SN Computer Science (2020) 1:134
https://doi.org/10.1007/s42979-020-00131-0

ORIGINAL RESEARCH

# Multiple Imputation Ensembles (MIE) for Dealing with Missing Data

**Aliya Aleryani[1] · Wenjia Wang[1] · Beatriz de la Iglesia[1]**

Received: 31 December 2019 / Accepted: 30 March 2020


**Abstract**
Missing data is a significant issue in many real-world datasets, yet there are no robust methods for dealing with it appropriately. In this paper, we propose a robust approach to dealing with missing data in classification problems: Multiple Imputation Ensembles (MIE). Our method integrates two approaches: multiple imputation and ensemble methods and compares two types of ensembles: bagging and stacking. We also propose a robust experimental set-up using 20 benchmark datasets from the UCI machine learning repository. For each dataset, we introduce increasing amounts of data Missing Completely at Random. Firstly, we use a number of single/multiple imputation methods to recover the missing values and then ensemble a number of different classifiers built on the imputed data. We assess the quality of the imputation by using dissimilarity measures. We also evaluate the MIE performance by comparing classification accuracy on the complete and imputed data. Furthermore, we use the accuracy of simple imputation as a benchmark for comparison. We find that our proposed approach combining multiple imputation with ensemble techniques outperform others, particularly as missing data increases.

**Keywords** Missing data · Multiple imputation · Dissimilarity measures · Classification algorithms · Ensemble techniques

## Introduction

Many real-world datasets have missing or incomplete data [22, 23, 75]. Since the accuracy of most machine learning algorithms for classification, regression and clustering could be affected by the completeness of datasets, processing and dealing with missing data is a significant step in data mining and machine learning processes. Yet, this is still underexplored in the literature [11, 28, 49, 61, 68–70].

A few strategies have been commonly used to handle incomplete data [30, 34, 48]. For regression problems specifically where missing data has been more widely studied [36, 38, 39, 48, 55], multiple imputation (MI) has shown advantage over other methods [48, 72] because the multiple imputed values give a mechanism to capture the uncertainty reflected in missing data. However, work is still needed to address the problem of missing data in the context of data mining algorithms. Particularly, it is timely to experiment with the concept of multiple imputation and how to apply to classification problems.

The aim of this work is therefore to conduct a thorough investigation on how to effectively apply MI for classification algorithms. We propose an ensemble that combines multiple models produced by MI, and we investigate the ways for combining different ensemble mechanisms with MI methods to achieve best results.

Our proposed method, MIE, is evaluated and compared with other alternatives under some simulated scenarios of increasing uncertainty in terms of missing data. For this, we create an experimental environment using datasets selected from the University of California Irvine (UCI) machine learning repository [47]. For each dataset, we use a mechanism called Missing Completely at Random (MCAR) to generate missing data by removing the values of chosen attributes and instances with a variable probability. Therefore, we produce several experimental datasets which contain increasing amount of data MCAR.

In those scenarios, we investigate how increasing the amount of missing data affects the performance of competing approaches for handling missing data. They include the algorithm's internal mechanism for handling missing data,

✉ Aliya Aleryani
A.Aleryani@uea.ac.uk

Wenjia Wang
W.Wang@uea.ac.uk

Beatriz de la Iglesia
B.Iglesia@uea.ac.uk

[1] University of East Anglia, Norwich NR4 7TJ, UK

Published online: 23 April 2020

single imputation, machine learning imputation and our proposed MIE.

### The Problem of Missing Data

Little and Rubin [48] have defined the missing data problem based on how missing data is produced in the first place and they proposed three main categories as follows: Missing Completely at Random (MCAR), Missing at Random (MAR) and Missing not at Random (MNAR). The categorisation is important because it affects the biases that may be inherent in the data, and therefore the safety of approaches such as imputation. Missing Completely at Random (MCAR) occurs when an instance missing for a particular variable is independent of any other variable and independent of the missing data. It can be said that for MCAR missing data is not related to any other factor known or unknown in the study. This represents the safer environment for imputation to operate. Missing at Random (MAR) happens when the probability of having missing value in a record may depend on the known values of other attributes but not on the missing data. There are some inherent biases in data MAR, but it may be still safe to analyse this type of data without explicitly accounting for the missing data. Missing not at Random (MNAR) occurs when the probability of the instance having a missing value depends on unobserved values. This is also termed a *non-ignorable* process and is the most difficult scenario to deal with. In this paper we focus on addressing MCAR data, the safest environment in which imputation could operate and one that is often encountered. Further work will investigate the other mechanisms.

Horton et al. [39] have further categorised the patterns of missing data into monotone and non-monotone. Monotone patterns of missing data imply that the same data points have missing values in one or more features, so specific points are affected by missing data. They state that the patterns are concerned with which values are missing, whereas the mechanisms are concerned with why data is missing.

We focus in this study on non-monotone MCAR data, so our missing data affects multiple data points with no particular relation between data missing for different attributes for the same data points.

### Mechanisms for Dealing with Missing Data

In practice, there are four popular approaches that have been used to deal with incomplete data: complete analysis [48], statistical imputation methods [4, 24, 34, 59], machine learning algorithms for imputation [40, 51, 52, 57, 59] and algorithms with a built-in mechanism to deal with missing data [10, 25, 43, 51, 74]. The first three approaches rely on pre-processing of the data to either remove or replace missing values. The last approach comprises a mechanism in the algorithms themselves to produce models taking account of the missing data. We provided a detailed explanation of the different approaches in our previous work [2].

We also studied how different classification algorithms such as C4.5 [51, 52], Naïve Bayes (NB) [10, 45], support vector machines (SVMs) [5, 16, 74] and random forest (RF) [7, 21] and their implementations in Weka, our platform of choice, can treat missing values [2]. A number of classification algorithms (e.g. C4.5 [52] and RF [43]) have been constructed with a mechanism called *fractional* method to cope with missing data. Naïve Bayes ignores features with missing values; thus, only the complete features are used for classification [10, 45]. SVMs do not deal with missing values [46] but its implementation in Weka, SMO, performs simple imputation [32]. In this work we investigate PART, in addition to the previous classifiers, which is also capable of treating missing data when constructing a partial tree as C4.5 does [25].

The rest of this paper is organised as follows: "Related Work" section reviews related research; the methods used in our paper are described in "MIE for Classification" section followed by our experimental set-up in "Experimental Set-Up" section and "Results" section detail out the results of this study; this is followed by a discussion and conclusions in "Discussion and Conclusions" section.

## Related Work

MI has been studied in the context of statistical analysis [55, 57, 58]. After that, it has been widely applied in many studies such as in survival analysis [41, 73], epidemiological and clinical trials [44, 65], medical studies [56, 72] and longitudinal studies [50, 63]. The application of MI with ensemble learning for classification has rarely been used in the literature. We review a few published papers that have discussed the problem of missing data in the context of classification algorithms and the use of MI methods.

Silva-Ramírez et al. [62] proposed a method for simple imputation based on a multi-layer perceptron (IMLP) and a method for multiple data imputation that combines a multi-layer perceptron and $k$-nearest neighbour ($k$-NN algorithm to impute missing data (MIMLP). The problem under consideration was monotone MCAR missing data. The methods were compared with the traditional imputation methods such as mean, hot-deck and regression-based imputation. Their results showed that the MIMLP method performed best for numeric variables and the IMLP method performed better with categorical variables. Imputation by MLP methods offered some advantages for some datasets though statistical test for significance was not performed.

Liu et al. [49] proposed a credal classification method with adaptive imputation for incomplete pattern. In credal

classification objects can belong to multiple classes and meta-classes. The method has two stages. First, a record is classified based on the available information if the class is non-ambiguous. However, when the record is hard to classify, then it goes to the second step which involves imputation and later classification. In the imputation phase, self-organized map (SOM) is used in combination with *k*-NN to obtain good accuracy while reducing computational burden.

A correlation-based low-rank matrix completion (LRMC) method was developed by Chen et al. [12]. The method applies LRMC to estimate missing data then uses a weighted Pearson's correlation followed by K-nearest neighbour (*k*-NN) search to choose the most similar samples. Furthermore, they proposed an ensemble learning to integrate multiple imputed values for a specific sample to improve imputation performance. The proposed method was tested on both traffic flow volume data and benchmark datasets. Further investigation was conducted to test the performance of the imputation in the classification tasks. Their proposed correlation-based LRMC and its ensemble learning method achieved better performance than traffic flow imputation methods such as temporal nearest average imputation (TNAI), temporal average imputation (TAI), probabilistic principal component analysis (PPCA) and low-rank matrix completion (LRMC).

Tran et al. [69] proposed a method that introduces multiple imputation with an ensemble and compared the proposed method with others that use simple imputation. Ten datasets were collected from UCI repository. The ensemble achieved better classification accuracy than the other methods. However, they only applied C4.5 as a classification algorithm and used one method to perform multiple imputation on relatively small datasets.

Garciarena and Santana [31] studied the relationship between different imputation methods and missing data patterns using ten datasets from the UCI repository and a set of fourteen different classifiers such as decision trees, neural networks, support vector machines, *k*-NN and logistic regression. The result shows that the performance of individual classifiers is statistically different when using various imputation methods. They concluded that the key to selecting proper imputation methods is to check first the patterns of missing data.

Tran et al. [70] further proposed methods incorporating imputation (single/multiple) with feature selections and clustering to improve classification accuracy and also the computational efficiency of imputation.

A new hybrid technique based on a fuzzy *c*-means clustering algorithm, mutual information feature selection and regression models (GFCM) was developed by Sefidian and Daneshpour [61]. The aim was to find a set of similar records with high dependencies for a missing record and then apply regression imputation techniques within the group to estimate missing values for that record. The method showed statistically significant differences in most cases in comparison with mean imputation, kNNI, MLPI [62], FCMI [53] and IARI [64].

## MIE for Classification

MI is a promising method that has been used to replace missing values by randomly drawing several imputed values from the distribution of unknown data [48, 55]. Unlike in simple imputation, the uncertainty is reflected as the imputation process will result in various plausible values. There are a number of methods to impute data that we will explore in our work, and explain below. We also explore different methods to ensemble the results obtained from the different imputed values, as the ensemble represents a method for combining the evidence from the different models to arrive a final classification which should encompass the degree of missing data.

### Imputation Methods

#### Multivariate Imputation by Chained Equations (MICE)

Fully Conditional Specification (FCS) is a method of MI that was firstly developed by Kennickell [42]. It defines a conditional density function to specify an imputation model for each missing predictor (variable) one by one, then iterates the imputation over that model. Multivariate Imputation with Chained Equations *(MICE)*, an algorithm developed by Buuren [8], is based on FCS but the imputation can be also applied for data that has no multivariate distribution. For each variable with missing values, the algorithm starts by identifying an imputation model for each column with missing values. After that, the imputation will be performed based on random draws from the observed data. The process is repeated based on the number of iterations set-up and the number of variables with missing values.

#### Expectation–Maximisation with Bootstrapping (EMB)

Honaker et al. [38] have developed an EMB algorithm for handling missing data that combines the EM algorithm with a bootstrapping approach. The EM algorithm is an iterative approach developed by Dempster et al. [17]. Starting with the expectation and then the maximisation step, the algorithm aims to estimate the model parameters by iteratively performing the following. Firstly, in the expectation step (E-step), the likelihood function is evaluated by considering the current estimate of the model parameters. Second, in the maximisation step (M-step), the parameters are updated to maximise the likelihood function. Next, the

E-step updates the parameters from *M-step* to determine the new distribution.

On the other hand, bootstrapping is a mechanism used to estimate a sample distribution from original data with or without replacement. EMB works by repeatedly drawing a bootstrap with replacement from the original data *M* times, for the *M* required imputations; then, EM is run which firstly assumes a particular distribution, then initialise a mean and variance values for the missing data in each bootstrap generated. Then, the likelihood function is estimated by considering the current estimate of the model parameters (mean and covariance). Then, the parameters are updated to maximise the likelihood model. The expectation and the maximisation steps are repeated until the values converge [37].

## Ensemble Methods

An ensemble is a technique for combining models used in machine learning. It was introduced by Tukey [71] when he built an ensemble of two different regression models. Since then, it has been then broadly studied and reviewed in classification tasks [6, 19, 20, 76]. The idea of an ensemble is to induce a set of base learners (classifiers), then their predictions are aggregated in some way to obtain a better classification. This can have advantages over relying on a single model as a combined model may be more precise and accurate. Furthermore, Breiman [6] explained the usefulness of ensembles with unstable classifiers that are easily affected by changes to the training data such as decision trees and neural networks.

An ensemble can be categorised according to the underlying machine learning algorithms used into two main types: homogeneous and heterogeneous. A *homogeneous* ensemble is constructed from learners of the same type, e.g. a set of decision trees. On the other hand, when the strategy is to combine different types of learners such as decision trees, neural networks, or Bayesian networks, then we have a *heterogeneous* ensemble.

In general application, the aim of constructing an ensemble is to achieve a classification accuracy that is higher than any of the individual learner. Thus, the individual learners are expected to be accurate with an error rate better than random guess, and diverse so two classifiers make different errors when predicting a new instance. A number of methods for constructing a diverse ensemble have been developed [6, 13, 19, 27]. Below is an explanation of the most popular ensemble methods which are bagging and stacking.

### Bagging

Bagging (also known as bootstrap aggregation) is one of common ensemble methods that can be applied to classification and regression problems [19, 52]. It is used to reduce the variance between models by generating additional training sets from the original data [52]. One such method is where a proportion of data points are randomly chosen with replacement by using *bootstrap* mechanism which generates multiple training sets; each has approximately 63% of the training data points [52, 66]. Then, a same base learner (e.g. decision trees) is run in parallel on these training sets. As a result, an ensemble of different models will be generated. To make the prediction for a new data, the final decision is made by a majority vote of the individual predictions obtained from the different models [19, 52].

### Stacking

In the context of ensemble learning, meta-learning is the process of learning from the multiple learners and their outputs on the original training data. Such a method is efficient when individual classifiers misclassify the same patterns [54]. The method was introduced by Wolpert [76] and refers to a construction mechanism that uses the output of classifiers instead of the training data to build the ensemble. A stacking ensemble can be implemented in two or more layers. In the first layer, a number of base learners are trained on the entire training set then produce (level-0) models. Then, the predictions of the individual models are used as input attributes *(meta-level attributes)* to the ensemble. The target of the original training set is appended to the (meta-level attributes) to form a new set of predictions, (level-1) model. This set is used to train a meta-classifier in the ensemble. The meta-classifier can be trained based on the predicated class label or the probabilities generated from (level-0) models [67]. This model is used to estimate the final prediction in the ensemble.

## Framework for MIE

Our ensemble for MI works as follows. We first generate a series of increasing missing data under MCAR assumption. We then impute the artificial training datasets and generate five imputed datasets using two different MI techniques: MICE and EMB as described in "Imputation Methods" section. We next use these datasets to train classifiers and build our *bagging* and *stacking* ensembles. For our *bagging* ensemble, we train homogeneous classifiers (same classifiers) on the imputed datasets. We then combine the predictions of the models obtained from a separate test data using a majority vote method. This method aggregates the predictions from the individual models and chooses the class that has been predicted most frequently as the final prediction, as illustrated in Fig. 1. Therefore, the bagging ensemble is evaluated using a hold-out test set. This can be viewed as an alternative method for bagging in which multiple imputed datasets may be more dissimilar to each other hence generate

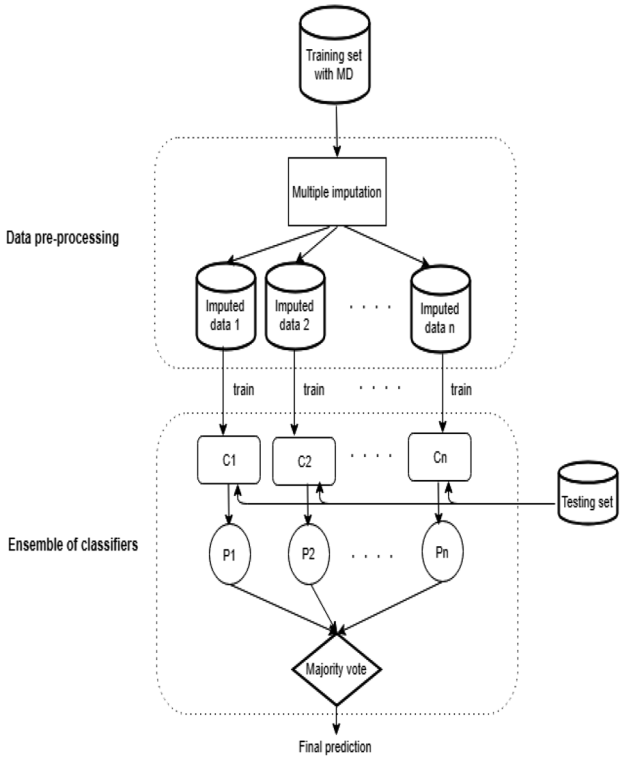


**Fig. 1** The bagging ensemble framework takes imputed datasets as inputs to train different classifiers $C_1$, ...,$C_n$ in layer 1. The predictions made by individual classifiers, $P_1$, ...,$P_n$, are combined by the majority vote method

more diverse models. On the other hand, Fig. 2 represents the construction of our *stacking* ensemble showing two layers. The first one involves the multiple imputed datasets trained by a number of learners (heterogeneous classifiers) to generate different models. The models are tested then against a separate test set to make a new dataset of predictions. This new dataset is combined with the actual class of the test set to construct the (level-1) dataset which is used as an input for the second layer. In this layer we train a meta-classifier and then we evaluate the performance of the ensemble using 10-fold cross-validation.

## Experimental Set-Up

### Datasets

For our study, a collection of 20 benchmark datasets were obtained from the UCI machine learning repository [47]. The datasets have different sizes and feature types (numerical real, numerical integer, categorical and mixed) as shown in Table 1. They were all complete datasets, that is they have no missing values, except PostOperativePatient dataset where three records with missing values have been deleted.

### Data Preparation

Before conducting experiment we solved the problem of the sparse datasets we have, LSVT and Forest Cover Type. LSVT has five attributes with zero values, so we deleted those features. On the other hand, we transform Forest Cover Type by taking the attributes that represent Wilderness_Area (4 binary columns) and Soil_Type (40 binary columns) then reducing each to a single column with multiple values. So the first new column has a numerical value of (1–4) which represents the presence of a particular area while the second indicates the soil type with a value (1–40). Additionally, we followed Clark et al. [15] mechanism of treating Abalone as a three-category classification by grouping classes 1–8, 9 and 10, and 11 so that we can improve the classification process.

Some datasets are provided with separate train and testing sets. The rest have been partitioned using *StratifiedRemoveFolds* filter in Weka to retain the class distributions with a ratio of 70% training and 30% testing, except Forest Cover Type, our largest dataset, where 60% of data is used for training and 40% for testing.

Table 2 shows the mean accuracy and the standard deviation of the classifiers (J48, NB, PART, SMO and RF) obtained on the testing sets by training on the original data with no missing values. Those classification results for the complete data are used then as the benchmarks to study how missing data affects the accuracy and performance of the algorithms when various methods for dealing with missing data are used. Note that we run RF five times with different random seeds as it obtains different results in each run given its stochastic nature. The RF classifier performs better than other classifiers in nine of the datasets. Then, SMO is the second best classifier, working best on five datasets out of twenty, while J48 and PART work best on three datasets each. The performance of NB is the worst compared to the other classifiers.

Then, we test the performance of multiple classifiers on multiple datasets using Friedman test to check which classifiers outperform others. The test shows a significant difference ($p$ value $< 0.05$) in performance so we proceed with post hoc test, Nemenyi test. The Critical Difference diagram as a result of applying Nemenyi test is shown in Fig. 3. The figure illustrates that RF behaves significantly better than PART and NB although there are no statistical differences within each group, i.e. no statistically significant between RF, SMO and J48. Similarly, the performance difference for J48, PART, NB and SMO is not statistically significant.

### Missing Data Generation

To create scenarios for testing with increasing missing data, some values in the training sets are removed completely at

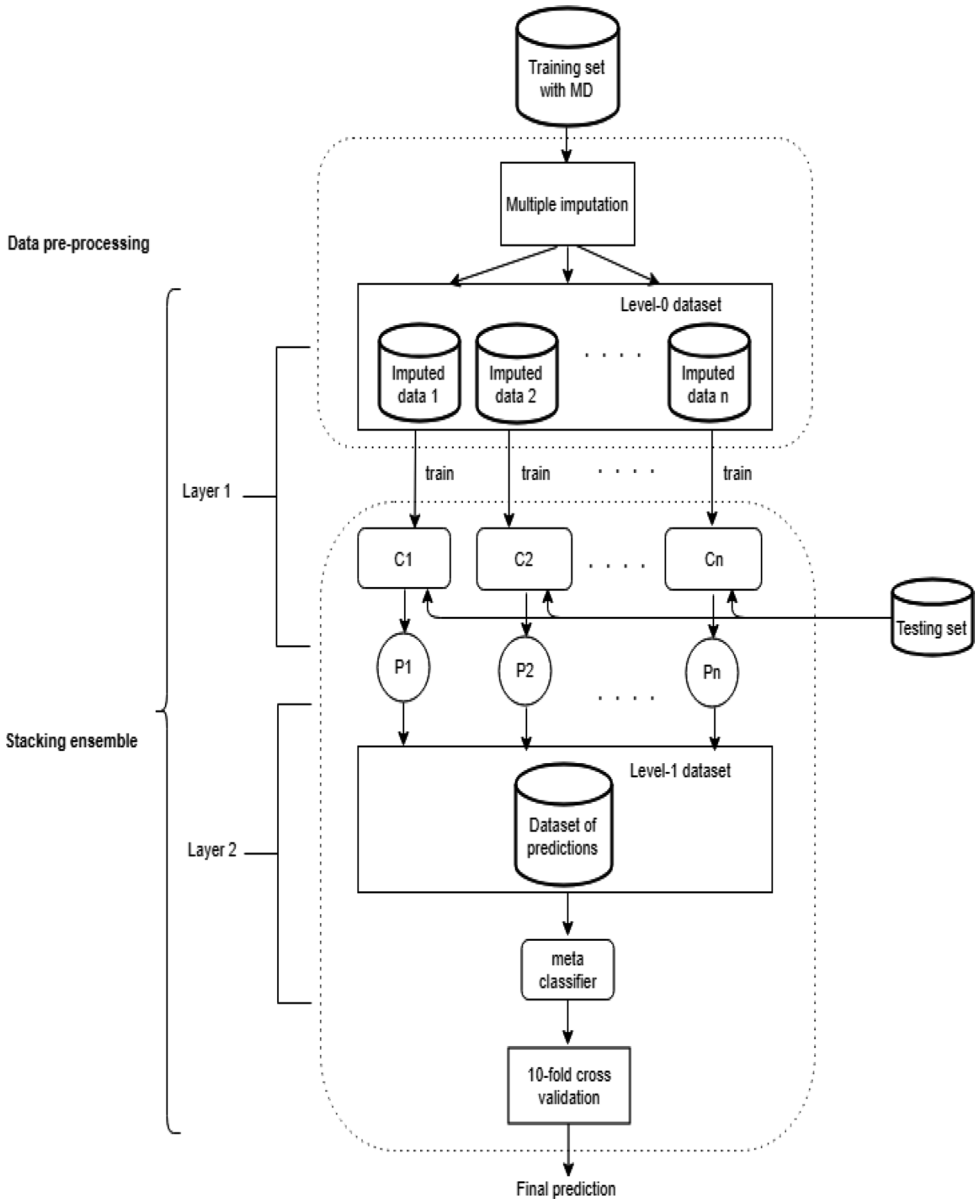


**Fig. 2** The stacking ensemble framework takes imputed datasets as inputs to train classifiers $C_1$, ...,$C_n$. The predictions made by individual classifiers, $P_1$, ...,$P_n$, are used to form a new data to be used to train a meta-classifier in the second layer

random as follows: Firstly, 10% (then 20%, 50%) of the attributes are randomly selected to remove data with the following chosen rates 5%, 15%, 30% and 50% of the records, respectively. We repeat the process of selection and removing five times so different features/records may be affected by missing data each time. As a result, 12 artificial datasets are produced from each of the original datasets each time and those have multiple levels of missing data. In total, we generate ($20 * 12 * 5 = 1260$) datasets. Table 3 summarises the experimental scenarios artificially created.

In our experiments, the models are tested on separated test data. However, for the stacking ensemble, the results reported represent tenfold cross-validation as the predictions of the separated test sets are used to construct a new dataset for the second layer of the stacking ensemble.

## Comparative Methods

### Building Models with Missing Data (MD)

In "Mechanisms for Dealing with Missing Data" section we discussed that the chosen algorithms have their own way of dealing with missing data internally. We therefore pass all the data including missing data to the algorithms without pre-processing. Such models are referred to as J48_MD, NB_MD, PART_MD, SMO_MD and RF_MD.

### Simple Imputation (SI)

To test simple imputation, the numerical attributes are replaced with their mean and the categorical attributes with

**Table 1** The details of the datasets collected for the experiments

| No. | Dataset | #Features | #Instances | #Classes | Feature types |
|---|---|---|---|---|---|
| 1 | PostOperativePatient | 8 | 87 | 2 | Integer, categorical |
| 2 | Ecoli | 8 | 336 | 8 | Real |
| 3 | Abalone | 8 | 4177 | 3 | Integer, real and categorical |
| 4 | TicTacToe | 9 | 958 | 2 | Categorical |
| 5 | BreastTissue | 10 | 106 | 6 | Real |
| 6 | Statlog | 20 | 1000 | 2 | Integer, categorical |
| 7 | Spect # | 22 | 276 | 2 | Categorical |
| 8 | Flags | 30 | 194 | 8 | Integer, categorical |
| 9 | BreastCancer | 31 | 569 | 2 | Real |
| 10 | Chess | 36 | 3196 | 2 | Categorical |
| 11 | Connect-4 | 42 | 67,557 | 2 | Categorical |
| 12 | ForestCoverType | 54 | 581,012 | 7 | Integer, categorical |
| 13 | ConnectionistBench | 60 | 208 | 2 | Real |
| 14 | HillValley # | 101 | 606 | 2 | Real |
| 15 | UrbanLandCover # | 148 | 168 | 9 | Integer, real |
| 16 | EpilepticSeizure | 179 | 11,500 | 5 | Integer, real |
| 17 | Semeion | 265 | 1593 | 2 | Integer |
| 18 | LSVT | 309 | 126 | 2 | Real |
| 19 | HAR # | 561 | 10,299 | 6 | Real |
| 20 | Isolet # | 617 | 7797 | 26 | Real |

The # symbol next to the dataset denotes that it has come with a separate test set

**Table 2** The mean accuracy of the classifiers and standard deviation for the complete datasets obtained based on test set

| Dataset | J48 | NB | PART | SMO | RF | Avg |
|---|---|---|---|---|---|---|
| PostOperativePatient | **71.43** | 64.29 | **71.43** | 60.71 | 64.29 (0.00) | 66.43 (0.00) |
| Ecoli | 82.14 | **83.04** | 82.13 | **83.04** | 80.89 (1.20) | 82.25 (0.24) |
| Abalone | 63.43 | 58.91 | 62.21 | 65.23 | **65.36** (0.71) | 63.028 (0.14) |
| TicTacToe | 84.33 | 72.73 | 88.09 | **98.75** | 95.17 (1.14) | 87.81 (0.23) |
| BreastTissue | **65.71** | 57.14 | 62.86 | 57.14 | 64.57 (1.56) | 61.48 (0.31) |
| Statlog | 72.37 | **76.28** | 69.97 | 76.27 | 74.95 (1.22) | 73.97 (0.24) |
| Spect | 66.84 | 64.71 | 65.24 | 67.91 | **69.95** (1.48) | 66.93 (0.30) |
| Flags | 57.81 | 48.44 | 53.13 | 35.94 | **60.00** (2.61) | 51.06 (0.52) |
| BreastCancer | 95.24 | 93.65 | 92.06 | **97.88** | 95.98 (0.80) | 94.96 (0.16) |
| Chess | **99.25** | 88.08 | 98.97 | 95.40 | 99.06 (0.16) | 96.15 (0.03) |
| Connect-4 | 79.31 | 72.11 | 78.39 | 76.06 | **81.94** (0.07) | 77.56 (0.01) |
| ForestCoverType | 93.71 | 62.16 | 91.4 | 71.70 | **96.58** (0.02) | 83.11 (0.01) |
| ConnectionistBench | 72.46 | 69.57 | 65.22 | 81.15 | **84.35** (1.89) | 74.55 (0.38) |
| HillValley | 48.15 | 51.44 | 48.15 | 53.08 | **56.71** (1.80) | 51.51 (0.36) |
| UrbanLandCover | 67.65 | 77.91 | 69.83 | 74.56 | **81.30** (0.53) | 74.25 (0.11) |
| EpilepticSeizure | 48.53 | 43.33 | 49.62 | 27.52 | **68.17** (0.40) | 47.43 (0.08) |
| Semeion | 92.84 | 91.34 | **98.49** | 97.74 | 95.40 (0.10) | 95.16 (0.02) |
| LSVT | 66.67 | 51.19 | **97.62** | 76.19 | 85.71 (1.69) | 75.48 (0.34) |
| HAR | 93.85 | 75.85 | 94.47 | **98.31** | 97.95 (0.20) | 92.09 (0.04) |
| Isolet | 83.45 | 82.36 | 82.81 | **95.83** | 93.97 (0.31) | 87.68 (0.06) |

Best accuracy values for each dataset are in bold

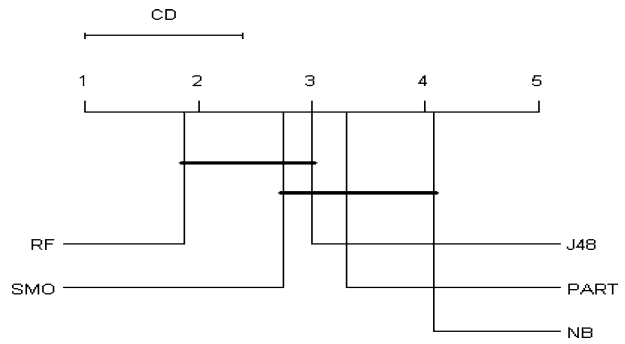


**Fig. 3** Critical Difference diagram showing statistically significant differences between classifiers. The bold line connecting classifiers means that they are not statistically different

**Table 3** Experimental scenarios with missing data artificially created

| Scenario | %Features selected | %Records affected by MD |
|---|---|---|
| Sce1 | 10 | 5 |
| Sce2 | | 15 |
| Sce3 | | 30 |
| Sce4 | | 50 |
| Sce5 | 20 | 5 |
| Sce6 | | 15 |
| Sce7 | | 30 |
| Sce8 | | 50 |
| Sce9 | 50 | 5 |
| Sce10 | | 15 |
| Sce11 | | 30 |
| Sce12 | | 50 |

their mode. Then, the produced datasets after imputation are used for classification model building. In our results the models with single imputation are referred to as J48_SI, NB_SI, PART_SI, SMO_SI and RF_SI.

### Random Forest Imputation (RFI)

We use a RF imputation package (missForest) implemented in *R* to replace missing values using a RF algorithm. The algorithm starts with filling incomplete data by median if they are numeric or mode if they are categorical. Then, it updates missing values by using proximity from random forest and iterates the imputation a number of times. Finally, the imputed value for an attribute with missing values is the weighted average of non-missing values if it is numeric or the mode if it is nominal. We set up the number of iterations to perform the imputation to 5 and the number of trees that grow in each forest to 300. In our results the models that used RFI are referred to as J48_RFI, NB_RFI, PART_RFI, SMO_RFI and RF_RFI.

## Proposed MIE Methods

The following steps have been undertaken to test the bagging and stacking ensemble. First, MICE and Amelia packages in *R*, which implement Multivariate Imputation with Chained Equation and Expectation–Maximisation with Bootstrap algorithms, respectively, are applied to generate five imputed datasets. For MICE, we set the predictive mean match as the imputation method, the number of iterations to perform the imputation to 20 and the number of the imputed datasets to 5. For Amelia, we used 5 as the number of imputations, too. Additionally, we perform the imputation in parallel when processing large and high dimensional datasets.

The multiple imputed datasets are used as inputs to train the classifiers and for our bagging ensemble, we aggregate the predictions obtained by the models by the majority vote method. One base learner is used as the classifier. Our classifiers of choice for the bagging ensemble are J48 [51], NB [45], PART [26], SMO [60] and RF [43] (as implemented in *Weka*) with their default options for classifying the data. On our results, such models are referred to as MICE_Hom and EMB_Hom, depending on the method to perform the imputation in the first place.

A second ensemble approach tested is to build a stacking ensemble, where the training datasets are used to perform the imputation then to train all chosen classifiers to generate several models. These models are used as inputs to the first layer of the stack. The predictions of the different models on the testing set are used to form a new dataset for level-1 in the ensemble. Then, we train a meta-classifier in this new dataset in the second layer. For testing the stacking ensemble, we perform 10-fold cross-validation to evaluate the performance of models. These are referred to as MICE_SE and EMB_SE depending on the MI method.

## Evaluating Classification Methods

We use the classification accuracy as the metric for our comparisons of performance. We compare between all the approaches looking for differences in the algorithms' performance on each scenario separately. We perform Wilcoxon signed-rank test with Finner's procedure for correcting the $p$ values for pairwise comparison testing with a significance level of $\alpha = 0.05$ [18, 29] with a number of controls separately as follows:

We perform two different statistical tests when evaluating the performance of classifiers over the datasets as follows:

1. When comparing multiple classifiers over multiple datasets, we use the method described by Demšar [18],

including the Friedman test and the post hoc Nemenyi test which is represented as a Critical Difference (CD) diagram, with a significance level of $\alpha = 0.05$.

2. The Wilcoxon signed-rank test is applied for performing pairwise comparison. We use a control algorithm, the performance of the classifier on SI, with a significance level at $\alpha = 0.05$.

### Evaluating Imputation Methods

Numerous statistical methods are used to check the variation between and within the multiple imputed datasets [1, 3, 8, 35]. These involve graphical representations such as histogram, density and quantile–quantile plots [1]. Others suggest the use of numerical comparisons such as means and standard deviations [35]. In this study instead we propose the use of dissimilarity, as used in the context of clustering algorithms [9, 9, 33], to evaluate the quality of the imputation methods. *Dissimilarity* is a numeric measurement of the degree of difference between data points. We check the quality of the imputation by comparing each imputed data point with its original counterpart. In that way we can measure the dissimilarity between each pair of points (original/imputed). We can then aggregate dissimilarity across the whole dataset to arrive at a measure of quality of the imputation, with imputations that produce points closer to the original being considered better than those where the dissimilarity is greater.

As we have different data types in our datasets, i.e. numeric, categorical, mixed, we use the weighted overall dissimilarity formula proposed by Gower [33], the *Gower Coefficient*, to compute the distance *dis(a,b)* between each data point, $a$, in the original dataset and the corresponding data point, $b$, in the artificial dataset after performing MI as follows:

$$dis(a,b) = \frac{\sum_{f=1}^{N} w_{ab}^{(f)} dis_{ab}^{(f)}}{\sum_{f=1}^{N} w_{ab}^{(f)}} \tag{1}$$

where $N$ denotes the total number of features in a dataset, $w$ is the assigned weight to a feature (we set $w$=1 for each feature) and $f$ is a feature which can be either numerical or categorical.

Before we apply the formula to measure distance, we standardise each numerical attribute, $f$, into a comparable range using the standardised measure, $z\_score$, to avoid attributes with a larger range having a bigger effect on the distance measurement. For this we use the following equation:

$$x^{'} = z(x_f) = (x_f - m_f)/s_f \tag{2}$$

where $x$ denotes a value in an attribute, $m$ the mean of attribute and $s$ is the mean absolute deviation for that attribute. Then, we compute the distance, $dis_{ab}^{(f)}$, as follows:

$$dis_{ab}^{(f)} = \left| x_{af}^{'} - x_{bf}^{'} \right| \tag{3}$$

The contribution of each categorical attribute to the overall dissimilarity $dis_{ab}^{(f)} = 0$ if $x_{af}$ and $x_{bf}$ are identical otherwise $dis_{ab}^{(f)} = 1$.

The overall aggregated dissimilarity function remains in the same range [0,1]. Finally, we average the distance of all records to obtain the mean distance between the original and imputed data.

## Results

In order to understand how different algorithms behave under different imputation regimes, we began by investigating each algorithm separately. In particular, we applied our proposed methods that combine MI with ensemble techniques, MIE, along with the comparative approaches as described in "Comparative Methods" section. We therefore study the performance of the internal mechanism of the algorithms for handling missing data (e.g. for J48, J48_MD, NB_MD, PART_MD, SMO_MD and RF_MD), the simple imputation (J48_SI, NB_SI, PART_SI, SMO_SI and RF_SI), the RF imputation (J48_RFI, NB_RFI, PART_RFI, SMO_RFI and RF_RFI). Our proposed MIE methods are represented by the combination between MI methods with bagging (MICE_Hom, EMB_Hom) and stacking ensembles (MICE_SE and EMB_SE). The details of the results can be found from the authors.

### Classification Performance

For each of the classifier/imputation methods studied, we applied the Friedman statistical test [18] to compare the performance of the imputation methods including our proposed approach. The test compares the mean ranks of the classifiers on a number of datasets as follows: with 7 algorithms (i.e. variations on imputation regimes) and 20 datasets, $F$ it is distributed according to the $F$ distribution with $7 - 1 = 6$ and $(7 - 1) * (20 - 1) = 114$ degrees of freedom. If we use a significance level of $\alpha = 0.05$, the critical value of $F$ is 2.18.

For the J48 algorithm, Table 4 summarises the mean rank for the different imputation/ensemble methods on each of the artificial datasets in each scenario separately. Hint: lowest rank means better performance. On average, for J48 the stacking ensemble with EMB (EMB_SE) obtained a better rank hence better overall classification accuracy, with MICE_SE second best. J48_SI was the worst.

**Table 4** The mean rank for J48 on different imputation methods along with proposed approach on all dataset affected by missing data in all scenarios

| Scenario | J48_MD | J48_SI | J48_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 5.08 | 5.38 | 4.13 | 4.88 | 3.60 | 2.50 | **2.45** |
| 2 | 5.35 | 5.30 | 4.25 | 4.50 | 3.78 | 2.60 | **2.23** |
| 3 | 5.58 | 5.58 | 4.10 | 4.18 | 3.93 | **2.10** | 2.55 |
| 4 | 4.93 | 5.53 | 4.68 | 4.50 | 3.70 | **2.33** | 2.35 |
| 5 | 5.38 | 5.60 | 3.98 | 4.53 | 3.68 | 2.45 | **2.40** |
| 6 | 5.25 | 5.68 | 4.43 | 4.10 | 3.55 | **2.48** | 2.53 |
| 7 | 5.35 | 5.20 | 5.05 | 4.15 | 3.48 | 2.40 | **2.38** |
| 8 | 5.20 | 5.53 | 4.58 | 4.80 | 3.05 | 2.48 | **2.38** |
| 9 | 4.93 | 5.33 | 4.80 | 4.75 | 3.60 | 2.48 | **2.13** |
| 10 | 5.30 | 5.13 | 4.70 | 4.50 | 3.43 | **2.23** | 2.73 |
| 11 | 5.00 | 5.33 | 4.58 | 4.50 | 2.90 | **2.63** | 3.08 |
| 12 | 4.88 | 5.65 | 4.15 | 4.20 | 3.13 | 3.30 | **2.70** |
| Avg rank | 5.19 | 5.44 | 4.45 | 4.47 | 3.49 | 2.50 | **2.49** |

The value in bold indicates that the algorithm performs better than others

**Table 5** The mean rank of NB in combination with different imputation methods and of our proposed approach on all dataset affected by missing data for different scenarios

| Scenario | NB_MD | NB_SI | NB_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 4.40 | 5.05 | 4.98 | 4.98 | 5.23 | 1.73 | **1.65** |
| 2 | 4.55 | 4.73 | 5.05 | 4.95 | 5.40 | 1.88 | **1.45** |
| 3 | 4.63 | 4.88 | 4.93 | 5.05 | 5.05 | **1.63** | 1.85 |
| 4 | 4.60 | 5.00 | 5.40 | 4.38 | 5.18 | **1.60** | 1.85 |
| 5 | 4.55 | 4.95 | 5.40 | 4.93 | 4.68 | 1.88 | **1.63** |
| 6 | 4.68 | 4.63 | 5.15 | 5.20 | 4.93 | 1.75 | **1.68** |
| 7 | 4.63 | 4.63 | 5.18 | 5.03 | 5.20 | 1.80 | **1.55** |
| 8 | 4.80 | 4.63 | 5.20 | 4.90 | 4.83 | 1.85 | **1.80** |
| 9 | 4.18 | 4.80 | 5.28 | 4.85 | 5.00 | 2.05 | **1.85** |
| 10 | 4.48 | 5.15 | 5.25 | 4.53 | 4.95 | 2.08 | **1.58** |
| 11 | 3.90 | 5.20 | 5.55 | 4.90 | 5.03 | 1.73 | **1.70** |
| 12 | 4.08 | 5.18 | 4.90 | 4.63 | 4.85 | **2.08** | 2.30 |
| Avg rank | 4.45 | 4.90 | 5.19 | 4.86 | 5.03 | 1.84 | **1.74** |

The value in bold indicates that the algorithm performs better than others

**Table 6** The mean rank of PART in combination with different imputation methods and of our proposed approach on all dataset affected by missing data for different scenarios

| Scenario | PART_MD | PART_SI | PART_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 4.48 | 5.45 | 4.78 | 4.63 | 3.65 | **2.45** | 2.58 |
| 2 | 4.43 | 5.58 | 4.78 | 4.58 | 3.35 | 2.93 | **2.38** |
| 3 | 4.80 | 5.43 | 4.78 | 4.35 | 3.33 | 2.88 | **2.45** |
| 4 | 4.85 | 5.55 | 4.85 | 4.28 | 3.65 | **2.40** | 2.43 |
| 5 | 4.30 | 5.43 | 4.75 | 5.40 | 3.10 | 2.58 | **2.45** |
| 6 | 4.73 | 5.48 | 4.55 | 4.73 | 3.28 | 2.70 | **2.55** |
| 7 | 4.75 | 5.13 | 4.70 | 4.90 | 3.03 | 3.13 | **2.38** |
| 8 | 4.35 | 5.93 | 5.03 | 4.23 | 2.93 | 2.95 | **2.60** |
| 9 | 4.98 | 5.53 | 5.15 | 4.58 | 3.10 | 2.73 | **1.95** |
| 10 | 4.63 | 5.50 | 5.00 | 5.00 | 3.03 | 2.50 | **2.35** |
| 11 | 5.28 | 5.70 | 4.85 | 4.25 | 2.60 | 2.75 | **2.58** |
| 12 | 4.38 | 5.75 | 4.88 | 4.30 | 3.18 | 2.93 | **2.60** |
| Avg rank | 4.66 | 5.54 | 4.84 | 4.60 | 3.18 | 2.74 | **2.44** |

The value in bold indicates that the algorithm performs better than others

**Table 7** The mean rank of SMO in combination with different imputation methods and of our proposed approach on all dataset affected by missing data for different scenarios

| Scenario | SMO_MD | SMO_SI | SMO_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 4.68 | 5.00 | 3.95 | 4.30 | 3.75 | 3.53 | **2.80** |
| 2 | 4.83 | 5.10 | 4.05 | 3.78 | 4.00 | 3.30 | **2.95** |
| 3 | 4.60 | 5.15 | 3.63 | 4.63 | 3.83 | 3.40 | **2.78** |
| 4 | 4.80 | 5.15 | 3.48 | 4.43 | 3.60 | **3.18** | 3.38 |
| 5 | 4.20 | 4.85 | 3.90 | 3.80 | 4.30 | **3.25** | 3.70 |
| 6 | 4.80 | 5.05 | 4.05 | 3.75 | 4.08 | **3.00** | 3.28 |
| 7 | 4.93 | 5.65 | 3.65 | 4.05 | 3.78 | 3.13 | **2.83** |
| 8 | 4.90 | 5.08 | 3.48 | 4.00 | 3.75 | 3.53 | **3.28** |
| 9 | 4.85 | 5.03 | 3.93 | 3.90 | 4.03 | 3.45 | **2.83** |
| 10 | 4.85 | 4.98 | 4.00 | 3.75 | 4.10 | 3.08 | **3.25** |
| 11 | 4.38 | 4.93 | **2.95** | 4.58 | 4.03 | 3.80 | 3.35 |
| 12 | 5.03 | 5.25 | **2.93** | 3.93 | 3.83 | 3.48 | 3.58 |
| Avg rank | 4.74 | 5.10 | 3.66 | 4.07 | 3.92 | 3.34 | **3.16** |

The value in bold indicates that the algorithm performs better than others

Similar results were obtained for the NB, PART and SMO algorithms as illustrated in Tables 5, 6 and 7. In each case the EMB_SE algorithm produced the best performance in terms of ranking and hence overall accuracy on different datasets. For both NB and PART, MICE_SE was second best. However, for SMO in Table 7, RFI was a close match to MICE_SE. For RF, shown in Table 8 EMB_Hom was the best in most scenarios, whereas the internal mechanism of RF for handling MD showed worse performance than others in most cases.

So far we have used the Friedman test to compute the average ranks. The test also gives us the ability to compute a $p$ value, to discern if the algorithm performs significantly different to others according to the average rank obtained. Table 4 presents the $p$ values resulting from application of the Friedman test for each scenario and each algorithm and shows that the performance of the different imputation methods when combined with a given classifier were significantly different. The symbol $*$ denotes that the test was significant $p < 0.05$. J48, NB and PART were statistically different when different imputation methods were applied in all scenarios tested. The performance of SMO was significant in most cases while RF was the same in half of the cases.

We also used the Wilcoxon signed-rank test for pairwise comparison with a control algorithm. This test computes the median (not average) accuracy among all datasets. We chose the performance of the classifiers applied to data imputed by SI as a control as this is a form of naive imputation which may be frequently used and is often used as a control against new data imputation methods (Table 9). MI combined with an ensemble in the case of the J48 algorithm (i.e. EMB_Hom, MICE_SE and EMB_SE) was statistically significant better than the control in all cases as shown in Table 10. On

**Table 8** The mean rank of RF in combination with different imputation methods and of our proposed approach on all dataset affected by missing data for different scenarios

| Scenario | RF_MD | RF_SI | RF_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 5.10 | 4.65 | 3.58 | 3.90 | **3.20** | 4.00 | 3.58 |
| 2 | 4.95 | 4.78 | 3.85 | 4.10 | **2.93** | 4.12 | 3.23 |
| 3 | 4.88 | 4.35 | 3.60 | 4.5 | **3.03** | 3.63 | 4.03 |
| 4 | 4.65 | 4.80 | 3.75 | 4.55 | **2.85** | 4.00 | 3.40 |
| 5 | 4.63 | 4.40 | 3.43 | **3.28** | 4.70 | 3.88 | 3.70 |
| 6 | 5.38 | 4.45 | 3.58 | 4.08 | **3.05** | 3.45 | 4.03 |
| 7 | 5.10 | 4.33 | 3.68 | 4.05 | **3.00** | 4.15 | 3.70 |
| 8 | 4.55 | 4.23 | 3.45 | 4.23 | **2.88** | 4.33 | 4.35 |
| 9 | 5.63 | 4.90 | 3.70 | 3.63 | **3.03** | 4.03 | 3.10 |
| 10 | 5.08 | 4.48 | 4.03 | 3.45 | **3.25** | 3.93 | 3.80 |
| 11 | 4.88 | 4.60 | 3.08 | 3.83 | **2.90** | 4.15 | 4.58 |
| 12 | 5.25 | 4.20 | **2.85** | 3.93 | 3.03 | 4.30 | 4.45 |
| Avg rank | 5.00 | 4.51 | 3.55 | 3.96 | **3.15** | 4.00 | 3.83 |

The value in bold indicates that an algorithm performs better than others

**Table 9** The $p$ values resulting from the Friedman test for comparing the performance of each classifier with different imputation methods separately

| Scenario | *Classifiers* | | | | |
|---|---|---|---|---|---|
| | J48 | NB | PART | SMO | RF |
| 1 | 6.33E−07* | 7.03E−13* | 5.34E−06* | 0.03* | 0.07 |
| 2 | 3.76E−07* | 5.50E−13* | 5.88E−06* | 0.02* | 0.03* |
| 3 | 1.34E−08* | 9.59E−12* | 1.21E−05* | 0.01* | 0.11 |
| 4 | 1.71E−07* | 3.00E−12* | 5.79E−07* | 0.01* | 0.04* |
| 5 | 7.05E−08* | 4.37E−12* | 8.65E−08* | 0.34 | 0.19 |
| 6 | 3.59E−07* | 4.00E−12* | 6.29E−06* | 0.02* | 0.02* |
| 7 | 8.73E−08* | 1.59E−12* | 1.33E−05* | 0.00* | 0.09 |
| 8 | 2.46E−08* | 1.24E−10* | 5.22E−07* | 0.04* | 0.15 |
| 9 | 6.39E−08* | 6.13E−10* | 3.54E−09* | 0.02* | 0.00* |
| 10 | 1.06E−06* | 3.58E−11* | 3.76E−08* | 0.03* | 0.13 |
| 11 | 1.74E−05* | 4.04E−13* | 1.82E−08* | 0.05 | 0.02* |
| 12 | 8.55E−05* | 1.37E−07* | 6.46E−06* | 0.01* | 0.01* |

The symbol (*) shows that the performance of the classifiers are statistically different when applying different imputation/ensemble methods

the other hand, MICE_HOM models and J48_RFI were significantly different in a few scenarios. The internal mechanism of J48 for handling MD was not different than SI.

For the NB algorithm, the results are shown in Table 11. We can see that only the combination between MI and stacking (MICE_SE and EMB_SE) performed statistically different from the control while other methods showed no difference.

For PART, as shown in Table 12, the combination between MI with ensembles (EMB_Hom, MICE_SE and EMB_SE) was better than the control in all cases. On the other hand, MICE_Hom, PART_RFI and the internal method were significantly different from the control in a few scenarios.

For SMO, performance for the EMB_SE approach is better in most but not all cases and similarly for MICE_SE, as illustrated in Table 13. SMO_RFI was better than the control only when the ratio of missingness increases. The EMB_Hom method was significantly better than the control when low missing values were encountered.

For RF, Table 14 presents the comparison with the control and shows some improvements when EMB_Hom was used. For all other approaches to missing data there appears to be little difference.

## Quality of the Imputed Data

Here we first evaluate the quality of imputation methods used, i.e. how far is the imputed data from the real data. We used the normalised Euclidean distance as explained in "Evaluating Imputation Methods" section to compute the mean dissimilarity between the imputed and the original data. We divide our analysis by the feature type (i.e. numerical, categorical or mixed) as imputation may work differently for different data types. The number of datasets in each group is 10, 5 and 5, respectively.

The three plots at the top of Fig. 4 represent the mean dissimilarity between the real and the imputed values using EMB, MICE, RFI and SI with respect to the numerical datasets. In most of the scenarios RFI produced imputed data closer to the real data as the mean dissimilarity was very close to 0. EMB was a close match to RFI followed by MICE. However, imputed data by SI was the worst as it was further from the real data specially with increasing uncertainty. With respect to the categorical data, the plots in the middle of the figure show that the mean dissimilarity for all imputation methods devised was close to each other and to the real data though EMB produced the best performance for most scenarios. In the case of categorical data, all methods tested were efficient in terms of recovering missing values.

**Table 10** The median accuracy for J48 on different imputation/ensemble methods along with proposed approach resulting from Wilcoxon signed-rank test

| Scenario | J48_MD | J48_SI | J48_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 75.32 | 75.30 | 75.43 | 75.57 | 76.07* | 81.05* | 81.31* |
| 2 | 75.24 | 75.24 | 75.69* | 75.69* | 76.11* | 80.65* | 81.34* |
| 3 | 74.90 | 74.95 | 75.40* | 75.40* | 75.98* | 81.01* | 80.67* |
| 4 | 75.03 | 74.86 | 75.08* | 75.20* | 75.86* | 80.34* | 80.21* |
| 5 | 75.19 | 75.13 | 75.64* | 75.78* | 76.39* | 80.86* | 80.85* |
| 6 | 75.39 | 75.11 | 75.60* | 75.64* | 76.28* | 80.61* | 80.44* |
| 7 | 75.00 | 75.10 | 75.37 | 75.54 | 76.25* | 80.15* | 80.65* |
| 8 | 74.54 | 73.93 | 74.54* | 74.27 | 76.38* | 79.71* | 79.44* |
| 9 | 75.63 | 75.21 | 75.31 | 75.34 | 76.70* | 81.08* | 81.09* |
| 10 | 74.82 | 75.36 | 75.14 | 75.33 | 76.92* | 80.74* | 80.28* |
| 11 | 74.31 | 73.84 | 74.91* | 74.41 | 76.63* | 78.78* | 78.33* |
| 12 | 73.04 | 71.92 | 73.50* | 73.74* | 75.63* | 77.06* | 77.66* |

The symbol (*) indicates that the algorithm performs better than the control

**Table 11** The median accuracy for NB on different imputation/ensemble methods along with proposed approach resulting from Wilcoxon signed-rank test

| Scenario | NB_MD | NB_SI | NB_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 69.91 | 69.94 | 69.77 | 69.91 | 69.76 | 81.80* | 81.84* |
| 2 | 69.99 | 70.0 | 69.80 | 69.92 | 69.83 | 81.83* | 82.12* |
| 3 | 70.09 | 70.07 | 69.78 | 69.88 | 69.93 | 81.42* | 81.58* |
| 4 | 69.74 | 69.92 | 69.54 | 69.67 | 69.78 | 81.15* | 81.32* |
| 5 | 69.89 | 70.04 | 69.56 | 69.96 | 69.92 | 81.16* | 81.62* |
| 6 | 69.90 | 70.06 | 69.50 | 69.77 | 69.90 | 81.47* | 81.53* |
| 7 | 70.07 | 70.14 | 69.62 | 69.85 | 69.91 | 81.22* | 81.47* |
| 8 | 69.66 | 69.94 | 69.35 | 69.53 | 69.74 | 80.41* | 80.11* |
| 9 | 70.13 | 70.31 | 69.37 | 70.15 | 70.013 | 81.04* | 81.36* |
| 10 | 70.03 | 69.91 | 69.18 | 70.31 | 69.79 | 81.35* | 81.33* |
| 11 | 70.25 | 69.73 | 69.07 | 69.83 | 69.62 | 79.86* | 79.44* |
| 12 | 70.25 | 69.07 | 69.09 | 69.54 | 69.49 | 78.11* | 78.20* |

The symbol (*) indicates that the algorithm performs better than the control

**Table 12** The median accuracy for PART on different imputation/ensemble methods along with proposed approach resulting from Wilcoxon signed-rank test

| Scenario | PART_MD | PART_SI | PART_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 75.89 | 75.51 | 75.56 | 75.46 | 77.11* | 80.92* | 81.03* |
| 2 | 75.68* | 75.14 | 75.60* | 75.35 | 77.18* | 79.81* | 80.80* |
| 3 | 75.62 | 74.99 | 75.34 | 75.61 | 77.34* | 79.75* | 80.08* |
| 4 | 75.29 | 74.93 | 74.84 | 75.44 | 76.64* | 79.86* | 80.54* |
| 5 | 75.98* | 75.40 | 75.64 | 75.41 | 77.53* | 80.27* | 80.70* |
| 6 | 75.55 | 75.09 | 75.65 | 75.34 | 77.26* | 80.15* | 80.54* |
| 7 | 75.20 | 75.21 | 75.36 | 74.66 | 77.68* | 79.42* | 80.38* |
| 8 | 75.19* | 73.40 | 74.24* | 74.74* | 77.23* | 78.32* | 79.15* |
| 9 | 75.58 | 75.23 | 75.11 | 75.35 | 77.65* | 80.33* | 81.02* |
| 10 | 75.28* | 74.80 | 74.92 | 74.69 | 77.55* | 79.81* | 80.22* |
| 11 | 74.21* | 73.15 | 74.20* | 75.00* | 77.75* | 77.98* | 78.40* |
| 12 | 73.85* | 71.70 | 72.85* | 72.57 | 75.82* | 76.45* | 77.57* |

The symbol (*) indicates that the algorithm performs better than the control

**Table 13** The median accuracy for SMO on different imputation/ensemble methods along with proposed approach resulting from Wilcoxon signed-rank test

| Scenario | SMO_MD | SMO_SI | SMO_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 76.12 | 75.61 | 75.93 | 76.52 | 76.36 | 81.37* | 81.61* |
| 2 | 75.75 | 75.62 | 75.74 | 76.27 | 76.12 | 81.55* | 81.75* |
| 3 | 75.66 | 75.54 | 76.27* | 76.24 | 76.32* | 81.11* | 81.22* |
| 4 | 75.46 | 75.40 | 76.38* | 75.50 | 76.33* | 80.95* | 80.82* |
| 5 | 76.23 | 75.99 | 75.61 | 76.35 | 76.26 | 80.93 | 81.25 |
| 6 | 75.71 | 75.79 | 76.01 | 76.14 | 75.85 | 81.12** | 81.10* |
| 7 | 75.28 | 75.04 | 75.84* | 76.01 | 76.08* | 80.94* | 81.13* |
| 8 | 74.99 | 75.00 | 75.51 | 75.94 | 75.44 | 79.75 | 79.95* |
| 9 | 76.14 | 76.00 | 75.94 | 76.46 | 76.44 | 81.16* | 81.50* |
| 10 | 75.21 | 75.16 | 75.17 | 76.41 | 75.42 | 81.17* | 80.93* |
| 11 | 74.89 | 74.61 | 75.43 | 74.93 | 74.28 | 79.23 | 79.16* |
| 12 | 73.27 | 73.26 | 75.20* | 73.19 | 73.67 | 77.81* | 78.23 |

The symbol (*) indicates that the algorithm performs better than the control

**Table 14** The median accuracy for RF on different imputation/ensemble methods along with proposed approach resulting from Wilcoxon signed-rank test

| Scenario | RF_MD | RF_SI | RF_RFI | MICE_Hom | EMB_Hom | MICE_SE | EMB_SE |
|---|---|---|---|---|---|---|---|
| 1 | 80.42 | 80.63 | 81.33* | 81.25 | 81.45* | 80.87 | 81.35 |
| 2 | 80.90 | 80.71 | 81.04 | 81.07 | 81.50* | 80.86 | 81.43 |
| 3 | 80.42 | 80.54 | 80.74 | 80.53 | 81.29 | 80.98 | 80.70 |
| 4 | 80.04 | 79.99 | 80.45 | 80.45 | 80.88* | 80.26 | 80.74 |
| 5 | 80.04 | 79.99 | 80.45 | 80.45 | 80.88 | 80.26 | 80.74 |
| 6 | 80.67 | 80.92 | 81.18 | 81.10 | 81.39 | 81.21 | 81.01 |
| 7 | 79.80 | 80.30 | 80.78 | 80.80 | 81.02 | 80.15 | 80.65 |
| 8 | 79.49 | 79.79 | 80.12 | 79.68 | 80.58* | 79.50 | 79.28 |
| 9 | 80.37 | 80.45 | 80.88 | 81.13 | 81.43* | 80.57 | 81.15 |
| 10 | 80.28 | 80.30 | 80.59 | 80.97 | 81.31 | 80.70 | 80.68 |
| 11 | 79.13 | 79.29 | 79.97 | 79.54 | 80.38* | 79.00 | 78.36 |
| 12 | 77.91 | 78.38 | 78.91 | 78.55 | 79.11 | 77.26 | 77.52 |

The symbol (*) indicates that the algorithm performs better than the control

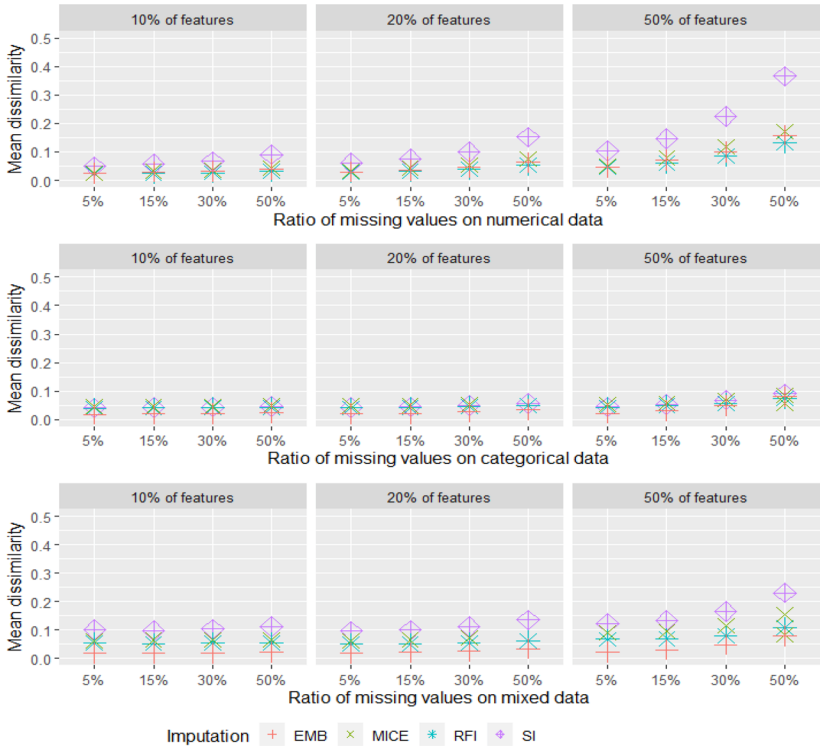


**Fig. 4** The mean dissimilarity between the original and imputed data points as a result of applying different imputation methods on numerical datasets where different percentage of features affected by missing data at different levels. The first row represents numerical data, the second represents categorical data, and the last represents mixed data

On the other hand, different imputation methods behave differently with the mixed data type as shown at the bottom of the figure EMB produced data that was more similar to the real data as the mean dissimilarity did not exceed 0.1 in the worst case. RFI became second best followed by MICE. Again the SI was the worst in all cases.

## Classification Results by Datatype

Finally, we analyse the efficiency of the imputation method based on different data types and we relate this to the performance of different classifiers. We do this separately for each classification algorithm. We present box plots showing the range of accuracies (max, min, median and any outliers) obtained for all the datasets of a given data type. The different scenarios in terms of % of missing data are represented in the x-axis, though we combined the results from the missing data affecting 10%, 20% and 50% features in one box plot as the same patterns were observed for each. The grey box plot in each graph represents the accuracy on the complete dataset, before any data is removed.

Figure 5 shows the range of accuracies as box plots for the J48 algorithm applied on numerical (left), categorical (centre) and mixed (right) datasets. On numerical datasets

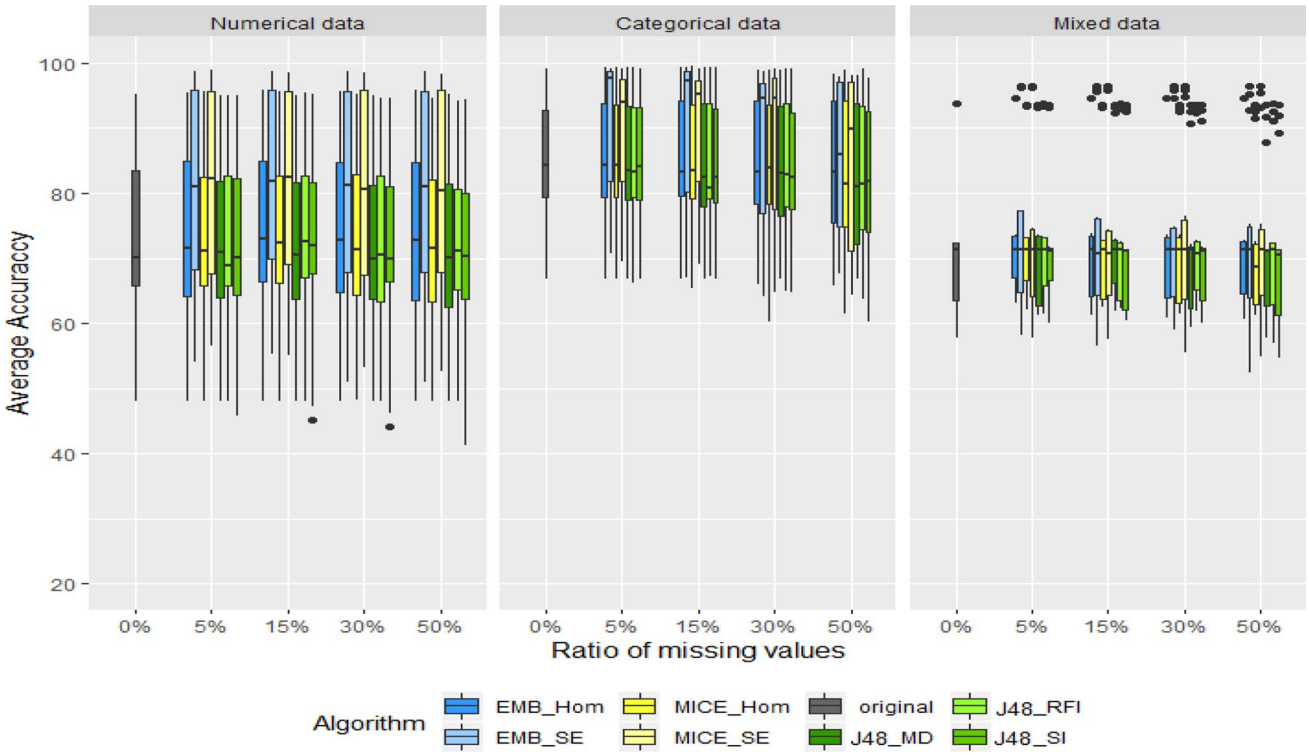


**Fig. 5** This figure contains box plots describing the overall average accuracy of J48 applied on imputed datasets using different imputation approaches along with the average accuracy on the complete data

(the plot to the left), there are two clear methods that stand out as the median value for EMB_SE and MICE_SE was much higher than that of other methods and of the complete data for all levels of missing data. Also, the maximum accuracy of both EMB_SE and MICE_SE increased by about 10% compared with the complete data also for all levels of missing data. The EMB_Hom method also shows some improved performance though not so marked. The other methods perform similarly to one another and to the complete data. For the categorical data (centre plot), a similar pattern for median accuracy is observed with EMB_SE and MICE_SE showing best median performance, with some but not so marked improvement for maximum accuracy too. Overall median accuracy of most approaches decreased with increasing uncertainty but for EMB_SE and MICE_SE it was both higher than the complete data and that the other methods for most scenarios. On mixed datasets (right plot), the median accuracy of all approaches seemed to be similar to the complete data but the maximum average accuracy increased when applying MICE_SE and EMB_SE. Outliers, represented by dots in the plot, were presented in all methods tested for mixed data.

For the NB algorithm, similar results are shown in Fig. 6. However, for NB, MICE_SE and EMB_SE show performance improvements both in terms of maximum and medium average accuracies with respect to the mixed datasets.

For PART, shown in Fig. 7 the median accuracy of PART_MD, PART_SI, PART_RFI, MICE_Hom and EMB_Hom deteriorated for numerical datasets in comparison with the original data while the performance improves when MICE_SE and EMB_SE are applied. On categorical datasets, all different methods helped to keep performance similar to that of the complete data for low % of missing values but not when increasing the uncertainty. The performance of EMB_Hom, MICE_SE and EMB_SE was the best on both categorical and mixed data.

For SMO, shown in Fig. 8 the median accuracy of all methods on numerical datasets was similar to each other and to the complete data while the minimum accuracy of MICE_SE and EMB_SE increased by up to 10% compared with the completed data. Similarly, all approaches tested on the categorical data were relatively close. On mixed datasets, the median accuracy of the classifier seemed to be equal to the complete data but the maximum accuracy increased when applying MICE_Hom, MICE_SE and EMB_SE.

Finally, the performance of RF with respect to different data types is shown in Fig. 9. All approaches tested were relatively similar so for this algorithm the method of imputation produced minor or no improvements. MICE_SE and EMB_SE improved on maximum average accuracy for the

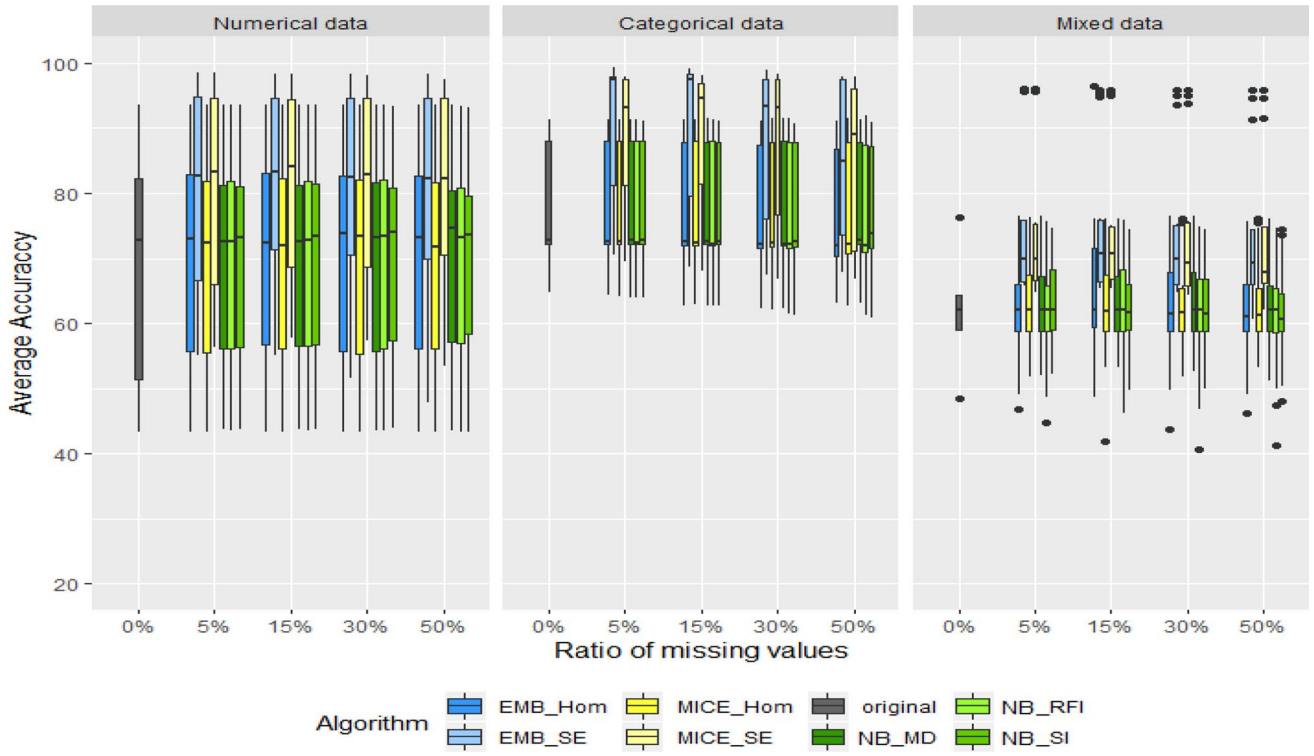


**Fig. 6** This figure contains box plots describing the overall average accuracy of NB applied on imputed datasets using different imputation approaches along with the average accuracy on the complete data

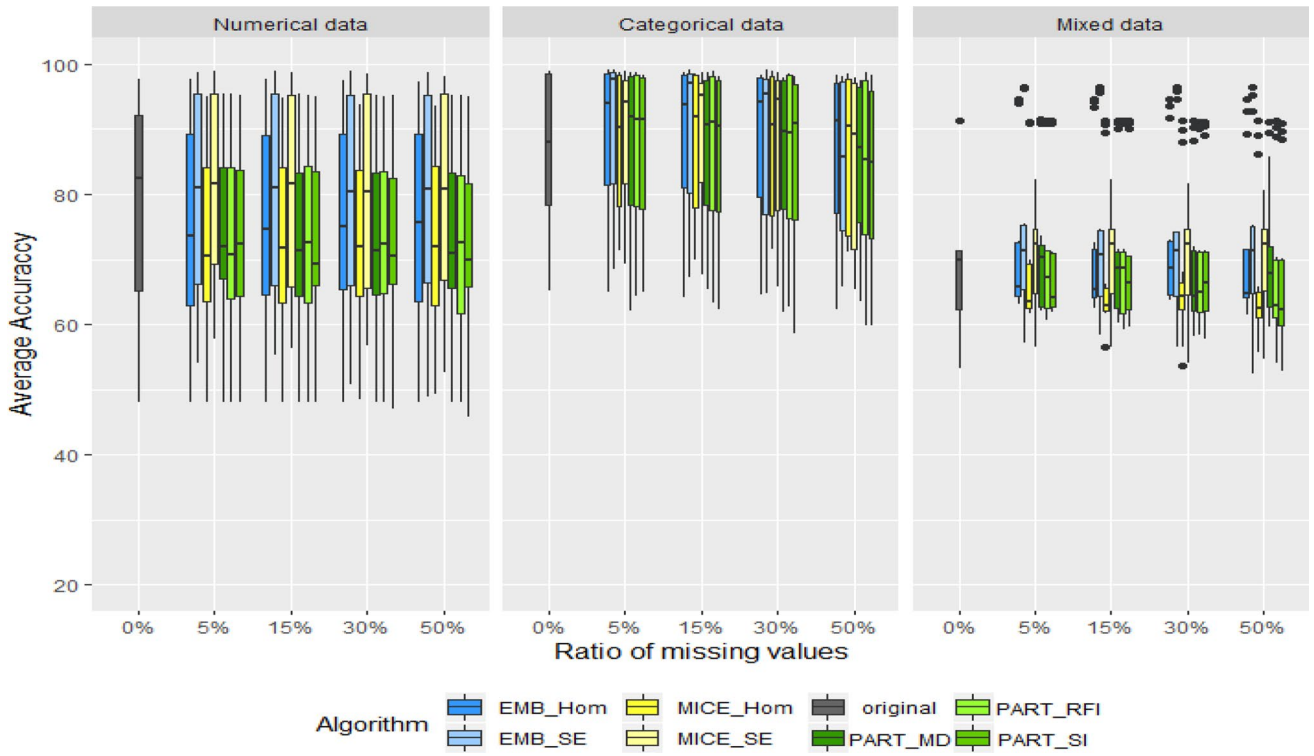


**Fig. 7** This figure contains box plots describing the overall average accuracy of PART applied on imputed datasets using different imputation approaches along with the average accuracy on the complete data

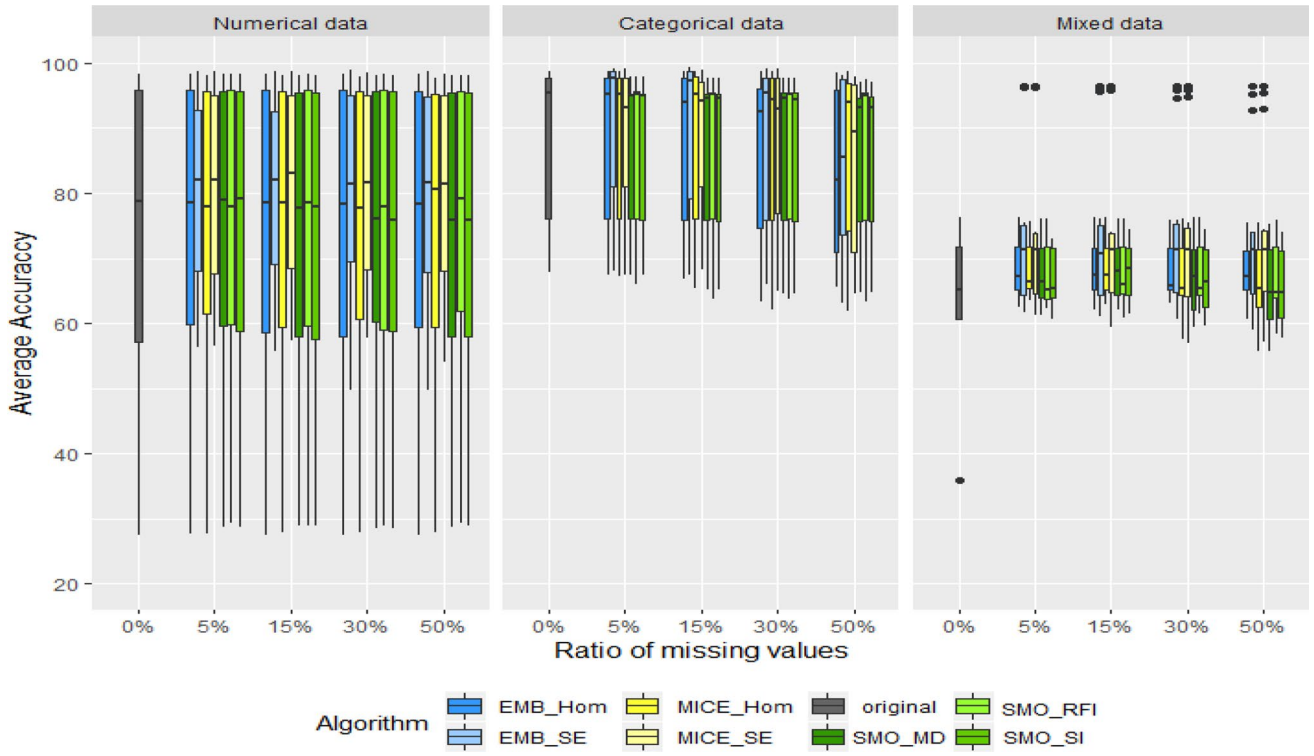


**Fig. 8** This figure contains box plots describing the overall average accuracy of SMO applied on imputed datasets using different imputation approaches along with the average accuracy for the complete data

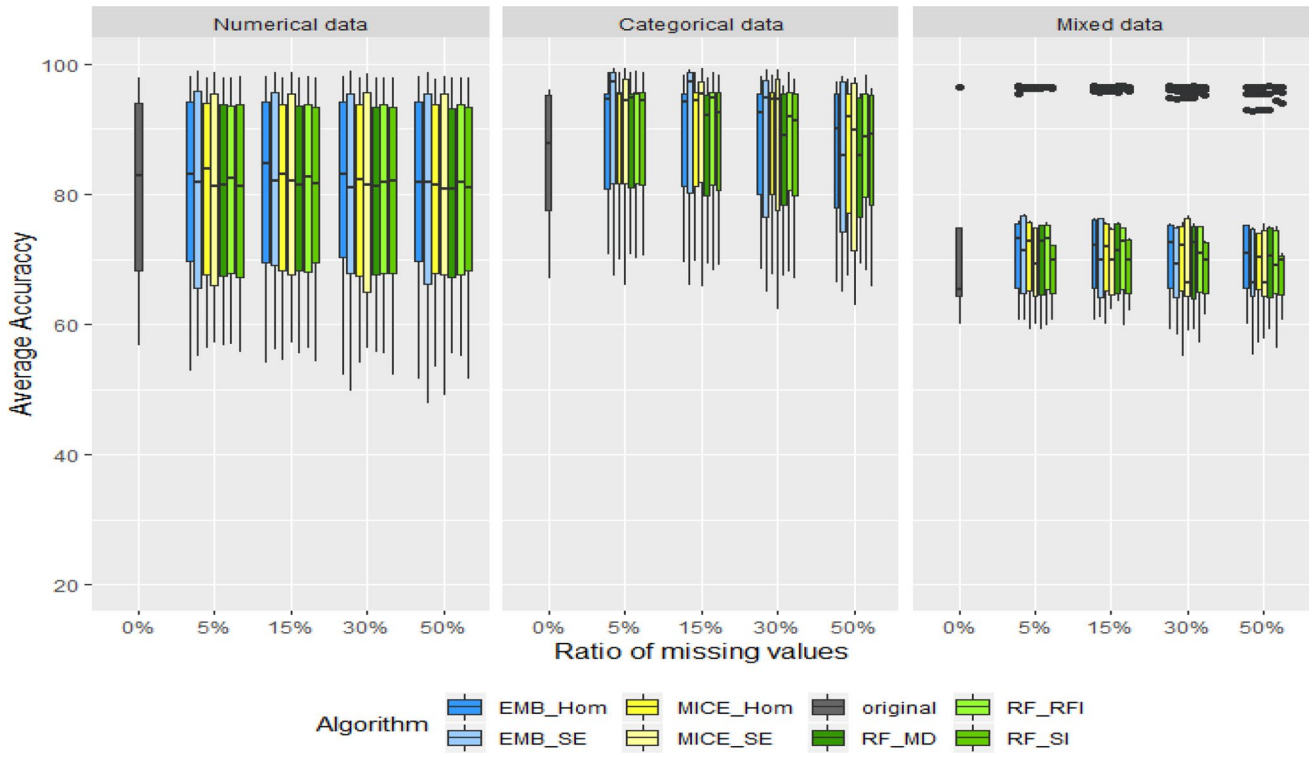


**Fig. 9** This figure contains box plots describing the overall mean accuracy of RF applied on imputed datasets using different imputation approaches along with original data

categorical data but did not perform so well when increasing missing data was present. For the mixed data all approaches seemed similar.

## Discussion and Conclusions

In this study, we investigate how different classification algorithms behave when using various methods for missing values imputation. We propose our MIE approach to improve classification with missing data and compare it with other methods for dealing with missing data.

For J48, NB, PART and to a large extent for SMO, the proposed EMB_SE produced the best performance for most levels of missing data with MICE_SE being a closed second. For high levels of missing data SMO worked well with RFI. For the RF algorithm, however, EMB_Hom produced the best performance in most cases. The differences in performance were statistically significant in all cases for J48, NB, PART and in the majority of cases for SMO. For RF, they were statistically significant in some scenarios only.

On the other hand, when comparing different approaches with a control method for imputation in the form of SI, we found that in most cases the proposed MIE techniques that rely on stacking (MICE_SE and EMB_SE) obtain statistically significantly better classification accuracy than the control when working with J48, NB and PART. This was also true for SMO in the majority of scenarios but not for RF where EMB_Hom showed more significant improvements, consistently with our previous results.

It is not possible to directly compare our results to others working on related work due to different datasets and experimental set-up. However, some comparisons are possible. For example, our findings, particularly for J48, are consistent with similar work done by Tran et al. [69] where they combined data imputed by MICE with an ensemble by using the majority vote method. Their proposed work achieved an improvement in terms of the classification accuracy. In our work we obtained further improvements on accuracy when using EMB imputation and stacking ensembles (MICE_SE and EMB_SE).

We proposed the use of dissimilarity to assess how far is the imputed data from the real data so that we can relate this to the performance of the algorithms. For numerical data RFI seems to perform best particularly for growing percentages of missing data. For categorical data EMB appears best by a very small margin, except for the highest missing data scenario. For mixed data EMB seems always best.

However, from further analysis of performance on each data type we can see that the imputation that recovers data best does not necessarily lead to a better classification performance. From the box plot analysis, we find that for all algorithms and data types except for RF, EMB_SE and MICE_SE produce consistently better performance than the others, hinting at the fact that the ensemble plays a big part in producing good results. For RF again most methods seem to perform similarly though EMB_SE and MICE_SE are still consistently good performers. This indicates that the ensemble in itself produces improvements irrespective of the quality of the imputation. As RF is already an ensemble algorithm, the advantages of MI for RF appear less obvious than for the others.

One of our important findings is that even in scenarios of increasing uncertainty, it is possible to obtain results similar or in some cases better than those obtained with the complete data, if the right imputation technique is used. This is an important finding as reasoning with missing data becomes then a lesser problem in the context of MCAR data. In this sense our proposed MIE methods, particularly those using stacking as the ensemble method produce consistently good results. Although multiple imputation may consume time and memory particularly with large datasets, its advantages in terms of representing the uncertainty as well as the ability to introduce diversity for the ensemble classifiers enables us to improve classification accuracy for scenarios with large levels of missing data and for most classification algorithms tested. If an algorithm such as RF is used, then the imputation method appears less relevant, although a poor imputation method like SI can produce deteriorated performance particularly for mixed data.

As a future work, we can increase the number of imputed datasets to test if more diversity produces further improvements. We could also test the implications of MAR data, by providing a different experimental set-up.

**Acknowledgements** We acknowledge support from Grant Number ES/L011859/1, from The Business and Local Government Data Research Centre, funded by the Economic and Social Research Council to provide economic, scientific and social researchers and business analysts with secure data services.

## Compliance with Ethical Standards

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

## References


1. Abayomi K, Gelman A, Levy M. Diagnostics for multivariate imputations. J R Stat Soc Ser C (Appl Stat). 2008;57(3):273–91.
2. Aleryani A, Wang W, De La Iglesia B. Dealing with missing data and uncertainty in the context of data mining. In: International conference on hybrid artificial intelligence systems, Springer, p. 289–301; 2018.
3. Azur MJ, Stuart EA, Frangakis C, Leaf PJ. Multiple imputation by chained equations: what is it and how does it work? Int J Methods Psychiatr Res. 2011;20(1):40–9.
4. Batista GE, Monard MC. An analysis of four missing data treatment methods for supervised learning. Appl Artif Intell. 2003;17(5–6):519–33.
5. Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory, ACM, p. 144–152; 1992.
6. Breiman L. Bagging predictors. Mach Learn. 1996;24(2):123–40.
7. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.
8. Buuren Sv, Groothuis-Oudshoorn K. Mice: multivariate imputation by chained equations in r. J Stat Softw. 2010; 1–68
9. Chae SS, Kim JM, Yang WY. Cluster analysis with balancing weight on mixed-type data. Commun Stat Appl Methods. 2006;13(3):719–32.
10. Chai X, Deng L, Yang Q, Ling CX. Test-cost sensitive naive Bayes classification. In: ICDM'04. Fourth IEEE international conference on data mining, IEEE, p. 51–58; 2004.
11. Che Z, Purushotham S, Cho K, Sontag D, Liu Y. Recurrent neural networks for multivariate time series with missing values. Sci Rep. 2018;8(1):6085.
12. Chen X, Wei Z, Li Z, Liang J, Cai Y, Zhang B. Ensemble correlation-based low-rank matrix completion with applications to traffic data imputation. Knowl Based Syst. 2017;132:249–62.
13. Cherkauer KJ. Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks. In: Working notes of the AAAI workshop on integrating multiple learned models, vol. 21, Citeseer; 1996.
14. Choi SS, Cha SH, Tappert CC. A survey of binary similarity and distance measures. J Syst Cybern Inform. 2010;8(1):43–8.
15. Clark D, Schreter Z, Adams A. A quantitative comparison of dystal and backpropagation. In: Australian conference on neural networks; 1996.
16. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3):273–97.
17. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Mthodol); 1977, 1–38.
18. Demšar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res. 2006;7(Jan):1–30.
19. Dietterich TG. Ensemble methods in machine learning. In: International workshop on multiple classifier systems, Springer, p. 1–15; 2000.
20. Dietterich TG. Ensemble learning. In: The handbook of brain theory and neural networks, vol. 2, p. 110–25; 2002.
21. Dittman D, Khoshgoftaar TM, Wald R, Napolitano A. Random forest: a reliable tool for patient response prediction. In: 2011 IEEE international conference on bioinformatics and biomedicine workshops (BIBMW), IEEE, p. 289–296; 2011.
22. Dong Y, Peng CYJ. Principled missing data methods for researchers. SpringerPlus. 2013;2(1):222.
23. Farhangfar A, Kurgan L, Dy J. Impact of imputation of missing values on classification error for discrete data. Pattern Recognit. 2008;41(12):3692–705.
24. Fichman M, Cummings JN. Multiple imputation for missing data: making the most of what you know. Organ Res Methods. 2003;6(3):282–308.
25. Frank E, Witten IH. Generating accurate rule sets without global optimization. In: Shavlik J (ed.) Fifteenth international conference on machine learning, Morgan Kaufmann, p. 144–151; 1998.
26. Frank E, Witten IH. Generating accurate rule sets without global optimization; 1998.
27. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci. 1997;55(1):119–39.
28. Gao H, Jian S, Peng Y, Liu X. A subspace ensemble framework for classification with high dimensional missing data. Multidimens Syst Signal Process. 2017;28(4):1309–24.
29. García S, Fernández A, Luengo J, Herrera F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: e. Inf Sci. 2010;180(10):2044–64.
30. García-Laencina PJ, Sancho-Gómez JL, Figueiras-Vidal AR. Pattern classification with missing data: a review. Neural Comput Appl. 2010;19(2):263–82.
31. Garciarena U, Santana R. An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. Expert Syst Appl. 2017;89:52–65.
32. George-Nektarios T. Weka classifiers summary. Athens: Athens University of Economics and Bussiness Intracom-Telecom; 2013.
33. Gower JC. A general coefficient of similarity and some of its properties. Biometrics. 1971; p. 857–871.
34. Grzymala-Busse JW, Hu M. A comparison of several approaches to missing attribute values in data mining. In: International conference on rough sets and current trends in computing, Springer, p. 378–385; 2000.
35. He Y, Zaslavsky AM, Landrum M, Harrington D, Catalano P. Multiple imputation in a large-scale complex survey: a practical guide. Stat Methods Med Res. 2010;19(6):653–70.
36. van der Heijden GJ, Donders ART, Stijnen T, Moons KG. Imputation of missing values is superior to complete case analysis and the missing-indicator method in multivariable diagnostic research: a clinical example. J Clin Epidemiol. 2006;59(10):1102–9.
37. Honaker J, King G. What to do about missing values in time-series cross-section data. Am J Polit Sci. 2010;54(2):561–81.
38. Honaker J, King G, Blackwell M, et al. Amelia ii: a program for missing data. J Stat Softw. 2011;45(7):1–47.
39. Horton N, Kleinman KP. Much ado about nothing: a comparison of missing data methods and software to fit incomplete data regression models. Am Stat. 2007;61:79–90. https://EconPapers.repec.org/RePEc:bes:amstat:v:61:y:2007:m:february:p:79-90.
40. Horton NJ, Kleinman KP. Much ado about nothing: a comparison of missing data methods and software to fit incomplete data regression models. Am Stat. 2007;61(1):79–90.
41. Kelly PJ, Lim LLY. Survival analysis for recurrent event data: an application to childhood infectious diseases. Stat Med. 2000;19(1):13–33.
42. Kennickell A.B. Imputation of the 1989 survey of consumer finances: stochastic relaxation and multiple imputation. In: Proceedings of the survey research methods section of the American Statistical Association, vol. 1; 1991.
43. Khalilia M, Chakraborty S, Popescu M. Predicting disease risks from highly imbalanced data using random forest. BMC Med Inform Decis Making. 2011;11(1):51.
44. Klebanoff MA, Cole SR. Use of multiple imputation in the epidemiologic literature. Am J Epidemiol. 2008;168(4):355–7.
45. Kohavi R, Becker B, Sommerfield D. Improving simple bayes; 1997.

46. Kotsiantis SB, Zaharakis I, Pintelas P. Supervised machine learning: a review of classification techniques; 2007.
47. Lichman M. UCI machine learning repository; 2013. http://archive.ics.uci.edu/ml
48. Little RJ, Rubin DB. Statistical analysis with missing data. New York: Wiley; 2014.
49. Liu Z, Pan Q, Dezert J, Martin A. Adaptive imputation of missing values for incomplete pattern classification. Pattern Recognit. 2016;52:85–95.
50. Newman DA. Longitudinal modeling with randomly and systematically missing data: a simulation of ad hoc, maximum likelihood, and multiple imputation techniques. Organ Res Methods. 2003;6(3):328–62.
51. Quinlan JR. C4. 5: programs for machine learning. Amsterdam: Elsevier; 2014.
52. Quinlan JR, et al. Bagging, boosting, and c4. 5. In: The association for the advancement of artificial intelligence (AAAI), vol. 1, p. 725–730; 1996.
53. Raja P, Thangavel K. Soft clustering based missing value imputation. In: Annual convention of the computer society of India, Springer, p. 119–133; 2016.
54. Rokach L. Ensemble-based classifiers. Artif Intell Rev. 2010;33(1–2):1–39.
55. Rubin DB. Multiple imputation after 18+ years. J Am Stat Assoc. 1996;91(434):473–89.
56. Rubin DB, Schenker N. Multiple imputation in health-are databases: an overview and some applications. Stat Med. 1991;10(4):585–98.
57. Schafer JL. Analysis of incomplete multivariate data. Boca Raton: CRC Press; 1997.
58. Schafer JL. Multiple imputation: a primer. Stat Methods Med Res. 1999;8(1):3–15.
59. Scheffer J. Dealing with missing data. Res Lett Inf Math Sci. 2002;3(1):153–60.
60. Schölkopf B, Burges CJ, Smola AJ. Advances in kernel methods: support vector learning. New York: MIT press; 1999.
61. Sefidian AM, Daneshpour N. Missing value imputation using a novel grey based fuzzy c-means, mutual information based feature selection, and regression model. Expert Syst Appl. 2019;115:68–94.
62. Silva-Ramírez EL, Pino-Mejías R, López-Coello M. Single imputation with multilayer perceptron and multiple imputation combining multilayer perceptron and k-nearest neighbours for monotone patterns. Appl Soft Comput. 2015;29:65–74.
63. Spratt M, Carpenter J, Sterne JA, Carlin JB, Heron J, Henderson J, Tilling K. Strategies for multiple imputation in longitudinal studies. Am J Epidemiol. 2010;172(4):478–87.
64. van Stein B, Kowalczyk W. An incremental algorithm for repairing training sets with missing values. In: International conference on information processing and management of uncertainty in knowledge-based systems, Springer, p. 175–186; 2016.
65. Sterne JA, White IR, Carlin JB, Spratt M, Royston P, Kenward MG, Wood AM, Carpenter JR. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. BMJ. 2009;338:b2393.
66. Tan PN, et al. Introduction to data mining. Bengaluru: Pearson Education India; 2006.
67. Ting KM, Witten IH. Issues in stacked generalization. J Artif Intell Res. 1999;10:271–89.
68. Tran CT, Zhang M, Andreae P. A genetic programming-based imputation method for classification with missing data. In: European conference on genetic programming, Springer, p. 149–163, 2016.
69. Tran CT, Zhang M, Andreae P, Xue B, Bui LT. Multiple imputation and ensemble learning for classification with incomplete data. In: The 20th Asia Pacific symposium on intelligent and evolutionary systems, IES 2016, Canberra, Australia, November 2016, Proceedings, Springer, pp. 401–415; 2017.
70. Tran CT, Zhang M, Andreae P, Xue B, Bui LT. Improving performance of classification on incomplete data using feature selection and clustering. Appl Soft Comput. 2018;73:848–61.
71. Tukey JW. Exploratory data analysis, vol. 2. Reading, MA; 1977.
72. Van Buuren S. Multiple imputation of discrete and continuous data by fully conditional specification. Stat Methods Med Res. 2007;16(3):219–42.
73. Van Buuren S, Boshuizen HC, Knook DL, et al. Multiple imputation of missing blood pressure covariates in survival analysis. Stat Med. 1999;18(6):681–94.
74. Vapnik V. The nature of statistical learning theory. Berlin: Springer; 2013.
75. Witten IH, Frank E, Hall MA, Pal CJ. Data mining: practical machine learning tools and techniques. Burlington: Morgan Kaufmann; 2016.
76. Wolpert DH. Stacked generalization. Neural Netw. 1992;5(2):241–59.