This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2994033, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Iterative Robust Semi-Supervised Missing Data Imputation

Nikos Fazakis[1], **Georgios Kostopoulos**[2], **Sotiris Kotsiantis**[3], **and Iosif Mporas**[4]
[1]Department of Electrical and Computer Engineering, University of Patras, Rion, CO 26504 GR
[2,3]Department of Mathematics, University of Patras, Rion, CO 26504 GR
[4]Centre for Engineering Research, University of Hertfordshire, Hatfield, CO AL10 9AB UK

Corresponding author: Nikos Fazakis (e-mail: fazakis@ece.upatras.gr).

ABSTRACT In many real-world applications scientists are often confronted with the problem of incomplete datasets due to several reasons. The direct analysis of datasets with missing values in attributes inevitably results in inaccurate learning models and erroneous results. Facing effectively the challenge of missing values is an essential step of the data mining process. Imputation is often employed to overcome the shortcomings incurred by missing data during the pre-process stage of data analysis. Therefore, a plethora of statistical and machine learning methods have been proposed and employed with a view to imputing the missing values in incomplete data with their potential or actual values. In this context, the main objective of this paper is to put forward an iterative stepwise imputation method based on the semi-supervised learning approach, called IRSSI. Semi-supervised methods have proved to be particularly effective for exploiting incomplete or partially labeled data with regard to the values of the target attribute. The proposed algorithm was experimentally evaluated on real-world benchmark datasets and artificially generated datasets using different high ratios of missing data. The experimental results demonstrate the efficiency of IRSSI algorithm compared to typical imputation methods.

INDEX TERMS Missing values, imputation, classification, semi-supervised learning.

I. INTRODUCTION

In many real-world applications scientists are often confronted with the problem of incomplete datasets [1]. This phenomenon is particularly intense on medical, clinical data, industrial and survey data [2], [3], [4]. Incompleteness or deficiency [3] is a frequent phenomenon which refers to the presence of missing values in one or more attributes of a dataset due to a variety of reasons, including manual data entry mistakes, equipment faults, devise failure, inaccurate measurements during data collection, accidental deletion, non-response, admission limitations, unwillingness to provide personal information and so on [5], [6]. The analysis of datasets with missing values in attributes is infeasible in most cases, since conventional data methods are not directly applicable [3]. Even if such a method is workable, it inevitably results in inaccurate learning models and erroneous results [7]. On the other hand, knowledge quality and data quality are inextricably linked. Therefore, poor data quality has a negative effect on both descriptive and predictive statistics [8].

A very interesting aspect of the missing values analysis concerns the mechanisms which result in missing data. The so-called “missingness mechanisms” [9] describe dependencies or non-dependencies between the distribution of an instance having one or more missing values and the distributions of observed and missing data [10]. Moreover, these mechanisms have a material impact on selecting the proper method for handling the missing data which occur in many real-world datasets [11]. We mainly consider three different assumptions of missing data [7]: Missing at Random (MAR), Missing Completely at Random (MCAR) and Missing not at Random (MNAR). MAR means that the probability of a single attribute value missing depends on the values of the observed data but not on the missing ones. In this case, the distribution of the observed data is the same as the distribution of the missing values [2]. Data are MCAR when the probability of a single attribute value missing is contingent neither on the values of the observed data nor upon the missing ones. Essentially, MCAR forms a special case of MAR [12]. In this case, the distributions of the observed and missing data are different. Finally, data are MNAR if the probability of a single attribute value to be missing depends on the missing data. Practically, if data are neither MAR nor MCAR, then they are deemed to be MNAR [13].

Facing effectively the challenge of missing values is an essential step of the data mining process. Missing values can be handled through specific strategies such as deletion of partially recorded instances or supplementation with potential or actual values [12], a method referred to as imputation [9]. Deleting instances with missing values is somewhat a straightforward approach which, however, results to loss of valuable information [9]. Unlike deletion, imputation is a very common approach to overcome the shortcomings caused by missing data during the pre-process stage of a data mining task. Therefore, a plethora of statistical and machine learning methods have been proposed with a view to imputing missing values in incomplete datasets according to a specific technique [14].

Machine learning based imputation methods have been shown to be very effective for addressing the missing values phenomenon in recent years. The prevalent idea behind these methods is to train a classification or regression model based upon observed data and subsequently apply it to predict all missing values of the dataset attributes [10]. A plethora of familiar supervised and unsupervised methods including a variety of classification, regression and clustering techniques have been effectively used in a wide range of studies, as easily identified in the pertinent literature [15].

From that perspective, the main objective of this study is to propose an imputation method integrating the semi-supervised learning (SSL) approach which is also known as "*weakly*" or "*incomplete supervised learning*" [16]. SSL methods have proved to be particularly effective for exploiting a small pool of labeled examples together with a large pool of unlabeled ones to improve learning performance. In this context, unlabeled examples may be considered as a form of incomplete or partially labeled instances [17] as regards the missing values of the target attribute (Fig. 1). Although SSL methods have been widely used for solving a variety of data mining problems, there is no similar work on the imputation field. The proposed algorithm, which we call Iterative Robust Semi-Supervised based Imputation (IRSSI), is a new hybrid imputation method based on robust semi-supervised ensembles, thereby harnessing the benefits of SSL. The experimental results on real-world benchmark datasets and artificially generated datasets, with respect to different and high ratios of missing data, demonstrate the efficiency of IRSSI algorithm compared to familiar imputation methods. Since SSL methods have not yet been used in the imputation process, we consider our proposal as an initial, yet promising step towards this direction.

The rest of this paper is structured as follows: In Section II, we discuss several methods for handling incomplete data, especially those concerning imputation methods. The proposed algorithm is described in detail in Section III, while the experimental procedure and the corresponding results are presented and analyzed in Section IV. Finally, the study concludes considering some thoughts for future directions.

II. METHODS FOR HANDLING MISSING DATA

A plethora of methods have been proposed to tackle the incompleteness problem, each one having its own advantages and disadvantages [18]. These methods can be grouped into two main categories: deletion and imputation.

A. DELETION METHODS

Dropping attributes or instances from incomplete data is considered to be a naive and convenient method for handling missing values, especially in the case where data are missing completely at random [10]. In the following paragraphs, we discuss the two major types of deletion methods: complete-case analysis and available-case analysis.

1. COMPLETE-CASE ANALYSIS

A very simple and commonly used method for handling incomplete data is to delete all instances having one or more missing values. Complete-case analysis (CCA) or case-wise deletion [11] is indeed a preferable and quite effective method

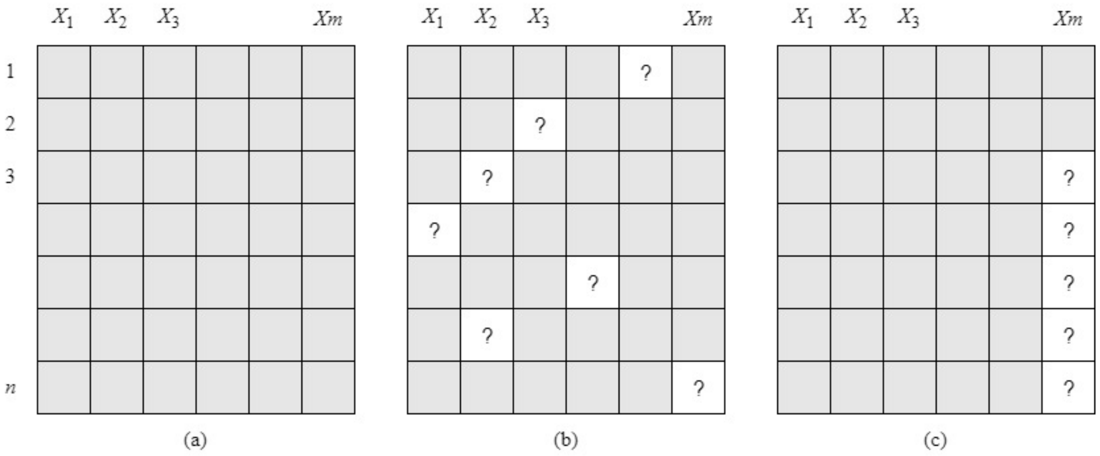

FIGURE 1. (a) A complete dataset consisting of *n* instances and *m* attributes X_1, X_1,…, X_m, (b) An incomplete dataset with missing values, (c) An incomplete dataset with missing values (unlabeled data) only in the target attribute X_m.

for data analysts, especially in cases where missing data constitutes a small part of the whole dataset [19]. A general acceptable rule is to omit all incomplete instances if data is missing for less than a predefined threshold, for example 5% [20]. At this point, we should emphasize in the following extreme case: Suppose that each one of the *n* instances of a dataset with *m* attributes has only one missing value, while all missing values correspond to different attributes. This implies the deletion of the whole dataset, while the missing ratio is only 1/*m*.

Nevertheless, although CCA yields a fully observed dataset which is available for further data analysis, it presents significant shortcomings that clearly affect its effectiveness. The first one is the loss of information, since it directly results to a subset of the initial dataset [9]. The second one is that the remaining subset is no longer representative of the parent population [12], while a bias is produced in the model if the missingness mechanism is not MCAR [10].

2. AVAILABLE-CASE ANALYSIS

Available-case analysis (ACA) or pair-wise deletion is another familiar deletion method which exploits instances with missing values in a flexible manner. More specifically, instead of dropping out an incomplete instance, the specific instance is used for analyzing the rest attributes with non-missing values, thereby utilizing all available information [7]. In this case, different analyses of data are produced based on different subsamples of instances which are depended only on the attributes employed each time. A major disadvantage of this method is that it also leads to biased estimates if the missingness mechanism is not MCAR [11].

B. IMPUTATION METHODS

The term "*imputation*" refers to the process of replacing the missing values of instances in a given incomplete dataset with their potential or actual values according to a specific strategy [12]. Therefore, several statistical and machine learning methods have been proposed and employed with a view to approximating the missing values occurred in incomplete datasets as effectively as possible [15]. Regardless of the method used, imputation is considered both an essential and sensitive step of data preprocessing [10], which clearly affects the performance of the data mining task [9]. Imputation methods are usually categorized into three main classes: single, machine-learning based and multiple (Fig. 2).

1. SINGLE IMPUTATION

Single or univariate imputation [21] deals with methods for replacing one missing value for each attribute with only an imputed one [9]. Four commonly used single imputation methods are: mean and mode, regression, hot deck and expectation-maximization.

MEAN AND MODE IMPUTATION

A particularly efficient and widely used statistical approach of replacing missing values of numerical or categorical attributes is through the mean and mode imputation technique [22]. According to the mean approach, the missing values of a single numerical attribute are replaced with the corresponding arithmetic mean of the observed ones of that attribute, while the mode approach fills out the missing values of a discrete or categorical attribute with the most frequent observed value (i.e. the mode of the attribute values) [6]. A slightly differentiated approach is to replace missing values through the mean and mode approach based solely on the instances with the same output class, known as concept average value and concept most common value respectively [23]. Note that, in both approaches, missing values are filled up with estimated ones, which inevitably introduces an additional bias [7], especially when data are not MCAR.

REGRESSION IMPUTATION

In accordance with this method, a regression model is built from the observed data of a specific instance and subsequently is used to predict the values of the missing values of that instance. The regression model (linear or non-linear) that best fits on the data depends on the nature of relationships between attributes [20]. Linear regression is usually applied for estimating the missing values of numerical attributes, while logistic regression or multinomial logistic regression is usually used for estimating the missing values of categorical ones [15].

HOT DECK IMPUTATION

Hot deck imputation is based on similar but complete instances of data for replacing the missing values of incomplete ones [20]. A considerable advantage of hot deck imputation is that, it does not alter the distribution of observed data after the imputation process, unlike mean and mode imputation [9]. A very similar approach, namely cold deck imputation, is to make use of similar complete data coming from an external source [9].

EXPECTATION-MAXIMIZATION

The Expectation-Maximization (EM) algorithm is an iterative method for imputing missing values in incomplete numerical datasets, originally introduced by Dempster et al. [24]. The concept behind the EM algorithm is "*impute, estimate and iterate until convergence*" [9]. Each iteration consists of two steps: expectation and maximization. The expectation step concerns the estimation of missing values given the observed data, while, in the maximization step, the current estimated values are used to maximize the likelihood of all the data. The estimated values are updated, the two steps are iterated until convergence of the maximum likelihood of data, and the final estimates are used as the imputation values [15].

2. MACHINE-LEARNING BASED IMPUTATION

Machine learning-based imputation [25] concerns the process of building a predictive learning model based on the observed data for estimating the values of the missing ones [20]. In particular, the attribute with missing values is considered to be the target attribute, while the rest ones are used to train a learning algorithm which is subsequently used to predict the unknown missing values [22]. According to [15], clustering, *k*-Nearest Neighbors (*k*-NN), decision trees and Random Forests are the top four machine learning-based imputation methods, while a plethora of machine learning based imputation methods are presented and discussed in [10] and which fall into five main categories: clustering, *k*-NN, decision trees, support vector machines and Artificial Neural Networks imputation methods. A short overview of some of the most characteristic machine learning imputation methods is presented below:

k-NN IMPUTATION

It is a simple and quite effective similarity-based imputation approach which relies on the *k*-NN technique [8]. For each missing value of a specific instance, the *k* most similar instances are selected according to the shared non-missing values and a predefined similarity measure (e.g. Euclidean distance, Manhattan distance or Minkowski norm). For categorical attributes, the imputed value is the most common among the *k* most similar instances, while the average value is used for numerical ones [10]. A slightly modified approach is the weighted *k*-NN method [26], that weights the distances of neighbors (weighted average) on the basis of a similarity measure. Obviously, both *k*-NN approaches are typical hot deck methods [20], whereas their effectiveness depends on the appropriate selection of the *k* parameter, which is often empirically selected [27].

DECISION TREES IMPUTATION

Decision trees form a commonly used supervised approach for imputing missing values in attributes. In general, a classification or regression tree is built for each attribute trained on observed data, which is subsequently used for estimating the missing values of a particular attribute [28].

CLUSTERING IMPUTATION

Typical clustering methods, such as *k*-means, hierarchical clustering [29] and *k*-means clustering with weighted distance [30] have been generally employed to improve the imputation performance in incomplete datasets. However, clustering methods are not robust enough to missing data [26].

3. MULTIPLE IMPUTATION

Multiple imputation or multivariate imputation [21] or repeated imputation [4] deals with methods for replacing one missing value for each attribute with *k>1* different imputed ones, thus creating *k* simulated and complete datasets which reflect the uncertainty of missing data [9]. More specifically, for each missing value, the estimated values are stored in a 1x*k* row vector, while the corresponding components of vectors along with the observed data constitute the *k* simulated datasets [3]. Imputation of missing data may be carried out by applying a specific technique, such as regression model, or even a sequence of regression models, such as linear, logistic and Poison, as has been shown in [31]. The *k* simulated datasets are analyzed separately, and the results are finally combined. Multiple imputation is a statistical technique which was originally proposed by Rubin for handling the problem of non-response in sample surveys [3]. The idea behind the creation of *k* multi-imputed datasets is to reflect both variation inside a single imputed dataset and sensitivity of inferences from the *k* different imputed datasets [32], contrary to the single imputation approach. Unfortunately, multiple imputation approach is more complex, time demanding and requires large data storage capabilities [9]. An important issue in multiple imputation is the appropriate selection of *k*, which is usually set equal to 3 or 5 [33].

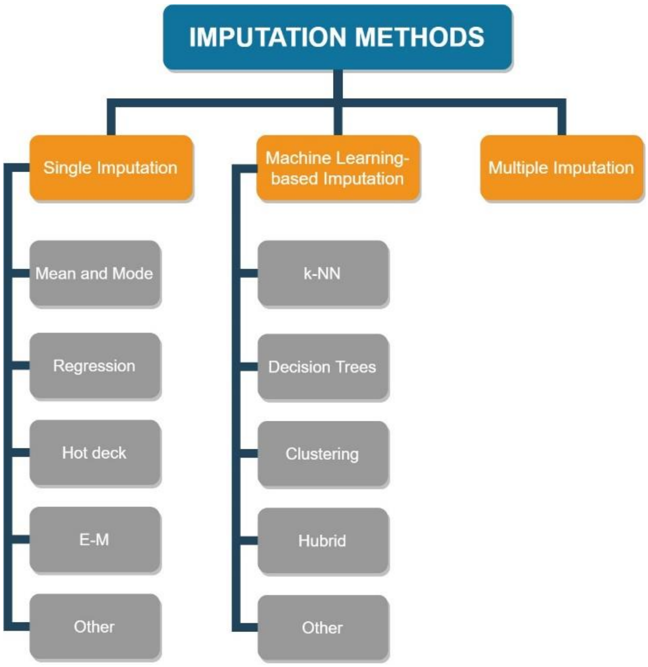

FIGURE 2. A representative taxonomy of imputation methods

To recapitulate, a vast array of differentiated methodologies has been put forward for efficiently handling the missing values problem as it can be promptly signaled out in the pertinent literature. These methods may be sorted out into different types [15]: simple and straightforwardly applicable approaches such as deletion methods, statistical methods, machine learning based methods and hybrid methods such as the one in [34], which combines C4.5, a well-known decision tree classifier, with the expectation maximization method. Even if the last ones have emerged recently in the imputation field, their effectiveness has been amply demonstrated in a wide range of studies. At this point, we should pinpoint that there is no universal imputation method that performs best for all datasets [35]. Utilization of datasets with different structure, the difference lying in the number of instances and attributes, as well as the percentage variation of missing data make it slightly difficult to recognize a widely approved method.

Motivated by the recent trend concerning the machine-learning based imputation methods, we propose a predictive model aiming to estimate missing values in incomplete datasets utilizing the SSL approach.

III. THE PROPOSED METHOD

As stated above, the main objective of this study is to present an imputation algorithm incorporating the SSL approach. The proposed algorithm is established on the basis of the Iterative Robust Model-based Imputation (IRMI) [21] algorithm. The IRMI algorithm is an improved variant of the Sequential Regression Multivariate Imputation (SRMI) approach proposed in [31], an effective and quite robust imputation technique for complex data structure, especially when data are MAR or MCAR. To harness the potential of SSL, two self-training techniques are employed within the attribute fitting loop of the IRMI algorithm, thus constructing a new hybrid imputation method based on robust semi-supervised ensembles, which we now call Iterative Robust Semi-Supervised based Imputation (IRSSI).

Simulating the IRMI algorithm behavior, IRSSI is an iterative algorithm which loops through all available attributes of a dataset, setting each time one of them as response attribute and all the others as independent ones. Essentially, the response attribute in each iteration is the dataset feature which is going to be imputed by the algorithm. The proposed IRSSI algorithm is introduced below in 7 basic steps.

Step 1: The missing values of a specific attribute are initialized by replacing them with the mean or the mode value of the observed ones, whereas, at the same time, the original positions of missing values are recorded.

Step 2: The attributes are sorted in ascending order according to the total number of missing values in each one. For simplicity, we consider the following notation for the sorted attributes:

$$M(x_1) \leq M(x_2) \leq \ldots \leq M(x_p), \qquad (1)$$

where $M(x_i)$ denotes the number of missing values for the attribute x_i. In addition, let $I = \{1,\ldots,p\}$ denote the set of all attribute indices.

Step 3: A pointer $l = 1$ is initialized and used as an attribute index.

Step 4: The indices of the initially missing values for attribute x_l are denoted as m_l and the rest observed ones as o_l, where $m_l \cup o_l = \{1,\ldots,p\}$. Utilizing o_l and m_l, two matrices are constructed containing the observed and missing cells respectively, denoted as $X^{o_l}_{I\backslash\{l\}}$ and $X^{m_l}_{I\backslash\{l\}}$ related to attribute x_l. The $X^{o_l}_{I\backslash\{l\}}$ matrix together with the response $x_l^{o_l}$ attribute constitute the labeled set L_l^0 of observed attributes for the current l. Using the same notation, $X^{m_l}_{I\backslash\{l\}}$ matrix along with $x_l^{m_l}$ constitute the current unlabeled set U_l^0 for l. According to the type of attribute x_l, two different procedures follow:

I. If x_l is a numerical attribute the sets L_l^0 and U_l^0 are passed in a multi-scheme semi-supervised regression (SSR) procedure [36]. This scheme utilizes three regression algorithms (hereinafter referred to as regressors) in order to efficiently augment L_l^0 with the U_l^0 instances [37].

II. If x_l is a categorical one, the sets L_l^0 and U_l^0 are passed in a self-trained classification procedure, an improved variant of [38], to augment L_l^0 with the U_l^0 instances.

The multi-scheme SSR procedure and the self-trained classification are briefly described below.

Step 5: According to the type of x_l, response $x_l^{m_l}$ is computed utilizing the internal base learners defined in procedures I or II, trained on the augmented labeled set (L_l) produced in step 4. In case of a numerical x_l, the averaged predictions of the three regressors are used as response values.

Step 6: Loop through steps 4 and 5 for $l = 2,\ldots,p$.

Step 7: Repeat steps 3 to 6 until the imputed cells are steady, according to the type:

$$\sum_i (\hat{x}_{l,i}^{m_l} - \tilde{x}_{l,i}^{m_l})^2 < \varepsilon, \qquad (2)$$

where $\hat{x}_{l,i}^{m_l}$ is the i-th imputed value of the current iteration and $\tilde{x}_{l,i}^{m_l}$ the previous imputed value. The constant ε is a small convergence parameter.

Speaking of the regression and classification procedures outlined on step 4, they both utilize the semi-supervised method of self-training [39]. The following paragraphs discuss briefly their logic flow.

MULTI-SCHEME SSR PROCEDURE

Starting with the first procedure, a multi-scheme SSR algorithm is employed using L_l^0 and U_l^0 as input. The applied algorithm is a variant of [37] and is based on an ensemble of three base regressors (bRegS) which are combined in a self-training loop using an instance selection function (MRL) [37]. The employed base regressors are described below in brief.

- Random Forests (RFs) is a simple, powerful and robust ensemble method for both classification and regression problems [40]. RFs creates multiple decision trees using different and randomly selected subsamples of features for splitting each tree node and aggregates their results via majority voting. A main advantage of the RFs algorithm is that it can efficiently handle overfitting phenomena.
- Linear Regression model (LR) [41].
- M5 is a model tree algorithm proposed by Quinlan [42], which induces trees of multivariate linear regression models. M5 is very effective and can successfully handle missing values and high dimensional datasets [43]. In brief, M5 learner grows regression trees with the leaves being themselves linear regression models.

In general, the combination of multiple regression models has a positive impact in the reduction of the generalization error. The selected models are widely referenced in the literature and are both efficient and robust. In each loop iteration, bRegS is trained on the current labeled set L^{iter} (with L^0 being equal to L_l^0). The trained models are then applied on U^{iter} (with U^0 being equal to U_l^0) and a matrix containing the predicted values is generated. Subsequently, the matrix is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2994033, IEEE Access

sorted using MRL and a percentage (T) of the unlabeled observations is added on L^{iter}, and removed from U^{iter}, using as target values the average predictions of bRegS for each observation. After the termination of the loop the augmented labeled set L_l ($\equiv L^{iter=last}$) is constructed.

SELF-TRAINED CLASSIFICATION PROCEDURE

The second procedure is used to exploit categorical attributes and is based on the algorithm presented in [38]. The base learner used inside the self-training classification loop is RFs algorithm and was picked for consistency reasons. In the same manner, in each self-training iteration, RFs is trained on L^{iter} and applied on U^{iter} and the predictions are produced. The most confident predictions are obtained in a matrix (M_{MCP}) considering the prediction probabilities of the observations of U^{iter}. Only a percentage (T) of the most confident predictions is added on L^{iter} (and removed from U^{iter}). After the self-training exiting criteria are met, the augmented labeled set is contained in L_l.

The pseudocode of IRSSI imputation algorithm is presented in Alg. 1, which summarizes the employed techniques.

IV. EXPERIMENTAL PROCESS AND RESULTS

Two basic approaches were used to validate the efficacy of the proposed algorithm. The first one is based on experimentation with real-world benchmark datasets. In the second, artificial datasets were constructed in order to further explore different aspects of IRSSI performance.

A. EXPERIMENTATION ON BENCHMARK DATASETS

The experiments were based on fifteen benchmark datasets from a variety of domain problems and were extracted from the UCI [44] repository, while a brief description of their structure is presented in Table I. We considered datasets with different structure: datasets with mixed type of attributes (categorical and numerical) and datasets consisting only of categorical attributes or numerical ones. Moreover, we considered both binary and multiclass classification problems.

TABLE I
STRUCTURE OF DATASETS

Dataset	*#Instances*	*#Categorical Features*	*#Numerical Features*	*#Classes*
anneal	898	32	6	6
audiology	226	69	0	24
breast-cancer	286	9	0	2
chess	3196	36	0	2
cmc	1473	7	2	3
credit-rating	690	9	6	2
haberman	306	1	2	2
horse-colic	368	15	7	2
housevotes	435	16	0	2
iris	150	0	4	3
kr-vs-kp	3196	36	0	2
solar-flare	323	12	0	6
soybian	683	35	0	19
tae	151	2	3	3
vote	435	16	0	2

Algorithm 1: Pseudocode of IRSSI algorithm

```
Input:
    D: the initial dataset containing missing values
    Epochs: number of maximum iterations allowed
Initialization:
    I.   Initialize missing values in D using the features' Medians and Modes values (Store original missing value positions).
    II.  Sort features analogous to their amount of missing values in ascending order.
    III. Construct matrices containing all observed ($X^{o_l}_{I\setminus\{l\}}$) and missing ($X^{m_l}_{I\setminus\{l\}}$) cells for each feature (l).
Main Loop:
  For currIter = 0; currIter ≤ Epochs; currIter++:
    For each l in D:
      IF l has no missing values or imputed cells are stable [as in (2)]:
        Skip l feature
      ELSE:
        Combine $X^{o_l}_{I\setminus\{l\}}$ with the observed cells of l ($x^{o_l}_l$) to construct the labeled set ($L^0_l$)
        Combine $X^{m_l}_{I\setminus\{l\}}$ with the missing cells of l ($x^{m_l}_l$) to construct the unlabeled set ($U^0_l$)
        IF l type is numeric: Apply SSR procedure on $L^0_l$ and $U^0_l$
        ELSE: Apply STC procedure on $L^0_l$ and $U^0_l$
        Re-calculate imputed cells stability for l [as in (2)]
SSR Procedure:
    Initialization:
        a. Train bRegS as base models on $L^0$ ($\equiv L^0_l$) and Set iter = 0
        b. Declare $f_{decision}$ ≡ MRL [37]
    Self-Train:
      Loop for a maximum of ten iterations (iter) or until $U^{iter}$ is empty (with $U^0 \equiv U^0_l$):
        a. Apply bRegS to $U^{iter}$ and select T*size($L^0$) instances with the most high-confident predictions ($X_{MCP}$) per iteration using $f_{decision}$
        b. Compute the average target values for $X_{MCP}$ instances using bRegS's predictions
        c. Remove $X_{MCP}$ from $U^{iter}$ and add them to $L^{iter}$
        d. Re-train bRegS as base model on new enlarged $L^{iter}$
        e. Set iter = iter + 1
      Train bRegS using the augmented labeled set ($L_l \equiv L^{iter=last}$)
      Predict feature l missing values ($x^{m_l}_l$) using bRegS on $U^0_l$
      Return
STC Procedure:
    Initialization:
        a. Train RFs as base model on $L^0$ ($\equiv L^0_l$) and set iter = 0
    Self-Train:
      Loop for a maximum of ten iterations (iter) or until $U^{iter}$ is empty (with $U^0 \equiv U^0_l$):
        a. Apply RFs on $U^{iter}$ and use the prediction probabilities to construct $M_{MCP}$
        b. Remove $M_{MCP}$ from $U^{iter}$ and add them to $L^{iter}$
        c. Re-train RFs as base model on new enlarged $L^{iter}$
        d. Set iter = iter + 1
      Train RFs using the augmented labeled set ($L_l \equiv L^{iter=last}$)
      Predict feature l missing values ($x^{m_l}_l$) using RFs on $U^0_l$
      Return
Output: Imputed dataset $D_{imputed}$
```

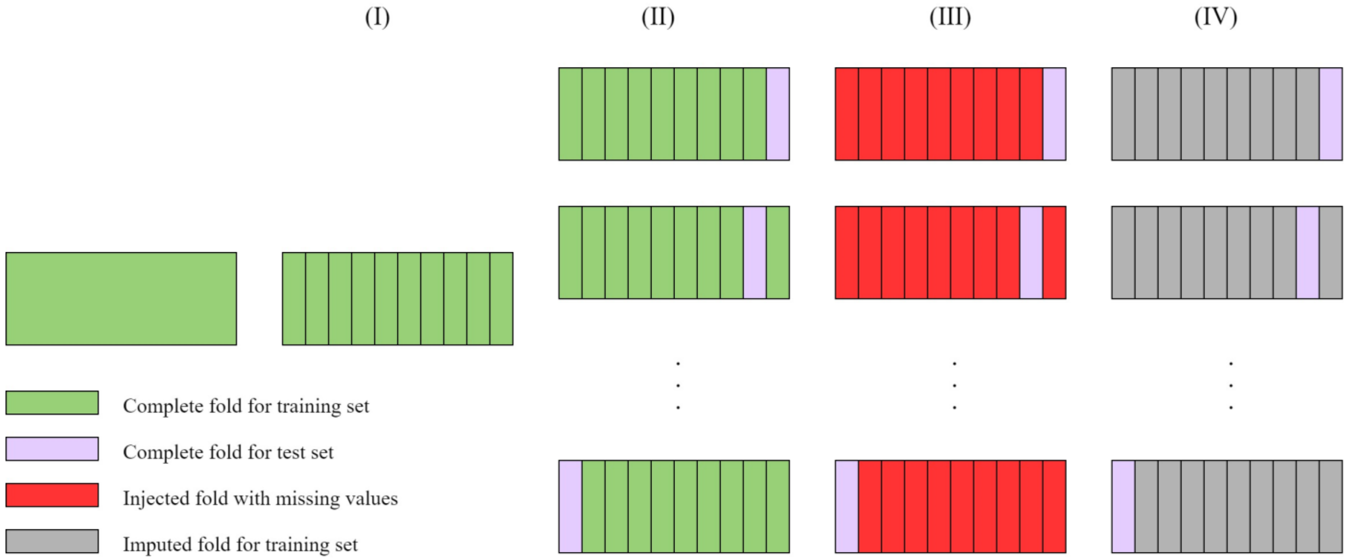
(I)
(II)
(III)
(IV)
Complete fold for training set
Complete fold for test set
Injected fold with missing values
Imputed fold for training set

FIGURE 3. Missing Values Injection and Imputation Process.

The experimental process consisted of four consecutive steps as illustrated in Fig. 3. Initially, each complete dataset was partitioned into ten equally sized folds using the 10 cross-validation resampling technique, thus ensuring the same distribution in each fold as in the full dataset. Subsequently, each fold was injected randomly with missing values employing the MCAR missingness mechanism. Three different proportions of missing values were considered in each dataset, hereinafter called missing ratio (MR), and in particular: 30%, 40% and 60%. The choice of missing ratio was based on relevant studies. These studies are primarily focused on small missing ratios, usually from 5% to 30%, while there is a lack of prior studies that consider missing ratios greater than 50% [15].

The next step was the simulation of missing values. This process was carried out employing six common and state-of-the-art imputation methods which can handle both categorical and numerical attributes, and in particular:

- The Mean/Mode method, which is regarded as one of the most representative baseline statistical missing values imputation techniques [15].
- The Fuzzy *k*-means (FKMeans) Clustering imputation method with the following values of input parameters: *k*=3, *m*=1.5 and the Euclidean metric as distance measure [45].
- The Local Least Squares (LLSimpute) imputation method [46].
- The Singular Value Decomposition (SVDimpute) imputation method [26].
- The IRMI machine learning based imputation algorithm.
- The proposed IRSSI algorithm.

In addition, for assessing the performance of the imputation methods employed in the experiments, two popular classification algorithms, belonging to representative machine learning families, were finally trained in the simulated and fully completed datasets. Hence, the classification process relies on the assumption that the imputed datasets simulate the real ones [12]. The two classification algorithms deployed after the imputation process, were the following:

- Rotation Forest (RF), a powerful ensemble of independent decision trees, based on feature extraction. Each base classifier is trained on a randomly selected subset of features, while Principal Component Analysis is applied to each one of the subsets [47].
- JRip, an implementation of RIPPER (Repeated Incremental Pruning to Produce Error Reduction), a very effective and interpretable rule-based induction learning algorithm based on incremental reduced error pruning [48].

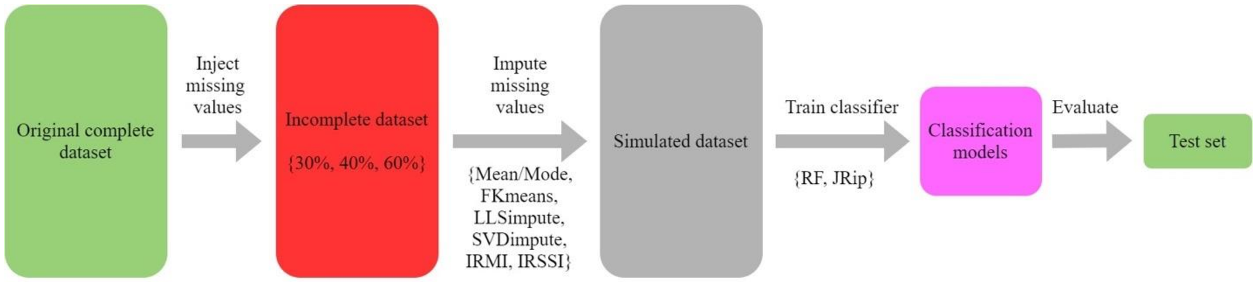

FIGURE 4. The Experimental Procedure

After the imputation process, each classifier was trained on nine simulated folds forming the training set, while the rest one, complete but non-simulated, was used for testing the performance of the classifier. This process was repeated ten times, until all folds were used as test set, and the results were averaged [49]. Therefore, we computed the overall accuracy of each learning model, a commonly used metric for classification problems, which corresponds to the percentage of correctly classified instances. In fact, accuracy is considered to be one of the most weighty metrics for evaluating different imputation methods for classification problems [20]. According to [20], the best imputation method gives better accuracy results for a specific classifier and a predefined missing ratio. The complete experimental procedure of our study is illustrated in Fig. 4.

A total of 45 incomplete and different datasets were finally included in the experimental process (3 missing ratios for each one of the 15 datasets). The average accuracy results regarding the three different missing ratios considered are summarized in Tables III, IV and V, while the supreme values for each dataset are highlighted in bold (including ties). Moreover, the standard deviation results in each case are presented in the same tables below each dataset accuracy. For simplicity, we make use of the notations acc_p, std_p for the accuracy and the standard deviation measure respectively for a missing ratio of p%.

It is clearly shown that the IRSSI algorithm performs better than all other imputation methods for most datasets, regardless of the missing ratio and the classifier deployed after the imputation process. The total number of wins for each imputation method, according to the missing ratio and the classifier deployed after the imputation process, are shown in Table II, while the best scores are bold highlighted.

TABLE II
TOTAL WINS OF EACH IMPUTATION METHOD

Imputation Method	***Missing Ratio***			***Classifier***	
	30%	**40%**	**60%**	**RF**	**JRip**
Mean Mode	3	4	0	6	1
FKMeans	2	2	6	5	5
LLSimpute	7	6	4	9	8
SVDimpute	3	1	0	1	3
IRMI	5	3	4	3	9
IRSSI	**11**	**14**	**16**	**22**	**19**

In more detail:

- Depending upon the missing ratio, IRSSI is found to prevail in all three scenarios. More precisely, IRSSI scores the highest accuracy values in 11, 14 and 16 datasets using a missing ratio of 30%, 40% and 60% respectively, followed by LLSimpute (7 and 6 datasets) and Fuzzy *k*-means (6 datasets).

TABLE III
ACCURACY AND STANDARD DEVIATION RESULTS (ACC_{30} %, STD_{30} %)

Dataset	***Rotation Forest***						***JRip***					
	Mean Mode	***FKMeans***	***LLS***	***SVD***	***IRMI***	***IRSSI***	***Mean Mode***	***FKMeans***	***LLS***	***SVD***	***IRMI***	***IRSSI***
anneal	98.441	97.883	97.773	84.522	98.105	**98.552**	95.989	95.767	96.217	76.408	96.437	**96.993**
	±1.134	±1.164	±0.997	±1.436	±1.122	±1.223	±1.670	±2.324	±1.803	±24.051	±1.195	±1.728
audiology	67.708	67.688	**71.739**	39.051	68.202	**71.739**	57.549	58.439	63.379	32.826	58.439	**64.229**
	±8.244	±8.305	±7.659	±14.164	±7.071	±10.400	±8.905	±9.090	±9.797	±8.116	±9.460	±11.577
breast-cancer	71.281	71.256	70.973	67.759	71.010	**72.020**	68.879	68.522	69.926	69.544	**71.675**	70.591
	±8.971	±8.443	±7.395	±7.250	±4.780	±6.879	±6.302	±6.248	±8.227	±10.177	±8.183	±4.917
chess	96.057	96.057	97.309	58.664	97.278	**97.340**	96.527	96.527	96.902	54.539	96.496	**96.871**
	±1.555	±1.555	±1.644	±5.972	±1.102	±1.191	±0.834	±0.834	±1.005	±2.878	±0.849	±0.895
cmc	51.400	48.548	**53.230**	43.583	50.782	50.652	48.943	48.340	**50.509**	49.628	47.590	50.110
	±3.607	±3.569	±3.690	±2.588	±2.519	±3.673	±4.015	±3.191	±3.849	±3.990	±2.218	±4.815
credit-rating	84.928	85.507	83.623	70.145	**85.652**	85.362	83.478	82.899	83.478	76.812	**84.493**	84.348
	±6.190	±3.490	±4.852	±11.470	±5.517	±4.604	±6.190	±3.417	±5.947	±8.500	±4.895	±5.177
haberman	**75.161**	72.871	72.215	73.516	73.527	74.516	73.882	72.570	72.882	**74.505**	72.559	73.204
	±2.983	±4.355	±5.333	±4.263	±1.724	±2.360	±5.921	±7.731	±6.204	±3.531	±4.062	±4.595
horse-colic	**83.138**	83.116	81.246	79.069	81.269	81.794	80.420	80.105	79.084	**81.502**	78.236	81.494
	±6.323	±7.180	±6.702	±5.024	±6.748	±2.100	±6.751	±8.683	±9.747	±6.445	±8.962	±7.718
housevotes	93.328	93.328	**97.389**	93.628	95.865	97.019	92.265	92.265	**97.019**	91.517	94.109	96.648
	±5.996	±5.996	±3.688	±5.653	±5.462	±3.592	±5.696	±5.696	±3.592	±9.588	±6.299	±3.830
iris	94.000	**96.667**	94.667	90.667	95.333	94.667	94.667	94.667	93.333	82.000	94.667	**95.333**
	±8.138	±4.472	±8.327	±8.537	±6.000	±7.775	±4.989	±4.000	±7.888	±8.969	±5.812	±6.000
kr-vs-kp	96.903	96.903	97.872	59.828	97.529	**98.123**	96.778	96.778	**97.340**	57.510	96.747	96.841
	±1.214	±1.214	±0.814	±3.992	±1.139	±0.655	±0.751	±0.751	±0.952	±4.616	±1.194	±1.371
solar-flare	69.309	69.309	68.409	67.462	71.193	**71.837**	68.371	68.371	69.347	60.047	**72.727**	68.693
	±6.305	±6.305	±5.979	±5.537	±6.085	±4.834	±5.558	±5.580	±8.579	±5.846	±5.914	±5.415
soybian	**92.675**	92.675	84.028	29.280	89.454	92.387	84.171	85.341	73.483	19.618	**85.345**	83.593
	±2.795	±2.795	±4.841	±5.424	±4.554	±2.344	±6.499	±6.057	±5.992	±2.272	±4.338	±7.191
tae	49.042	**55.542**	50.917	43.625	52.875	50.833	38.458	41.125	39.083	**44.375**	43.708	38.458
	±9.643	±13.208	±12.923	±10.325	±11.605	±14.630	±4.314	±9.986	±8.680	±11.928	±8.491	±10.788
vote	93.330	93.092	**95.613**	87.532	94.688	95.375	92.172	92.400	94.699	91.221	93.996	**94.931**
	±3.162	±2.926	±3.353	±8.029	±3.141	±2.946	±4.146	±3.875	±2.567	±6.237	±3.616	±2.262
Total wins	3	2	4	0	1	**6**	0	0	3	3	4	**5**

TABLE IV
ACCURACY AND STANDARD DEVIATION RESULTS (ACC$_{40}$ %, STD$_{40}$ %)

Dataset	***Rotation Forest***						***JRip***					
	Mean Mode	***FKMeans***	***LLS***	***SVD***	***IRMI***	***IRSSI***	***Mean Mode***	***FKMeans***	***LLS***	***SVD***	***IRMI***	***IRSSI***
anneal	96.770	96.881	97.328	82.291	97.104	**97.549**	94.876	95.765	94.655	81.846	95.546	**96.428**
	±1.955	±1.853	±2.342	±1.931	±1.511	±2.158	±2.547	±1.920	±2.265	±2.963	±1.650	±2.879
audiology	64.565	67.668	63.281	29.269	64.111	**69.466**	55.336	53.043	53.518	16.759	52.668	**57.984**
	±7.189	±9.191	±7.636	±13.453	±10.722	±7.616	±9.523	±7.961	±7.469	±12.773	±10.516	±5.437
breast-cancer	67.808	67.438	69.224	66.798	70.973	**72.414**	70.616	68.830	68.190	70.628	71.330	**72.365**
	±6.105	±6.437	±6.050	±6.664	±6.643	±6.797	±11.865	±5.103	±6.839	±4.482	±8.320	±10.348
chess	93.898	93.898	96.308	54.506	95.337	**97.028**	95.369	95.369	95.338	54.600	**95.557**	95.182
	±1.588	±1.588	±1.074	±2.202	±1.288	±1.275	±1.421	±1.421	±1.438	±1.481	±1.735	±1.731
cmc	**50.770**	50.044	48.207	43.249	48.470	49.897	46.432	**48.606**	47.448	47.521	48.265	48.132
	±4.701	±4.124	±4.282	±4.055	±4.328	±2.900	±3.747	±4.056	±5.630	±3.384	±4.366	±3.088
credit-rating	**85.797**	84.058	84.493	61.594	83.768	83.913	82.174	82.899	83.768	56.232	82.319	**85.072**
	±2.029	±4.443	±4.809	±7.195	±2.884	±4.913	±4.203	±4.140	±6.210	±8.642	±2.493	±4.895
haberman	73.194	72.860	71.570	**73.849**	73.505	73.516	74.151	73.871	**74.516**	73.828	74.172	74.495
	±2.543	±3.375	±7.676	±4.119	±3.258	±3.138	±3.923	±7.179	±5.043	±4.320	±3.108	±2.539
horse-colic	82.327	79.610	79.077	79.069	81.787	**84.512**	79.092	80.405	74.992	81.231	**83.138**	81.224
	±3.720	±6.622	±6.523	±9.462	±5.432	±4.544	±3.345	±7.432	±6.180	±6.285	±5.296	±5.574
housevotes	93.023	93.023	96.492	92.942	90.199	**97.019**	93.405	92.598	96.250	88.767	88.246	**97.019**
	±4.820	±4.820	±4.219	±8.680	±6.475	±3.592	±5.946	±5.573	±5.603	±9.209	±8.243	±3.592
iris	94.667	92.667	**96.000**	86.000	94.000	94.667	92.667	91.333	93.333	81.333	91.333	**94.000**
	±6.532	±6.289	±6.110	±10.088	±5.538	±4.989	±8.667	±7.916	±7.888	±9.333	±6.000	±4.667
kr-vs-kp	94.367	95.527	**97.090**	56.947	96.370	96.558	94.930	95.744	**95.871**	52.410	95.119	95.120
	±2.101	±1.616	±1.019	±3.800	±1.155	±1.018	±1.284	±1.031	±1.415	±4.575	±1.349	±1.740
solar-flare	68.741	68.703	67.140	68.409	**72.737**	71.828	69.669	**69.953**	65.616	67.169	68.106	69.015
	±4.404	±6.080	±5.975	±8.847	±5.952	±5.088	±5.090	±4.699	±7.513	±9.393	±4.416	±4.531
soybian	91.355	92.229	81.974	24.738	88.866	**92.240**	77.415	78.781	72.163	16.839	79.060	**83.001**
	±2.040	±4.122	±5.388	±6.192	±3.106	±2.632	±9.007	±6.510	±5.048	±6.527	±2.189	±4.557
tae	**55.583**	49.458	49.542	39.708	50.250	51.667	**41.000**	37.792	38.417	37.042	39.667	37.042
	±10.904	±16.351	±10.352	±8.245	±14.260	±8.851	±6.675	±6.192	±10.662	±9.740	±9.939	±5.714
vote	92.870	92.400	**96.305**	89.175	93.124	94.255	91.052	91.940	**95.846**	88.436	94.033	94.725
	±3.794	±4.006	±2.983	±6.750	±3.365	±4.000	±7.109	±3.888	±2.917	±10.164	±3.989	±3.673
Total wins	3	0	3	1	1	**7**	1	2	3	0	2	**7**

TABLE V
ACCURACY AND STANDARD DEVIATION RESULTS (ACC$_{60}$ %, STD$_{60}$ %)

Dataset	***Rotation Forest***						***JRip***					
	Mean Mode	***FKMeans***	***LLS***	***SVD***	***IRMI***	***IRSSI***	***Mean Mode***	***FKMeans***	***LLS***	***SVD***	***IRMI***	***IRSSI***
anneal	94.211	**94.434**	88.421	79.075	94.100	93.206	92.096	**93.879**	89.534	71.077	90.860	90.964
	±3.098	±1.853	±2.342	±1.931	±2.583	±1.966	±3.531	±1.920	±2.265	±2.963	±3.507	±6.940
audiology	52.194	**59.368**	49.111	35.889	55.692	54.407	40.119	41.700	40.692	22.213	31.403	**42.964**
	±12.037	±9.191	±7.636	±13.453	±7.346	±11.399	±10.054	±7.961	±7.469	±12.773	±6.257	±7.307
breast-cancer	69.581	68.805	66.367	62.931	71.712	**72.020**	70.271	67.426	68.190	58.091	67.845	**71.687**
	±8.481	±6.437	±6.050	±6.664	±6.966	±3.149	±8.960	±5.103	±6.839	±4.482	±5.702	±2.782
chess	90.771	90.771	89.299	51.779	88.362	**93.555**	89.364	89.364	88.705	52.877	90.645	**93.710**
	±3.368	±1.588	±1.074	±2.202	±1.854	±1.354	±2.749	±1.421	±1.438	±1.481	±1.288	±1.275
cmc	48.274	47.723	48.135	41.206	47.315	**49.290**	45.694	46.642	44.609	43.109	44.265	**47.323**
	±4.729	±4.124	±4.282	±4.055	±3.629	±3.921	±4.487	±4.056	±5.630	±3.384	±2.643	±3.338
credit-rating	83.768	83.478	82.029	57.391	82.029	**83.913**	78.986	81.014	82.609	54.928	78.116	**83.333**
	±5.826	±4.443	±4.809	±7.195	±3.562	±5.667	±7.137	±4.140	±6.210	±8.642	±6.130	±6.618
haberman	73.204	72.194	66.602	70.602	**73.849**	72.882	74.140	71.892	71.925	±73.215	**75.785**	74.484
	±2.790	±3.375	±7.676	±4.119	±2.534	±1.944	±4.921	±7.179	±5.043	4.320	±3.796	±4.217
horse-colic	79.077	**79.332**	79.077	70.165	73.664	78.784	75.255	77.200	73.649	71.209	70.653	**77.980**
	±7.927	±6.622	±6.523	±9.462	±5.450	±6.991	±8.332	±7.432	±6.180	±6.285	±6.489	±5.273
housevotes	90.404	92.509	**95.809**	55.285	90.061	90.602	86.920	86.764	**96.264**	61.434	88.104	93.852
	±7.950	±4.820	±4.219	±8.680	±6.258	±4.233	±5.289	±5.573	±5.603	±9.209	±6.475	±3.592
iris	90.667	91.333	92.000	66.000	92.000	**94.667**	88.000	**96.000**	89.333	60.000	90.000	90.667
	±6.799	±6.289	±6.110	±10.088	±5.812	±4.989	±8.327	±7.916	±7.888	±9.333	±9.545	±11.235
kr-vs-kp	88.674	91.240	91.208	48.216	89.550	**91.802**	89.612	90.583	92.178	48.092	**92.304**	90.053
	±3.574	±1.616	±1.019	±3.800	±2.974	±2.357	±3.804	±1.031	±1.415	±4.575	±3.210	±5.013
solar-flare	65.606	66.231	62.197	64.688	61.544	**70.275**	64.375	**67.756**	61.004	59.763	63.759	65.322
	±6.397	±6.080	±5.975	±8.847	±8.349	±10.267	±5.159	±4.699	±7.513	±9.393	±5.084	±5.360
soybian	83.738	83.734	67.466	14.352	83.892	**85.339**	63.534	65.126	55.309	7.020	**74.235**	71.281
	±4.917	±4.122	±5.388	±6.192	±3.527	±5.339	±7.595	±6.510	±5.048	±6.527	±2.918	±5.436
tae	44.292	41.625	38.917	28.542	43.083	**44.958**	39.000	35.750	39.750	34.458	33.750	**41.667**
	±8.894	±16.351	±10.352	±8.245	±11.353	±11.815	±12.828	±6.192	±10.662	±9.740	±6.092	±10.028
vote	92.172	89.905	**95.846**	60.428	89.884	93.584	85.756	87.579	**95.835**	65.465	89.218	94.260
	±3.780	±4.006	±2.983	±6.750	±4.397	±3.905	±4.213	±3.888	±2.917	±10.164	±4.415	±3.867
Total wins	0	3	2	0	1	**9**	0	3	2	0	3	**7**

TABLE VI
FRIEDMAN TEST AND LI POST HOC TEST (α=0.05) FOR 30% MISSING RATIO

Classifier after Imputation	***Friedman test***					***Li post hoc test***	
	Statistic	***p-value***	***Null Hypothesis***	***Imputation Method***	***Ranking***	***p-value***	***Null Hypothesis***
RF	9.62331	0.00000	rejected	IRSSI	2.20000	-	-
				IRMI	2.86667	0.32911	accepted
				Mean/Mode	3.30000	0.13794	accepted
				LLSimpute	3.40000	0.10533	accepted
				FKMeans	3.50000	0.07836	accepted
				SVDimpute	5.73333	0.00000	rejected
JRip	4.18707	0.00219	rejected	IRSSI	2.23333	-	-
				LLSimpute	2.83333	0.37978	accepted
				IRMI	3.23333	0.18761	accepted
				Mean/Mode	3.93333	0.02026	rejected
				FKMeans	4.10000	0.01003	rejected
				SVDimpute	4.66667	0.00059	rejected

TABLE VII
FRIEDMAN TEST AND LI POST HOC TEST (α=0.05) FOR 40% MISSING RATIO

Classifier after Imputation	***Friedman test***					***Li post hoc test***	
	Statistic	***p-value***	***Null Hypothesis***	***Imputation Method***	***Ranking***	***p-value***	***Null Hypothesis***
RF	14.13218	0.00000	rejected	IRSSI	1.86667	-	-
				FKMeans	2.83333	0.15705	accepted
				Mean/Mode	3.00000	0.10330	accepted
				IRMI	3.66667	0.00988	rejected
				LLSimpute	3.83333	0.00471	rejected
				SVDimpute	5.80000	0.00000	rejected
JRip	5.38973	0.00030	rejected	IRSSI	2.23333	-	-
				IRMI	3.10000	0.20456	accepted
				FKMeans	3.26667	0.14082	accepted
				LLSimpute	3.46667	0.08195	accepted
				Mean/Mode	3.70000	0.03844	rejected
				SVDimpute	5.23333	0.00001	rejected

TABLE VIII
FRIEDMAN TEST AND LI POST HOC TEST (α=0.05) FOR 60% MISSING RATIO

Classifier after Imputation	***Friedman test***					***Li post hoc test***	
	Statistic	***p-value***	***Null Hypothesis***	***Imputation Method***	***Ranking***	***p-value***	***Null Hypothesis***
RF	10.42943	0.00000	rejected	IRSSI	1.76667	-	-
				Mean/Mode	3.10000	0.05096	accepted
				LLSimpute	3.33333	0.02248	rejected
				IRMI	3.40000	0.01740	rejected
				FKMeans	3.86667	0.00222	rejected
				SVDimpute	5.53333	0.00000	rejected
JRip	12.75322	0.00000	rejected	IRSSI	1.73333	-	-
				FKMeans	3.03333	0.05704	accepted
				LLSimpute	3.40000	0.01535	rejected
				IRMI	3.53333	0.00885	rejected
				Mean/Mode	3.56667	0.00766	rejected
				SVDimpute	5.73333	0.00000	rejected

- Employing the RF classifier in the simulated datasets to evaluate the performance of the imputation methods, it is observed that IRSSI performs better in 22 datasets, followed by LLSimpute (9 datasets) and FKmeans (5 datasets). Similar results were obtained in the case of JRip. The IRSSI algorithm obtains better results in 19 datasets, followed by IRMI (9 datasets) and LLSimpute (8 datasets).
- Considering the influence of missing ratio, we can see that the following inequality holds:

  $$acc_{30} > acc_{40} > acc_{60}$$

  as could logically be expected. Besides that, it is worth noting that the IRSSI efficiency is improved as the missing ratio increases, thereby confirming the potential of SSL.
- As regards the estimated deviations, we observe small values in most datasets for all missing ratios, meaning that the average accuracies are close enough to the real ones.

In addition, an extensive statistical analysis of the results was carried out to confirm the performance of IRSSI. Therefore, we applied the Friedman non-parametric test [50] followed by the Li post hoc test [51] (significance level α=0.05). Both statistical tests are commonly used for comparing the performance of more than two methods [52]. According to the Friedman test results (Tables VI-VIII), the three imputation methods were sorted from the best performer (lowest ranking value) to the worst one (highest ranking value) for each of the six scenarios (three different missing ratios and two different classifiers). It is clearly shown in these tables that IRSSI prevails in all scenarios, while the remainder methods consistently have the lowest scores.

Since the null hypothesis H_o was rejected (i.e. the means of the results of the six methods are the same), the Li post hoc test was used for detecting the specific differences among them. Li's test is very powerful and produces better results than other tests, especially when testing multiple hypotheses. The post hoc results are also displayed in Tables VI-VIII using IRSSI as control method. It is worth noting that from the remainder methods, no one seems to prevail. To be more specific, LLSimpute and IRMI perform relatively well for the first two scenarios (i.e. for 30% missing ratio), while they lag behind IRSSI. For the next two scenarios (i.e. for 40% missing ratio) there does not appear a method which can compete IRSSI. Finally, in the last two scenarios (i.e. for 60% missing ratio), FKMeans is also achieving good results.

In addition, the performance of IRSSI is higher from its main rival (i.e. IRMI), as demonstrated by the experimental results and the statistical tests. It is therefore evident that IRMI benefits from the integration of the self-training process employed within the attribute fitting loop, thereby yielding a more robust and accurate imputation method, especially in datasets with large proportion of missing values. So, it becomes clear that IRSSI can efficiently handle the deficiency phenomenon in incomplete datasets of different structure and missing ratio values.

B. EXPERIMENTATION ON ARTIFICIAL DATA

In addition, a series of experiments were conducted utilizing artificially constructed data in order to reveal the efficiency of IRSSI in comparison with its main rival (i.e. IRMI) and against LLSimpute. Therefore, five random generated artificial datasets were applied. For each dataset, a random five-class classification problem was constructed. The feature values of the datasets were drawn from clusters of points normally distributed about vertices of an *n*-dimensional hypercube, where *n* is the number of informative features. A set of general parameters were selected with a view to generating robust random datasets. Table IX summarizes them.

Each artificial dataset is composed of seven features with five of them being informative and the rest of ones containing random noise. At the time of generation, all features where numerical, thus a discretization process was applied in five of the features. Since the initial features were drawn from a normal distribution with a standard deviation of 1.00, six bins were defined dividing the numerical values in six categorical ones with the following intervals:

(-100,-1.5),(-1.5,-0.75),(-0.75,0),(0,0.75),(0.75,1.5), (1.5,100)

Gaussian Noise was also applied in all features with a standard deviation of 0.4.

TABLE IX
ARTIFICIAL DATASETS PARAMETERS

Parameter	***Values***	***Details***
Sample Size	500	
Continuous Features	2	Numeric Features Real values
Discrete Features	5	Nominal Features Categorical values
Total Num. of Features	7	
Num. of Features Containing Information	5	Informative Features
Num. of Uninformative Features	2	
Gaussian Noise	0.4	Standard Deviation of injected Gaussian Noise
Num. of Artificial Datasets	5	Random Number Seeds: 15, 16, 17, 18, 19

In order to compare the performance of the rivals, the average root mean square error (Mean RMSE) of the feature vector differences between each original and imputed dataset instance was calculated for each imputation algorithm according to the following formula:

$$Mean\ RMSE = 1/n\sum_{i=1}^{n}\sqrt{1/k\sum_{j=1}^{k}|v_{i,j}^{original} - v_{i,j}^{imputed}|^2}, \quad (3)$$

where *n* is the number of instances, *k* is the number of features and $v_{i,j}$ is the corresponding feature value. All categorical features were transformed to their one-hot encoding [53] equivalents, thus the above equation could be easily applied. Since the generated artificial datasets were five, every error presented in this section on the figures is the averaged calcula-

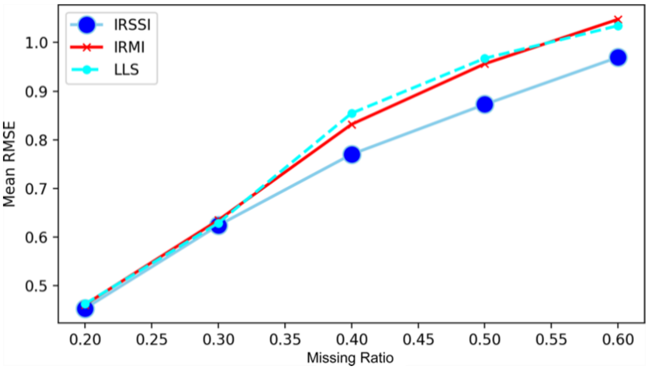

FIGURE 5. Comparison of imputation errors for IRSSI, IRMI and LLS on five different missing value ratios.

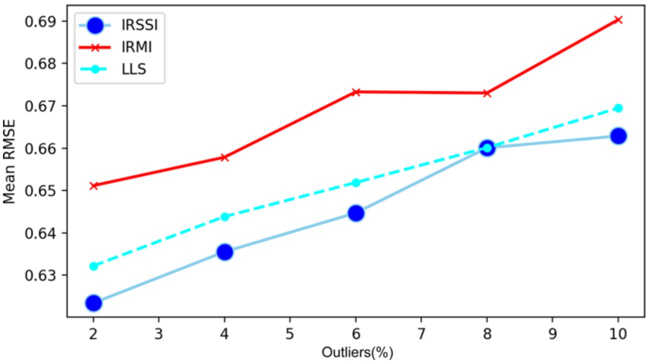

FIGURE 6. Comparison of imputation errors for IRSSI, IRMI and LLS on five different outlier ratios along with 30% ratio of missing values.

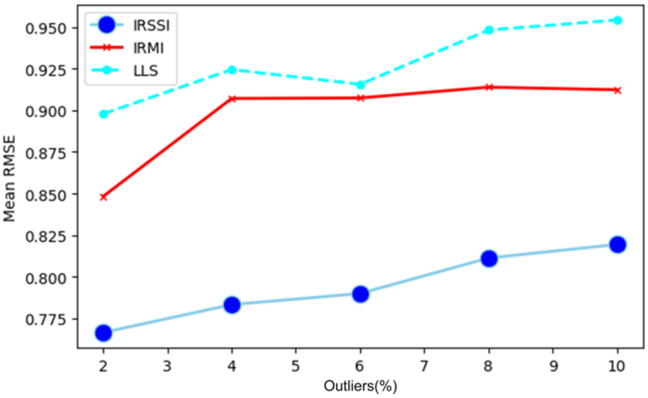

FIGURE 7. Comparison of imputation errors for IRSSI, IRMI and LLS on five different outlier ratios along with 40% ratio of missing values.

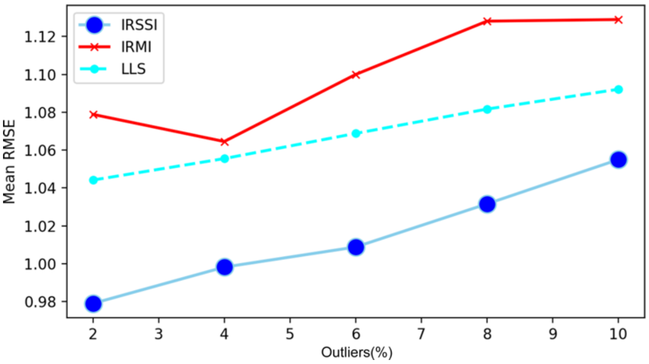

FIGURE 8. Comparison of imputation errors for IRSSI, IRMI and LLS on five different outlier ratios along with 60% ratio of missing values.

ted error over the five artificial datasets.

In the first experiment, we compare the behavior of the algorithms over different missing value ratios. In more detail, the five original artificial datasets were injected with missing values in ratios varying from 20% to 60% and the resulting datasets were fitted using the three imputation algorithms. The computed mean errors for each ratio are presented in Fig. 5. The three algorithms are close in terms of generated errors for very low missing ratios, whereas IRSSI is steadily producing lower imputation errors as the missing ratio increases.

In the second set of experiments, we compare the performance of the three algorithms regarding the presence of outliers. Therefore, the original artificial datasets were injected with outliers in five different ratios ranging from 2% to 10% (Fig. 6-8) and accordingly were injected missing values (30%, 40% and 60%). There is a clear predominance of IRSSI confirming that ensemble schemes tend to better handle outlier values [54].

Moreover, in order to observe the imputation capability of IRSSI and IRMI, a sixth artificial dataset was generated containing only three informative features (two numeric and one categorical), three classes and a hundred samples. This dataset was injected with 50% missing values and was imputed using the two algorithms. The original dataset clusters along with the imputation-generated clusters can be observed in Fig. 9. For comparing the quality (compactness and separation) of the generated clusters, two comparison indices were applied on the two imputed datasets. The first index is the Dunn index (DI) [55], an internal cluster valuation scheme. Higher index values indicate better clustering and is calculated as follows:

$$DI = \min_{1\le i\le c}\left\{ \min_{1\le j\le c,\, j\ne i}\left\{ \frac{\delta(X_i, X_j)}{\max\limits_{1\le k\le c}(\Delta X_k)} \right\}\right\}, \qquad (4)$$

where c is the number of clusters, $\delta(X_i, X_j)$ is the inter-cluster distance between clusters X_i and X_j, and ΔX_k is the intra-cluster distance of cluster X_k.

The second one is the Davies-Bouldin index (DBI) [56], formulated in (5). The clustering quality is judged using quantities and features inherent to the dataset. Lower DBI values indicate better separation and tightness of the clusters.

$$DBI = 1/k\sum_{i=1}^{k}\max_{i\ne j}\left\{ \frac{\Delta X_i + \Delta X_j}{\delta(X_i, X_j)} \right\}, \qquad (5)$$

where $\delta(X_i, X_j)$ and ΔX_i, ΔX_j as above symbolize the inter-cluster and instar-cluster distances accordingly.

Table X summarizes the computed indices, which reveal a slightly better clustering behavior for the IRSSI algorithm.

TABLE X
DUN AND DAVIES-BOULDIN INDICES

Dataset	***DI***	***DBI***
IRMI Imputed	0.138	3.137
IRSSI Imputed	0.142	2.890

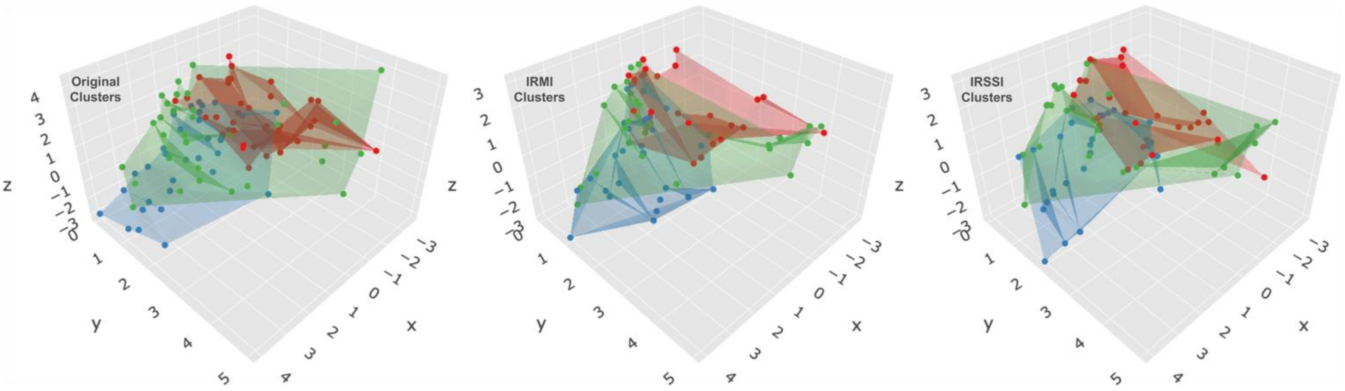

FIGURE 9. Original dataset clusters versus IRMI and IRSSI imputed dataset clusters.

Finally, a meta-dataset was constructed with a view to extracting meaningful rules regarding the performance of IRSSI in connection with the structure of datasets. To this end, Tables I, III, IV and V were combined, thus producing the meta-dataset. The derived binary class feature indicates whether the IRSSI outperforms the rest compared methods in each case. The rules were automatically extracted using Decision trees and RIPPER algorithms. In summary, two general rules were constructed: (1) If the dataset structure is mainly consisting of nominal features, then IRSSI displays strong performance characteristics. (2) The performance of IRSSI is significantly increasing as the missing ratio increases.

V. CONCLUSIONS

In the present study, we proposed a new hybrid imputation method based on robust semi-supervised ensembles. The Iterative Robust Semi-Supervised based Imputation algorithm (IRSSI) is a refined version of the IRMI algorithm, harnessing the benefits of SSL in a simple and efficient manner. The experimental results on fifteen benchmark datasets, using different and high ratios of missing data (30%, 40% and 60%) and two typical classifiers after the imputation process (RF and JRip), favor IRSSI compared to familiar imputation methods: the Mean/Mode statistical method, the Fuzzy k-means single imputation method, LLSimpute, SVDimpute and the IRMI as the baseline method. Furthermore, the behavior of the rivals was examined on artificially generated datasets, considering a variety of missing value ratios and the presence of extreme outliers. The comparison between IRSSI, IRMI and LLSimpute verifies the superiority of the proposed method, as statistically confirmed by the Friedman non-parametric test and the Li post hoc test.

It is worth considering a few ideas to further improve the proposed algorithm. The first one concerns the utilization of parallel execution capabilities of the modern processing units. Several design changes in the flow of the algorithm (e.g. employ a more sophisticated flow for the calculation of multiple depended attribute responses at once) would enable IRSSI to impute the dataset's attributes in a more parallelized manner and increase its throughput. Another step on this direction is the modification of the inner procedures of IRSSI (step 4) to embrace prediction models that are suitable to be executed in GPUs. Such advancements could make the algorithm suitable for big data analysis or data streaming problems in combination with deep learning methods.

In addition, the investigation of the performance of IRSSI on tackling other machine learning problems seems an interesting area for future research. For example, the examination of algorithm efficacy when applying an imputation method together with clustering algorithms like density-based spatial clustering (DBSCAN) [57] or balanced iterative reducing and clustering (BIRCH) [58]. Furthermore, the application of IRSSI as imputation method to enhance regression datasets could also increase the data correlation on regression or even on time series-based problems.

Finally, embedding filters for handling outliers and extreme values for the imputed data, would have an immediate positive effect on the accuracy of the IRSSI. Filtering algorithms, such as local outlier factor [59] for detecting anomalous values based on neighboring data or Isolation Forest [60], a tree-based outlier detector, can be a perfect fit for application within the proposed algorithm.

APPENDIX

A full implementation of the IRSSI algorithm was developed in java and implemented for the WEKA [61] software tool, which offers a user-friendly graphical interface and supports a plethora of classification, regression and clustering algorithms. Our implementation is publicly available as a separate package at https://github.com/fazakis/semi-supervised-missing-values-imputation-weka-package , while the algorithm is located under the filters menu of WEKA.

REFERENCES

[1] M. Li and X. Zhang, "Information fusion in a multi-source incomplete information system based on information entropy," *Entropy*, vol. 19, no. 11, p. 570, 2017.

[2] A. Farhangfar, L. Kurgan, and J. Dy, "Impact of imputation of missing values on classification error for discrete data," *Pattern*

Recognit., vol. 41, no. 12, pp. 3692–3705, 2008.

[3] D. B. Rubin, *Multiple imputation for nonresponse in surveys*, vol. 81. John Wiley & Sons, 2004.

[4] M. Saar-Tsechansky and F. Provost, “Handling missing values when applying classification models,” *J. Mach. Learn. Res.*, vol. 8, no. Jul, pp. 1623–1657, 2007.

[5] M. L. Yadav and B. Roychoudhury, “Handling missing values: A study of popular imputation packages in R,” *Knowledge-Based Syst.*, vol. 160, pp. 104–118, 2018.

[6] J. W. Grzymala-Busse and W. J. Grzymala-Busse, “Handling missing attribute values,” in *Data mining and knowledge discovery handbook*, Springer, 2009, pp. 33–51.

[7] S. Van Buuren, *Flexible imputation of missing data*. Chapman and Hall/CRC, 2018.

[8] L. Beretta and A. Santaniello, “Nearest neighbor imputation algorithms: a critical evaluation,” *BMC Med. Inform. Decis. Mak.*, vol. 16, no. 3, p. 74, 2016.

[9] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. Wiley, 2019.

[10] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*, vol. 72. Springer, 2015.

[11] A. N. Baraldi and C. K. Enders, “An introduction to modern missing data analyses,” *J. Sch. Psychol.*, vol. 48, no. 1, pp. 5–37, 2010.

[12] J. L. Schafer, *Analysis of incomplete multivariate data*. Chapman and Hall/CRC, 1997.

[13] A. Suyundikov, J. R. Stevens, C. Corcoran, J. Herrick, R. K. Wolff, and M. L. Slattery, “Accounting for dependence induced by weighted KNN imputation in paired samples, motivated by a colorectal cancer study,” *PLoS One*, vol. 10, no. 4, p. e0119876, 2015.

[14] X. Xu, W. Chong, S. Li, A. Arabo, and J. Xiao, “MIAEC: Missing data imputation based on the evidence Chain,” *IEEE Access*, 2018.

[15] W.-C. Lin and C.-F. Tsai, “Missing value imputation: a review and analysis of the literature (2006--2017),” *Artif. Intell. Rev.*, pp. 1–23, 2019.

[16] Z.-H. Zhou, “A brief introduction to weakly supervised learning,” *Natl. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2017.

[17] X. J. Zhu, “Semi-supervised learning literature survey,” 2005.

[18] C.-B. Lu and Y. Mei, “An Imputation Method for Missing Data Based on an Extreme Learning Machine Auto-Encoder,” *IEEE Access*, vol. 6, pp. 52930–52935, 2018.

[19] J. L. Schafer and J. W. Graham, “Missing data: our view of the state of the art.,” *Psychol. Methods*, vol. 7, no. 2, p. 147, 2002.

[20] P. J. García-Laencina, J. L. Sancho-Gómez, and A. R. Figueiras-Vidal, “Pattern classification with missing data: A review,” *Neural Comput. Appl.*, vol. 19, no. 2, pp. 263–282, 2010.

[21] M. Templ, A. Kowarik, and P. Filzmoser, “Iterative stepwise regression imputation using standard and robust methods,” *Comput. Stat. Data Anal.*, vol. 55, no. 10, pp. 2793–2806, 2011.

[22] G. E. Batista and M. C. Monard, “An analysis of four missing data treatment methods for supervised learning,” *Appl. Artif. Intell.*, vol. 17, no. 5–6, pp. 519–533, 2003.

[23] J. W. Grzymala-Busse, L. K. Goodwin, W. J. Grzymala-Busse, and X. Zheng, “Handling missing attribute values in preterm birth data sets,” in *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, 2005, pp. 342–351.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. R. Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–22, 1977.

[25] M. M. Jenghara, H. Ebrahimpour-Komleh, V. Rezaie, S. Nejatian, H. Parvin, and S. K. S. Yusof, “Imputing missing value through ensemble concept based on statistical measures,” *Knowl. Inf. Syst.*, vol. 56, no. 1, pp. 123–139, 2018.

[26] O. Troyanskaya *et al.*, “Missing value estimation methods for DNA microarrays,” *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[27] V. Kumutha and S. Palaniammal, “An enhanced approach on handling missing values using bagging k-NN imputation,” in *2013 International Conference on Computer Communication and Informatics*, 2013, pp. 1–8.

[28] B. Twala, “An empirical comparison of techniques for handling incomplete data using decision trees,” *Appl. Artif. Intell.*, vol. 23, no. 5, pp. 373–405, 2009.

[29] X. Feng, S. Wu, and Y. Liu, “Imputing missing values for mixed numeric and categorical attributes based on incomplete data hierarchical clustering,” in *International Conference on Knowledge Science, Engineering and Management*, 2011, pp. 414–424.

[30] B. M. Patil, R. C. Joshi, and D. Toshniwal, “Missing value imputation based on k-mean clustering with weighted distance,” in *International Conference on Contemporary Computing*, 2010, pp. 600–609.

[31] T. E. Raghunathan, J. M. Lepkowski, J. Van Hoewyk, P. Solenberger, and others, “A multivariate technique for multiply imputing missing values using a sequence of regression models,” *Surv. Methodol.*, vol. 27, no. 1, pp. 85–96, 2001.

[32] D. B. Rubin, “Multiple imputations in sample surveys-a phenomenological Bayesian approach to nonresponse,” in *Proceedings of the survey research methods section of the American Statistical Association*, 1978, vol. 1, pp. 20–34.

[33] P. Royston, “Multiple imputation of missing values,” *Stata J.*, vol. 4, no. 3, pp. 227–241, 2004.

[34] G. Rahman and Z. Islam, “A decision tree-based missing value imputation technique for data pre-processing,” in *Proceedings of the Ninth Australasian Data Mining Conference-Volume 121*, 2011, pp. 41–50.

[35] J. Luengo, S. García, and F. Herrera, “On the choice of the best imputation methods for missing values considering three groups of classification methods,” *Knowl. Inf. Syst.*, vol. 32, no. 1, pp. 77–108, 2012.

[36] G. Kostopoulos, S. Karlos, S. Kotsiantis, and O. Ragos, “Semi-supervised regression: A recent review,” *J. Intell. Fuzzy Syst.*, vol. 35, no. 2, pp. 1483–1500, 2018.

[37] N. Fazakis, S. Karlos, S. Kotsiantis, and K. Sgarbas, “A multi-scheme semi-supervised regression approach,” *Pattern Recognit. Lett.*, vol. 125, pp. 758–765, Jul. 2019.

[38] N. Fazakis, S. Karlos, S. Kotsiantis, and K. Sgarbas, “Self-trained Rotation Forest for semi-supervised learning,” *J. Intell. Fuzzy Syst.*, vol. 32, no. 1, pp. 711–722, 2017.

[39] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, 2009.

[40] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[41] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[42] J. R. Quinlan, “Learning with continuous classes,” *Mach. Learn.*, vol. 92, pp. 343–348, 1992.

[43] Y. Wang and I. H. Witten, “Induction of model trees for predicting continuous classes,” 1996.

[44] A. Asuncion and D. Newman, “UCI machine learning repository.” 2007.

[45] D. Li, J. Deogun, W. Spaulding, and B. Shuart, “Towards missing data imputation: a study of fuzzy k-means clustering method,” in *International conference on rough sets and current trends in computing*, 2004, pp. 573–579.

[46] H. Kim, G. H. Golub, and H. Park, “Missing value estimation for DNA microarray gene expression data: local least squares imputation,” *Bioinformatics*, vol. 21, no. 2, pp. 187–198, 2005.

[47] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, “Rotation forest: A new classifier ensemble method,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, 2006.

[48] W. W. Cohen, “Fast effective rule induction,” in *Machine Learning Proceedings 1995*, Elsevier, 1995, pp. 115–123.

[49] M. Kubat, *An introduction to machine learning*, vol. 2. Springer, 2017.

[50] M. Friedman, “The use of ranks to avoid the assumption of normality implicit in the analysis of variance,” *J. Am. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.

[51] J. D. Li, “A two-step rejection procedure for testing multiple hypotheses,” *J. Stat. Plan. Inference*, vol. 138, no. 6, pp. 1521–

1527, 2008.

[52] S. García and A. Fernández, “Advanced nonparametric tests for multiple comparisons in the design of experiments in computational ...,” *Inf. Sci. (Ny).*, vol. 180, no. 10, pp. 2044–2064, 2017.

[53] R. A. Horn, C. R. Johnson, R. A. Horn, and C. R. Johnson, “Norms for vectors and matrices,” in *Matrix analysis*, 2013.

[54] D. M. Harris and S. L. Harris, *Digital Design and Computer Architecture*. 2007.

[55] M. Mohandes, M. Deriche, and S. O. Aliyu, “Classifiers Combination Techniques: A Comprehensive Review,” *IEEE Access*. 2018.

[56] J. C. Dunn, “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters,” *J. Cybern.*, 1973.

[57] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 1979.

[58] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, and others, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *Kdd*, 1996, vol. 96, no. 34, pp. 226–231.

[59] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: an efficient data clustering method for very large databases,” in *ACM Sigmod Record*, 1996, vol. 25, no. 2, pp. 103–114.

[60] W. Wang and P. Lu, “An efficient switching median filter based on local outlier factor,” *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 551–554, 2011.

[61] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[62] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

NIKOS FAZAKIS is a PhD candidate in the Department of Electrical and Computer Engineering at the University of Patras, Greece. He received his diploma from the same department and he also holds an MBA. He has participated in numerous European and National research programs and he has a variety of publications in the fields of machine learning and data mining.

Dr. **GEORGIOS KOSTOPOULOS** is a high school mathematics teacher. He received his master’s and doctor's degrees in Computer Science and Educational Data Mining respectively from University of Patras, Greece (Department of Mathematics). He has several publications with enough citations. His research interests are in the field of machine learning, learning analytics and educational data mining.

SOTIRIS KOTSIANTIS is an Assistant Professor in the Department of Mathematics at the University of Patras, Greece. He is a mathematician and holds a Master as well as a PhD in Computer Science from the University of Patras, Greece. He has a lot of publications with numerous citations. His research interests are in the field of data mining, machine learning and educational data mining.

IOSIF MPORAS is Senior Lecturer in Information Engineering at the University of Hertfordshire UK. His research interests are in machine learning and processing of heterogeneous information. He has participated in more than 10 R&D projects and has served as PC member, Chair of international conferences and Guest Editor in special issues of journals related to AI applications, respectively. He has authored and co-authored more than 100 scientific articles in international journals and conference proceedings, which have been cited more than 893 times.