

# Handling incomplete data classification using imputed feature selected bagging (IFBag) method

Ahmad Jaffar Khan<sup>a</sup>, Basit Raza<sup>a,\*</sup>, Ahmad Raza Shahid<sup>a</sup>, Yogan Jaya Kumar<sup>b</sup>,  
Muhammad Faheem<sup>c</sup> and Hani Alquhayz<sup>d</sup>

<sup>a</sup>*Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan*

<sup>b</sup>*Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia*

<sup>c</sup>*Department of Computer Engineering, Abdullah Gul University, Kayseri, Turkey*

<sup>d</sup>*Department of Computer Science and Information, College of Science in Zulfi, Majmaah University, Al-Majmaah, Saudi Arabia*

**Abstract.** Almost all real-world datasets contain missing values. Classification of data with missing values can adversely affect the performance of a classifier if not handled correctly. A common approach used for classification with incomplete data is imputation. Imputation transforms incomplete data with missing values to complete data. Single imputation methods are mostly less accurate than multiple imputation methods which are often computationally much more expensive. This study proposes an imputed feature selected bagging (IFBag) method which uses multiple imputation, feature selection and bagging ensemble learning approach to construct a number of base classifiers to classify new incomplete instances without any need for imputation in testing phase. In bagging ensemble learning approach, data is resampled multiple times with substitution, which can lead to diversity in data thus resulting in more accurate classifiers. The experimental results show the proposed IFBag method is considerably fast and gives 97.26% accuracy for classification with incomplete data as compared to common methods used.

Keywords: Incomplete data, machine learning, data classification, feature selection, ensemble learning

## 1. Introduction

Data classification is one of the tasks performed in data mining and machine-learning. Classification consists of training and testing phase [1]. The training phase is used to build a classifier on a specific training data set, while testing phase is used to classify new instances using learned classifier. Various classification strategies in domains such as medical diagnosis, credit card fraud and facial recognition have been applied successfully. Predictive algorithms such as supporting vector machines, neural network and logistic regression mostly require complete data for classification problems, while these algorithms do not work with incomplete data which has missing values. It is quite common to find missing values in real-world datasets [2]. About 45% of all the datasets available on UCI machine learning repository, which

---

\*Corresponding author: Basit Raza, Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad 45550, Pakistan. E-mail: basit.raza@comsats.edu.pk.

is the most commonly used repository, have missing values [3]. During the data collection process, results can be missing due to various reasons. For example, machine failure during the industrial experiment, respondents ignoring some questions during social surveys, not all tests are carried out on every patient, and data change in the financial datasets can cause missing values in datasets [4,5]. Thus, handling missing values is an important part of data mining. Inappropriate handling of missing values can have serious impact on performance of predictive models which could lead to invalid conclusions.

The simplest way to handle missing value problem is by deleting instances [6], which contain even a single missing attribute. However, this approach of listwise deletion can introduce biases in the study as demonstrated in [7] and this is more likely to happen when the missing data are not randomly distributed. Thus, most commonly used approach of classification with incomplete data are to substitute missing values with probable values using imputation methods [11]. Thus far, many of imputation methods have been developed [8–10]. For example, when mean imputation is applied it substitutes features all missing values with median of all the values of the same feature, thus providing a complete data on which any predictive algorithm can be used. Simple/single imputation methods like mean imputation are not very accurate as they treat all values including imputed values as true values while they do not take into consideration the uncertainty of the missing value [12]. However, powerful methods of imputation such as multiple imputations methods are often more precise than single imputation, while being more computationally intensive [13]. When combining both classification algorithms and imputation, it is not easy to determine whether it will be both effective and efficient, mostly during the testing phase.

Ensemble learning is a machine learning model in which a collection of base classifiers is used for classification problem and could improve classification accuracy [14]. Ensemble learning model, when implemented for classification with missing data, builds multiple classifiers in the training phase. In testing phase, each trained classifier classifies incomplete instance while giving majority voted class as class output [15]. Nonetheless, classification with incomplete data is often not possible for the datasets with large number of missing values using available ensemble methods [16]. Hence, work must be done to enable ensemble learning models to classify instances even when a dataset has numerous missing values. The process of selecting significant features from initial features is known as feature selection, which is extensively used on complete datasets in order to improve classification [17]. Studies on the selection of features have also been done on incomplete data [18], but incomplete instances in current approaches are usually replaced by calculating missing values using imputation before classification. Feature selection has the ability to decrease the number of missing instances by removing redundant information. Thus, helping in increasing the precision and speed of the incomplete instance classification.

We propose IFBag that uses multiple imputation, feature selection and bagging ensemble for classification which improves the performance significantly. The proposed IFBag method is tested using eight incomplete real-world datasets taken from UCI. Each dataset contains incomplete instances varying from low as well as high in number. IFBag method is tested using three different classifiers algorithms to analyze the proposed method performance. The proposed IFBag method decreases computational complexity. This is achieved by using bagging ensemble to use set of base classifiers instead of using missing pattern to make large number of classifiers particularly where data has large quantity of missing values. In this study, we will present how to increase accuracy and accelerate the testing phase for classifying missing values. Multiple imputation method is used to compensate for the uncertainty due to missing data to transform missing training data into full training data [19]. The training data features were further pruned by utilizing feature selection. Afterwards, the proposed method uses bagging ensemble learner on the training data. Where bootstrapping helps in minimizing effects of uncertainty due to sampling fluctuation. While ensemble creates multiple base classifiers, which are trained on sub-datasets made by

bootstrapping complete training data. These base classifiers are able to classify incoming incomplete instances without any need for imputation. We used four benchmark methods to compare and validate the relative performance of the proposed imputed feature selected bagging (IFBag) methods. We used three different learning classifier algorithms on the proposed IFBag method with eight different UCI datasets. The results show considerable performance improvement as compared to other benchmark methods.

The study is aimed at improving classification performance of existing classification methods. Its main objectives are as follows. Firstly, to use an imputation approach which gives better accuracy and has less computation complexity when used for classifying incomplete data. Secondly, the feature selection method with an ensemble learner is to be selected in such a way that they give high accuracy and takes less execution time to classify new incomplete instances. Lastly, the proposed method will be evaluated to show that it gives higher accuracy and is less computationally expensive as compared to other ensemble methods.

The remaining study is organized as follows. The Section 2 presents related works. In Section 3 the proposed method IFBag is described. Section 4 provides an insight of the experimental setup. All the results are discussed in Section 5. Section 6 concludes the study with and provides future direction.

## 2. Related work

In this section, existing traditional approaches are discussed that have been generally used for classifying incomplete data. Then, it discusses classification of incomplete data using ensemble learning techniques. In the end, it describes how feature selection has been used with incomplete data in general.

Numerous approaches have been proposed for classification with incomplete data. Most popular approach for classifying incomplete data is imputation [20]. Imputation can be of two types: single imputation and multiple imputations. Single imputation methods only estimate single value to replace each missing value. Ratolojanahary et al. [21] showed that the  $k$ -nearest neighbors (kNN) imputation is the most effective single imputation method to determine missing values. kNN imputation checks the closest  $k$  neighbor to approximate missing values by taking an average of values contained in  $k$  instances that are used as substitutes for the missing values. kNN is often more accurate and is computationally more expensive than mean imputation which simply replaces missing values with mean value of a column. This is due to the fact; it takes considerably more time to find the nearest instance when data has large number of instances or when the value of  $k$  is large. While imputation methods estimate multiple values for each missing value by taking into account missing value uncertainty [12]. Multivariate imputation by chained equations (MICE) is flexible and powerful than most other multiple imputation as pointed out by Stead and Wheat [22]. MICE imputation method uses regression method to fill missing values with estimated values. MICE operate by substituting all missed values for a random value from the same feature. Every feature with substituted values is then regressed with other features to approximate replacement values. This is done multiple times for all feature with replaced values to get a single imputed dataset. The above steps are repeated  $n$  number of times to get  $n$  imputed datasets whose values are averaged to get a final imputed dataset. MICE perform significantly better on data sets which contain large amounts of missing values [21]. The focus of previous proposed imputation researches for classification is to improve classification accuracy. However, it takes considerable time to predict missing values by applying any imputation methods. Thus, how to use imputation effectively and efficiently must be further studied.

Ensemble learner is a method that builds a collection of classifiers in the training phase for classification task. Each trained base classifiers then predict the class label of instance which are combined to predict

a single class label in the testing phase. As compared to traditional classifiers, multiple classifier sets have been proven to be much more accurate. Ensemble learning has also been used historically for classification of missing data. Tran et al. [23] proposed an approach in which feature selection and imputation are applied on incomplete training data after which based on pattern of missing values it creates an ensemble of base classifiers. This approach benefits by classifying without any imputation of new incomplete instances. Empirical results sometimes show it fails to classify the new instances when classifiers feature pattern does not match with trained classifiers. Khan et al. [24] proposed a multiple imputation method with bagging ensemble to classify instances with missing values. This ensemble is able to model uncertainty and diversity in the data while being able to handle high number of missing values. Results showed that bootstrapping with multiple imputation using expectation maximization gives better performance at high missingness ratio while performing similarly at a low missingness ratio as compared to single imputation. Twala and Cartwright [25] proposed bagging ensemble approach. This ensemble approach helps in creating sub-samples of incomplete data all of which are used to train a decision tree classifier. This results in giving an optimized ensemble in size as it only selects decorrelated decision trees while combined outputs of classifiers give its result. No imputation was used in this approach, but a few multiple imputation methods were added later. However, it is not clear how multiple imputation was used in this approach. Baneshi and Talei [26] used multiple imputation using MICE method on incomplete data. On which bootstrapping is applied and the results from it are aggregated using statistical techniques. Valdiviezo et al. [27] approach used a combination of missing data procedures with tree-based classification methods after single and multiple imputation methods. They came to a conclusion that for small missingness single imputation is enough while for large missingness multiple imputation should be used with tree-bagging. Schomaker and Heumann [28] suggested that the best way for calculating randomization valid confidence intervals when using multiple imputation on bootstrapped samples and bootstrapped samples on multiple imputed datasets. He also suggested that multiple imputation of bootstrap sampling should be used when there is low missingness and bootstrapping of multiple imputation should be used during high missingness. The researcher also explored other types classifier techniques to handle incomplete data. Su et al. [29] presented an approach which used an ensemble classifier to handle missing values. They treated incomplete data by removing a fixed amount of attribute values thus creating multiple versions of original incomplete dataset. All the datasets are then imputed and trained on separate classifiers and their classification results are combined. The results of their study showed that ensemble learners with Bayesian as base classifier while expectation maximization as imputation gives improved performance as compared to single classifiers. A draw back in this approach was that it removes more missing values from already incomplete data thus reducing on accuracy. Nanni et al. [30] proposed an approach which uses random sub spaces with multiple imputation. The main idea behind their approach was to divide incomplete data into fixed number of clusters. In each cluster, they replace missing values of missing data objects with its center value or mean value of the cluster. This approach reduces information loss by mean imputation in case of full data replacement by mean. Random subspace is executed multiple time on imputed data to make an ensemble. Increased performance is shown by this method on several medical datasets while not dropping when missingness increases to 30%. However, it is necessary to further investigate ensemble methods for classifying incomplete data.

Feature selection is an approach that is used to extract relevant subsets of features from dataset features by removing unnecessary and insignificant features [17]. Thus, reducing training time and improving classification accuracy. Feature selection can be divided in to two parts: evaluation technique and a search technique. In order to evaluate feature subsets value, we use evaluation technique. Search technique is used to search for new feature subsets. Both techniques affect the result we get after using feature selection

as pointed by Chandrashekar and Sahin [31]. We can subdivide evaluation measure into wrapper and filter methods. The wrapper approach tests subsets of feature using a classification algorithm. While mutual information is used to score feature subsets by filter approach. As wrapper approaches use classification algorithms to evaluate feature subsets, and it happens to be more accurate. Filter approaches are often more effective when opposed to the wrapper approach, because they are independent of any classification algorithm. Hancer et al. [32] explains that Search technique is divided into deterministic and evolutionary search techniques for feature selection. Example of deterministic search techniques is the sequential forward selection (SFS) and sequential backward selection (SBS). On the other hand, for feature selection, Evolutionary search techniques like Particle Swarm optimization (PSO), Genetic Programming (GP) and Genetic algorithms (GAs) are used. Evolutionary search techniques can find the global best solution without any assumptions related to search space or domain know how. Qian and Shu [33] points out that although feature selection is applied on data with missing values, most of the classification algorithms still cannot use them directly. Thus, it must be further studied on how we can use feature selection with data having missing values in such a way that it improves performance of classification algorithms.

### 3. Proposed IFBag method

In this part, we will discuss the proposed IFBag method. It starts with a section which presents the overall structure and concept behind the proposed method. Next section explains comprehensively on the training and testing phase.

#### 3.1. A detailed overview of proposed IFBag method

The proposed IFBag method can be divided into two phases: a training phase and testing phase. Training phase of the IFBag method builds a collection of classifiers to be trained by training data. The testing phase classifies new instances using the trained base classifiers. This proposed method does not require any prior domain knowledge to use. The flowchart of IFBag method is shown in Fig. 1. The proposed IFBag method works on three fundamental concepts. The first concept is to build ensemble of base classifiers. These base classifiers are trained on sub-dataset extracted from bootstrapping training data. This is done as bootstrapping introduces diversity in the training data thus making diverse and accurate classifiers. It allows incomplete instances to be classified by the ensemble without any need for estimating missing values using imputation.

The other fundamental concept is based on using imputation on incomplete instances during training phase, and not during testing phase. Powerful imputation method tends to create high quality training data, so classifiers built on imputed data tend to be more accurate. The biggest drawback of using powerful imputation is that they are computationally expensive to perform. But, in most situations, the training phase has no time limit. Therefore, use of multiple imputation methods in the training phase is not a problem with high computation costs. On the other hand, in the testing phase there can be limited time to classify instances, thus high computation cost of using multiple imputation makes it impractical to be used. Last thing is to apply feature selection to further enhance training data by removing unnecessary and insignificant features. Using feature selection can also remove missing values, thus reducing time needed to approximate missing values using imputation.

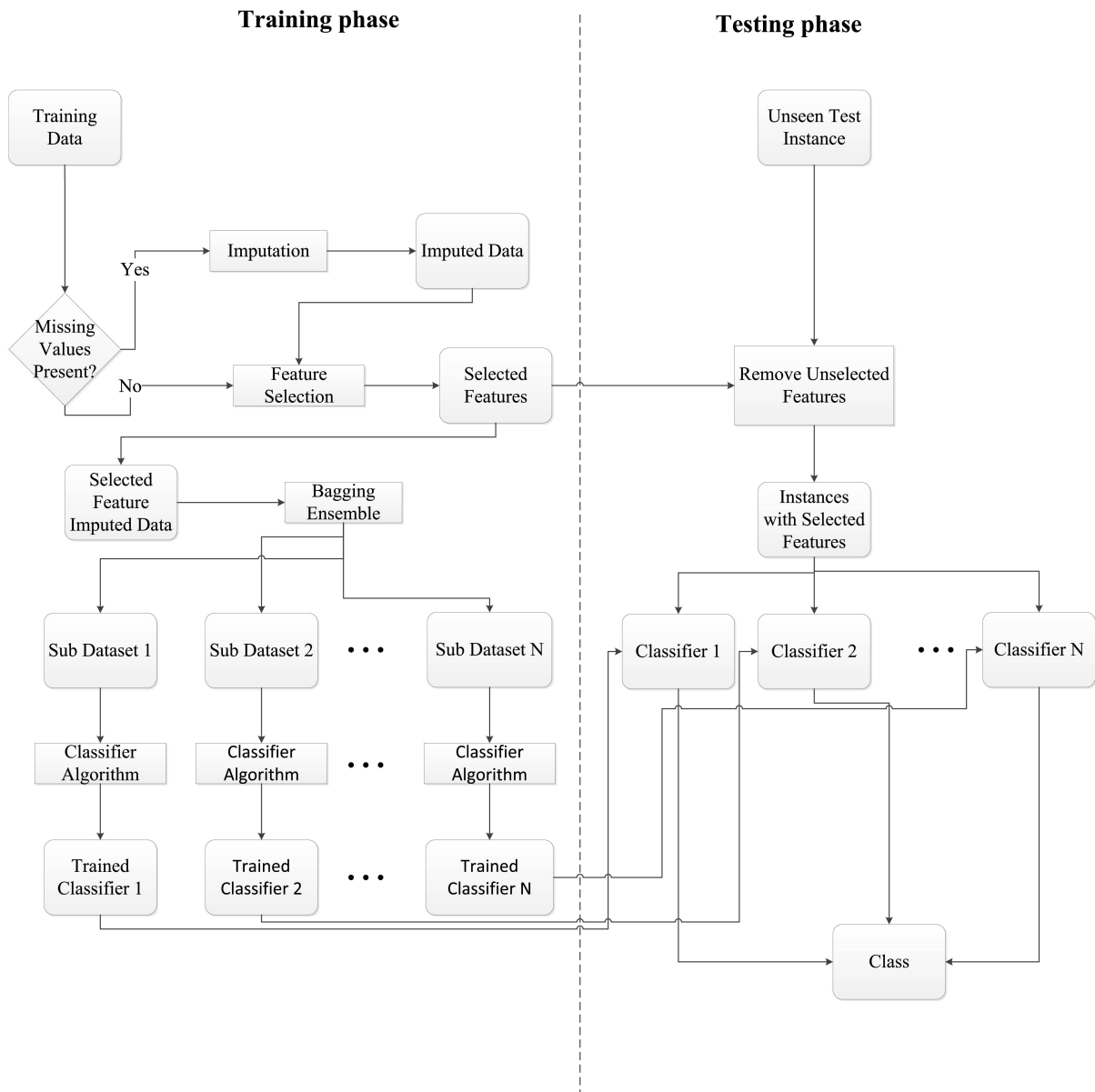


Fig. 1. Complete overview of the proposed IFBag method.

### 3.2. Training phase

The main purpose of the training phase is to create a collection of trained classifiers that are used to classify instances with missing values. The basic steps of training phase are shown in Algorithm 1 which are domain independent in nature. The algorithm takes training data  $D$ , a classifier learning algorithm  $C$  and the number of base classifiers  $n$  to be made in the ensemble as inputs. To generate a complete imputed dataset  $ImpD$ , the algorithm uses imputation on the training data. Then to remove unnecessary and insignificant features from  $ImpD$ , the feature selection is applied to get selected features  $SF$ .  $ImpD$

is then reduced to  $ImpD_{sf}$  by removing all features not present in  $SF$ . Afterward  $n$  number of base classifiers are trained on sub-dataset made by bootstrap samples of  $ImpD_{sf}$ . Thus, resulting in generation of a set of trained classifiers. Training phase can be divided into three parts: imputation, feature selection and ensemble learner.

---

**Algorithm 1** Training phase
 

---

```

1: Input:
2: D, an original training dataset
3: C, classifier algorithm
4: n, number of base classifiers
5: Output:
6: TC, a set of trained classifiers
7: SF, a set of selected features
8:  $ImpD \leftarrow \text{Imputation}(D)$ 
9:  $SF \leftarrow \text{Feature Selection}(ImpD)$ 
10:  $ImpD_{sf} \leftarrow SF(ImpD)$ 
11:  $TC \leftarrow \{ \}$ 
12: For  $i = 1:n$  do
13:    $Bag \leftarrow \text{Bootstrapping}(ImpD_{sf})$ 
14:    $Classifier_i \leftarrow C(Bag)$ 
15:    $TC \leftarrow TC \cup Classifier_i$ 
16: end
17: return TC and SF;

```

---

Training phase can be divided into three parts: imputation, feature selection and bagging ensemble.

### 3.2.1. Imputation

We used MICE to estimate the missing values in training data. However, the MICE can be computationally expensive and thus is used only in training phase. Algorithm 2 describes MICE imputation [34]. The MICE algorithm initially initializes all missing values with mean values of each column to get a complete the dataset  $X_{(full)}$ . Afterwards, it sets one of the estimated missing value attributes to missing while denoting all indexes with no missing values as  $X_{(k)}^m$ . Then  $X_{(k)}^m$  data set is fitted on a regression model. On the other hand, all the indexes with missing values are denoted as  $X_{(k)}^{-m}$ . Later, the fitted regressor model is used to predict all the missing value in  $X_{(k)}^{-m}$ . This is done for each attribute till all missing values are estimated. This process is repeated in cycles and stopped when estimated data coverage. Figure 2 shows the working steps of MICE imputation.

### 3.2.2. Feature selection

Feature selection is done to remove redundant and irrelevant features from the dataset. If the search and evaluation methods are not chosen carefully feature selection can become an expensive process. Evaluation methods such as wrapper methods while being more accurate are often computationally expensive as compared to filter methods. This happens as wrapper methods use classification algorithm meanwhile filter methods use mutual information to score feature subsets. In case of a large training data or when using classifiers like multiple layer perceptron wrapper methods tends to be computationally expensive. Some filter methods like correlation feature selection (CFS) [35] and minimal-redundancy-maximal-relevance criterion (mRMR) [36] have accuracy close to wrapper method while being computationally less expensive. Thus, as the datasets we are using contains numerous features and instances, we will be using CFS filter methods. The heuristic approach [37] used by CFS is given in Eq. (1).

$$Merit_S = \frac{f\bar{r}_{cf}}{\sqrt{f + f(f-1)\bar{r}_{ff}}} \quad (1)$$

**Algorithm 2** MICE

---

```

1: Input:
2: X, a matrix of incomplete data having size total_obs X total_features
3: cycles, number of times repeated
4: Output:
5:  $X_{full}$ , a matrix of completed data
6: For  $i = 1$  to cycles do
7:   For  $k = 1$  to total_features do
8:      $y_i = X_k$  // k th column
9:      $X_{(k)} = X \setminus X_k$ 
10:     $m \subset \{1, n\} = \{i \mid X_k \neq \text{NaN}\}$  //where  $X_k$  indices are not missing is denoted by m
11:     $\text{reg} = \text{linear\_regressor}()$ 
12:     $\text{reg.fit}(X_{(k)}^m, y_k^m)$  //subset of dataset where  $X_k$  is not missing is fitted on the model
13:     $y_{(k)}^{\rightarrow m} = \text{reg.predict}(X_{(k)}^{\rightarrow m})$  // where  $X_k$  indices are missing is denoted by  $\rightarrow m$ 
14:   end for
15: end for
16: Return  $X_{full}$ 

```

---

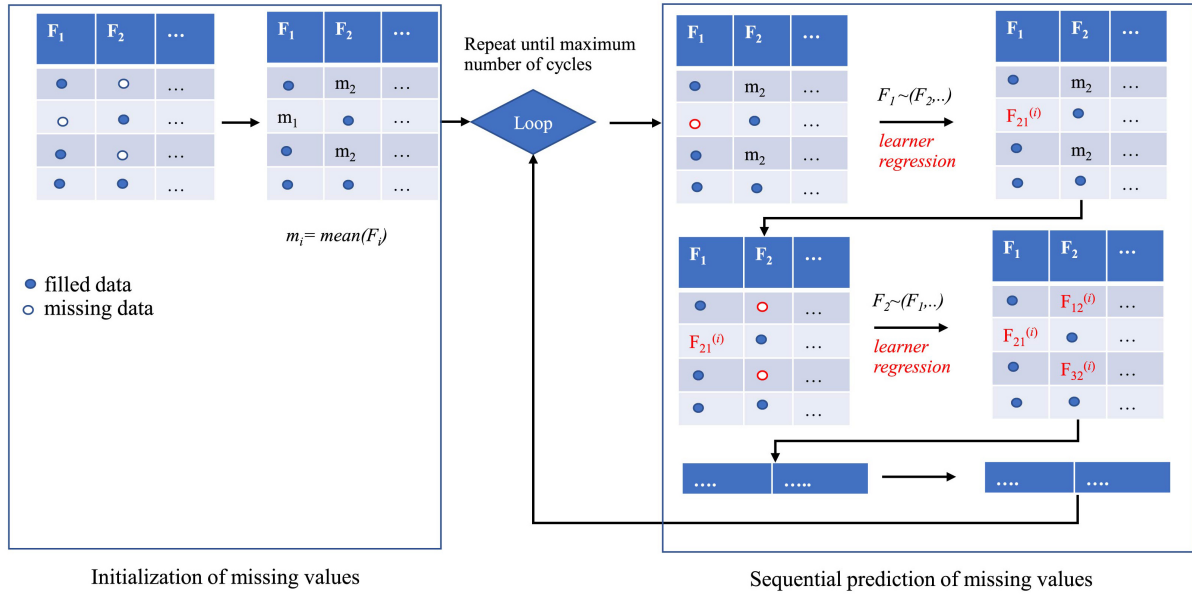


Fig. 2. Shows working of MICE imputation.

In Eq. (1),  $\text{Merit}_S$  is the “score” of the feature subsets,  $\bar{r}_{cf}$  being the average feature class correlation,  $f$  being features and  $\bar{r}_{ff}$  being average feature to feature inter-correlation.

Evolutionary search techniques such as GA and PSO are proven to give better performance for feature selection [23]. Thus, to do feature selection efficiently, we used GA to search for feature subsets while CFS for evaluation of subset. Figure 3a shows in detail the working of feature selection method. Where GA searches for feature subsets, which are evaluated by CFS evaluator on the basis of correlation between features and sends back  $\text{Merit}_S$  score of each feature subset, which is then used to select the best subset of features. The Fig. 3b shows how feature selection works with mammographic dataset, which is one of the eight dataset used for experiments. The features of mammographic dataset, including BI-RADS, Age, Margin and Density are shown in Fig. 3b. The GA search method searches for different feature subsets that are then sent for evaluation to CFS, in turns obtained the  $\text{Merit}_S$  score for each feature subset.

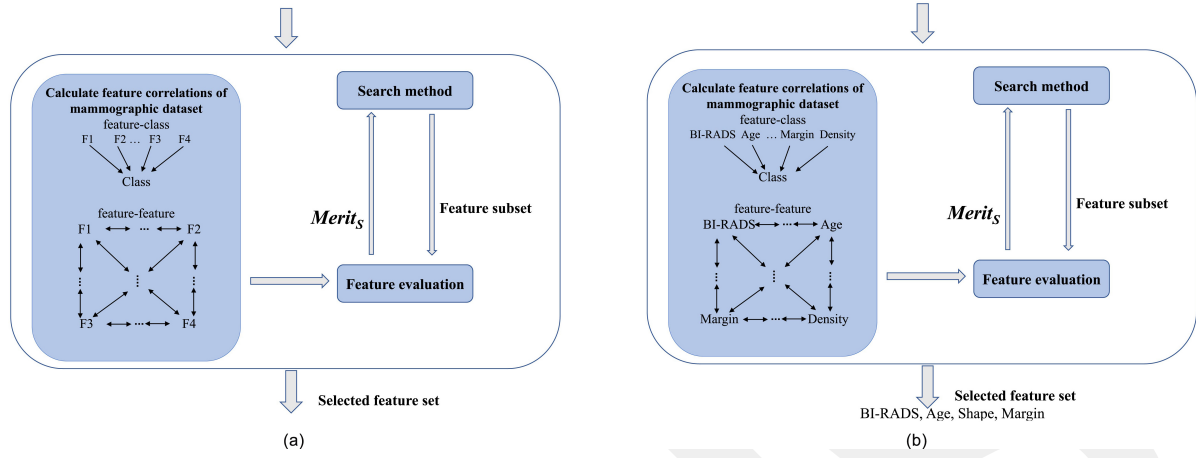


Fig. 3. Feature selection method working mechanism. (a) Detail working of feature selection method irrespective of dataset. (b) Working of feature selection method when using mammographic dataset.

GA keeps searching for ideal feature subset until 100th generation in the evolution process. In the end, BI-RADS, Age, Shape and Margin are the ideal feature subset are shown as output having high  $Merit_S$  as compared to other subsets.

### 3.2.3. Ensemble learner

After imputation and feature selection on the training data, we get a complete data free from irrelevant features and missing values. Then a bagging ensemble technique is applied to the training data, which creates twenty-five base classifiers and each having its own diverse sub dataset created by bootstrap sampling of training data with replacement. Bootstrap sampling helps in minimizing effects of uncertainty due to sampling fluctuation in the training dataset, which occurs when using cross-fold validation. Bagging ensemble combines weak learners sequentially to achieve better performance by introducing diversity in the training data thus making diverse and accurate classifiers.

This ensemble learner can use any classification algorithm where each base classifier is trained on different bootstrap sample which then majority votes over the result. Depending upon the number of base classifiers  $B$ ,  $B$  bootstrap samples ( $S_1, S_2, \dots, S_B$ ) are generated. Each bootstrap sample  $S_j$  is then used by base classifier  $C_j$ . The final output of class is given after aggregated  $B$  classifiers perform majority voting where each vote has same weight. Equation (2) shows how the final output of class label is obtained [38].

$$C^*(X) = \arg \max_{y \in Y} \sum_{j: C_j(X)=y} 1 \quad (2)$$

### 3.3. Testing phase

The testing phase is where new data instances are classified using ensemble of trained classifiers from the training phase. Testing phase steps are shown in Algorithm 3. Selected features  $SF$ , new data instance  $X$ , and a set of trained classifiers  $TC$  are taken as input by the algorithm. The dataset  $X$  is first reduced to  $SF$  features and all missing values are replaced by a static value. As an ample time is available during training, no such luxury is available during testing, which needs to be kept computationally less expensive. Thus, we have avoided imputation like the MICE during the testing phase. Also, from literature, it has

**Algorithm 3** Testing phase

---

```

1: Input:
2: X, dataset to be classified
3: TC, a set of trained classifiers
4: SF, a set of selected features
5: Output:
6: Class, instance class label
7:  $X_{sf} \leftarrow SF(X)$ 
8: replace missing values with static value
9: Apply each TC to reduced  $X_{sf}$ 
10: end
11: Return Class label by majority vote;

```

---

been observed that mean imputation adversely affects the missing value uncertainty in the testing dataset. Instead, we used static value of  $-999$  since it is not present in any of the datasets thus will not affect the outcome of results during classification in anyway. Thus, replacement of missing values with static value does not require the use imputation in testing phase. Afterward, all trained classifiers are applied to  $X_{sf}$  and the algorithm output majority voted class labels for each instance in the dataset. The features that are not in the selected feature set are first removed from the instance. Next, all the learned classifiers apply to the new instance to classify it. Afterwards the ensemble returns the appropriate class for the instance. Methods used to classify data with missing values typically use imputation to get better accuracy in test phase. In most cases the used imputation is quite powerful such as MICE. But as the proposed IFBag method does not use any imputation thus is considerably faster.

#### 4. Experimental setup

In this section, we explain the setup of the experiments, which are performed. We describe the datasets, methods used for comparison and parameter used in the experiments.

##### 4.1. Datasets

Experiments were performed on eight real-world incomplete datasets. These datasets were extracted from UCI machine learning repository [3]. Details regarding the datasets are summarized in Table 1 such as name, number of instances, feature numbers and type of data (Real/integer/Nominal), the percentage of missing instances, and number of classes. In this study we used quantitative and qualitative attributes. The quantitative attributes are easy to handle. However, for qualitative attributes, it requires pre-processing to convert into quantitative values for smooth handling. The selected datasets contain different percentages of missing instances ranging from 1.98% in case of heart-c dataset to 100% in heart-h dataset. Even a single missing feature value is construed as incomplete instance. In the heart-h dataset since all the instances have at least one missing feature value, thus all of them are considered as incomplete instances. The problem of high number of instances such as mammographic dataset with 961 total instances to small number of total instances such as hepatitis dataset having 155 instances. Another problem in the datasets is huge range between high to low dimensionality such as chronic and mammographic dataset has 24 and 5 features respectively. Datasets being used to benchmark this proposed method also have both binary and multiple-class labels.

During experimentation, we used ten-fold cross-validation. In which one-fold is used as testing while nine folds are used as training iteratively ten times as seen in Fig. 4. To eliminate any statistical variation during cross-validation, we performed cross-validation 30 times on given datasets.

Table 1  
List of datasets used

Dataset	Instances	Feature (R/I/N)	Class	Incomplete instances (%)
Chronic	400	24(11/0/13)	2	60.50
Credit	690	15(3/3/9)	2	5.36
Heart-c	303	13(13/0/0)	5	1.98
Heart-h	294	13(6/0/7)	2	100
Hepatitis	155	19(2/17/0)	2	48.39
Housevotes	435	16(0/0/16)	2	46.67
Mammographic	961	5(0/5/0)	2	13.63
Tumor	339	17(0/0/17)	22	61.01

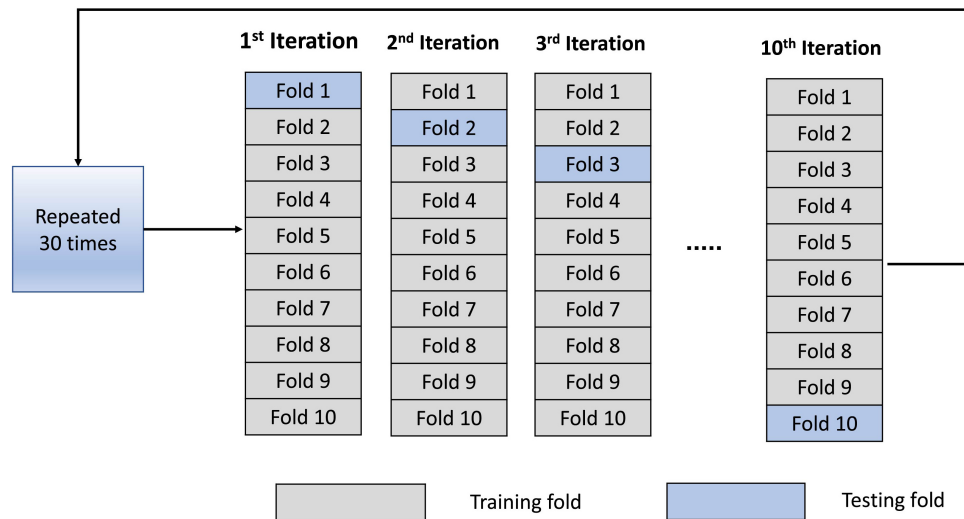


Fig. 4. Ten-fold cross-validation.

#### 4.2. Benchmark methods

To understand how our proposed IFBag method works, we compared it with four benchmark methods to its accuracy and run time. The first two benchmark methods use simple imputation on incomplete data before classification. The remaining two benchmark methods use ensemble approach for classifying incomplete data and thus do not need imputation on new incomplete instance. Following are the details of four benchmark methods used for comparison:

The first benchmark method is named kNNI. Single imputation kNN is used to approximate missing values for both training and test values. Both complete training and testing values provided by the benchmark method allows any classification algorithm to be used.

The second benchmark method is named MICEI. This benchmark method uses MICE. MICE is used to estimate missing values in training and testing instances. TIFBag method and MICEI benchmark method both use MICE to replace missing values. However, MICEI method uses MICE to replace missing data in test instances before classifying them while IFBag method replaces missing values with static values before classifying.

The third benchmark method is named Ensemble [23]. This is an ensemble method which can classify incomplete data. Ensemble [23] benchmark and proposed IFBag methods both apply feature selection

technique to reduce features in training data. These both methods also use powerful imputation to predict missing values to make the incomplete training data into a complete training data. The difference between IFBag method and this benchmark method is that IFBag method makes a fixed number of base classifiers which are trained on sub-dataset made from bootstrap with replacement so that each instance can be picked more than once. Meanwhile the benchmark method builds base classifier for each missing value pattern found in the training data. So, if training data has high amount of missing values, it can be translated into many base classifiers which can considerably slow down the method. On the other hand, the proposed classifier makes same set of base classifiers thus it would not consume much time based on each incomplete data instances its trained on.

The final benchmark method is named Bagging ensemble [24]. Which uses ensemble of classifiers to predict instances with missing values. The IFBag method and Bagging ensemble [24] method builds multiple sets of base classifiers which are trained on sub-datasets made from training data. These sub-datasets are made using bootstrap with replacement so that each instance, can be picked more than once. The difference between these two methods is that in the benchmark method, the imputation is applied on each incomplete sub-dataset to estimate missing values before training a base classifier. On the other hand, IFBag method uses imputation on incomplete training dataset to estimate missing values before it is divided into sub-datasets. Another difference between these methods is that no technique is used to reduce the number of missing values in benchmark method thus more time will be used to estimate values using imputation. While the proposed IFBag method uses feature selection technique which can reduce missing values in the incomplete training data thus reducing the amount of values to be estimated, thus saving more time. To demonstrate the edge of IFBag method on this Bagging ensemble [24] benchmark method a comparison is done.

### 4.3. Experimentation parameters

#### 4.3.1. Imputation

The experiments were performed using MICE imputation. The MICE imputation was implemented using the fancy impute python package during our experiments. MICE imputation was used to get complete training dataset by estimating all missing values.

#### 4.3.2. Feature selection

During experiments, instead of using wrapper method for feature selection we used filter-based method which takes less time. To evaluate feature subsets, we used the CFS measure [35]. CFS was used to evaluate feature subsets as its ability to evaluate both uncorrelation between features and correlation between each feature with a class simultaneously. CFS also has been proved to be accurate as wrapper methods in most cases while taking less execution time [35]. For searching feature subsets, we used GA as its been widely used for feature selection. Following are the parameters we set for GA. Population size 50, maximum number of generations 100 with probability 0.9 and mutation probability 0.1. Feature selection was implemented using python-weka-wrapper python package.

#### 4.3.3. Learning algorithms

For comparison of the proposed IFBag method with the previously proposed benchmark methods, we used three learning algorithms; tree algorithm (J48), kNN and multilayer perceptron (MLP). All the three classification algorithms are implemented in python sklearn package. The number of classifiers in IFBag method and Bagging ensemble [24] was set to 25 as mentioned in [24]. The Ensemble [23] meanwhile automatically identifies the number of classifiers to be made based on training data provided to it.

Table 2  
Different classification methods accuracy (mean and standard deviation) when using different classifiers

Data set	Classifier	IFBag method	kNNI	MICEI	Ensemble [23]	Bagging ensemble [24]
heart-c	J48	57.48 ± 8.41	54.76 ± 8.26↑	55.09 ± 8.10↑	56.86 ± 8.32↑	57.16 ± 8.40
	kNN	55.12 ± 8.17	53.36 ± 7.98↑	53.22 ± 8.60↑	54.58 ± 8.50↑	52.89 ± 8.43↑
	MLP	54.73 ± 8.94	57.59 ± 8.87↓	57.60 ± 8.75↓	58.10 ± 7.92↓	20.06 ± 16.22↑
chronic	J48	97.26 ± 2.5	93.77 ± 4.75 ↑	93.87 ± 4.61↑	61.97 ± 7.67↑	97.23 ± 2.37
	kNN	81.25 ± 12.2	63.30 ± 7.85↑	61.90 ± 7.69↑	55.16 ± 11.31↑	69.36 ± 7.47↑
	MLP	51.94 ± 11.5	75.60 ± 15.7 ↓	75.25 ± 17.4↓	52.99 ± 15.84↓	56.63 ± 8.14↓
credit	J48	85.66 ± 4.03	85.10 ± 3.77	85.19 ± 3.85	85.42 ± 4.33	85.07 ± 3.99
	kNN	72.40 ± 5.46	66.86 ± 5.06↑	67.45 ± 4.79↑	76.06 ± 4.77↓	67.86 ± 5.40↑
	MLP	85.66 ± 4.03	74.15 ± 6.45↑	73.97 ± 6.48↑	57.59 ± 11.47↑	62.76 ± 6.19↑
Heart-h	J48	80.44 ± 6.76	78.44 ± 7.37↑	78.10 ± 8.16↑	47.64 ± 23.87↑	81.29 ± 6.73↓
	kNN	68.67 ± 9.54	65.89 ± 7.93↑	65.34 ± 8.06↑	42.37 ± 24.15↑	68.32 ± 8.40
	MLP	54.23 ± 23.4	82.52 ± 6.82↓	81.74 ± 7.06↓	38.18 ± 26.65↓	57.85 ± 11.1↓
Housevotes	J48	95.08 ± 3.03	93.30 ± 4.16↑	93.50 ± 4.25↑	93.04 ± 4.75↑	95.28 ± 3.07
	kNN	91.49 ± 3.92	90.40 ± 4.39↑	90.37 ± 4.68↑	90.69 ± 4.47↑	90.14 ± 4.23↑
	MLP	91.04 ± 3.61	93.25 ± 3.87↓	93.08 ± 3.98↓	90.94 ± 4.14↑	91.61 ± 3.55
Mammographic	J48	82.35 ± 3.71	79.59 ± 4.11↑	79.46 ± 3.72↑	80.77 ± 4.05↑	82.41 ± 3.75
	kNN	79.51 ± 3.91	79.67 ± 3.72	79.67 ± 4.33	79.90 ± 4.12	78.58 ± 3.94↑
	MLP	81.48 ± 3.73	79.94 ± 4.09↑	80.14 ± 4.14↑	71.02 ± 11.56↑	59.47 ± 10.3↑
Tumor	J48	36.23 ± 7.32	34.54 ± 7.62↑	35.03 ± 8.04↑	33.52 ± 8.88↑	40.32 ± 7.47↓
	kNN	35.99 ± 7.44	43.11 ± 8.19↓	43.72 ± 8.09↓	33.70 ± 8.69↑	32.58 ± 7.32↑
	MLP	25.43 ± 6.69	44.85 ± 8.36↓	44.88 ± 8.55↓	33.53 ± 8.40↓	17.65 ± 6.45↑
hepatitis	J48	80.10 ± 10.6	80.42 ± 10.01	79.37 ± 9.54↑	77.10 ± 10.12↑	81.06 ± 10.24↓
	kNN	77.12 ± 12.0	78.67 ± 10.16↓	78.74 ± 10.01↓	77.63 ± 11.01	79.63 ± 9.81↓
	MLP	77.60 ± 12.0	80.29 ± 8.86↓	80.12 ± 10.4↓	78.84 ± 10.48↓	37.23 ± 13.3↑

## 5. Results and discussion

Here, we will discuss the experimental results of the proposed IFBag method in comparison to benchmark methods.

### 5.1. Accuracy comparison

Table 2 presents both proposed and benchmark methods classification accuracies. Each method's accuracy has a mean and standard deviation. The table's first column lists the dataset's name and the second column names the classifier algorithm used. The proposed IFBag method's accuracy is given in the third column. The last four columns show accuracies of the four benchmark methods. The first benchmark method named "kNNI" uses kNN bases imputation to predict missing values. The second benchmark method named "MICEI" uses MICE imputation to predict missing values. "Ensemble [23]" and "Bagging ensemble [24]" are the two ensemble benchmark methods. The table shows values which are accuracy results of classifier algorithm(row) with specific method used for classification (column) of the given datasets. The values show average classification accuracy ± standard deviation.

In order to correctly evaluate the results, we must choose a suitable statistical test. A test must be multiple tests instead of a pair test as we are comparing proposed IFBag method with multiple methods and it should also be a non-parametric test as it does not require a normal distribution of data as parametric tests. One of the popular multiple non-parametric test called Friedman test [39] is used. This test is helpful in making us understand the importance of the results. The Friedman test shows the existence of an important difference in the results of each method in each classifier and with each dataset. The holm procedure [39] is used which helps in performing pair tests between two methods. To show that IFBag

method is better, the symbol  $\uparrow$  is used while symbol  $\downarrow$  is used to show when its worse than benchmark methods as seen in Table 2.

The Fig. 5 shows the calculated Precision, Recall and F1 score of all methods on eight datasets using J48 classifier. Reason behind using J48 is that IFBag method when used with J48 outperforms other methods as compared to when used with other classifiers. IFBag method in mostly has higher Precision, Recall and F1 score as seen in Fig. 5. Precision gives us percentage of correctly predicted positive classes as compared to total positive predictions. The Eq. (3) shows how Precision is calculated. where  $TP$  is True Positive, and  $FP$  is False Positive.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Recall parameter for benchmark methods is calculated using Eq. (4) which helps us in finding percentage of positive classes as compared to total class predictions. Where  $TP$  is true positive, and  $FN$  is false positive. Recall adds  $TP$  and  $FN$  then divides  $TP$  with sum of  $TP$  and  $FN$ .

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

Lastly F1 score parameter is a weighted average of precision and recall which is calculated using Eq. (5). Where it first multiplies *Precision* and *Recall* then dividing it with sum of *Precision* and *Recall* which is then multiplied by 2 to get F1 score.

$$F1 \text{ score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (5)$$

#### 5.1.1. Comparison with computational methods

Figure 6 compares each individual benchmark methods results with proposed method and shows us the percentage of times it performs better, similar, or worse in performance. The IFBag performs much better than the other methods in most cases. IFBag method achieves similar or better results in 68% of the cases when compared to kNNI. Meanwhile, it performs better or similarly in 69% of cases when compared to MICE benchmark method.

This is possible due to feature selection which helps in removing unnecessary and insignificant features. This feature reduction helps in improving classification accuracy of IFBag method as compared to other imputation methods. Also, as the IFBag method uses multiple classifiers thus it is able to better generalize the results.

#### 5.1.2. Comparison with other ensemble methods

The proposed IFBag will in most cases achieve greater accuracy in comparison to other benchmark ensemble methods as seen in Fig. 6. The IFBag method when compared to bagging ensemble [24] is much more accurate in 46% and similar in 30% of the cases. While in comparison to ensemble [23] IFBag method is more accurate in 58% and similar in 6% of the cases. The proposed IFBag method has better accuracy than bagging ensemble [24] because feature selection used in IFBag removes redundant and irrelevant features. This removal of features helps in improving quality of training data for the IFBag method. The proposed IFBag method is comparatively more accurate then ensemble [23] as it uses bagging ensemble to make a set of base classifiers instead of making base classifiers based on missing value pattern than using majority voting ensemble. The bagging ensemble helps in reducing variance and helps in avoiding overfitting while in base of voting ensemble it simply outputs majority vote as output. This complete analysis with different datasets shows that the proposed IFBag method results vary based on the dataset and classifier used, sometimes giving slightly lower performance compared to other methods.

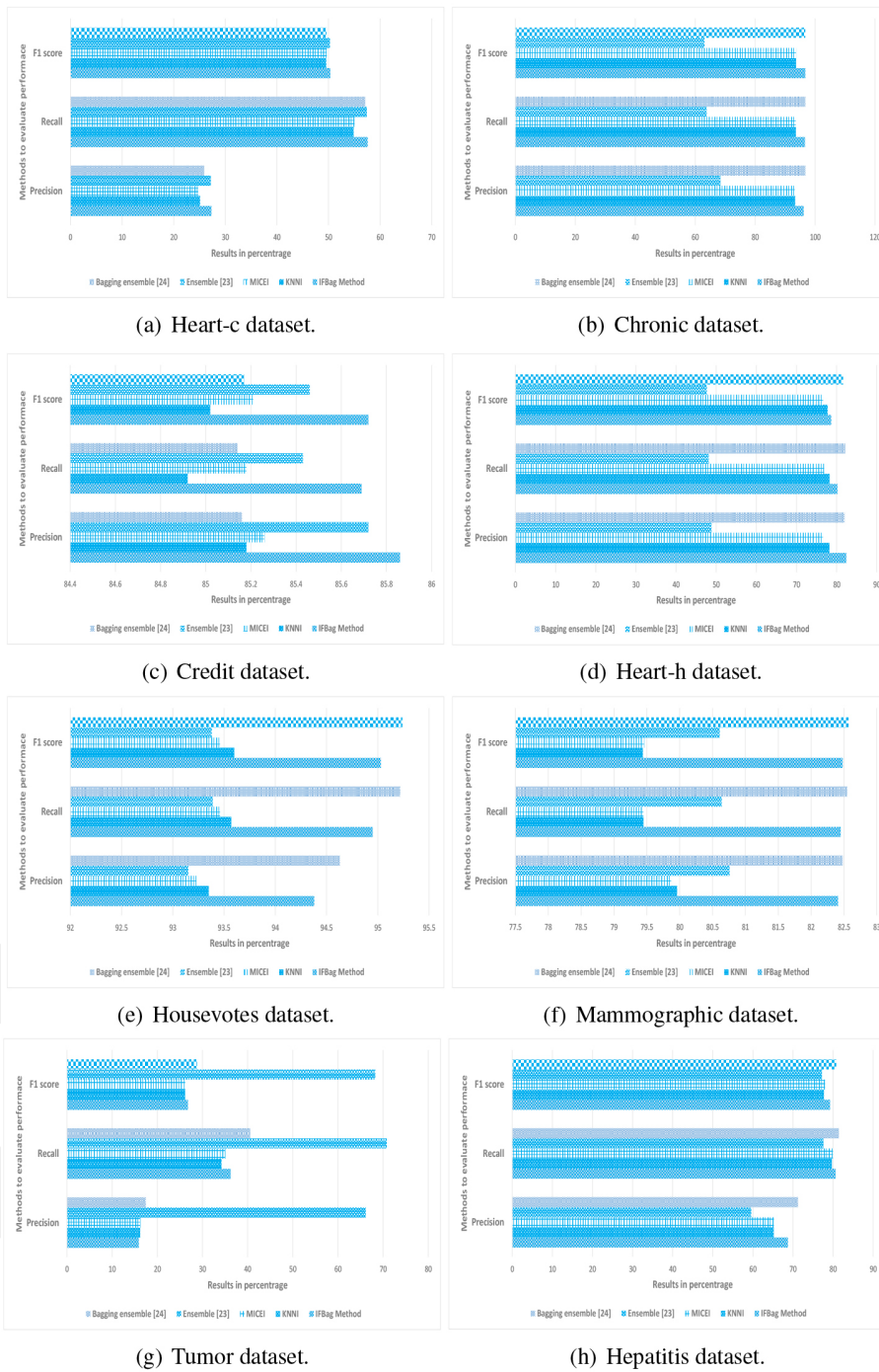


Fig. 5. Shows comparison of Precision, Recall and F1 score of different methods on eight different datasets.

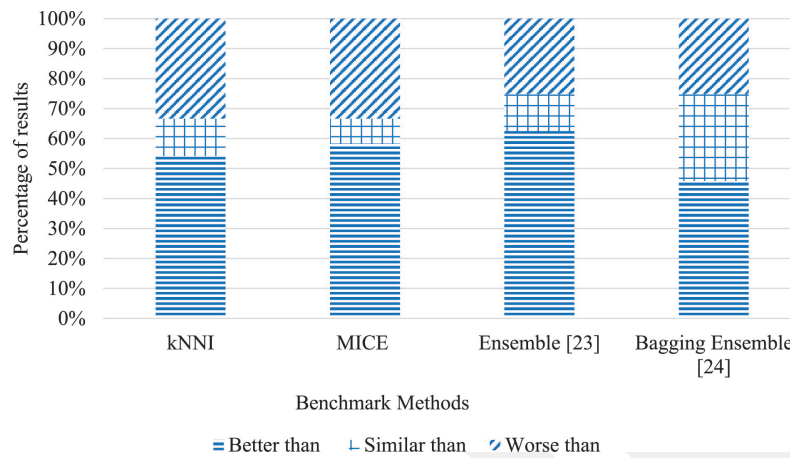


Fig. 6. Comparing results of individual benchmark methods with the proposed IFBag method and its performance in percentage.

### 5.1.3. Comparison with other classification algorithms

The proposed IFBag method gives better accuracy on datasets which have small number of missing values. In addition, IFBag method also gives better accuracy on datasets which have many missing values as can be seen in Table 2. For example, the IFBag method gives better accuracy on mammographic dataset containing only 13.63% missing values. On the other hand, IFBag method gives better accuracy with J48 or kNN on chronic dataset containing only 60.5% incomplete instances. Thus, we can say that IFBag method gives better performance in most cases when using kNN and J48 classifiers.

### 5.1.4. Comparison of different methods with various incomplete datasets

Different classification methods are compared using various incomplete datasets in Table 2. IFBag method is compared using heart-c dataset which has 1.98% incomplete instances. Table shows that IFBag method in comparison to other methods shows considerable improvement when using J48 and kNN classifiers while showing similar performance when using MLP. Comparison of IFBag method using chronic dataset having 60.5% incomplete instances is shown in Table 2. Results show that J48 and kNN classifiers with IFBag method show considerable performance gains. On the other hand, MLP decreases performance of the IFBag method as compared to other benchmark methods. Credit dataset with 5.36% incomplete instances when used with IFBag method shows higher performance as compared other benchmark methods as shown in Table 2. J48 and MLP classification algorithms show considerable improvement in performance. While kNN shows little performance gain in comparison to kNNI and MICEI while slightly worse in comparison to other ensemble methods. In Table 2 comparison is done using heart-h dataset which has 100% incomplete instances. The table shows that IFBag method shows an improvement on kNN and J48 while performs lower worse when compared to bagging ensemble [24] method. Using kNN classifier with IFBag method, it shows better results on housevotes dataset which contains 46.67% incomplete instances. The proposed method on mammographic dataset with 13.63% incomplete instances shows both MLP and J48 achieve good performance gains compared to kNN. Tumor dataset with 61.01% incomplete instances is shown in Table 2. IFBag method performs worse compared to other methods. Hepatitis dataset with 48.39% incomplete instances performs similarly to other methods with different classifiers while never outperforming any of them as seen in Table 2.

Overall, we can say that J48 with IFBag method gives best accuracy with heart-c, chronic, housevotes, mammographic datasets. MLP with IFBag method gives best accuracy with credit dataset alone. kNN

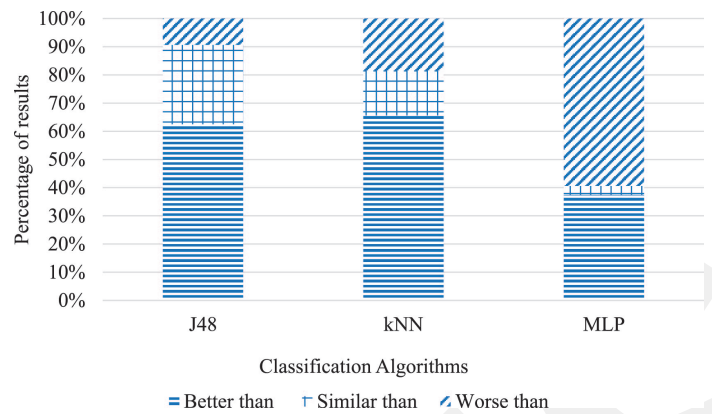


Fig. 7. Comparison of different classification algorithms with the proposed IFBag method and its performance in percentage.

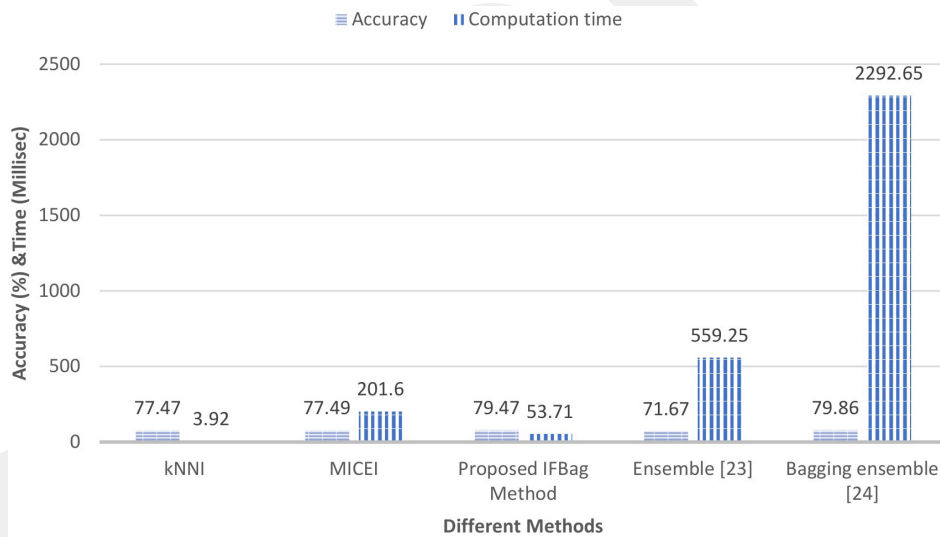


Fig. 8. Comparison of weighted average computation time (milliseconds) and weighted average classification accuracy using J48 classifier.

meanwhile does not give overall better performance than other classification algorithms. However, it gives best accuracy in heart-c, chronic, house votes, mammographic and heart-h datasets when we only compare kNN classifier results with another benchmark methods. Figure 7 shows that J48 algorithm gives 62% better and 28% similar performance with IFBag method. kNN algorithm gives 63% better and 15% similar performance when used with IFBag method according to Fig. 7. Among all the three classification algorithms, MLP classifier has the worst performance when used with IFBag method as compared to other benchmark methods. MLP gives 38% better while 2% similar performance which is lowest when compared to other classifiers used with the IFBag method. Thus, it is safe to say that J-48 seems to benefit the most from proposed IFBag method. The reason for this could be feature selection is able to remove most features which are irrelevant and redundant. But as J48 performs feature selection while constructing classifiers they can be more sustainable to missing values.

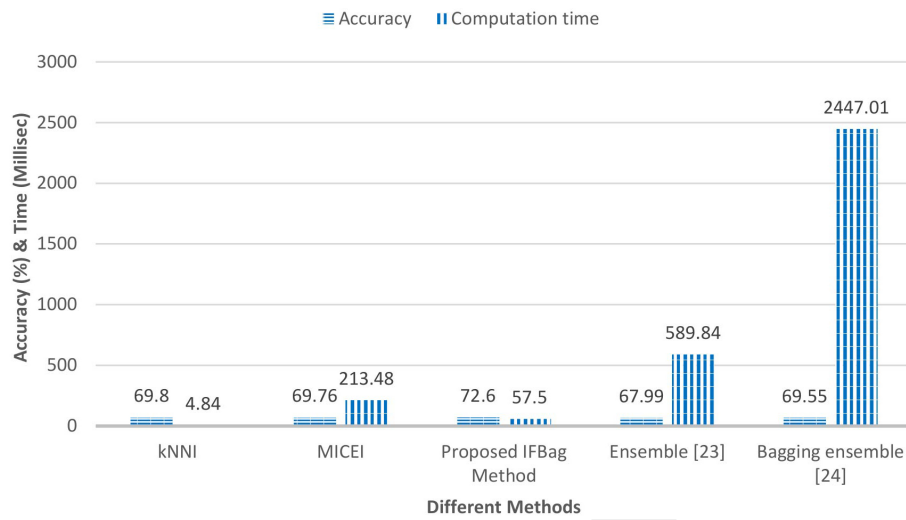


Fig. 9. Comparison of weighted average computation time (milliseconds) and weighted average classification accuracy using kNN classifier.

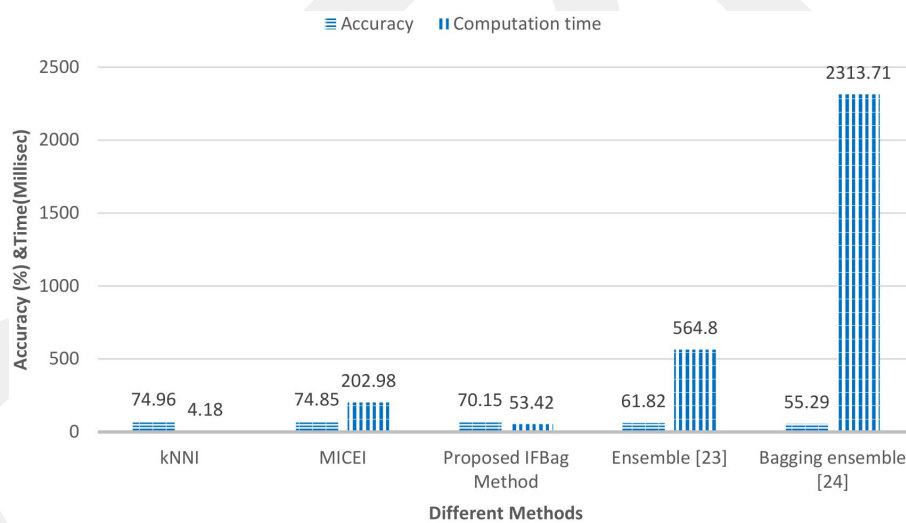


Fig. 10. Comparison of weighted average computation time (milliseconds) and weighted average classification accuracy using MLP classifier.

## 5.2. Computation time

When performing classification, we focus on computation time taken when classifying new instance in the testing phase. Figures 8–10 show a comparison of weighted average computation time for classifying instances in the testing phase and weighted average classification accuracy when using J48, kNN and MLP classifier, respectively. Equation (6) was used to find weighted average computation time and weighted average accuracy of each method, where  $x$  is the accuracy or computation time,  $w$  is the weights of observations while  $n$  being total number of observations.

$$\text{Weighted Average} = \frac{\sum_{i:1}^n w_i x_i}{\sum_{i:1}^n w_i} \quad (6)$$

### 5.2.1. Comparison with imputation methods

The proposed IFBag method takes less computation time than other methods, which use imputation in most cases as compared to other ensemble methods. It is clear from Figs 8–10 that IFBag method is multiple times faster than MICEI irrespective of which classifier is used. MICEI is slower as MICE being multiple imputation is slower in estimating missing values while the IFBag method does not need to estimate any missing values. Figures 8–10 also clearly shows that kNNI is less computationally expensive than IFBag method even though no imputation is used to estimate missing values. The long computation time in the IFBag method can be due to multiple base classifiers which are applied to classify instance.

### 5.2.2. Comparison with other ensemble methods

Figures 8–10 that the proposed IFBag method is computationally less expensive than other benchmark methods. This is possible due to usage of feature selection before the classifiers are built. Feature selection helps in removing redundant and irrelevant features from training data, thus helping in generating simpler classifiers. Feature selection can also help in reducing missing values in various datasets thus reducing the need for imputation. The IFBag method unlike ensemble [11] only builds a specific number of base classifiers as compared to building a classifier for each missing value pattern. So IFBag method can classify instances much fast as compared to other ensemble benchmark methods. It also helps in reducing instances in the testing phase thus taking less time. In short, the IFBag method does not only reduce computation time, but also improves the accuracy.

### 5.3. Critical discussion

Here, we will be discussing if and how our proposed IFBag method fulfils the desired requirements. The first requirement was that an imputation method used must give higher accuracy and take less computation time when used with incomplete data. So, instead of using imputation method such as MICE in both training and test phase we used it in training phase while using static value in testing phase to replace missing values. Using Fig. 8 to compare IFBag method with kNNI and MICEI we see at least 2.51% and 2.49% increase in accuracy respectively using J48 classifier. While 73.35% decrease in time computation for MICEI and almost similar for kNNI. Using kNN classifier we see at least 3.85% and 3.91% increase in accuracy in kNNI and MICEI methods respectively as compared to IFBag method as seen in Fig. 9. The execution time on the other hand decreases 73% for MICEI while giving comparable result for kNNI. MLP classifier results as seen in Fig. 10 shows that IFBag method gives comparable accuracy with kNNI and MICEI, taking 73.68% less execution time for MICEI. Thus, we can conclude from Figs 8–10 that our proposed method meets this requirement.

Our second requirement was use of feature selection method with ensemble learner in increasing accuracy and reducing execution time. The proposed IFBag method used CFS as evaluator and GA as search method for feature subsets while using Bagging ensemble learner for classification. Fig. 8 shows that in comparison with Ensemble [23] our IFBag method is accurate by 9.88% while taking 90.39% less computation time when using J48 classifier. Similarly, when using kNN classifier IFBag method increases accuracy by 6.34% and decreasing execution time by 90.25% as seen in Fig. 9 as compared to Ensemble [23]. Comparing MLP classifier, our IFBag method produced better results with increase of 11.88% accuracy and execution time reduction of 90.1% as seen in Fig. 10. This comparison shows that we met our second requirement as well.

The last requirement of IFBag method was that overall, the model must have higher accuracy and be less computationally expensive than other benchmark ensemble methods. When we look at results of the IFBag method in conjunction with other ensemble approaches like Ensemble [23] and Bagging ensemble [24] it gives average increase in accuracy of 9.33% and 7.87% while taking 90.40% and 97.56% average less execution time when using three different classifiers. Figures 8–10 shows results with different classifying algorithms from which we can conclude that the proposed IFBag method performs much better in both accuracy and execution time. Thus, the IFBag method achieved all the requirements set for this study.

## 6. Conclusion and future work

In data mining, classifying incomplete data is a challenging and difficult task. Thus, this study proposed IFBag method to classify incomplete data using imputation, feature selection and ensemble learning. The proposed IFBag method handles incomplete data by using imputation of training data to get complete training data. Then feature selection is applied to remove unnecessary and insignificant features from complete dataset to get a reduced complete dataset. The proposed IFBag method then uses bootstrapping which incorporates diversity in the training sub-dataset to build an ensemble of classifiers to classify incomplete instances. Combining these ideas together helps in achieving diverse classifier which is more accurate and computationally less expensive. Experimentation was done to compare four benchmark methods with the proposed IFBag method based on classification accuracy and computation time taken. Among the four benchmark methods, two methods used imputation in both training and testing phase while having a single classifier. The other two methods use ensemble learning with imputation only on training phase to classify incomplete data. The results indicate that the proposed IFBag methods achieved higher classification accuracy while taking comparatively less computation time in most cases as compared to other benchmark methods.

In future work, we would like to further investigate why datasets with multiple classes give lower accuracy during classification with both IFBag method and benchmark methods.

## Acknowledgments

First of all, thanks to Allah Almighty who gives me strength and confidence to complete this paper. After that, I would like to express my profound appreciation to my supervisor, Dr. Basit Raza, for providing guidance and feedback throughout this paper. I would also like to thank my parents for their love and support throughout my life, without whom, I would have never been in a position to do this research.

## References

- [1] C.T. Tran, M. Zhang, P. Andrae, B. Xue and L.T. Bui, Improving performance of classification on incomplete data using feature selection and clustering, *Applied Soft Computing* **73** (2018), 848–861.
- [2] X. Lai, X. Wu, L. Zhang, W. Lu and C. Zhong, Imputations of missing values using a tracking-removed autoencoder trained with incomplete data, *Neurocomputing* **366** (2019), 54–65.
- [3] D. Dua and C. Graff, Uci machine learning repository.irvine, ca: University of california, school of information and computer science., 2019. <http://archive.ics.uci.edu/m6> Accessed January 21, 2019.

- [4] M.G. Rahman and M.Z. Islam, Fimus: A framework for imputing missing values using co-appearance, correlation and similarity analysis, *Knowledge-Based Systems* **56** (2014), 311–327.
- [5] J. Sun, H. Fujita, P. Chen and H. Li, Dynamic financial distress prediction with concept drift based on time weighting combined with adaboost support vector machine ensemble, *Knowledge-Based Systems* **120** (2017), 4–14.
- [6] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R.B. Altman, Missing value estimation methods for DNA microarrays, *Bioinformatics* **17**(6) (2001), 520–525.
- [7] R.J. Little and D.B. Rubin, *Statistical analysis with missing data* (Vol. 793), John Wiley & Sons, 2019.
- [8] S. Liu, H. Dai and M. Gan, Information-decomposition-model-based missing value estimation for not missing at random dataset, *International Journal of Machine Learning and Cybernetics* **9**(1) (2018), 85–95.
- [9] H. Huang, H. Wang and M. Sun, Incomplete data classification with view-based decision tree, *Applied Soft Computing* **94** (2020), 106437.
- [10] R.K. Bania and A. Halder, R-Ensembler: A greedy rough set based ensemble attribute selection algorithm with kNN imputation for classification of medical data, *Computer Methods and Programs in Biomedicine* **184** (2020), 105122.
- [11] M.L. Yadav and B. Roychoudhury, Handling missing values: A study of popular imputation packages in R, *Knowledge-Based Systems* **160** (2018), 104–118.
- [12] R.D. Priya, R. Sivaraj and N.S. Priyaa, Heuristically repopulated bayesian ant colony optimization for treating missing values in large databases, *Knowledge-Based Systems* **133** (2017), 107–121.
- [13] I.R. White, P. Royston and A.M. Wood, Multiple imputation using chained equations: issues and guidance for practice, *Statistics in Medicine* **30**(4) (2011), 377–399.
- [14] R. Polikar, J. DePasquale, H.S. Mohammed, G. Brown and L.I. Kuncheva, Learn++. mf: A random subspace approach for the missing feature problem, *Pattern Recognition* **43**(11) (2010), 3817–3832.
- [15] S. Agarwal and C.R. Chowdary, A-stacking and a-bagging: Adaptive versions of ensemble learning algorithms for spoof fingerprint detection, *Expert Systems with Applications* **146** (2020), 113160.
- [16] Y.-T. Yan, Y.-P. Zhang, Y.-W. Zhang and X.-Q. Du, A selective neural network ensemble classification for incomplete data, *International Journal of Machine Learning and Cybernetics* **8**(5) (2017), 1513–1524.
- [17] J. Cai, J. Luo, S. Wang and S. Yang, Feature selection in machine learning: A new perspective, *Neurocomputing* **300** (2018), 70–79.
- [18] G. Doquire and M. Verleysen, Feature selection with missing data using mutual information estimators, *Neurocomputing* **90** (2012), 3–11.
- [19] C.T. Tran, M. Zhang, P. Andreae and B. Xue, Improving performance for classification with incomplete data using wrapper-based feature selection, *Evolutionary Intelligence* **9**(3) (2016), 81–94.
- [20] H. Hegde, N. Shimpi, A. Panny, I. Glurich, P. Christie and A. Acharya, Mice vs ppca: Missing data imputation in healthcare, *Informatics in Medicine Unlocked* **17** (2019), 100275.
- [21] R. Ratolojanahary, R.H. Ngouna, K. Medjaher, J. Junca-Bourié, F. Dauriac and M. Sebilou, Model selection to improve multiple imputation for handling high rate missingness in a water quality dataset, *Expert Systems with Applications* **131** (2019), 299–307.
- [22] A.D. Stead and P. Wheat, The case for the use of multiple imputation missing data methods in stochastic frontier analysis with illustration using English local highway data, *European Journal of Operational Research* **280**(1) (2020), 59–77.
- [23] C.T. Tran, M. Zhang, P. Andreae, B. Xue and L.T. Bui, An effective and efficient approach to classification with incomplete data, *Knowledge-Based Systems* **154** (2018), 1–16.
- [24] S.S. Khan, A. Ahmad and A. Mihailidis, Bootstrapping and multiple imputation ensemble approaches for classification problems, *Journal of Intelligent & Fuzzy Systems* (Preprint) (2019), 1–15.
- [25] B. Twala and M. Cartwright, Ensemble missing data techniques for software effort prediction, *Intelligent Data Analysis* **14**(3) (2010), 299–331.
- [26] M.R. Baneshi and A. Talei, Assessment of internal validity of prognostic models through bootstrapping and multiple imputation of missing data, *Iranian Journal of Public Health* **41**(5) (2012), 110.
- [27] H.C. Valdiviezo and S. Van Aelst, Tree-based prediction on incomplete data using imputation or surrogate decisions, *Information Sciences* **311** (2015), 163–181.
- [28] M. Schomaker and C. Heumann, Bootstrap inference when using multiple imputation, *Statistics in Medicine* **37**(14) (2018), 2252–2266.
- [29] X. Su, T.M. Khoshgoftar and R. Greiner, Making an accurate classifier ensemble by voting on classifications from imputed learning sets, *International Journal of Information and Decision Sciences* **1**(3) (2009), 301–322.
- [30] L. Nanni, A. Lumini and S. Brahnam, A classifier ensemble approach for the missing feature problem, *Artificial Intelligence in Medicine* **55**(1) (2012), 37–50.
- [31] G. Chandrashekar and F. Sahin, A survey on feature selection methods, *Computers & Electrical Engineering* **40**(1) (2014), 16–28.
- [32] E. Hancer, B. Xue and M. Zhang, Differential evolution for filter feature selection based on information theory and feature ranking, *Knowledge-Based Systems* **140** (2018), 103–119.

- [33] W. Qian and W. Shu, Mutual information criterion for feature selection from incomplete data, *Neurocomputing* **168** (2015), 210–220.
- [34] M.J. Azur, E.A. Stuart, C. Frangakis and P.J. Leaf, Multiple imputation by chained equations: what is it and how does it work? *International Journal of Methods in Psychiatric Research* **20**(1) (2011), 40–49.
- [35] N.D. Cilia, C. De Stefano, F. Fontanella and A.S. di Freca, A ranking-based feature selection approach for handwritten character recognition, *Pattern Recognition Letters* **121** (2019), 77–86.
- [36] H. Zhou, Y. Zhang, Y. Zhang and H. Liu, Feature selection based on conditional mutual information: minimum conditional relevance and minimum conditional redundancy, *Applied Intelligence* **49**(3) (2019), 883–896.
- [37] M.A. Hall, Correlation-based feature selection of discrete and numeric class machine learning, 2000.
- [38] Z. Wu, N. Li, J. Peng, H. Cui, P. Liu, H. Li and X. Li, Using an ensemble machine learning methodology-bagging to predict occupants' thermal comfort in buildings, *Energy and Buildings* **173** (2018), 117–127.
- [39] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research* **7** (2006), 1–30.