the weighting of the predictions by the training accuracy of the individuals.

4.4 Cleveland data set

For the Cleveland set, the otherwise comparably performing conventional Decision Trees achieved the lowest median test accuracy of 46.1%, as shown in Table 6. This time, there is also some spread around this value. The highest median test accuracy among the Grammatical Evolution approaches of 62.2% was achieved by the best-ensembles of all calibrations, independent of weighting. The unweighted version of the first calibration and the weighted version of the second calibration achieved the lowest spread in values, whereas the third calibration had a much higher spread. Random Forest attained a median test accuracy of 58.9% with an even higher spread than the best-ensembles.
The median test accuracy for the population-ensembles and the individual trees was lower. The highest median test accuracy was achieved by the population-ensemble which and the individual trees of the smaller calibration with a value of 60%. There

Table 6: *Comparison of the median test accuracy on the Cleveland data set*

	EnsBest	EnsBest weighted	EnsPop	EnsPop weighted	GEDT	RF	DT
Calibration 1	62.2 (0.2)	62.2 (0.3)	58.9 (5.3)	58.9 (5.3)	57.8 (3.9)	59.3 (1.9)	46.2 (1.5)
Calibration 2	62.2 (0.7)	62.2 (0.2)	60.0 (3.6)	60.0 (3.6)	60.0 (3.4)		
Calibration 3	62.2 (1.0)	62.2 (1.2)	56.7 (18.4)	58.9 (17.3)	58.9 (3.5)		

The five columns on the left show the median test accuracy for the Grammatical Evolution approaches. The last two columns show the performance of the comparison methods. The top numbers represent the median test accuracy in percent. The numbers in parentheses stand for the standard deviation of the accuracy.

was no difference between the median test accuracy of the weighted and the unweighted version. The individual trees of the third calibration had a lower median test accuracy with a value of 58.9% followed by the first calibration with a value of 57.8%. The first calibration yielded a median test accuracy of the population-ensembles of 58.9% each. There was a difference in the median test accuracy of the population-ensembles for the multi-objective calibration. The weighted version achieved a value of 58.9%, which is higher than the 56.7% of the unweighted version. Interestingly, there were several population-ensembles of the third calibration having a higher test accuracy than training accuracy. This is not the case for the population-ensembles of the other calibrations.
Between the population-ensembles, there were no changes in the weighted and unweighted versions for the first calibration. The highest value of 58.9% was achieved by multiple combinations, and the lowest value of 57.8% by the strictest cut-off and an initial size of 100. The ensembles with cut-offs 0.25 and 0.5 had the highest median test accuracy for four of the five ensemble sizes.
For the second calibration, there were no differences in the median test accuracy between the weighted and unweighted versions. The highest value of 61.1% was attained by the combinations with initial size 250 and cut-offs -0.25 and 0.5. The lowest value of 60.0% was achieved by many combinations.
For the multi-objective calibration, the weighted ensembles were at least as good as the unweighted versions for all but one combination. For the strictest cut-off and initial size 200 there was a decrease in the median test accuracy. The lowest median test accuracy of 15.6% was observed for the two largest unweighted ensembles with the strictest cut-offs and third and fourth-largest weighted ensembles. The highest value of 58.9% was seen for multiple weighted and unweighted ensembles.
The difference in the performance of the Grammatical Evolution Decision Trees and conventional Decision Trees can in part be explained by the difference in group sizes for this data set. This leads to problems in the decision-making process of the conventional Decision Trees. As Grammatical Evolution Decision Trees do not use guided decision-making, the performance of the individual trees and ensembles exceed the test accuracy of the conventional Decision Trees. The best-ensemble approaches for all calibrations do outperform Random Forest.

5 Conclusion

In this thesis, different approaches to creating ensemble classifiers using Grammatical Evolution Decision Trees were applied to four real-world data sets. Two calibrations were added to the baseline calibration used in Fitzgerald, Azad, and Conor Ryan (2017) to improve the diversity in the classifiers. The methods were compared to conventional Decision Trees and their ensemble method Random Forest. The Grammatical Evolution methods showed promise for the Cleveland data set with its classes of unequal sizes. For the other data sets, the performance of Random Forest was either comparable or better. The best ensembles performed comparable or better than the other methods for the Banknote, Iris and Cleveland data sets. For the Vehicle data set the approaches using Grammatical Evolution did not achieve a satisfying test accuracy. The population ensembles performed worse on the Banknote, Vehicle, and Cleveland data sets. One caveat to the performance of the best ensemble approach is the computational cost attached to it. With the best ensemble of the smaller calibration being as good or better than the ones from the other calibrations, a decrease in the computational demand is attainable. There was no advantage found in including a second objective function in the evolutionary process. This fact is worsened by the much higher computational demand. Focusing on the population ensembles, there were in most cases only slight changes in the performance between the cut-offs for exclusion from the ensembles and no big changes between the initial sizes of the ensembles. There was no clear configuration of cut-off and initial ensemble size that showed a persistent advantage over others. The same can be concluded for the inclusion of a weighted majority vote in the form it was implemented in this work. There is also the possibility that the differences between the cut-offs are driven by the randomness in the selection process of the remaining individuals.
Further research could be conducted into the performance of the best ensemble approach when performing classification on data sets with unequal group sizes. Additionally, a comparison with other classification methods could be of interest.

References

Alcalá-Fdez, J., A. Fernandez, J. Luengo, J. Derrac, L S. García, Sánchez, and F. Herrera. (2011). *KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework.* URL: `https://sci2s.ugr.es/keel/datasets.php`.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6(2), pp. 182–197.

Deodhar, Sushamna and Alison Motsinger-Reif (Mar. 2010). "Grammatical Evolution Decision Trees for Detecting Gene-Gene Interactions". In: vol. 6023, pp. 98–109.

Dua, Dheeru and Casey Graff (2017). *UCI Machine Learning Repository.* URL: `http://archive.ics.uci.edu/ml`.

Fenton, M., J. McDermott, D. Fagan, S. Forstenlechner, E. Hemberg, and M. O'Neill (2017). *PonyGE2: Grammatical Evolution in Python.* Version 0.2.0. URL: `https://github.com/PonyGE/PonyGE2`.

Fisher, R.A. (1988). *Iris.* UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Fitzgerald, Jeannie M., R. Muhammad Atif Azad, and Conor Ryan (2017). "GEML: A Grammatical Evolution, Machine Learning Approach to Multi-class Classification". In: *Computational Intelligence.* Ed. by Juan Julián Merelo, Agostinho Rosa, José M. Cadenas, António Dourado Correia, Kurosh Madani, António Ruano, and Joaquim Filipe. Springer International Publishing: Cham, pp. 113–134.

Hoover, Kristopher, Rachel Marceau, Tyndall Harris, Nicholas Hardison, David Reif, and Alison Motsinger-Reif (2011). "Optimization of Grammatical Evolution Decision Trees". In: *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation.* GECCO '11. Association for Computing Machinery: Dublin, Ireland, pp. 35–36.

Janosi, Andras, William Steinbrunn, Pfisterer Matthias, and Robert Detrano (1988). *Heart Disease.* UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Kuncheva, L.I. and C.J. Whitaker (2003). "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy". In: *Machine Learning* 51, pp. 181–207.

Lohweg, Volker (2013). *banknote authentication.* UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Mowforth, Pete and Barry Shepherd (n.d.). *Statlog (Vehicle Silhouettes).* UCI Machine Learning Repository. URL: `http://archive.ics.uci.edu/ml`.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Rokach, Lior and Oded Maimon (2014). *Data Mining with Decision Trees.* 2nd. WORLD SCIENTIFIC.

Ryan, C., M. O'Neill, and JJ Collins (2018). *Handbook of Grammatical Evolution.* Springer International Publishing.

Ryan, Conor, J. J. Collins, and Michael O'Neill (1998). "Grammatical Evolution: Evolving Programs for an Arbitrary Language". In: *Proceedings of the First European Workshop on Genetic Programming.* EuroGP '98. Springer-Verlag: Berlin, Heidelberg, pp. 83–96.

Yiu, Tony (2019). "Understanding Random Forest". In: *Towards Data Science*. URL: https://towardsdatascience.com/understanding-random-forest-58381e0602d2 (visited on 03/18/2022).

Yule, G. Udny (1900). "On the Association of Attributes in Statistics: With Illustrations from the Material of the Childhood Society, &c." In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 194, pp. 257–319. ISSN: 02643952. URL: http://www.jstor.org/stable/90759 (visited on 05/19/2022).

6 Appendix

The appendix to this thesis is structured accordingly to the results chapter. The first part contains general information that was used in all specifications of the experiments. The later sections provided additional information sorted by the various data sets.

6.1 Grammar

```
# Expression
<expr>::= np.where(<cond>, <label>, <label>)                                    |
          np.where(<cond>, <label>, (<expr>))                                   |
          np.where(np.logical_and(<cond>, <cond>),<label>,(<expr>))             |
          np.where(np.logical_or(<cond>,<cond>),<label>,(<expr>))               |
          np.where(<cond>, <expr>, <expr>)                                      |
          <expr>                                                                |
          <label>

# number of classes for classification task
<label> ::= -1|1

# Conditions
<cond> ::=  (x[:, <var_idx>] <= <const>)                |
            (x[:, <var_idx>] <= x[:, <var_idx>])        |
            (<subExpr> <= <const>)                      |
            (<subExpr> <= <subExpr>)

# Sub-expression (transformed variables)
<subExpr> ::=   x[:, <var_idx>]<op><const>              |
                x[:, <var_idx>]<op>x[:, <var_idx>]

# Operators
<op> ::= + | - | *

# Constant
<const> ::=  0.<digit>  | (-0.<digit>)

# Digits from zero to nine
<digit> ::= GE_RANGE:9

# number of variables in the dataset
<var_idx> ::= GE_RANGE:dataset_n_vars
```

Figure 3: *Grammar used for the creation of Grammatical Evolution Decision Trees in PonyGE2. The expressions are formed from nested NumPy.where function calls to replicate the decision structure of a conventional Decision Tree. Combined conditions were done by using the NumPy functions for and/or to enable comparisons of entire attribute vectors created in the condition part of the grammar. This grammar can be adapted to any number and specification of labels by changing the set of labels. Digits can be created using the capabilities of PonyGE2 to use ranges. The library can also infer the number of attributes from the data.*

6.2 Banknote data set

Table 7: *Comparison of the median training accuracy on the banknote data set*

	EnsBest	EnsBest weighted	EnsPop	EnsPop weighted	GEDT	RF	DT
Calibration 1	99.1 (0.2)	98.6 (0.1)	95.9 (2.3)	95.9 (2.3)	95.9 (2.6)	100 (0.3)	100 (0.0)
Calibration 2	99.0 (0.0)	98.5 (0.0)	95.4 (2.6)	95.4 (2.6)	95.7 (1.1)		

The five columns on the left show the median test accuracy for the Grammatical Evolution approaches. The last two columns show the performance of the comparison methods. The top numbers represent the median test accuracy in percent. The numbers in parentheses stand for the standard deviation of the accuracy.

6.2.1 Calibration 1

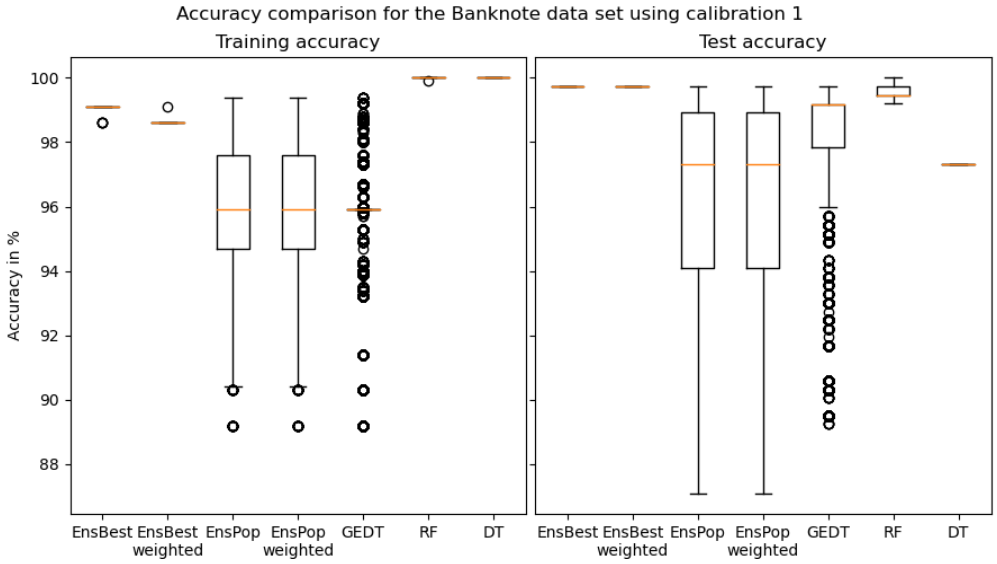

Figure 4: *Plot of the accuracy of the methods using the first calibration on the Banknote data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

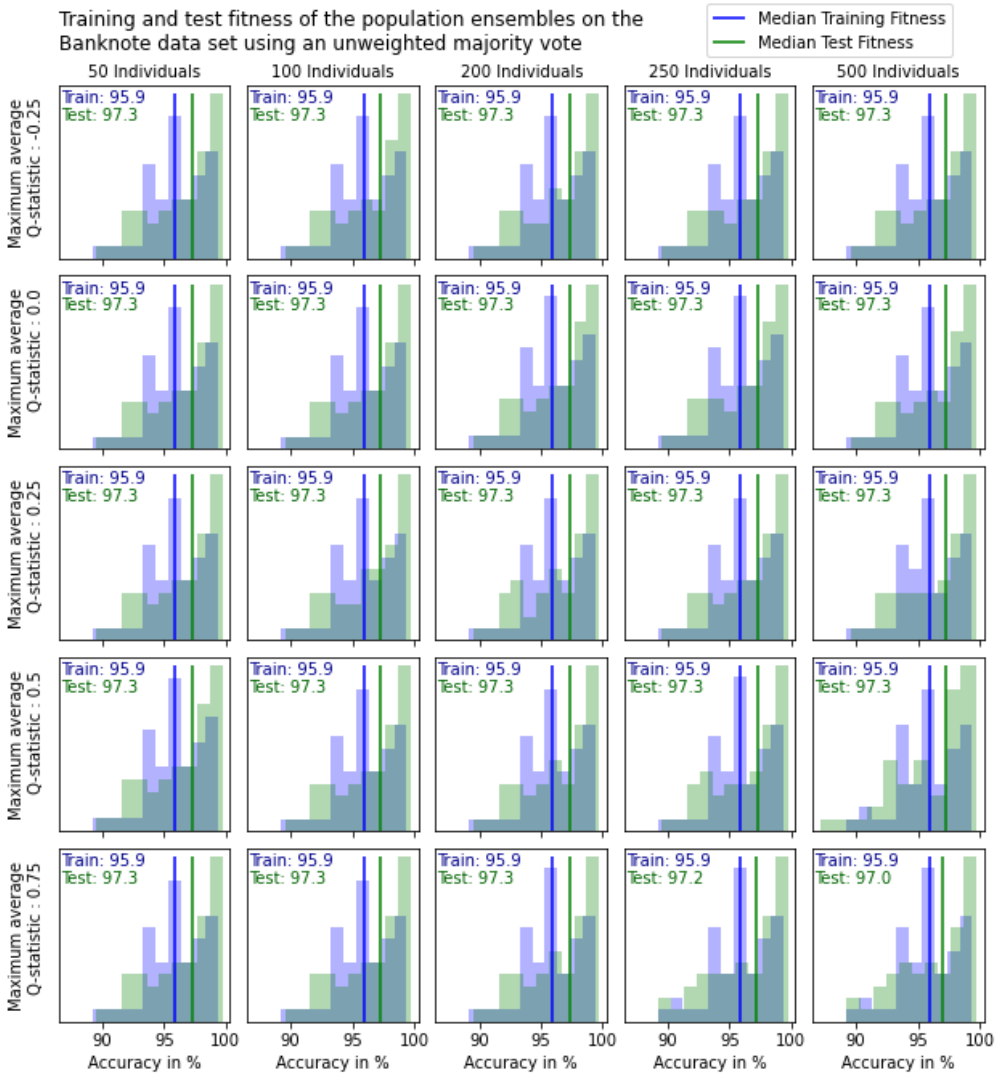

Figure 5: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

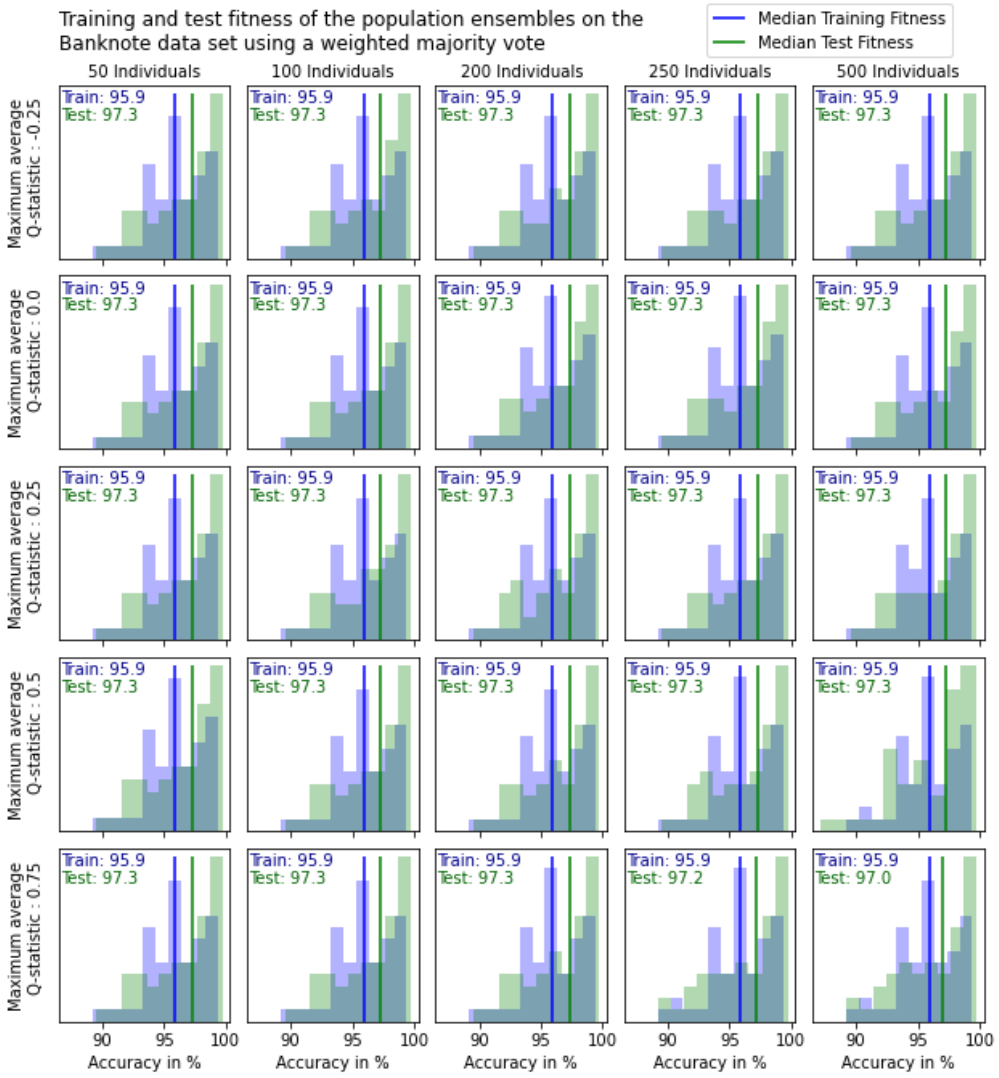

Figure 6: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

6.2.2 Calibration 2

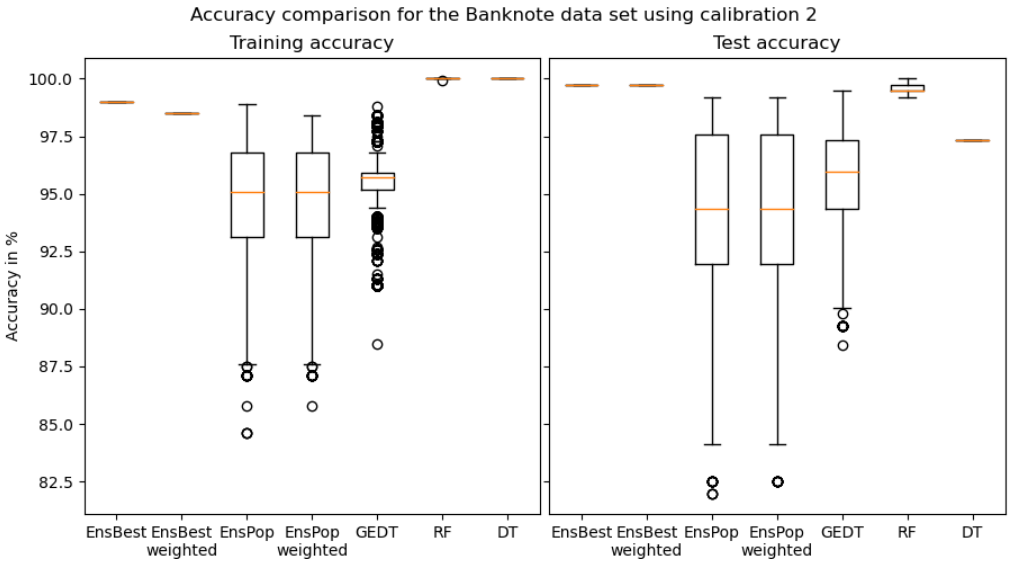

Figure 7: *Plot of the accuracy of the methods using the second calibration on the Banknote data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

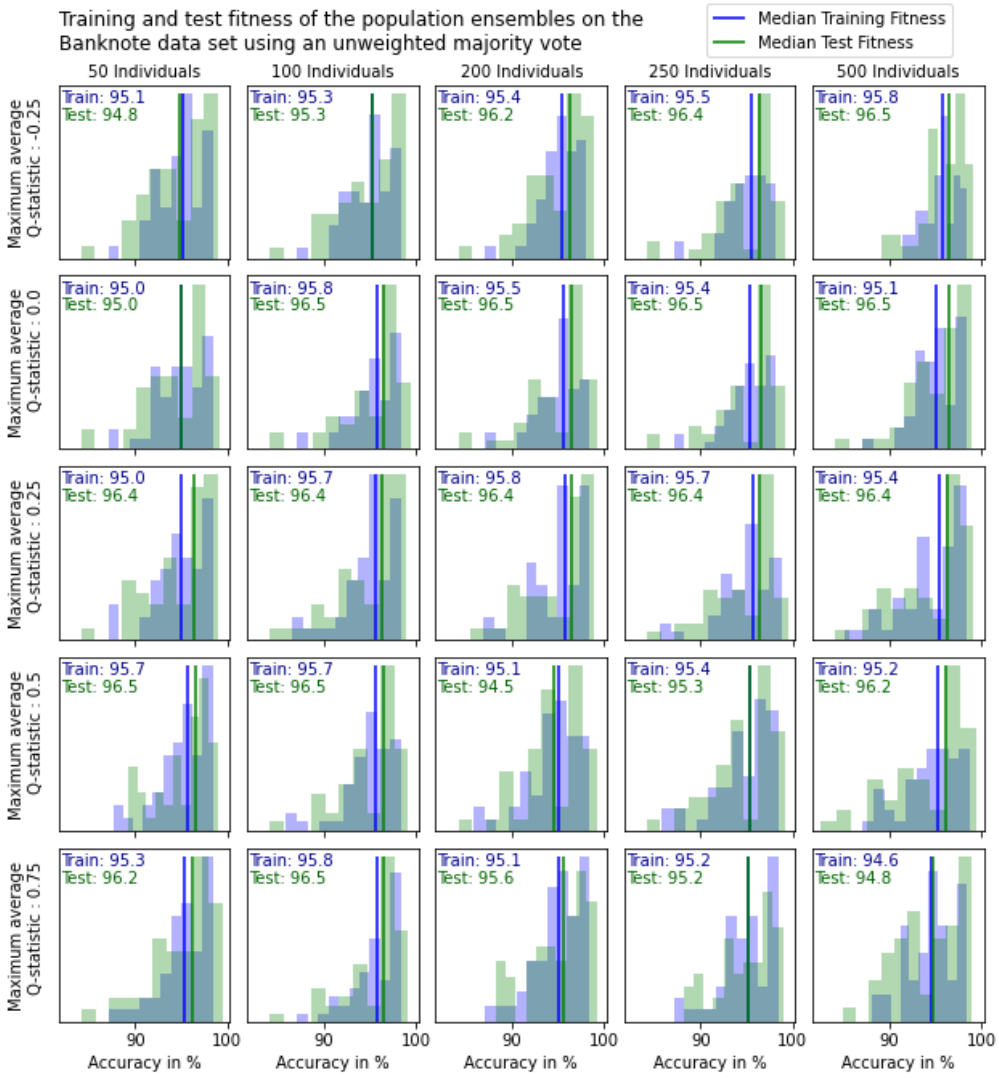

Figure 8: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

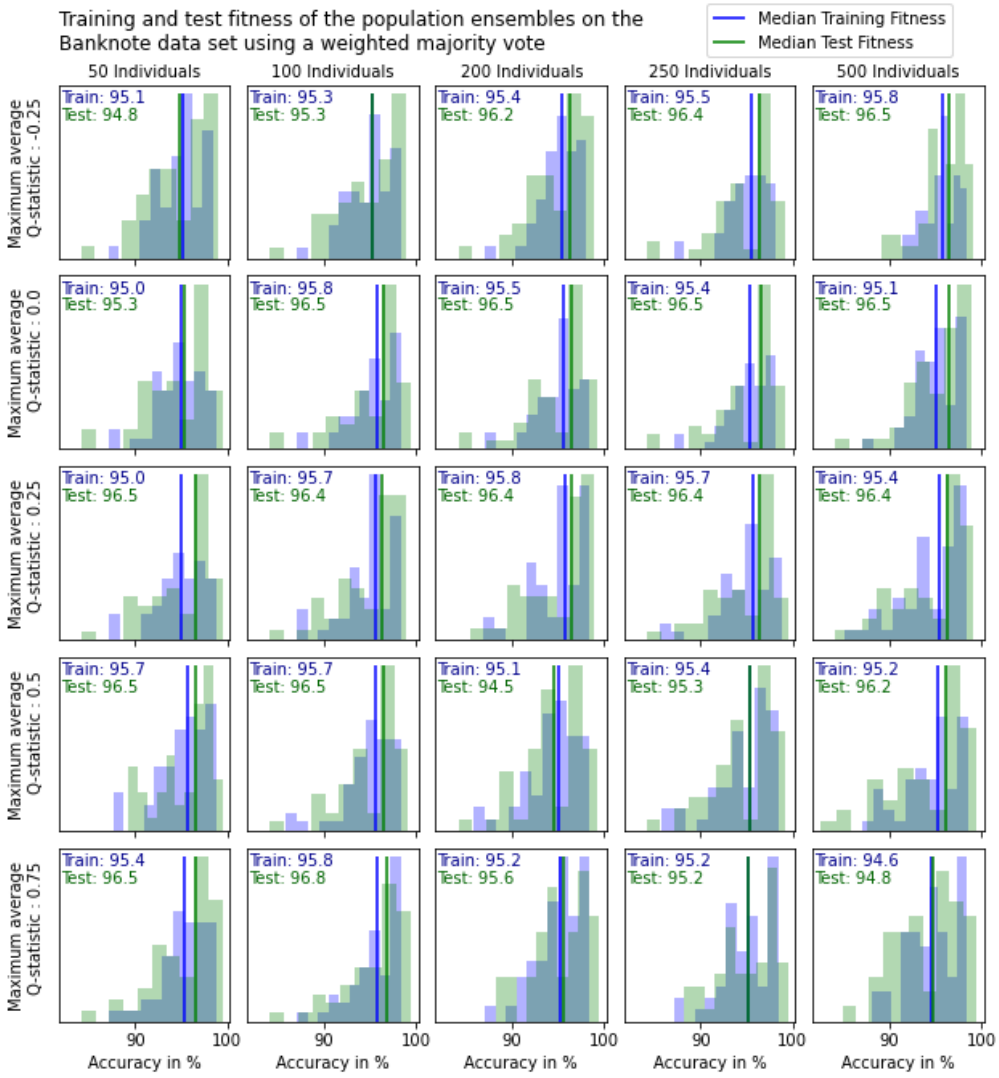

Figure 9: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

6.3 Iris data set

Table 8: *Comparison of the median training accuracy for the Iris data set*

	EnsBest	EnsBest weighted	EnsPop	EnsPop weighted	GEDT	RF	DT
Calibration 1	99.1 (0.0)	99.1 (0.0)	96.2 (2.4)	96.2 (2.4)	97.1 (1.4)	100 (0.1)	100 (0.0)
Calibration 2	97.1 (0.0)	97.1 (0.0)	95.2 (2.5)	95.2 (2.5)	95.2 (1.4)		

The five columns on the left show the median test accuracy for the Grammatical Evolution approaches. The last two columns show the performance of the comparison methods. The top numbers represent the median test accuracy in percent. The numbers in parentheses stand for the standard deviation of the accuracy.

6.3.1 Calibration 1

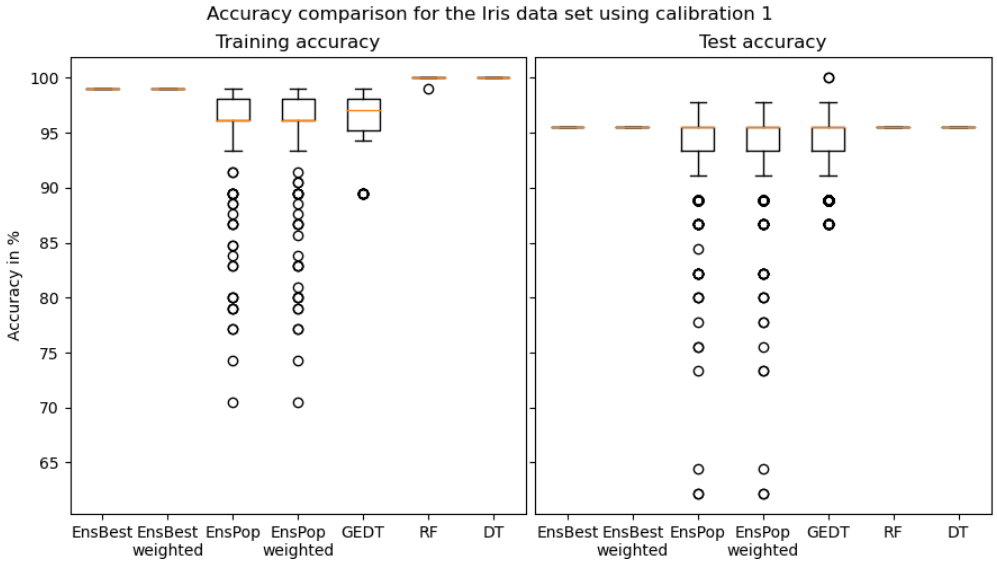

Figure 10: *Plot of the accuracy of the methods using the first calibration on the Iris data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

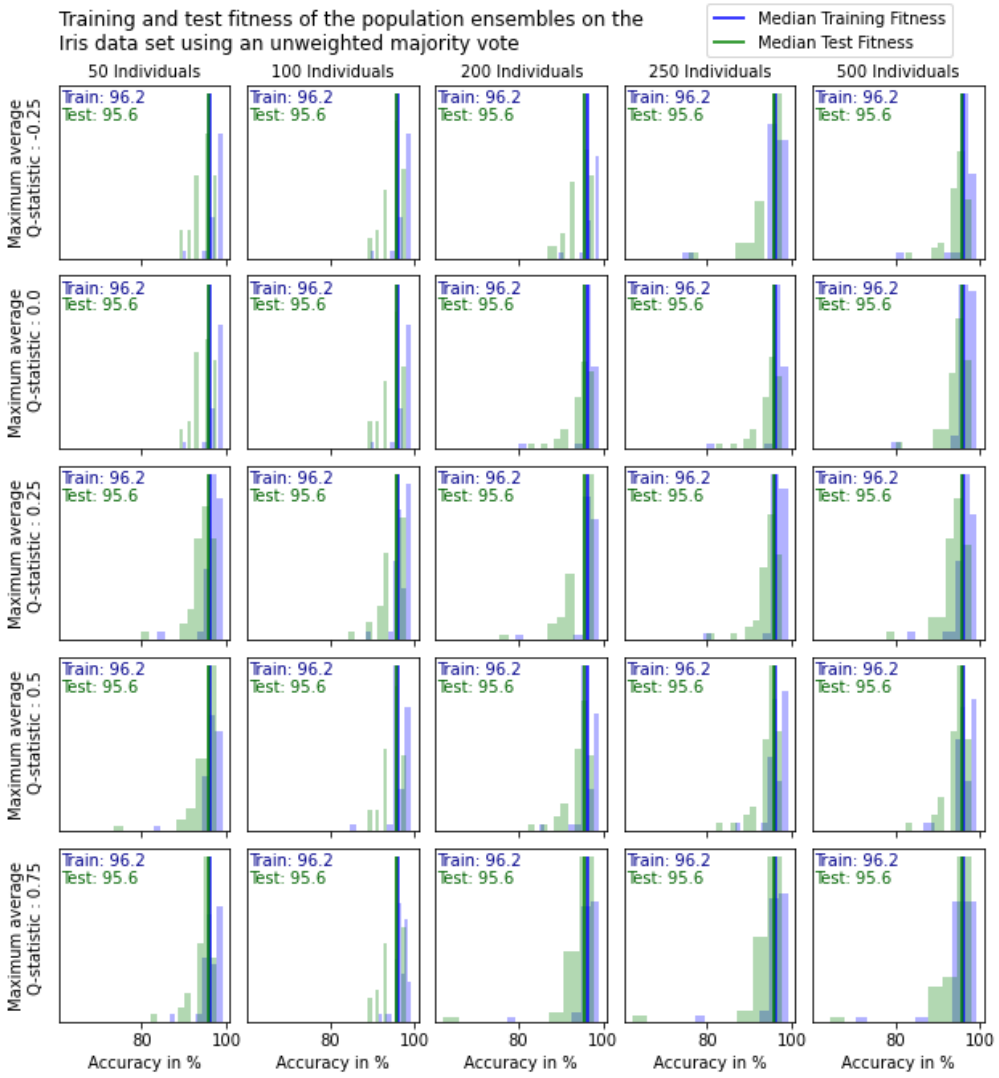

Figure 11: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

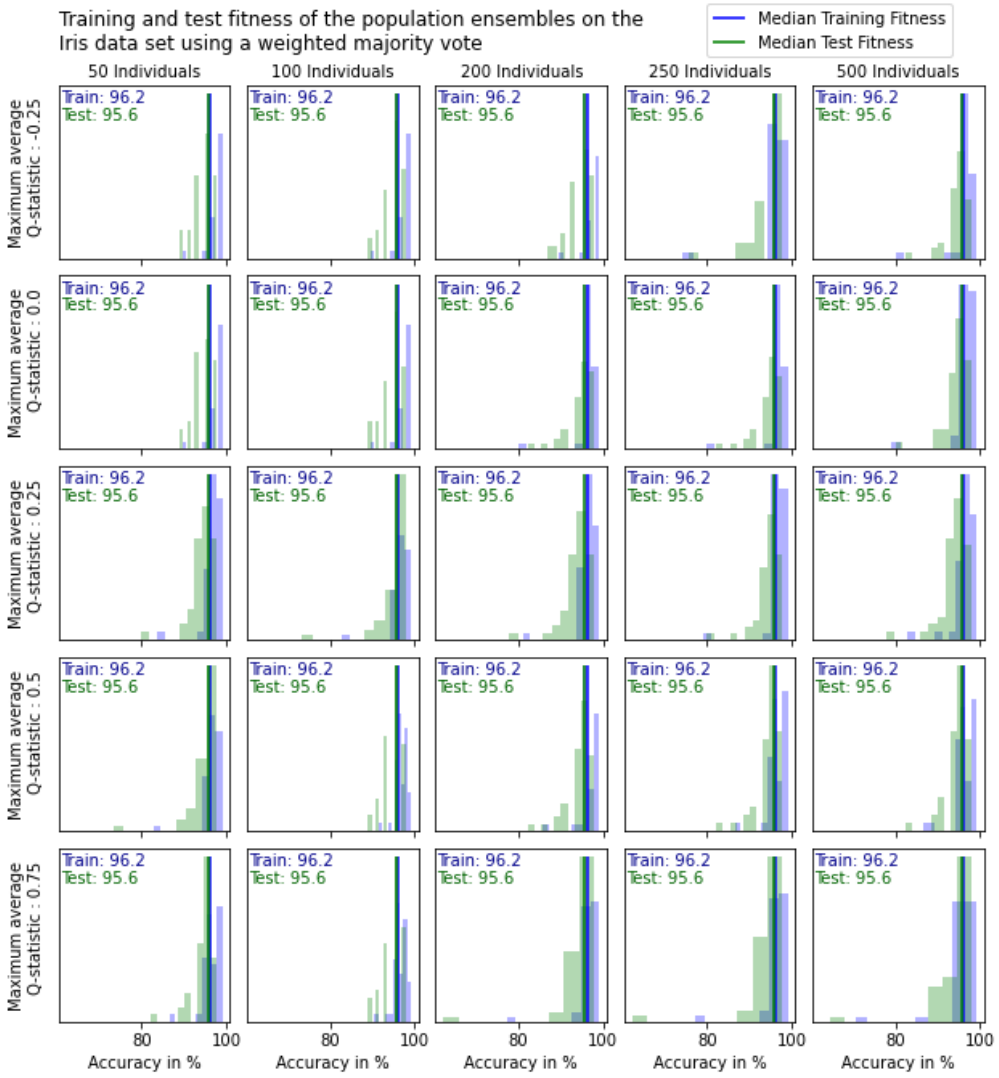

Figure 12: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

6.3.2 Calibration 2

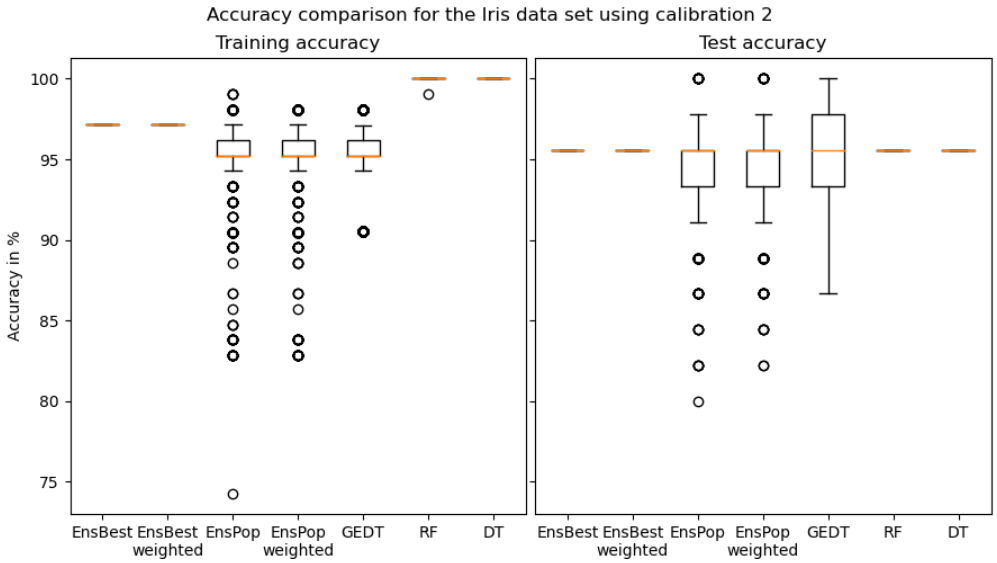

Figure 13: *Plot of the accuracy of the methods using the second calibration on the Iris data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

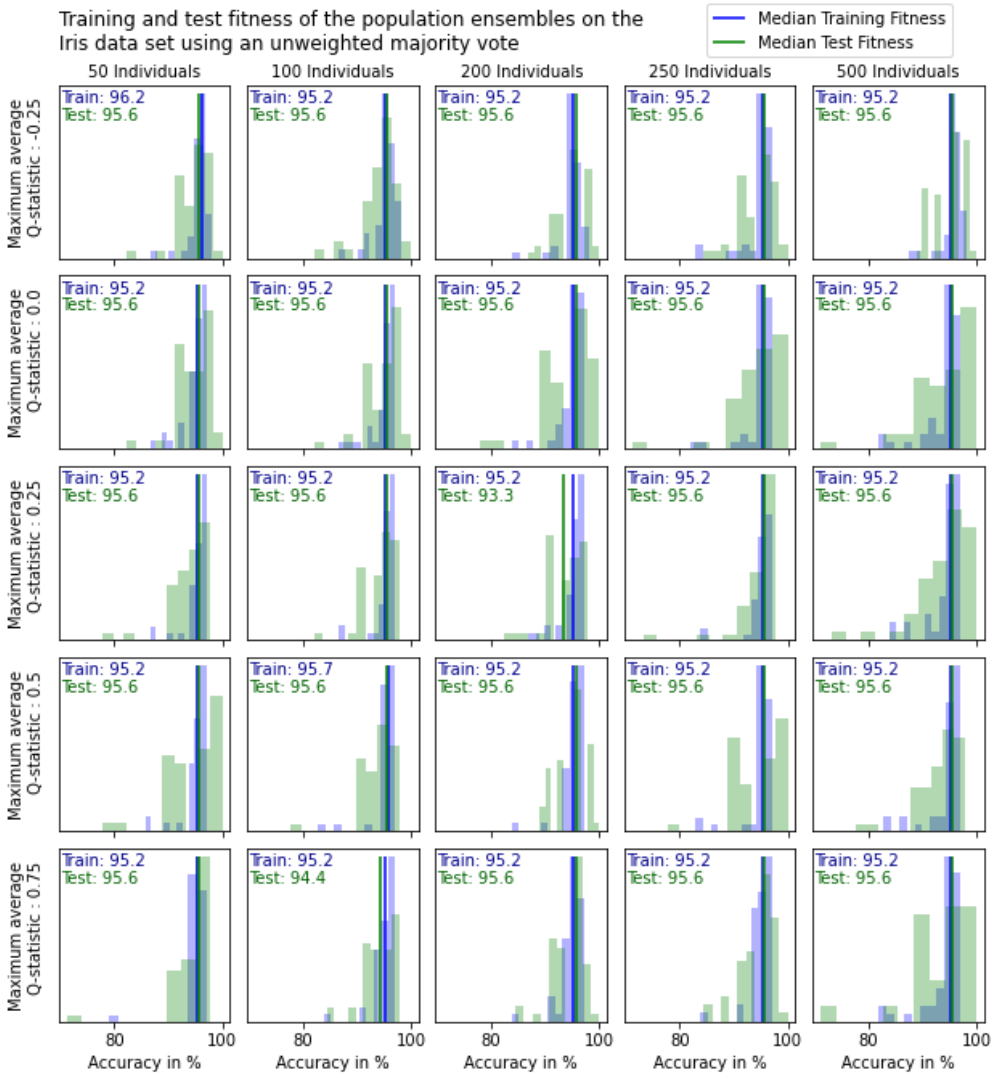

Figure 14: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

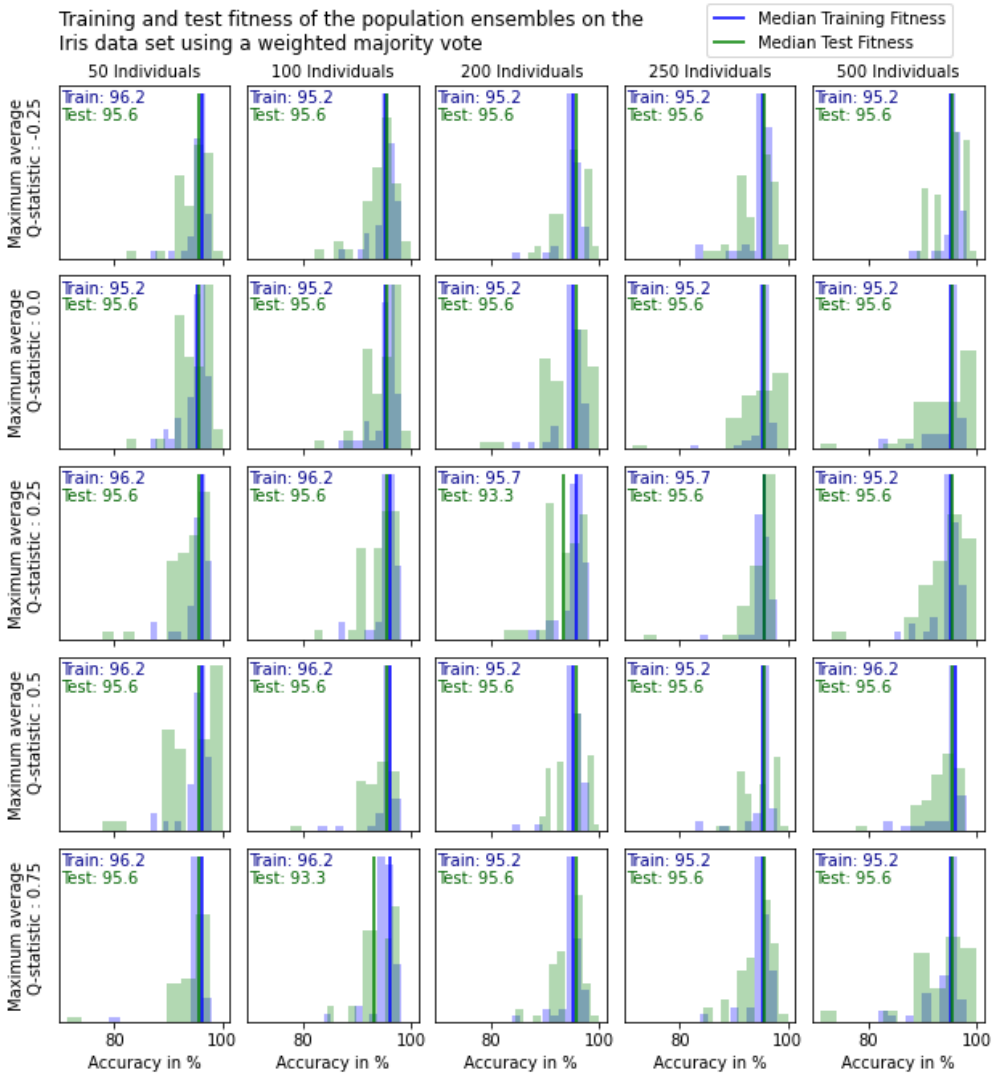

Figure 15: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

6.4 Vehicle data set

Table 9: *Comparison of the median training accuracy for the Vehicle data set*

	EnsBest	EnsBest weighted	EnsPop	EnsPop weighted	GEDT	RF	DT
Calibration 1	71.6 (0.1)	72.0 (0.1)	58.3 (5.0)	58.3 (4.9)	58.8 (3.9)	100 (0.0)	94.9 (0.0)
Calibration 2	71.4 (0.0)	71.4 (0.0)	58.3 (4.4)	58.3 (4.4)	56.6 (3.5)		
Calibration 3	69.3 (0.1)	69.6 (0.2)	50.7 (10.2)	51.5 (9.8)	57.4 (5.1)		

The five columns on the left show the median test accuracy for the Grammatical Evolution approaches. The last two columns show the performance of the comparison methods. The top numbers represent the median test accuracy in percent. The numbers in parentheses stand for the standard deviation of the accuracy.

6.4.1 Calibration 1

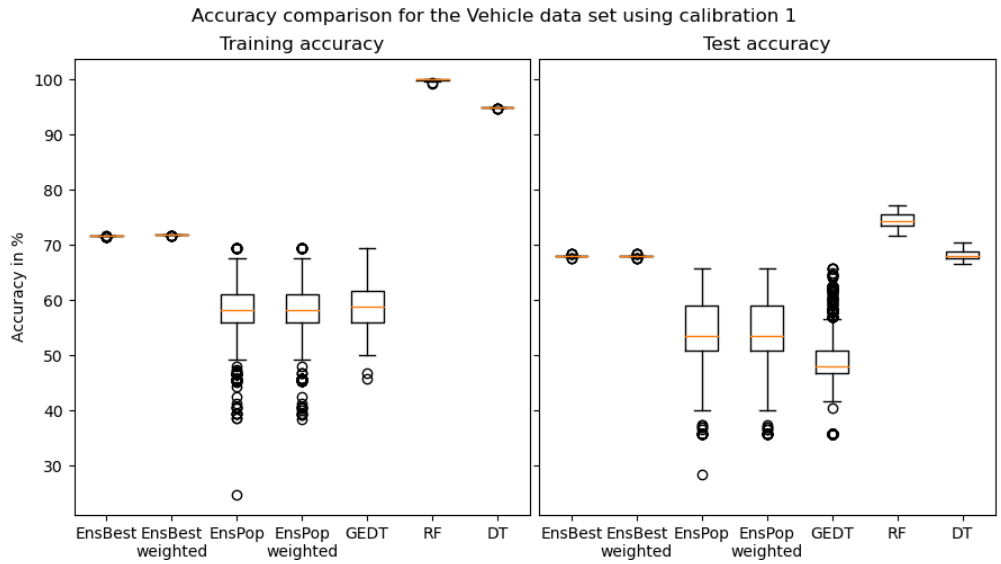

Figure 16: *Plot of the accuracy of the methods using the first calibration on the Vehicle data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

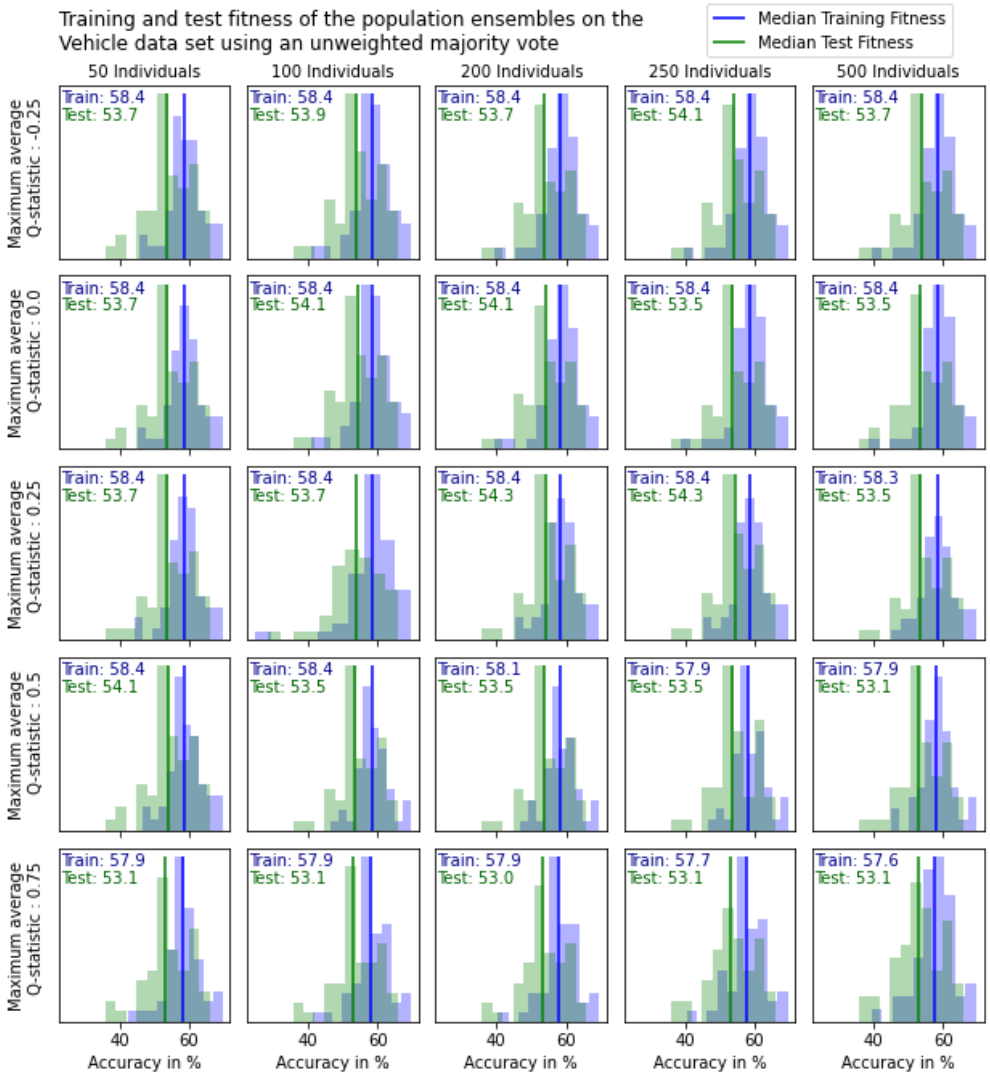

Figure 17: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

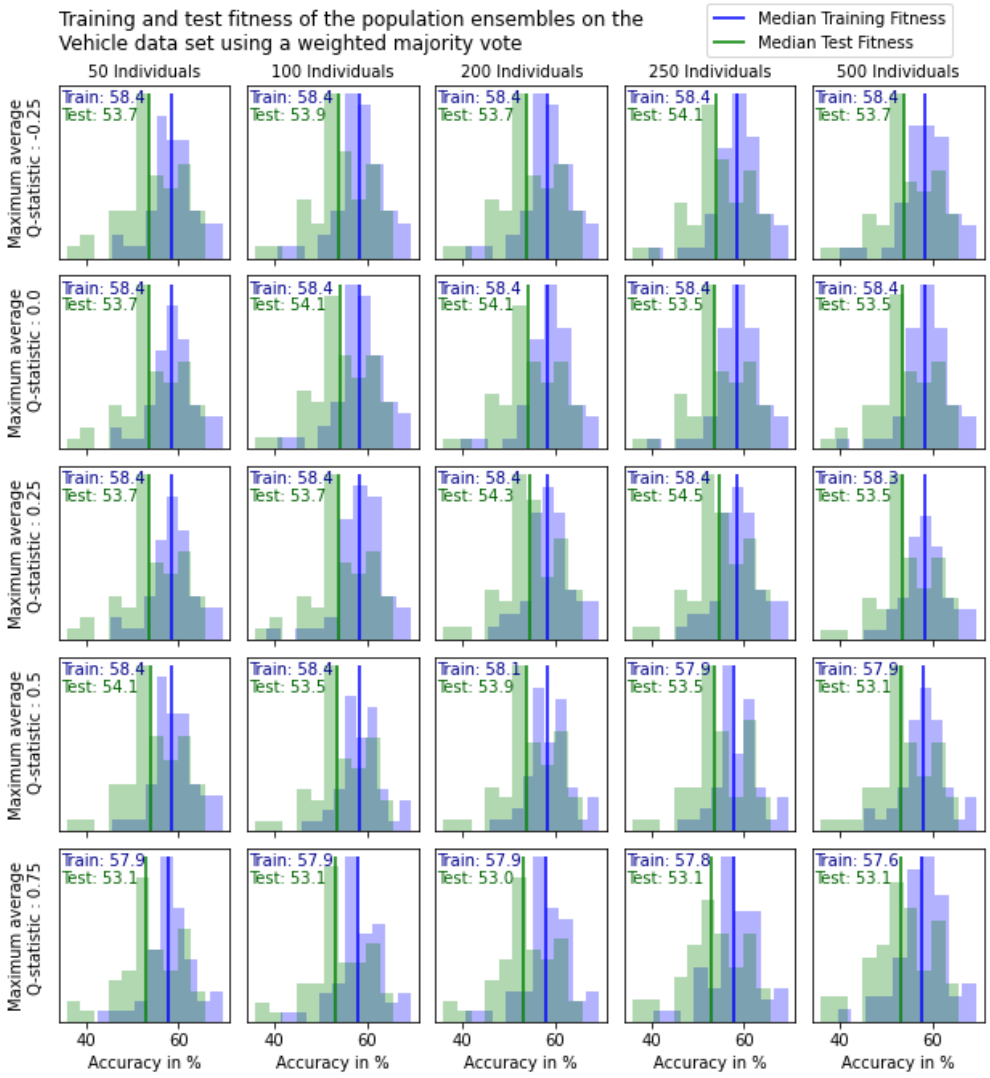

Figure 18: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

6.4.2 Calibration 2

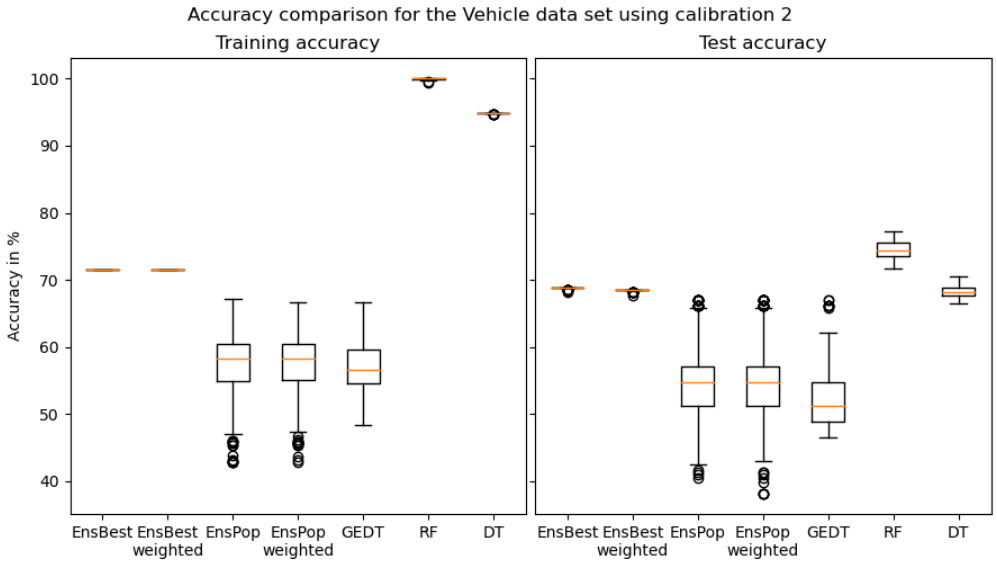

Figure 19: *Plot of the accuracy of the methods using the second calibration on the Vehicle data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

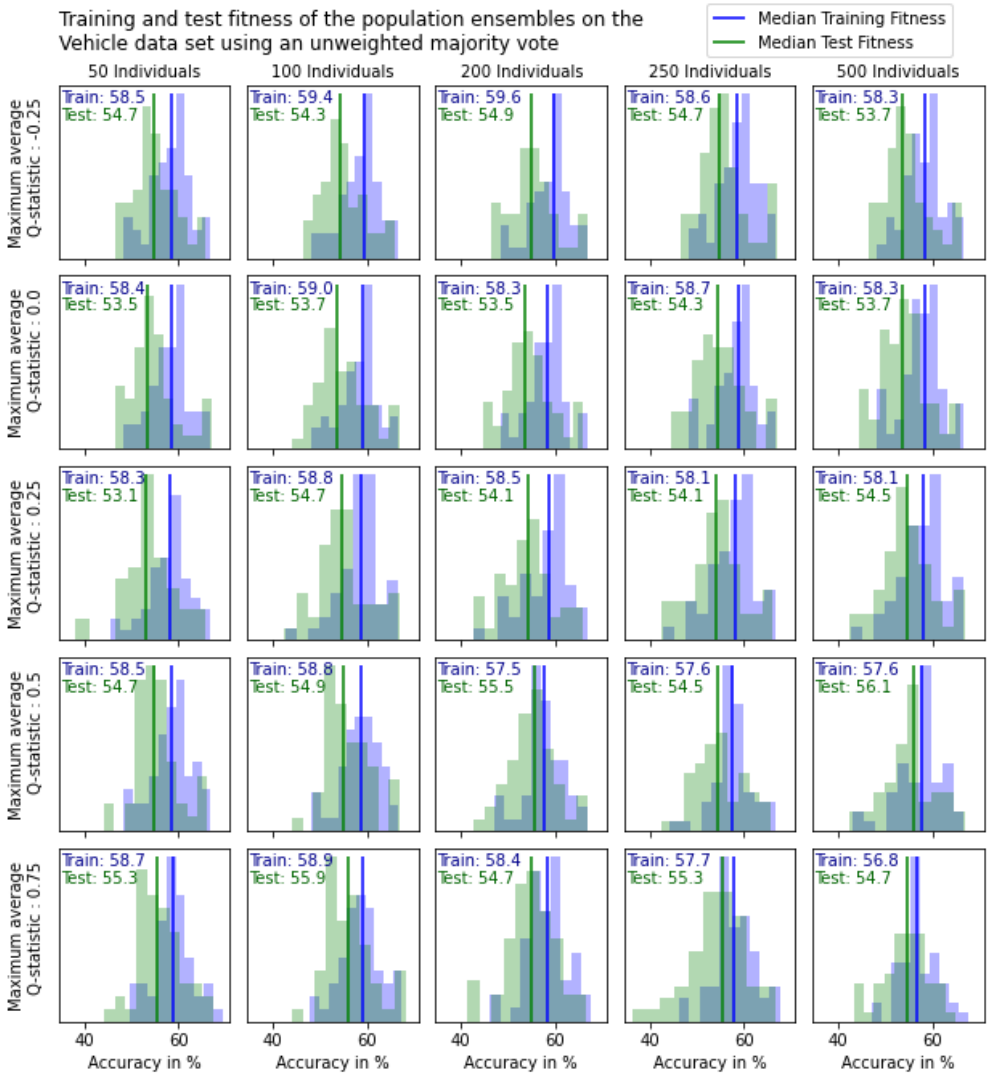

Figure 20: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

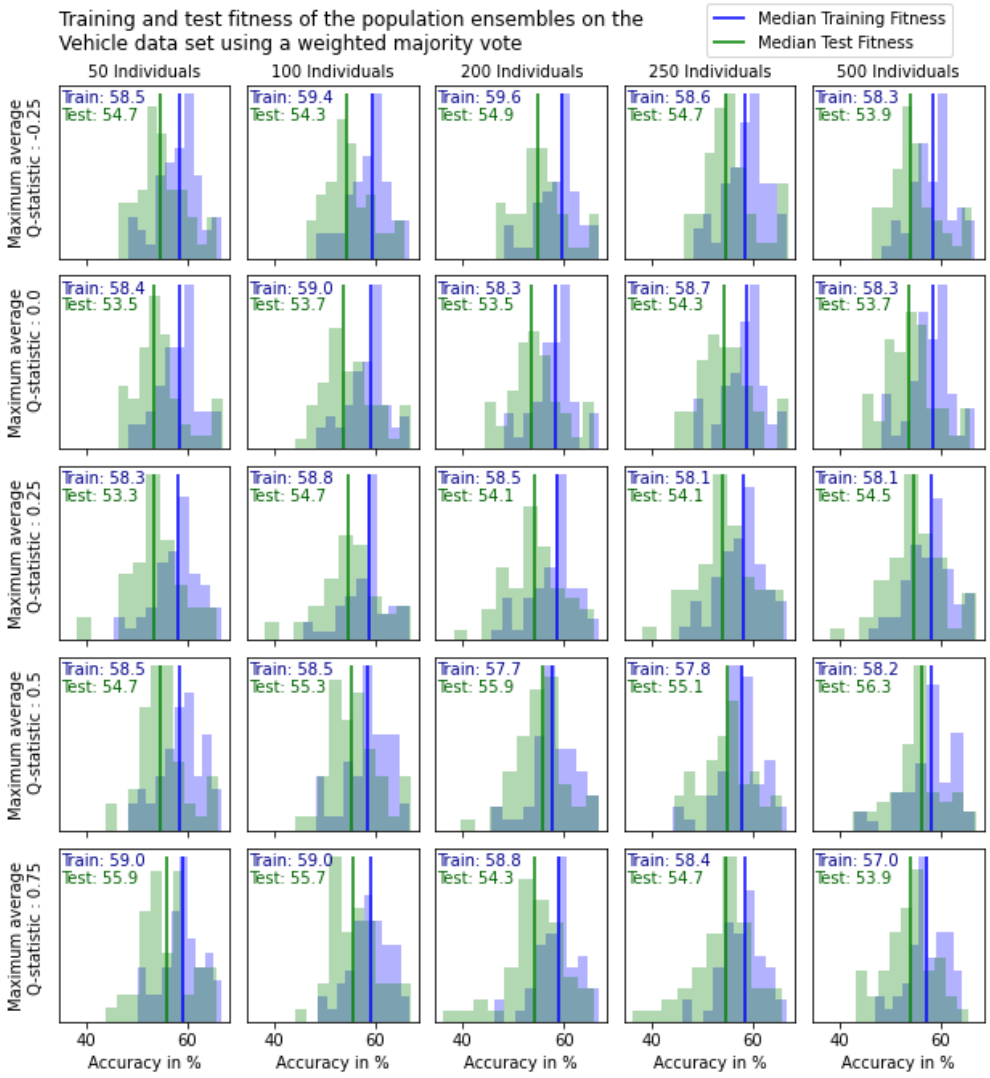

Figure 21: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

6.4.3 Calibration 3

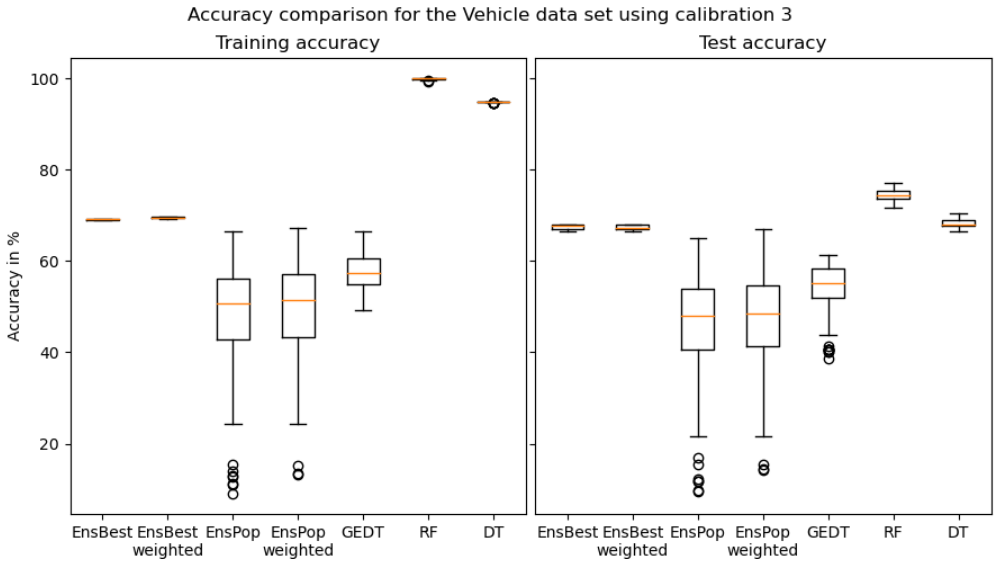

Figure 22: *Plot of the accuracy of the methods using the third calibration on the Vehicle data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

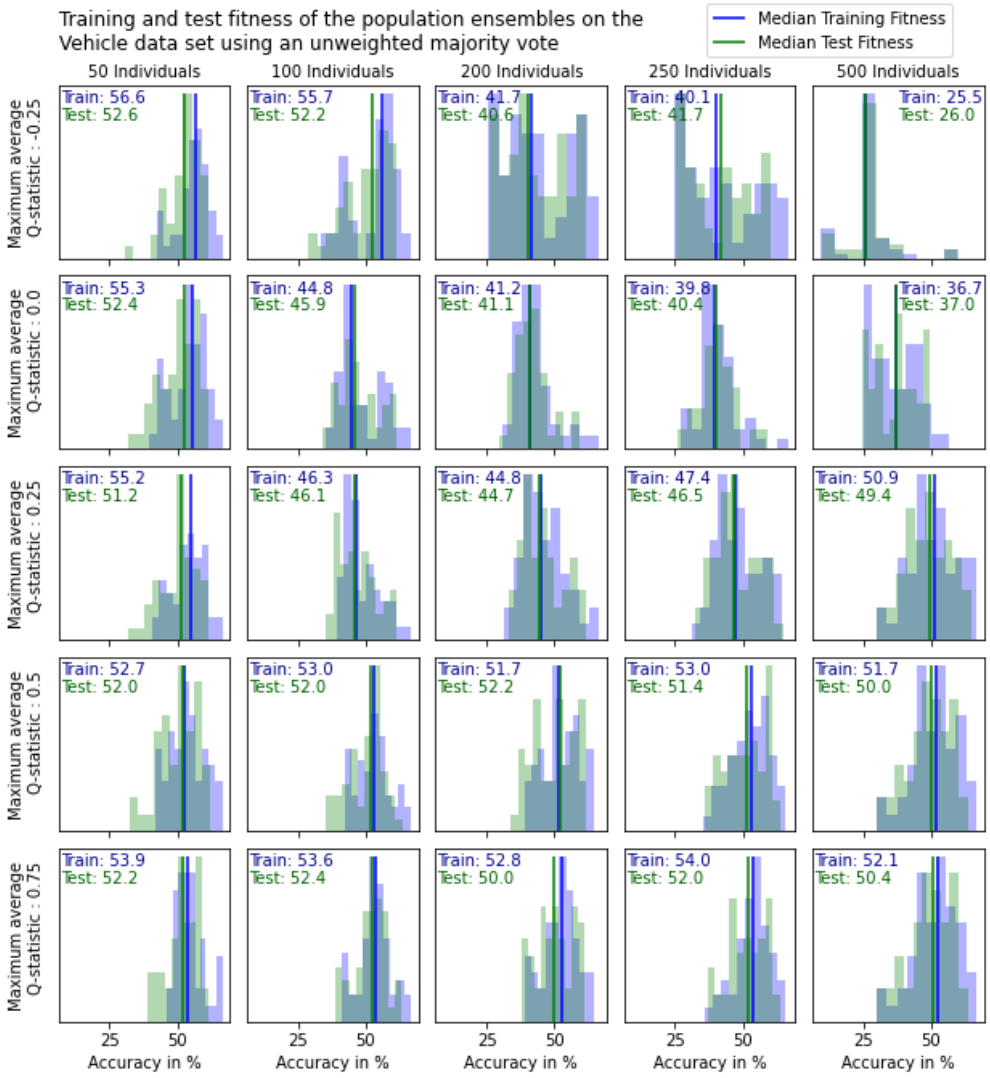

Figure 23: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

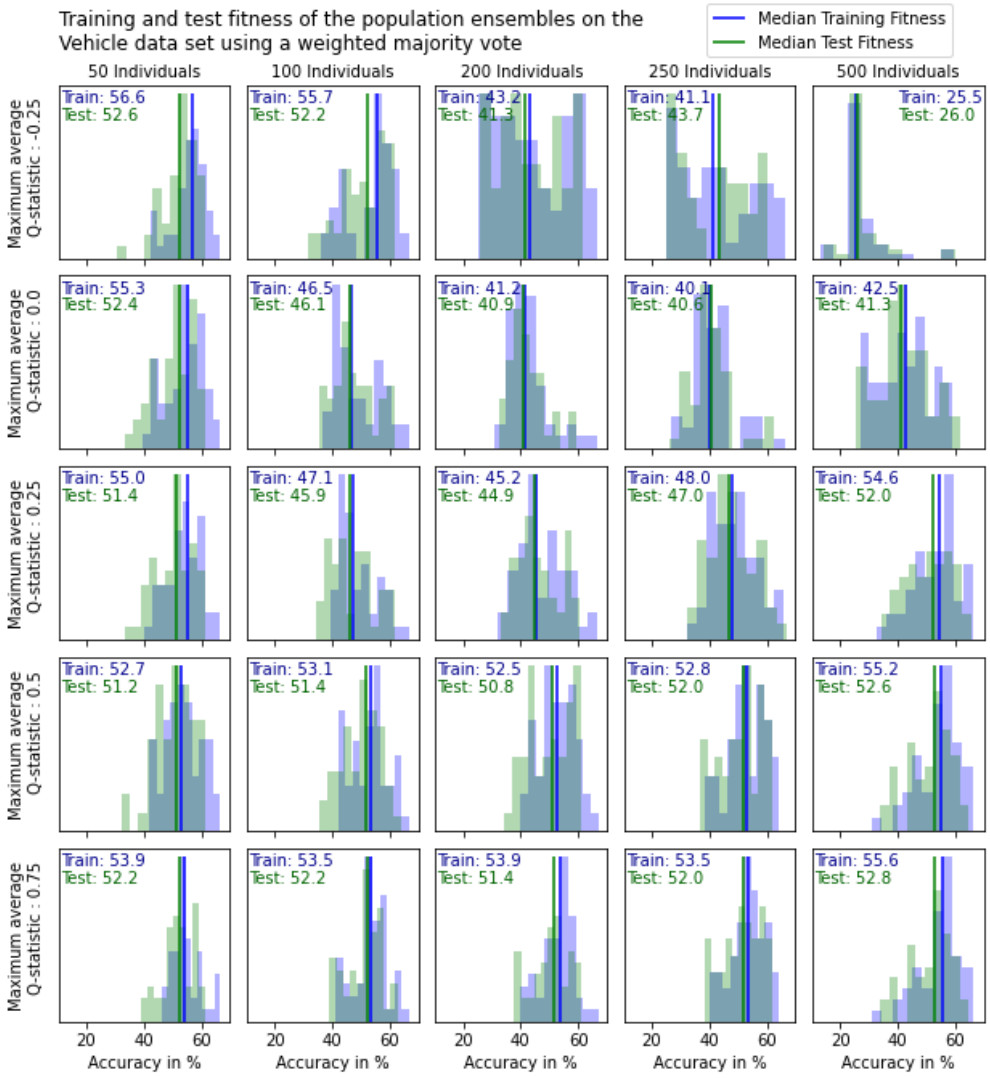

Figure 24: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

6.5 Cleveland data set

Table 10: *Comparison of the median training accuracy on the Cleveland data set*

	EnsBest	EnsBest weighted	EnsPop	EnsPop weighted	GEDT	RF	DT
Calibration 1	70.5 (0.4)	70.5 (0.4)	63.5 (1.4)	63.5 (1.4)	62.8 (1.6)	100 (0.0)	100 (0.0)
Calibration 2	61.4 (0.5)	61.4 (0.5)	62.3 (1.8)	62.3 (1.8)	62.3 (1.1)		
Calibration 3	61.8 (0.4)	62.8 (1.2)	56.5 (16.3)	55.1 (15.1)	59.9 (1.5)		

The five columns on the left show the median test accuracy for the Grammatical Evolution approaches. The last two columns show the performance of the comparison methods. The top numbers represent the median test accuracy in percent. The numbers in parentheses stand for the standard deviation of the accuracy.

6.5.1 Calibration 1

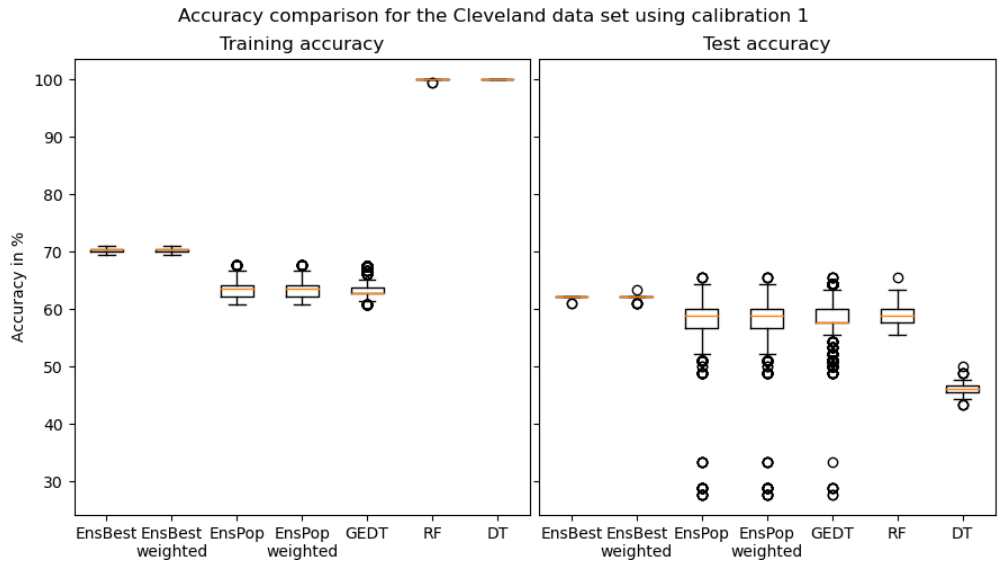

Figure 25: *Plot of the accuracy of the methods using the first calibration on the Cleveland data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

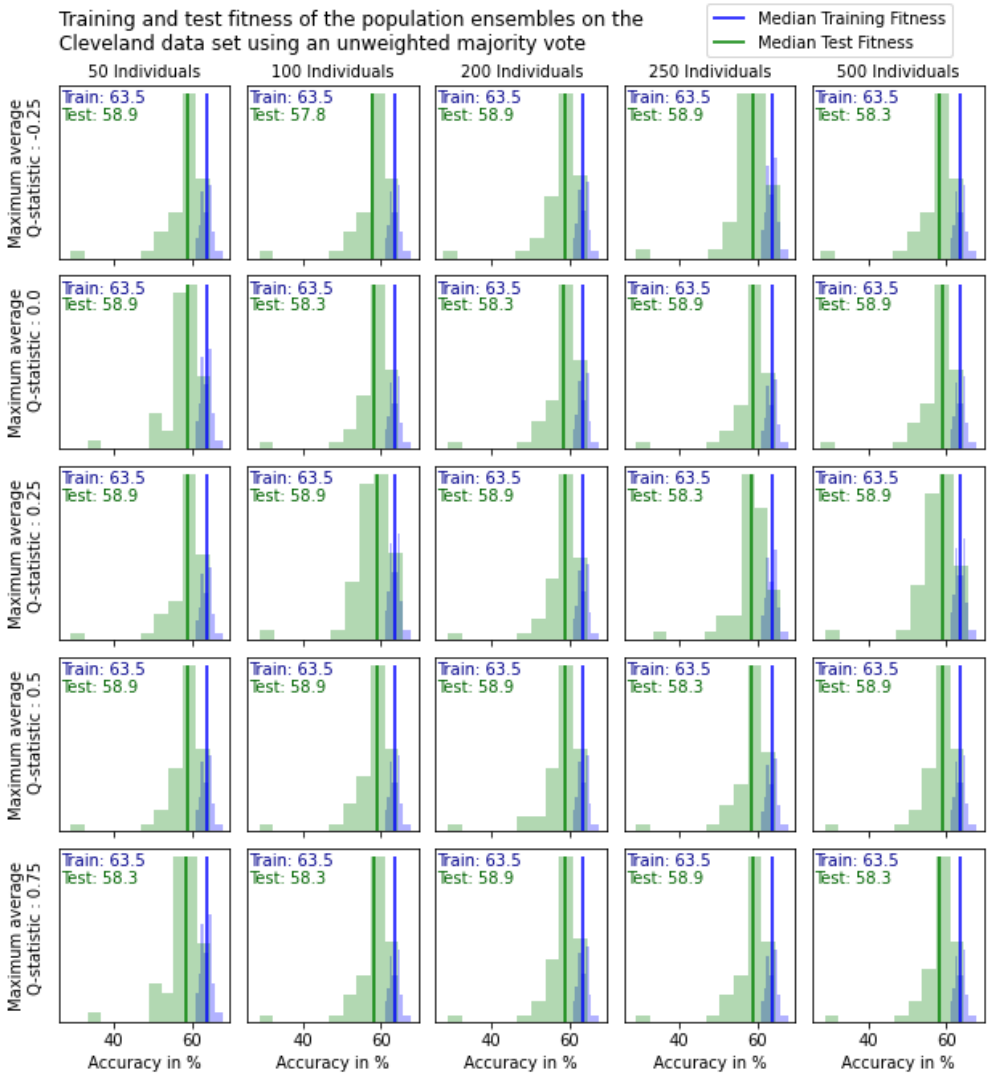

Figure 26: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

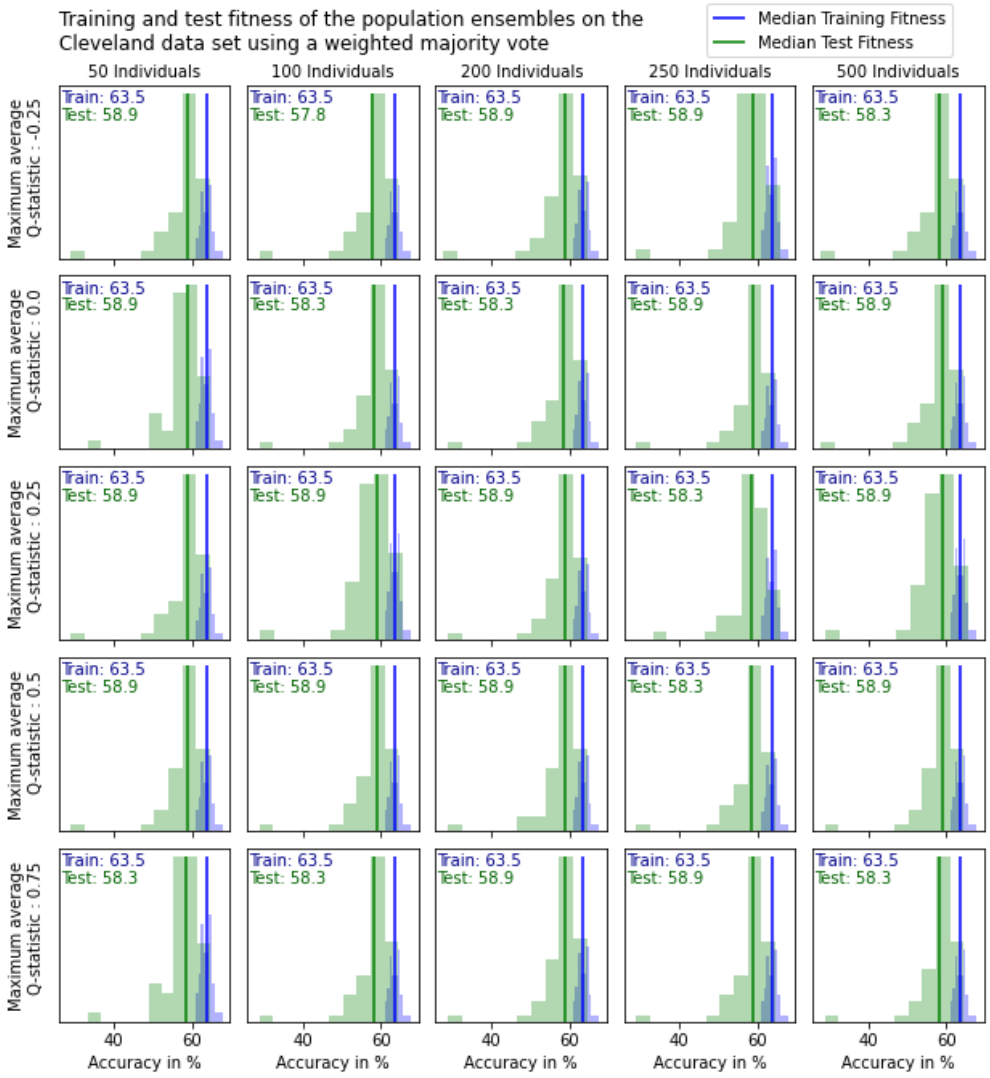

Figure 27: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

6.5.2 Calibration 2

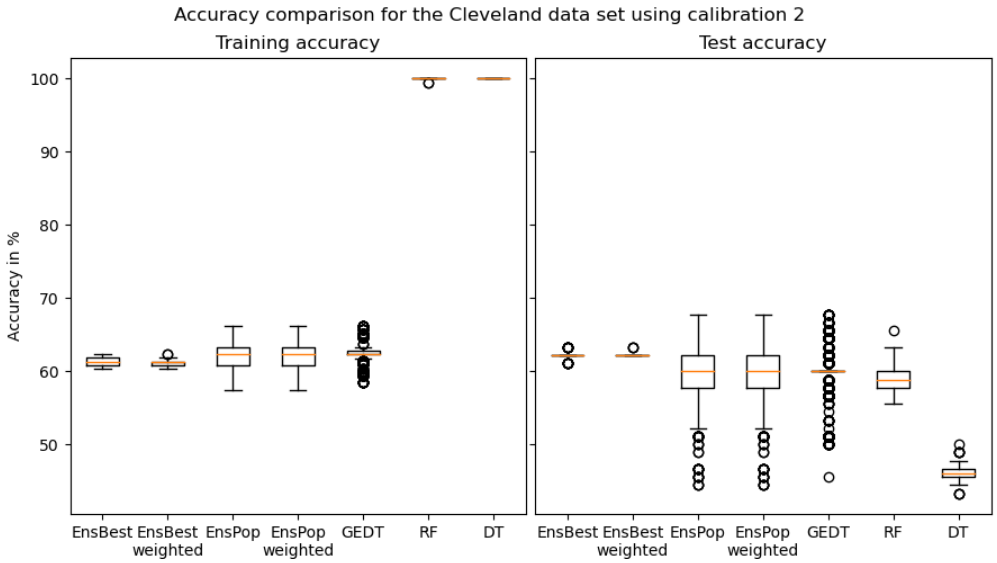

Figure 28: *Plot of the accuracy of the methods using the second calibration on the Cleveland data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

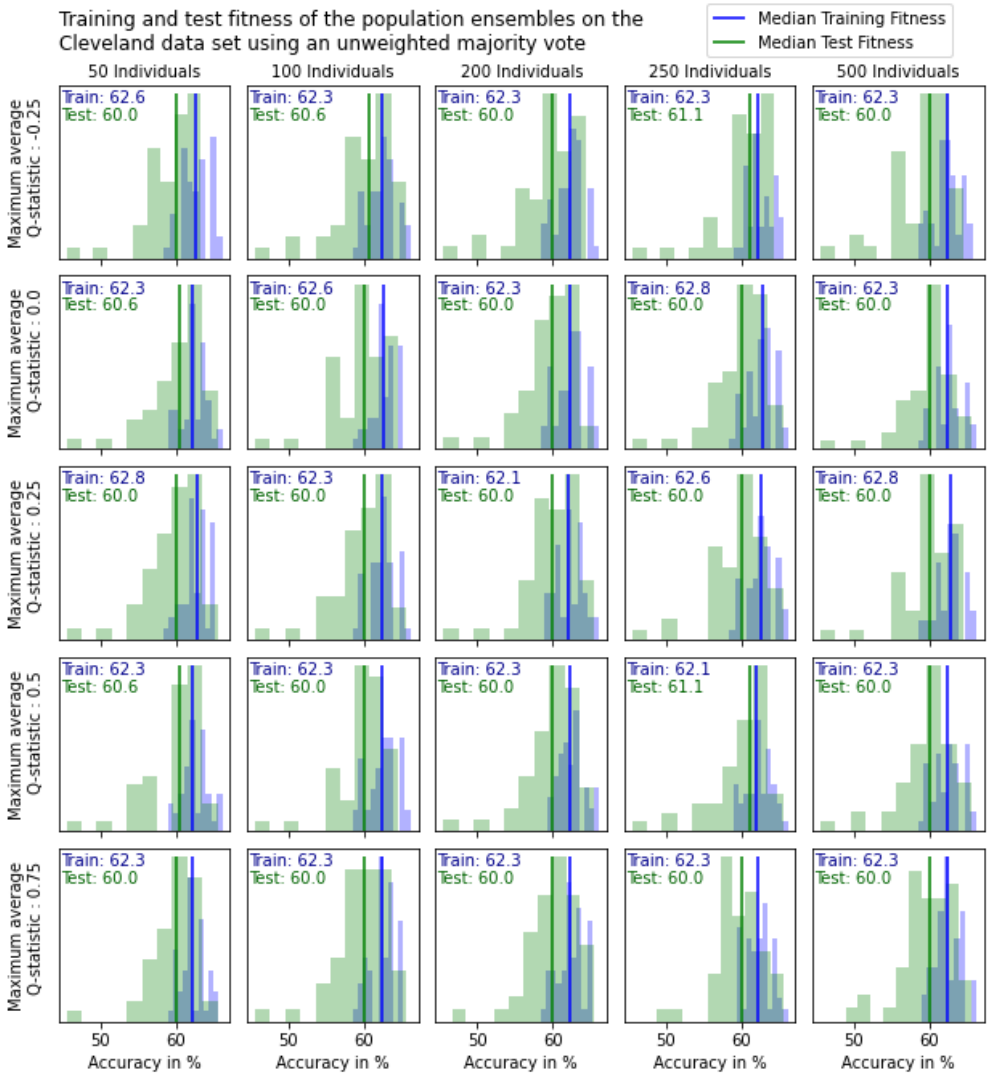

Figure 29: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

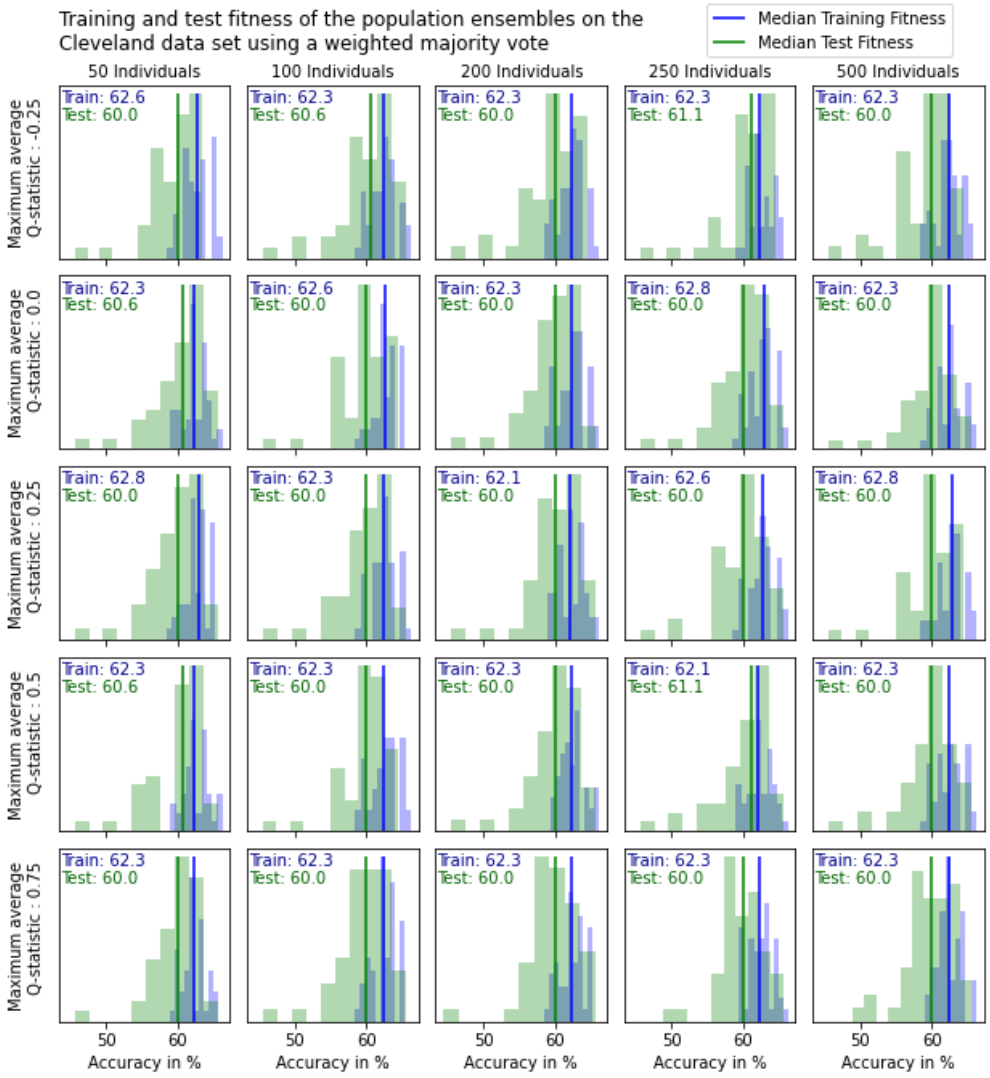

Figure 30: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

6.5.3 Calibration 3

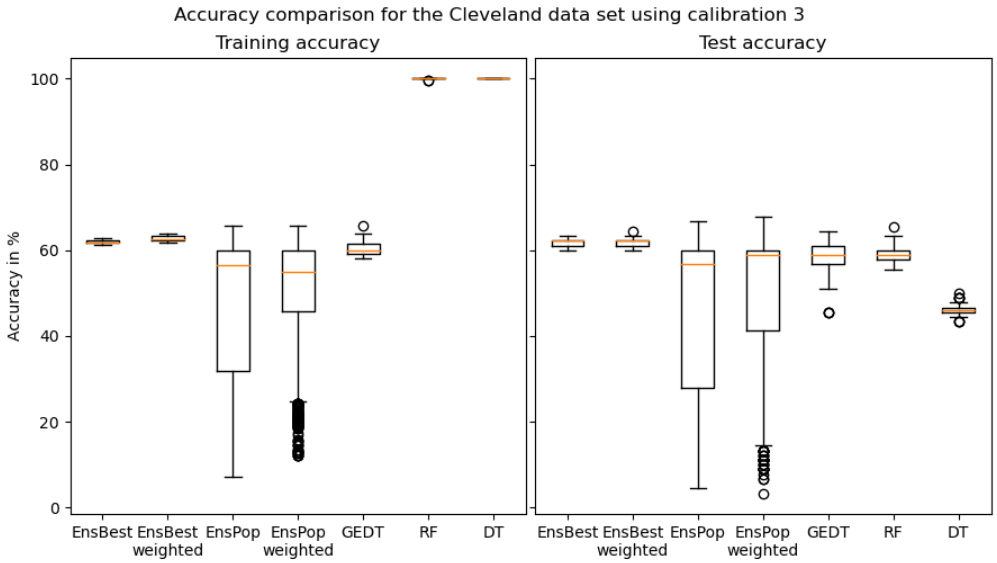

Figure 31: *Plot of the accuracy of the methods using the third calibration on the Cleveland data set. The left panel shows a box plot for the training accuracy. The right panel shows a box plot of the test accuracy.*

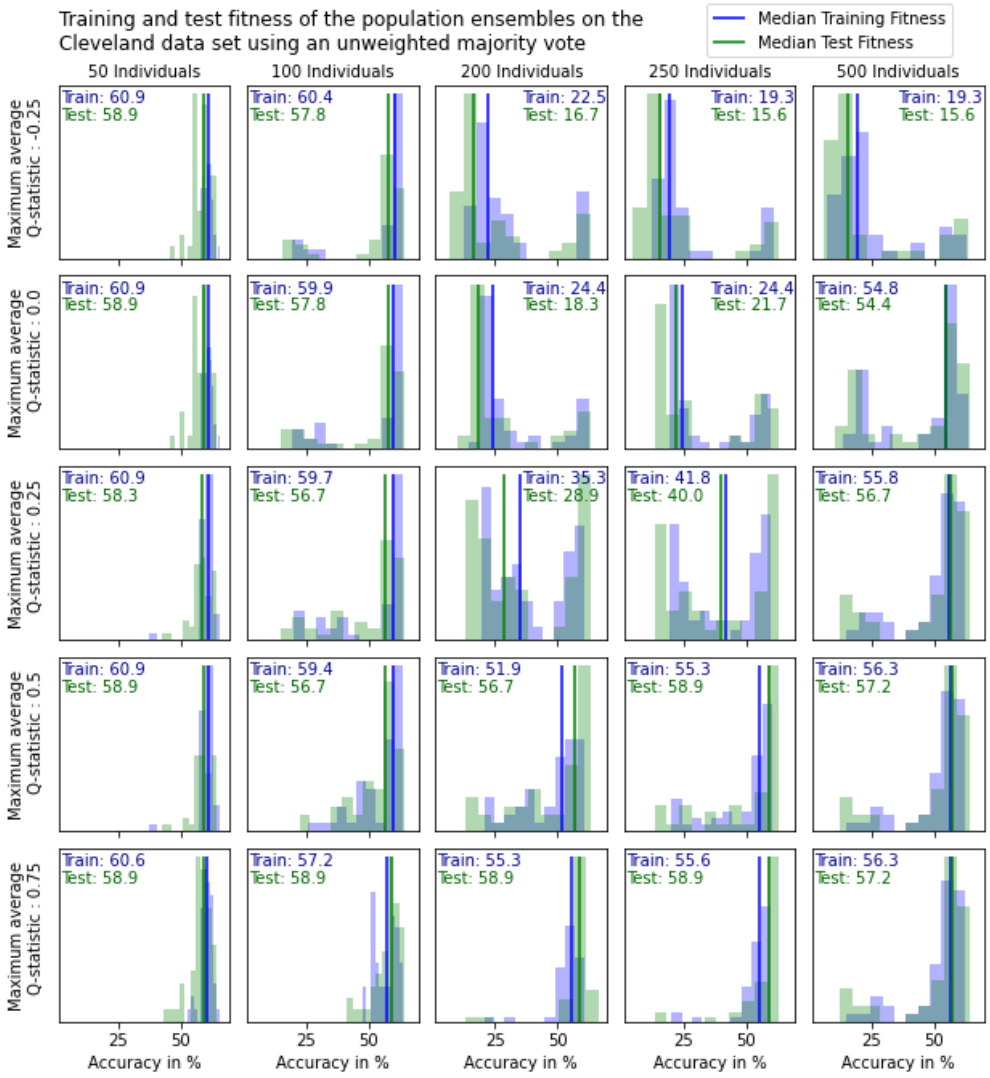

Figure 32: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

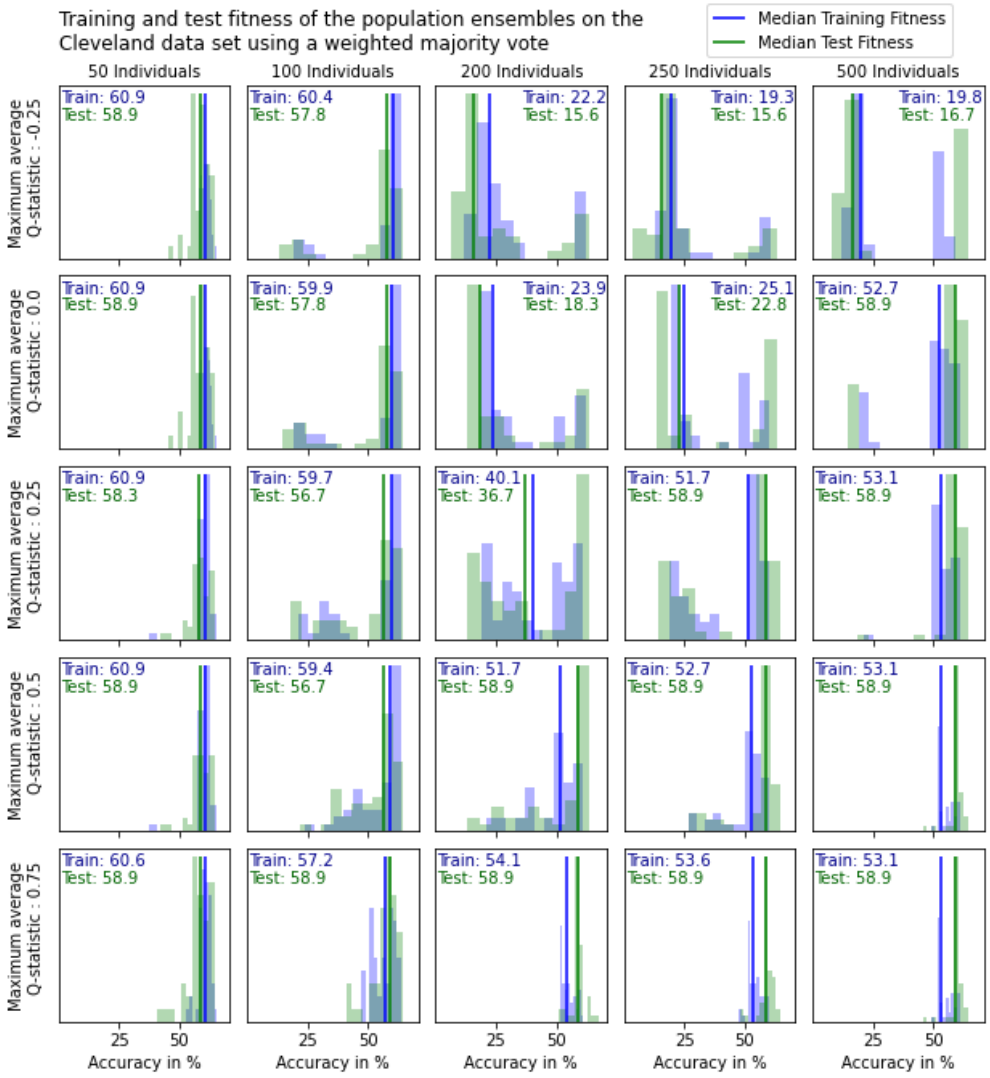

Figure 33: *Plot of the distributions of the training fitness (blue) and test fitness (green) of the fifty population ensembles created following the initial ensemble sizes indicated in the columns and Q-statistic as an exclusion cutoff indicated in the rows. Means of the training and test fitness are represented as vertical lines in the corresponding colors and in the upper corners of the subplots.*

7 Plagiatserklärung

Ich bezeuge mit meiner Unterschrift, dass meine Angaben über die bei der Abfassung meiner Arbeit benützten Hilfsmittel sowie über die mir zuteil gewordene Hilfe in jeder Hinsicht der Wahrheit entsprechen und vollständig sind. Ich habe das Merkblatt zu Plagiat und Betrug vom 22.02.2011 gelesen und bin mir der Konsequenzen eines solchen Handelns bewusst.

Dominik Gaiato