# A Novel Framework for Imputation of Missing Values in Databases

Alireza Farhangfar, Lukasz A. Kurgan, *Member, IEEE*, and Witold Pedrycz, *Fellow, IEEE*

***Abstract*—Many of the industrial and research databases are plagued by the problem of missing values. Some evident examples include databases associated with instrument maintenance, medical applications, and surveys. One of the common ways to cope with missing values is to complete their imputation (filling in). Given the rapid growth of sizes of databases, it becomes imperative to come up with a new imputation methodology along with efficient algorithms. The main objective of this paper is to develop a unified framework supporting a host of imputation methods. In the development of this framework, we require that its usage should (on average) lead to the significant improvement of accuracy of imputation while maintaining the same asymptotic computational complexity of the individual methods. Our intent is to provide a comprehensive review of the representative imputation techniques. It is noticeable that the use of the framework in the case of a low-quality single-imputation method has resulted in the imputation accuracy that is comparable to the one achieved when dealing with some other advanced imputation techniques. We also demonstrate, both theoretically and experimentally, that the application of the proposed framework leads to a linear computational complexity and, therefore, does not affect the asymptotic complexity of the associated imputation method.**

***Index Terms*—Accuracy, databases, missing values, multiple imputation (MI), single imputation.**

## I. Introduction

MANY of industrial and research databases are plagued by an unavoidable problem of data incompleteness (missing values). Behind this serious deficiency, there are a number of evident reasons, including imperfect procedures of manual data entry, incorrect measurements, and equipment errors. In many areas of application, it is not uncommon to encounter databases that have up to or even more than 50% of their entries being missing. For example, an industrial instrumentation maintenance and test database maintained by Honeywell [31] has more than 50% of missing data, despite the strict regulatory requirements for data collection. Another application domain overwhelmed by missing values arises in medicine; here, almost every patient record lacks some values, and almost every attribute used to describe patient's records is lacking values for some patient's record [17]. For example, a medical database of patients with cystic fibrosis with more than 60% of its entries missing was analyzed in [30]. One of the reasons why medical databases are so heavily exposed is that most medical data are collected as a by-product of patient care activities, rather than for an organized research protocol [17]. At the same time, the majority of prior studies related to missing data concern relatively low, usually below 20%, amounts of missing data [1], [4], [41]. In contrast, in this paper, we are concerned with databases with up to 50% of missing data.

Missing values make it difficult for analysts to realize data analysis. Three types of problems are usually associated with missing values: 1) loss of efficiency; 2) complications in handling and analyzing the data; and 3) bias resulting from differences between missing and complete data [3]. Although some methods of data analysis can cope with missing values on their own, many others require complete databases. Standard statistical software works only with complete data or uses very generic methods for filling in missing values [31]. Other data processing packages that are used for visualization and modeling often use and display only the complete records or map missing values to an arbitrary fixed value, e.g., $-1$ or 999 999, thus leading to distortion of the presented results. Hence, in all such cases, imputation plays an important role. It could also be invaluable in cases when the data needs to be shared, and the individual users may not have resources to deal with their incompleteness [33], [44].

There are two general approaches to deal with the problem of missing values: They could be ignored (removed) or imputed (filled in) with new values. The first solution is applicable only when a small amount of data is missing. Since in many cases databases contain relatively large amount of missing data, it is more constructive and practically viable to consider imputation. A number of different imputation methods have been reported in the literature. Traditional imputation methods use statistics and rely on some simple algorithms such as mean and hot-deck imputation, as well as complex methods including regression-based imputation and expectation–maximization (EM) algorithm. In recent years, a new family of imputation methods, which uses machine learning (ML) algorithms, was proposed. Another major development comes in the form of the multiple imputations (MIs) first described by Rubin in the 1970s [43]. In this case, each missing value is imputed $m$ times (usually, $m$ is between 3 and 5) by the same imputation algorithm, which uses a model that incorporates some randomness. As a result, $m$ "complete" databases are generated, and usually, the average of the estimates across the samples is used to generate the final imputed value. The development of such methods was mainly driven by a need to improve accuracy of the imputation. Early methods were very simple and computationally inexpensive. Newer methods use more complex procedures, which could improve the quality of imputation, but come at a higher computational effort. At the same time, we have witnessed a

Manuscript received October 1, 2004; revised May 25, 2005. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada. The work of W. Pedrycz was supported by the Canada Research Chairs Program. This paper was recommended by Associate Editor D. Zhang.

The authors are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: farhang@ece.ualberta.ca; lkurgan@ece.ualberta.ca; pedrycz@ece.ualberta.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2007.902631

| Attribute 1 | Attribute 2 | Attribute 3 | Attribute 4 | Attribute 5 |
|---|---|---|---|---|
| data | ? | data | data | data |
| ? | data | data | ? | data |
| data | data | data | ? | data |
| data | data | data | data | data |
| … | … | ... | … | ... |

Fig. 1. Database containing missing values.

rapid growth of size of databases. Recently published results of a 2003 survey on the largest and most heavily used commercial databases show that the average size of Unix databases experienced a 6-fold increase when compared to year 2001. For Windows databases, this growth was 14-fold. Large commercial databases now average ten billion data points [57].

The main objective of this paper is to build a new framework aimed at the improvement of the quality of existing imputation methods (we will be referring to them as *base* imputation methods). The framework should meet three requirements.

1) It should improve accuracy of imputation when compared to the accuracy resulting from the use of a single base imputation method.
2) An application of the framework to a base imputation method should not worsen its asymptotic computational complexity.
3) It should be applicable to a wide range of generic (base) imputation methods, including both statistical and ML-based imputation techniques.

To meet these requirements, in the proposed framework, we impute some of the missing values several times. Furthermore, the overall environment is characterized by several important features that clearly distinguish it from some other MI methods.

- It imputes only a subset of the missing values multiple times. The imputation is executed in an iterative manner. At each iteration, some high-quality imputed values are accepted, and the remaining lower quality missing values are imputed again (multi-imputed). Assuming that at each iteration half of the values are imputed (the framework uses mean-based parameter to select imputed values, which for data with normal distribution approximates to a half of values) and that ten iterations are executed, then the number of imputations becomes equal to

$$\sum_{i=0}^{9} \frac{1}{2^i} k = k + \frac{1}{2}k + \frac{1}{4}k + \cdots + \frac{1}{512}k < 2k$$

  where $k$ is the number of missing values.

In contrast, in the case of MIs, every missing value is imputed several times, and for the typical values of $m$, the number of imputations is not less than $3k$, but it can be as large as $10k$ [52]. Therefore, the framework is more efficient.

- It uses the high-quality accepted imputed values from the previous iterations to impute the remaining missing values. In contrast, MIs use the original database containing all the missing values and, thus, do not take advantage of already imputed values. Therefore, the imputation procedure of the proposed framework is possibly more accurate since, in each iteration, more data are used to infer the imputation model for imputing the remaining missing values. This hypothesis was confirmed experimentally in this paper.

Extensive experimental results presented in this paper show that the proposed imputation framework results in substantial improvement of imputation accuracy. We show that using the proposed framework with very simple imputation methods, such as hot deck, produces accuracy of imputation that surpasses quality of results generated by advanced statistical and MI methods while preserving low computational overhead. This advantage is clearly demonstrated with the use of the proposed framework to imputation technique of linear complexity (i.e., an ML-based imputation using Naïve Bayes). The resulting imputation method was also linear, and its accuracy is higher than that of any of several other imputation methods, including complex statistical and MIs techniques.

This paper is organized in the following manner. We first review a number of representative imputation methods (Section II). Section III elaborates on the structure of the proposed framework. In Section IV, we report on experimental results and offer an extensive comparative analysis. Conclusions and recommendations are covered in Section V. Throughout the text, the term database pertains to a relational data set.

## II. Background and Related Work

### A. Background

In what follows, we are concerned with databases consisting of one or multiple tables, where columns describe attributes (features), and rows denote records (examples or data points). Fig. 1 shows a typical database involving five attributes; note that some of them have missing values denoted by "?". In general, the attributes can be numerical discrete, numerical continuous, and nominal. In this paper, we are dealing with imputation procedures for discrete attributes, i.e., discrete numerical and nominal. We note that the two main application areas of missing data imputation procedures are concerned with equipment maintenance databases [31] and survey data [23], [29], [43], [45], both of which use discrete data.

Some of the missing data imputation algorithms are supervised; that is, they require some class attribute. They impute missing values one attribute at a time by setting it to be the class attribute and using data from the remaining attributes to generate a classification model, which is used to perform imputation.

The three different modes that lead to introduction of missing values are: 1) missing completely at random (MCAR); 2) missing at random (MAR); and 3) not missing at random (NMAR) [31], [33]. The MCAR mode applies when the distribution of a record having a missing value for an attribute does not depend on either the complete data or the missing data. This mode usually does not hold for nonartificial databases. Its relaxed version, i.e., the MAR mode, where the distribution depends on data but does not depend on the missing data itself,

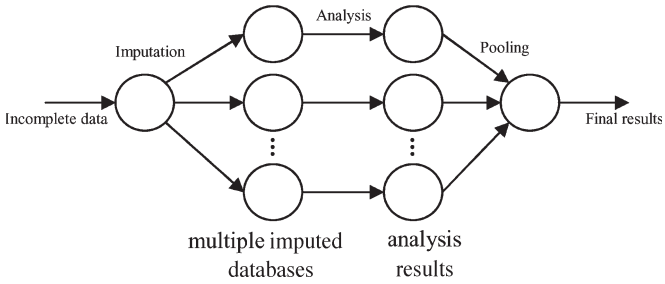


Fig. 2. Flow of operations in MI.

is assumed by most of the existing methods for imputation of missing data [51], and therefore, it is also assumed in this paper. In case of the MCAR mode, the assumption is that the distribution of missing and complete data are the same, whereas for the MAR mode, they are different, and the missing data can be predicted by using the complete data [33]. The third mode, i.e., the NMAR, where the distribution depends on the missing values, is rarely used in practice.

### B. Related Work

The existing methods for dealing with missing values can be divided into two main categories: 1) missing data removal and 2) missing data imputation. The removal of missing values is concerned with discarding the records with missing values or removing attributes that have missing entries. The latter can be applied only when the removed attributes are not needed to perform data analysis. Both removals of records and attributes result in decreasing the information content of the data. They are practical only when a database contains a small amount of missing data and when the ensuing analysis of the remaining complete records will not be biased by the removal [31]. They are usually performed in the case when dealing with missing data introduced in the MCAR mode. Another method belonging to the same category proposes substituting missing values for each attribute with an additional category. Although this method provides a simple and easy-to-implement solution, its usage results in substantial problems occurring during the subsequent analysis of the resulting data [55].

The imputation of missing values uses a number of different algorithms, which can be further subdivided into single-imputation and MI methods. In the case of single-imputation methods, a missing value is imputed by a single value, whereas in the case of MI methods, several, usually likelihood ordered, choices for imputing the missing value are computed [43]. Rubin defines MIs as a process where several complete databases are created by imputing different values to reflect uncertainty about the right values to impute. At the next step, each of the databases is analyzed by standard procedures specific for handling complete data. At the end, the analyses for each database are combined into a final result [11], [44]. Fig. 2 illustrates the flow of operations in MI procedure.

Several approaches have been developed to perform MIs. Li [32] and Rubin and Schafer [42] use Bayesian algorithms that support imputation by using posterior predictive distribution of the missing data based on the complete data. The Rubin–Schafer method assumes the MAR mode, as well as multivariate normal distribution for the data. Alzola and Harrell introduce a method that imputes each incomplete attribute by cubic spline regression given all other attributes, without assuming that the data can be modeled by a multivariate distribution [2]. The MI methods are computationally more expensive than the single-imputation techniques, but at the same time, they better accommodate for sample variability of the imputed value and uncertainty associated with a particular model used for imputation [31]. Detailed description of MI algorithms can be found in [45], [51], [52], and [59].

Both the single-imputation and MI methods can be divided into three categories: 1) data driven; 2) model based; and 3) ML based [31], [33], [38]. Data-driven methods use only the complete data to compute imputed values. Model-based methods use some data models to compute imputed values. They assume that the data are generated by a model governed by unknown parameters. Finally, ML-based methods use the entire available data and consider some ML algorithm to perform imputation.

The data-driven methods include simple imputation procedures such as mean, conditional mean, hot-deck, cold-deck, and substitution imputation [31], [49]. The mean and hot-deck methods are described in detail later in this paper, whereas the remaining methods are only applicable to special cases. The cold-deck imputation requires additional database, other than the database with missing values, to perform imputation, which is usually not available to data analyst. The substitution method is applicable specifically to survey data, which significantly narrows down its possible application domains.

Several model-based imputation algorithms are described in [33]. The leading methods include regression-based, likelihood-based, and linear discriminant analysis (LDA)-based imputation. In regression-based methods, missing values for a given record are imputed by a regression model based on complete values of attributes for that record. The method requires multiple regression equations, each for a different set of complete attributes, which can lead to high computational cost. Also, different regression models must be used for different types of data; that is, linear or polynomial models can be used for continuous attributes, whereas log–linear models are suitable for discrete attributes [31]. The likelihood-based methods can be considered to impute values only for discrete attributes. They assume that the data are described by a parameterized model, where parameters are estimated by maximum-likelihood or maximum *a posteriori* procedures, which use different variants of the EM algorithm [18], [33].

Recently, several ML algorithms were applied to the design and implementation of imputation methods. A probabilistic imputation method that uses probability density estimates and Bayesian approach was applied as a preprocessing step for an independent module analysis system [13]. Neural networks were used to implement missing data imputation methods [26], [55]. An association rule algorithm, which belongs to the category of algorithms encountered in data mining, was used to perform MIs of discrete data [58]. Recently, algorithms of supervised ML were used to implement imputation. In this case, imputation is performed one attribute at a time, where the selected attribute is used as a class attribute. An ML algorithm is used to generate a data model from data associated with complete portion of the class attribute, and the generated model is used to perform classification to predict missing values of the class attribute. Several different families of supervised ML algorithms, such as decision trees, probabilistic, and decision

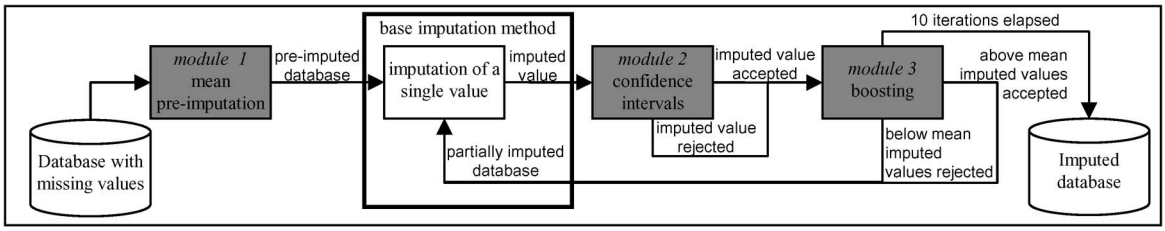


Fig. 3. Structure of the proposed framework.

rules [18], can be used; however, the underlying methodology remains the same. For example, a decision tree C4.5 [39], [40] and a probabilistic algorithm Autoclass [14] were used in [31], whereas a decision rule algorithm CLIP4 [15], [16] and a probabilistic algorithm Naïve Bayes were studied in [22]. A decision tree along with an information retrieval framework was used to develop incremental conditional mean imputation in [19]. In [4], a $k$-nearest neighbor algorithm was used. Statistical and ML-based imputation methods are briefly compared in [23]. Also recently, ML-based imputation methods were experimentally compared with data-driven imputation, showing their superiority in terms of imputation accuracy [22].

The development of the new missing data imputation methods was mainly driven by the need to improve accuracy of imputation. The simplest data-driven imputation methods were followed by model-based methods and MI procedures. As a result of this evolution, complex and computationally expensive algorithms, such as MI logistic regression, were developed. At the same time, because of recent rapid growth of database sizes, researchers and practitioners require imputation methods to be not only accurate but also scalable. MIs and ML-based imputation methods are characterized by a relatively high accuracy, but at the same time, they are often complex and computationally too expensive to be used for real-time imputation or for imputing in large databases [49]. We show, both theoretically and experimentally, that the proposed framework has linear asymptotic complexity with respect to the number of records. Therefore, as long as the base imputation method has linear or worse complexity (to the best of our knowledge, there are no sublinear imputation methods), the application of the framework does not worsen the base method's complexity. The proposed framework consists of three modules, which are concerned with performing mean pre-imputation, using confidence intervals, and applying boosting, respectively. Extensive experimental tests show that the application of the proposed framework improves accuracy of the base imputation method and, at the same time, preserves its asymptotic complexity. Applying the framework to a very simple imputation method, such as hot deck, on average, improves its accuracy to match accuracy of complex model-based imputation methods, such as multiple polytomous logistic regression imputation, while at the same time being significantly faster and easier to implement.

This paper concerns the imputation of discrete attributes. This limitation is imposed by the considered base imputation methods; that is, in the case of ML-based imputation, only discrete attributes can be imputed. We note that the proposed framework is applicable to imputation methods that handle continuous attributes, and its extension to these methods will be the subject of future work.

## III. Proposed Framework

The overall architecture of the proposed framework is visualized in Fig. 3. It consists of three main functional modules: 1) mean pre-imputation; 2) application of confidence intervals; and 3) boosting. All of those are visualized as shadowed boxes.

Let us briefly discuss the functionality of each of these modules. The missing values are first pre-imputed (module 1), i.e., temporarily filled with a value that is used to perform imputation, using a fast linear mean imputation method. Next, each missing pre-imputed value is imputed using a base imputation method, and the imputed value is filtered by using confidence intervals (module 2). Confidence intervals are used to select the most probable imputed values while rejecting possible outlier imputations. Once all the values are imputed and filtered, each of them is assigned with a value that quantifies its quality; that is, it might be expressed as a probability or a distance. Based on these values, the boosting module (module 3) accepts the best high-quality imputed values, whereas the remaining imputed values are rejected, and the process repeats with the new partially imputed database. After ten iterations, all the remaining imputed values are accepted, and the imputed database is outputted. We note that any imputation method, i.e., data driven, model based, or ML based, can be used as the base method.

### A. Imputation Methods

This section provides a short description of several imputation methods being used in the proposed framework or in the experimental section of this paper. A description of how the selected methods are incorporated in the proposed framework is also provided. The selection of the imputation methods was driven by the following principles. The base methods that will be tested with the proposed framework should be simple enough to show that they can be improved by the application of the framework to match or surpass the quality of complex high-quality model-based imputation methods. They should also represent both data-driven and ML-based categories. Therefore, hot-deck imputation and ML-based imputation that use Naïve Bayes algorithms were selected.

To provide comprehensive evaluation, the framework with the selected two base methods should be compared with advanced high-quality model-based imputation methods, as well as fast data-driven methods. Therefore, two MI methods, i.e., LDA-based method and multivariate imputation that combines logistic, polytomous, and linear regressions, and three data-driven methods, i.e., mean, hot deck, and MI by sampling, are used in the experimental section.

TABLE I
SUMMARY OF THE IMPUTATION METHODS USED IN THIS PAPER

| Method name | Imputation algorithm | Multiple/single imputation | Discrete data | Continuous data | Abbreviation |
|---|---|---|---|---|---|
| Naïve-Bayes | Naïve Bayes algorithm | single | Yes | No | NB |
| Hot deck | nearest neighbor | single | Yes | Yes | HD |
| Mean | attribute average (mode) | single | Yes | Yes | Mean |
| Polyreg | polytomous regression | multiple | Yes | No | POLYLOGREG |
| LDA | linear discriminant analysis | multiple | Yes | No | LDALOGREG |
| Logreg | logistic regression | multiple | Yes (Binary) | No | --- |
| Sampling | random sampling | multiple | Yes | Yes | SAM |
| Framework with Naïve-Bayes | Proposed framework with Naïve-Bayes algorithm | single | Yes | No | FNB |
| Framework with Hot deck | Proposed framework with Hot deck algorithm | single | Yes | Yes | FHD |

*1) Single-Imputation Methods:* In the *mean imputation*, the mean of the values of an attribute that contains missing data is used to fill in the missing values. In the case of a categorical attribute, the mode, which is the most frequent value, is used instead of the mean. The algorithm imputes missing values for each attribute separately. Mean imputation can be conditional or unconditional, i.e., not conditioned on the values of other variables in the record. Conditional mean method imputes a mean value, that depends on the values of the complete attributes for the incomplete record [8]. In this paper, the unconditional mean, which is computationally faster and therefore can be efficiently used with large data sets, is used both to impute the missing values as a stand-alone method and to perform pre-imputation of the missing values in the proposed framework.

In the *hot deck*, for each record that contains missing values, the most similar record is found, and the missing values are imputed from that record. If the most similar record also contains missing information for the same attributes as in the original record, then it is discarded, and another closest record is found. The procedure is repeated until all the missing values are successfully imputed or the entire database is searched. When no similar record with the required values filled in is found, the closest record with the minimum number of missing values is chosen to impute the missing values. Several distance functions can be used [23], [45], [48]. In this paper, a computationally fast distance function is used, which assumes a distance of 0 between two attributes if both have the same numerical or nominal value or, otherwise, assumes a distance of 1. A distance of 1 is also assumed for an attribute for which any of the two records has a missing value. In the case of supervised databases, which are used in this paper, the hot-deck method takes advantage of the class information to lower computational time. Since, usually, certain correlations exist between records in the same class, the distance is computed only between records within the same class.

In *regression*, imputation is performed by regression of the missing values using complete values for a given record [26]. Several regression models can be used, including linear, logistic, polytomous, etc. Logistic regression applies maximum-likelihood estimation after transforming the missing attribute into a logit variable, which shows changes in natural log odds of the missing attribute. Usually, logistic regression model is applied for binary attributes, polytomous regression for discrete attributes, and linear regression for numerical attributes.

*Naïve Bayes* is an ML technique based on computing probabilities [21]. The algorithm works with discrete data and requires only one pass through the database to generate a classification model, which makes it very efficient, i.e., linear with the number of records. Imputation based on the Naïve Bayes consists of two simple steps. Each attribute is treated as the class attribute, and the data are divided into two parts: 1) training database that includes all records for which class attribute is complete and 2) testing database for which the records are missing. First, prior probability of each non-class attribute value and frequency of each nonclass attribute value in combination with each class attribute value are computed on the basis of the training database. The computed probabilities are then used to perform prediction of class attribute for testing database, which constitute the imputed values.

*2) MI Methods:* One of the most flexible and powerful MI regression-based methods is the multivariate imputation by chained equations (MICE) [9], [10]. The method provides a full spectrum of conditional distributions and related regression models. MICE incorporates logistic regression, polytomous regression, linear regression and uses Gibbs sampler to generate MI [12]. MICE is furnished with a comprehensive state-of-the-art missing data imputation software package [28]. We will use it in the experimental section of this paper. It provides Bayesian linear regression for continuous attributes, logistic regression for binary attributes, and polytomous logistic regression for categorical data with more than two categories. MICE also delivers a comprehensive library of nonregression imputation methods, such as predictive mean, unconditional mean, multiple random sample imputation that is suitable for the attributes in the MCAR model, and LDA for categorical data with more than two categories. LDA is a commonly used technique for data classification and dimensionality reduction [34]. At the same time, it serves as a statistical approach to classification-based missing data imputation. The LDA method is particularly suitable for data where within-class frequencies are unequal, as it maximizes the ratio of between-class variance to the within-class variance to assure best separations.

Table I summarizes all methods that are used in this paper. Three single-imputation and four MI methods were used. The methods include data-driven, model-based, and ML-based

types. We also note that some of the considered imputation methods work only with discrete attributes. The experimental section used the following imputation methods: random sampling multiple imputation (SAM), mean single imputation (Mean) and hot-deck single imputation (HD), regression imputation that uses polytomous and logit MI (POLYLOGREG), LDA together with logit regression MI (LDALOGREG), ML-based Naïve Bayes single imputation (NB), and Naïve Bayes and hot-deck imputations combined with the proposed framework (FNB and FHD, respectively).

### *B. Detailed Description of the Proposed Framework*

In what follows, each module of the proposed framework (see Fig. 3) is explained, and its role in the overall framework and asymptotic complexity is described. We define $n$ to be the number of attributes, $r$ denotes the number of records, $m$ is the maximal number of missing values for an attribute, and $v$ is the maximal number of values for an attribute. We also make the following assumptions: $r \gg n$, $r \gg v$, and $r > m$, where $n$ and $v$ are small integers. Therefore, computational complexity is a function of $r$ and $m$, and the remaining variables are not included in the formation of the estimates.

*1) Mean Pre-imputation Module:* The mean pre-imputation module was developed based on the premise that the base imputation method would benefit, i.e., improve its accuracy, by having a complete database to develop a model and impute the missing data. Completion of the database enhances its information contents, which, if done correctly, ultimately results in the ability to generate a better imputation model. A simple and efficient way of completing the database is to initially impute the missing values and subsequently use the pre-imputed values to perform the actual imputation. The pre-imputation should not worsen the asymptotic complexity of the entire imputation procedure, and therefore, an efficient method should be selected. Mean imputation was selected as the best candidate for this purpose since it is computationally efficient, is simple, and performs imputation with a relatively high accuracy [22], [37], [47]. The benefits of using mean pre-imputation were experimentally verified by applying it to 15 databases when using two imputation methods, i.e., NB and HD, which are later combined with the proposed framework. The results given in Section IV-B2 show that the application of mean pre-imputations with some databases may result in worsening the imputation accuracy, but for majority of them, it results in improvements; that is, on average, 4.5% improvement in accuracy was observed for the HD method and 3.5% for the NB method. Also, since computing the mode or mean values for each attribute from a given database requires one sweep through the data (for computing the mode, the attribute values should be encoded into consecutive integers to avoid searching through all attribute values when computing frequencies), the complexity of performing pre-imputation is linear with respect to the number of records, i.e., $O(r)$, and does not depend on $m$.

*2) Confidence Interval Module:* Confidence interval module is used to filter out possible outlier imputation candidates that are generated by the base imputation method. The filter is based on the premise that imputed values, which are close to the mean (for numerical attributes) or mode (for nominal attributes) of an attribute, have the highest probability of being correct. The filter is designed by computing confidence intervals. Imputed values for a given attribute that fall within the interval are kept, whereas the values outside of the interval are discarded. The confidence intervals are defined as an interval estimate for the mean of an attribute [54]. Confidence intervals, which are related to Student's $t$-test, define a lower limit and an upper limit for the mean to be in the form

$$M - z\sigma_M \leq X \leq M + z\sigma_M$$

where $M$ is the sample mean, $\sigma_M = \sigma/\sqrt{r}$ is the standard error of the mean, $\sigma$ is the standard deviation, and the value of $z$ depends on the assumed level of confidence.

This definition applies to numerical attributes. In the case of nominal attributes, the mean is substituted by the modal value (mode), and the frequency of values for an attribute is computed and normalized as follows: A value of 1 is assigned to the most frequent value for the attribute, 0 is assigned to the frequency of zero, and the frequencies of the remaining attribute values are assigned a normalized value within [0,1]. By analogy to the confidence intervals for numerical attributes, the confidence intervals for nominal attributes are defined as follows:

$$f_{\text{avg}} \leq X$$

where $f_{\text{avg}}$ is the average value of the normalized frequencies for all attribute values.

In other words, imputed values with a frequency lower than the average will be filtered out. To further improve the quality of the filter, for all supervised databases, the confidence intervals are computed individually for each of the predefined classes; that is, a confidence interval is computed for each subset of the database that is associated with a given class value. Section IV-B1 shows experimental validation for the selection of the average value. Normalizing and computing the average frequencies for all values for each attribute from a given database require one sweep through the data (again, the attribute values should be encoded), and therefore, the complexity of computing confidence intervals is linear with respect to the number of records, i.e., $O(r)$, and does not depend on $m$. Also, filtering the imputed values using the confidence intervals requires $O(n * m)$ time since filtering each missing value takes $O(1)$ time. The complexity of the confidence interval module is $O(r) + O(n * m)$ and is linear with respect to the number of records and the total number of missing values. Computation and application of confidence intervals for numerical attributes also exhibit linear complexity.

### *C. Boosting*

Boosting is an ML procedure for improving the accuracy of classification algorithms [25], [50]; a comprehensive reference list can be found at http://www.boosting.org/. In its original version, boosting is a procedure where a set of data models is iteratively generated from a given data set based on modification of weights associated with records. The weights are modified to increase the focus of the next model on generating correct model for records that were misclassified by the preceding models. Classification that is generated by individual models is combined using a voting scheme. In general, both theoretical and experimental studies show that boosting a weak

TABLE II
DESCRIPTION OF THE DATABASES USED IN THE EXPERIMENTS

| Name | # Examples | # Attributes | # Classes | % Boolean attributes | Abbreviation |
|---|---|---|---|---|---|
| Soybean (small) | 47 | 36 | 4 | 36.1 | Soy |
| Postoperative Patient Data | 87 | 9 | 3 | 11.1 | Pos |
| Promoters | 106 | 58 | 2 | 1.7 | Pro |
| Monks1 | 432 | 7 | 2 | 43 | Mk1 |
| Monks2 | 432 | 7 | 2 | 43 | Mk2 |
| Monks3 | 432 | 7 | 2 | 43 | Mk3 |
| Balance | 625 | 5 | 3 | 0 | Bal |
| Tic-tac-toe | 958 | 10 | 2 | 11.1 | Tic |
| CMC | 1473 | 10 | 3 | 30 | Cmc |
| Car | 1728 | 7 | 4 | 0 | Car |
| Splice | 3190 | 61 | 3 | 0 | Spl |
| Kr-vs-kp | 3196 | 36 | 2 | 97.3 | Krs |
| LED | 6000 | 8 | 10 | 87.5 | Led |
| Nursery | 12960 | 9 | 5 | 11.1 | Nrs |
| Kr-V-K | 28056 | 7 | 17 | 0 | Krv |
| Synt256 | 256000 | 21 | 10 | 0 | Syn |

classification algorithm, i.e., algorithms that generated models whose performance is better than the one observed for a plain random drawing, results in a classification that is more accurate than a model that is generated by a single "strong" classification algorithm.

Our framework uses a boosting-like technique. The ultimate goal is to improve accuracy of the imputation by accepting only high-quality imputed values and using them, i.e., additional and reliable information, to impute the remaining values. In general, the module works iteratively and is appended at the end of the imputation process, when all imputed values have been already filtered out. At each iteration, high-quality imputed values are selected and accepted, whereas the remaining values are rejected. In this way, a partially imputed database is created and fed back to the base imputation algorithm. Next, the imputation is repeated, but this time, the concentration is on imputing the remaining values. The number of iterations is set to 10, where all the remaining imputed values are accepted at the last iteration. The number of iterations was established experimentally in Section IV-B1 to balance imputation accuracy and computational time. In general, this value gives, on average, the best accuracy of imputation, whereas the application of a higher number of iterations gives comparable results and requires more computations. The imputed values are accepted or rejected based on their weight and some threshold; that is, all values with weights more than the threshold are accepted, whereas the remaining values are rejected. The weights should reflect the quality of imputation. Their values are dependent on a particular base imputation method. In this paper, two base imputation methods, i.e., NB ML imputation and HD imputation, are investigated. In the case of NB ML imputation, the weights are defined as the probabilities of the selected class variables, i.e., the probability of the predicted class that becomes the imputed value. The threshold is set to be the mean value of the selected class probability for all imputed values. Similarly, for the HD imputation, the weights are defined as the distance between the record with imputed values and the records from which the imputed value was taken, i.e., the two records that are used to perform imputation. The threshold is set as the average distance between the records with missing data and their closest records for all the imputed values. Since weight values are taken directly and without additional computational cost from the base imputation methods, computation of the threshold requires $O(n * m)$ time as it involves computing the mean value among weights for all imputed values. The selection/rejection of imputed values takes $O(n * m)$ time since filtering each imputed value takes $O(1)$ time. Therefore, the complexity of the boosting module is $O(n * m)$. This complexity is not dependent on the number of records and is linear with respect to the total number of missing values.

The overall complexity of the proposed framework is computed as the complexity of pre-imputation, i.e., $O(r)$, summed with the complexity of the confidence interval module and boosting module multiplied by the number of boosting iterations, i.e., $10 * (O(r) + O(n * m) + O(n * m))$. Therefore, the total complexity is $O(r) + 10 * (O(r) + O(n * m) + O(n * m)) = O(r) + O(n * m)$ and is linear with respect to both the number of records and the total number of missing values. We note that the impact of the mean pre-imputation, confidence intervals, and boosting on the quality of imputation will be shown experimentally later in this paper.

## IV. EXPERIMENTS AND RESULTS

### A. Experimental Setup

The proposed framework was tested with a comprehensive set of sixteen databases. The databases were chosen from the University of California at Irvine ML repository [5] and the Knowledge Discovery in Databases repository [27] to ensure that full range of different characteristics, such as number and types of attributes, number of records and classes, is covered. The selected databases include only discrete attributes, as discussed earlier. The description of the selected databases, which is ordered by the number of examples, is shown in Table II.

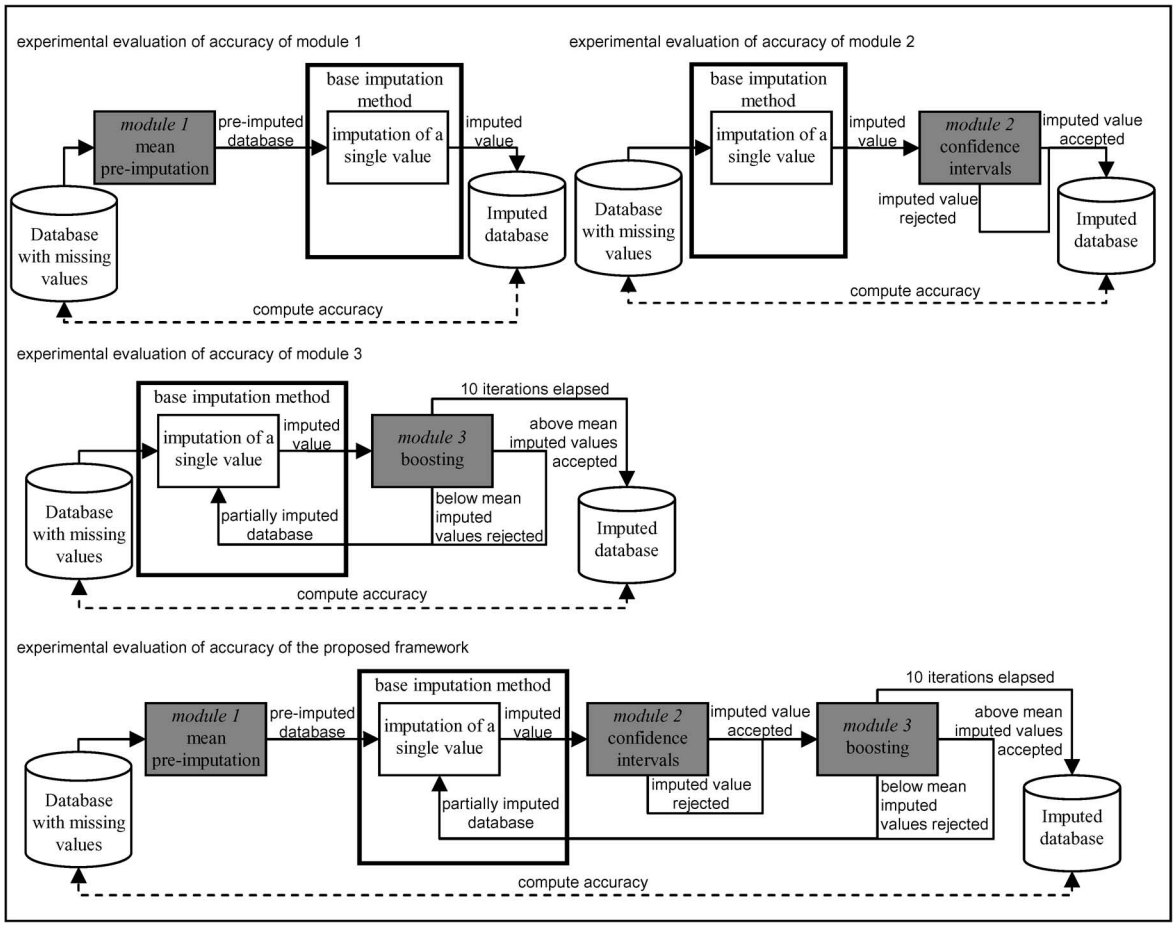


Fig. 4. Experimental evaluation of the proposed framework and its modules.

The *syn* is a synthetic data set being generated using a data set generator published at http://www.datasetgenerator.com and is used to evaluate complexity of the considered methods. The data set was built using the following settings: Number of predicting attributes was set up as 20, domain size of the attributes is equal to 20, number of rules is 10, and number of records was taken as 256 000. The databases originally are complete, and missing data were introduced randomly. This enables computing performance index, in terms of accuracy of imputation, which is defined as the number of correct imputations over the total number of missing values, by comparing imputed values with the original values. Missing values were introduced uniformly into all attributes, except the class attribute. The missing values were introduced at six different levels, i.e., 5%, 10%, 20%, 30%, 40%, and 50%, to demonstrate the impact of the amount of missing data on the quality of imputation.

The experimental section is divided into three parts.

1) *Framework module evaluation*. The goal is to provide motivation for the proposed design of the framework. The effect of each of the three framework modules on the accuracy of imputation improvement is experimentally demonstrated. Fig. 4 shows how the experimental evaluation was performed for each of the modules and for the entire framework.
2) *Experimental comparison with other imputation methods*. The goal is to experimentally compare the quality of imputation between the stand-alone base methods, i.e., NB ML-based imputation and hot deck, base methods in the framework, and other state-of-the-art imputation methods.
3) *Experimental complexity analysis*. The goal is to show that the computational complexity of the application of the proposed framework is linear and, therefore, does not worsen complexity of the base method. Running times for both base methods and base methods in the framework are compared between each other and with the theoretical complexity estimates.

### B. Framework Module Evaluation and Design

This section summarizes experiments that apply the mean pre-imputation, confidence intervals, and boosting modules of the framework in separation to show accuracy gain that corresponds to each of the modules. It also presents experimental results to support selection of the confidence interval size and number of boosting iterations.

*1) Selection of Confidence Interval Size and Number of Boosting Iterations:* Before presenting the results related to the application of individual modules, the confidence interval size and the number of boosting iterations are experimentally determined. The results are computed using 11 databases and both FHD and FNB imputation methods (the HD and NB methods in the proposed framework) since the impact of the

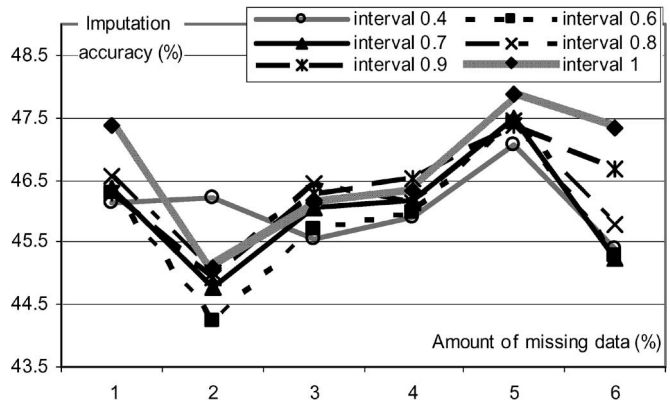


Fig. 5. Accuracy of imputation of FHD for different confidence interval sizes.

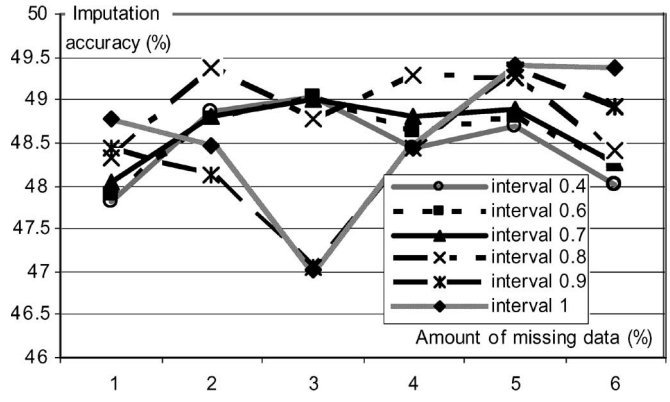


Fig. 6. Accuracy of imputation of FNB for different confidence interval sizes.

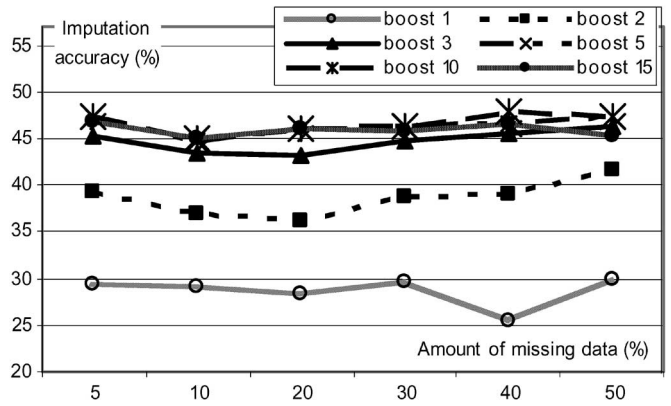


Fig. 7. Imputation accuracy of FHD for different number of boosting iterations.

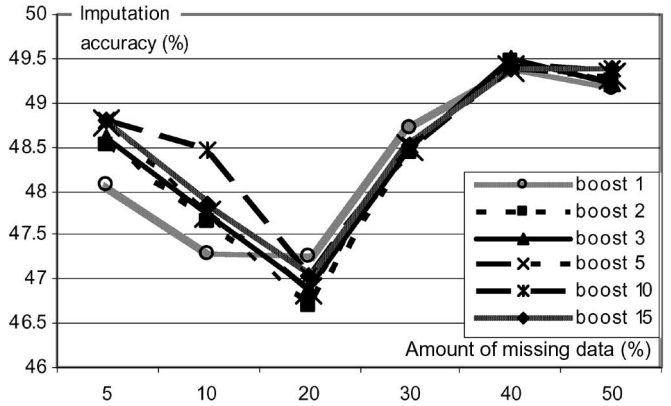


Fig. 8. Imputation accuracy of FNB for different number of boosting iterations.

parameters on quality of the final imputation system is considered. Databases with large number of attributes and/or values, i.e., *Spl*, *Krv*, *Nrs*, and *Krs*, were not used due to the large number of performed experiments (more than 1700) and, thus, very high computational cost.

As shown in Section III-B2, the most appropriate interval size is based on the mean value for continuous or mode value for discrete values. Selecting mean and mode values results in filtering out about half of the imputed values (given that their values are normally distributed), which have relatively high probability of being incorrect when compared to the other half that is kept. This way, the number of accepted values grows relatively fast with the subsequent imputation iterations, and at the same time, high-quality imputed values are accepted. To validate this claim, several other interval sizes based on the mean/mode value are also examined. The selected sizes are $k * \text{mean}(k * \text{mode})$ where $k = 0.4, 0.6, 0.7, 0.8, 0.9, 1,$ and $1.05$.

Figs. 5 and 6 illustrate the average accuracy of imputation for different amounts of missing values ($x$-axis) and different confidence interval sizes for FHD and FNB, respectively, over 11 databases. The results for FHD methods show average accuracies of 46.1%, 45.8%, 46.0%, 46.2%, 46.3%, and 46.7% for $k = 0.4, 0.6, 0.7, 0.8, 0.9,$ and $1$, respectively, over all databases and different amounts of missing data. For $k = 1.05$, the imputation could be completed only for four databases, i.e., *Bal*, *Car*, *Cmc*, and *Pro*, whereas for the remaining databases, too many values were filtered out to complete the imputation process. The results show that slightly better results are achieved with larger value of $k$, and the best accuracy is achieved for $k = 1$. The results for FNB show average accuracies of 48.5%, 48.6%, 48.6%, 48.9%, 48.4%, and 48.6% for the increasing values of $k$, whereas again for $k = 1.05$, imputation could be completed only for four databases, i.e., *Bal*, *Cmc*, *Led*, and *Krv*. Best results were achieved for $k = 0.8$, but results for other interval sizes are very close. We note that, in general, framework with both HD and NB exhibits marginal sensitivity to the interval size setting, as long as the value of $k$ is between 0.4 and 1.0. In general, between FHD and FNB, the best average accuracy of 47.6% was achieved for $k = 1$, and thus, the mean for continuous values or the mode for discrete values is the best setting for the confidence interval size.

The selection of number of boosting iterations is based on comparison of results when both FNB and FHD imputation methods are boosted at different number of times. In general, increasing the number of iterations should result in improving imputation accuracy, but at the same time, more computations are required. The selected number of iterations should be the minimal that gives relatively high accuracy, i.e., accuracy that either does not improve or improves very little when higher number of iterations is used. The experiments apply $k$ iterations, where $k = 1$ (no boosting), $2, 3, 5, 10,$ and $15$, to select the best value.

Figs. 7 and 8 present the average accuracy of imputation for different amounts of missing values ($x$-axis) and different number of boosting iterations for FHD and FNB, respectively,

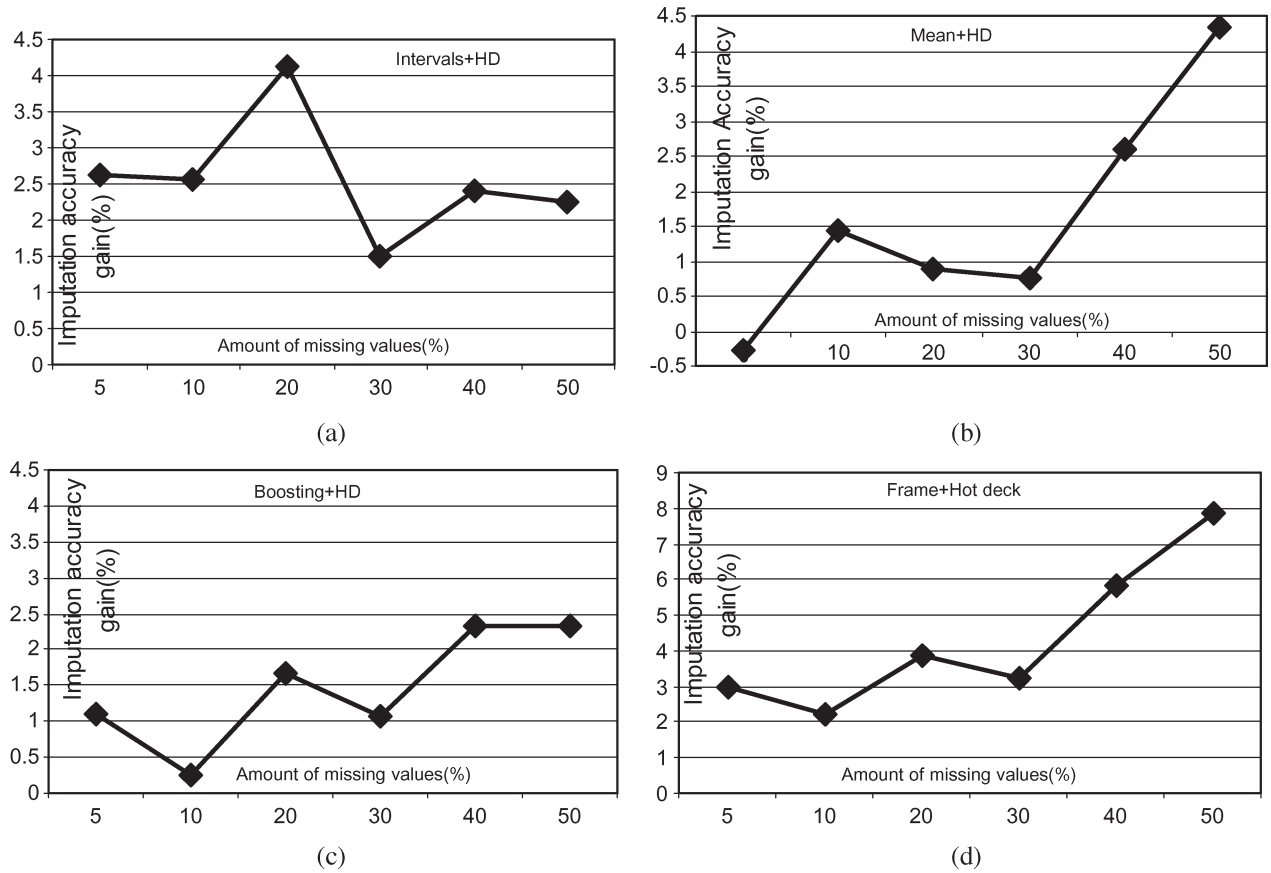


Fig. 9. Improvement in the imputation accuracy for the HD base imputation method. (a) Improvement by using the confidence interval module. (b) Improvement by using the mean pre-imputation module. (c) Improvement by using the boosting module. (d) Improvement by using the entire framework.

over 11 databases. The results for FHD method show that, on average, over all databases and different amounts of missing data, 28.6% accuracy is achieved when no boosting is applied, and with the increasing number of iterations, accuracies of 38.6%, 44.8%, 46.4%, 46.7%, and 45.9% are achieved. The best accuracy is achieved for ten iterations. The results for FNB method show average accuracy of 48.3% for no boosting and two iterations, 48.4% for three and five iterations, and 48.6% and 48.5% for ten and fifteen iterations. Again, the application of ten iterations gives slightly better accuracy, whereas the application of the larger number of fifteen iterations results in comparable results. We note that FHD is sensitive to a different number of iterations, whereas FNB is relatively insensitive. Based on the completed experiments, ten boosting iterations are chosen to be implemented in the framework.

The confidence interval size and the number of boosting iterations are determined separately; that is, we have not investigated the impact of one parameter on the other. While this en block treatment could be potentially beneficial, there are two main reasons to consider them separately: 1) Each module is independent, and the user can consider either all or some of them to implement the framework, which means that a modular design could be well preserved in this manner. 2) The results show that the modules are not sensitive with respect to the specific parameter settings; that is, any confidence interval size between $0.4k$ and $k$ and any number of boosting iteration more than ten can be used. This suggests that when considering both parameters at the same time, the framework is likely to be not sensitive to the optimal combined setting.

*2) Evaluation of the Individual Framework Modules:* The experiments compare the accuracy of imputation of the two base methods under consideration (i.e., HD and NB), the base methods with each of the framework's modules in separation, and, finally, the base methods combined with the entire framework. The mean interval size and ten boosting iterations are used. They are performed on 15 databases (*syn* database is omitted) with six different levels of missing values. The results report an average (over all databases) imputation accuracy gain, which is defined as the difference between the imputation accuracy of base method with one of the framework's modules or the entire framework and the imputation accuracy of the base method, for all considered levels of missing data.

Fig. 9 presents results for the HD imputation treated as the base imputation method. Fig. 9(a) illustrates that applying confidence intervals results in the average imputation accuracy gain of up to 4%. Fig. 9(b) shows that using mean pre-imputation results in the imputation accuracy gain by up to 4.5% and that the improvements are larger for larger amounts of missing data. This is related to poorer imputation accuracy of the base method with the increasing amount of missing data, which is compensated by better effectiveness of the framework's module. Fig. 9(c) shows the impact of boosting, which improves imputation by up to 2.5% and is also characterized by an increasing trend. Finally, the average imputation accuracy gain for the entire framework is shown in Fig. 9(d). It is evident that the use of the proposed framework results, on average, in the increase of the accuracy of imputation by up to 9%, which is a significant improvement. We note that the

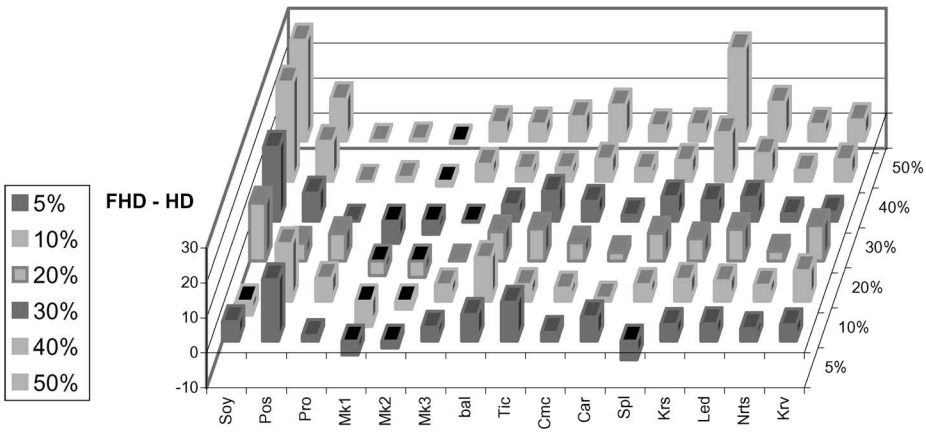


Fig. 10. Difference in the imputation accuracy between HD with and without the framework (FHD − HD) for six levels of missing values and for 15 databases.

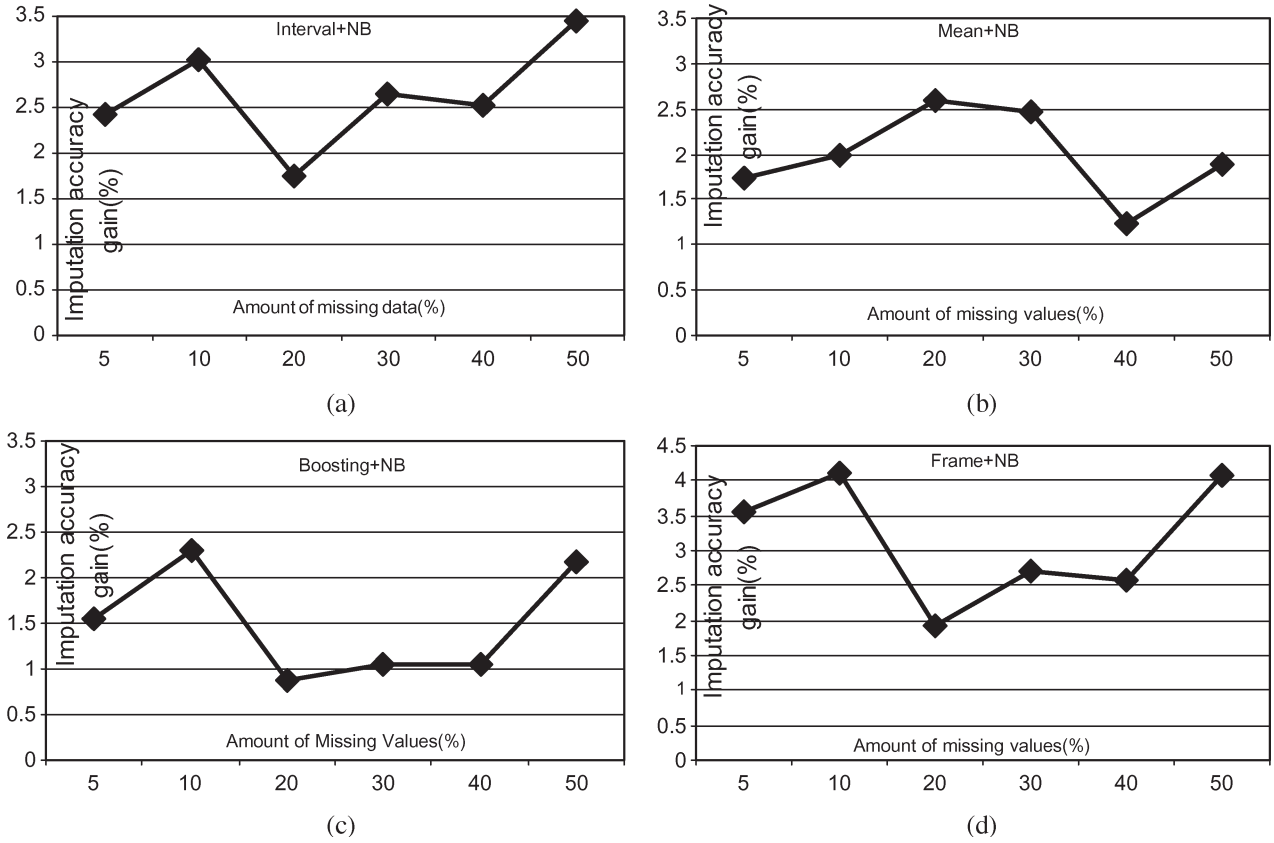


Fig. 11. Improvement in the imputation accuracy for the NB ML-based imputation method. (a) Improvement by using the confidence interval module. (b) Improvement by using the mean pre-imputation module. (c) Improvement by using the boosting module. (d) Improvement by using the entire framework.

individual effects of all modules are not cumulative, but the overall improvement shown by the framework is significantly larger than the improvements generated by each of the modules. In addition, the increasing trend in improvement associated with the increasing amount of missing values shows that the framework can effectively compensate for the degradation of accuracy of the base method.

Fig. 10 shows detailed results concerning the difference between imputation accuracy of the framework with HD as the base method and stand-alone HD imputation for six levels of missing values and 15 databases. The bars with black ceiling represent negative values, which result from the decrease of imputation accuracy related to the application of the framework, whereas gray ceilings show improvement. It is clear that for most databases and different levels of missing data, the imputation accuracy was improved by applying the framework: 13 times accuracy was worse, whereas 47 times it was improved. We note that the highest improvements were about 25%.

In the following graphs, a similar analysis is completed when using NB ML imputation method as the base method. The average improvement in accuracy of imputation is summarized in Fig. 11. Fig. 11(a) shows the impact of confidence interval, which results in the imputation accuracy gain by up to 3.5%. Similarly, Fig. 11(b) and (c) shows the average improvement of imputation accuracy due to applying mean pre-imputation

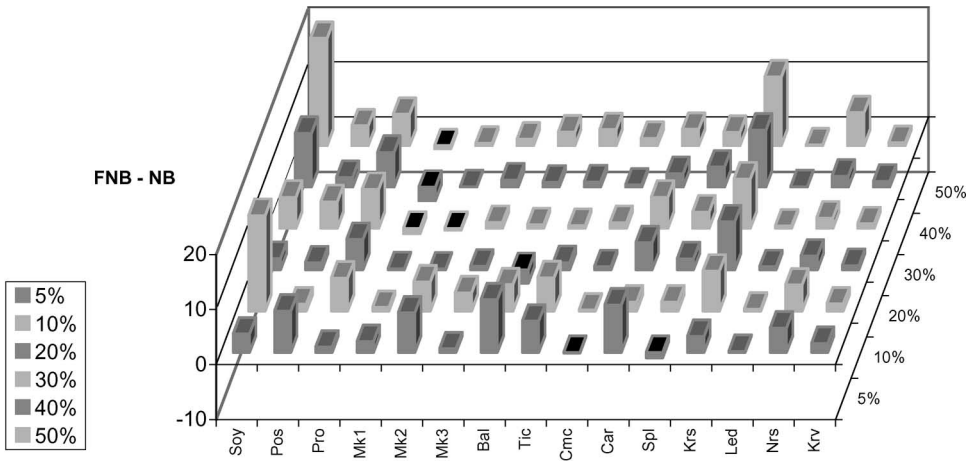


Fig. 12. Difference in the accuracy of imputation between the NB ML imputation method with and without the framework (FNB − NB) for six levels of missing values and for 15 databases.

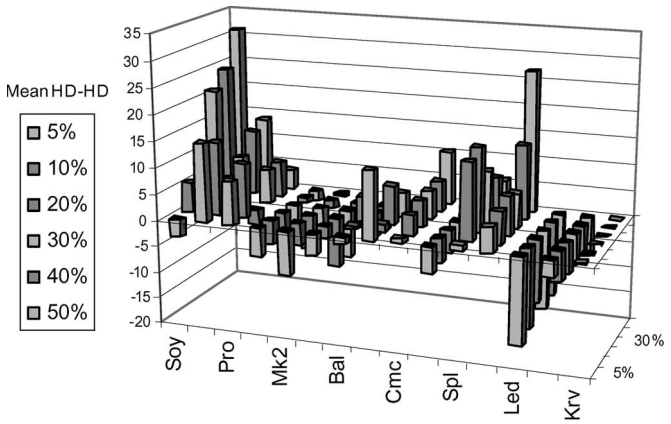


Fig. 13. Difference between HD with and without mean pre-imputation.

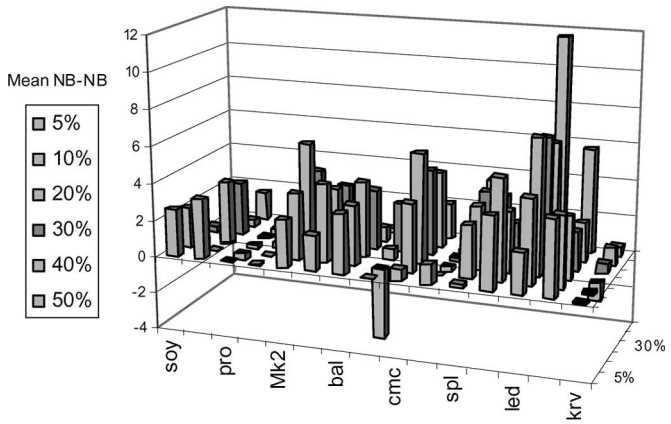


Fig. 14. Difference between NB with and without mean pre-imputation.

and boosting modules, respectively. In both cases, the achieved imputation accuracy gain ranges between 1% and 2.5%. Finally, Fig. 11(d) shows that the application of all modules in tandem results in the imputation accuracy gain up to 4%. Although the effects of all modules are not cumulative, the overall improvement is significantly larger than the improvement resulting from the application of the best module.

Fig. 12 shows the difference in the imputation accuracy between the NB ML imputation method with and without the framework for six levels of missing values and for 15 databases. Similarly to the results shown in Fig. 10, they show that the majority of the values (53 out of 60) are positive, which indicates an improvement in the accuracy of imputation that comes from using the framework.

A detailed breakdown of results related to the effect of mean pre-imputation on both HD [Fig. 9(b)] and NB [Fig. 11(b)] imputation methods is shown in Figs. 13 and 14, respectively. Positive values denote improvement, whereas negative values show when the application of the pre-imputation worsens the accuracy. The results show that for some databases like *Car* and *Led*, mean pre-imputation improves the accuracy by about 30% for the HD imputation method. However, for other databases such as *Mk1*, *Mk2*, and *Mk3* in case of the HD method, it worsens the results. In general, improved accuracy is more frequent for NB imputation method, although HD method is characterized by much higher improvements for some databases. On average, the mean pre-imputation improves the imputation accuracy by 4.5% for the HD method and by 3.5% for the NB method. These relatively high average improvements justify the use of the mean pre-imputation in the proposed framework. As a future work, which is beyond the scope of this paper, a criterion identifying the effectiveness of the mean pre-imputation would be worth developing.

We conclude that the application of each of the framework's modules, in separation and together, always results in some average improvement of imputation accuracy for both of the considered base imputation methods. It can be expected that the application of the framework should, on average, result in the improvement of the imputation accuracy. The level of improvement will be quantified and compared with the performance of other imputation methods in Section IV-C. In general, we note that the level of the imputation accuracy gain depends on the performance of the base method; that is, it is larger for low-quality imputation methods such as HD, whereas it gets smaller for better-quality base methods such as the NB ML algorithm.

Figs. 15 and 16 compare the average accuracy of imputation of the HD and NB ML imputation methods with and without the framework, respectively.

This comparison shows that the application of the framework results in the flattening of the accuracy curve with respect to the increasing amount of missing data, particularly for the

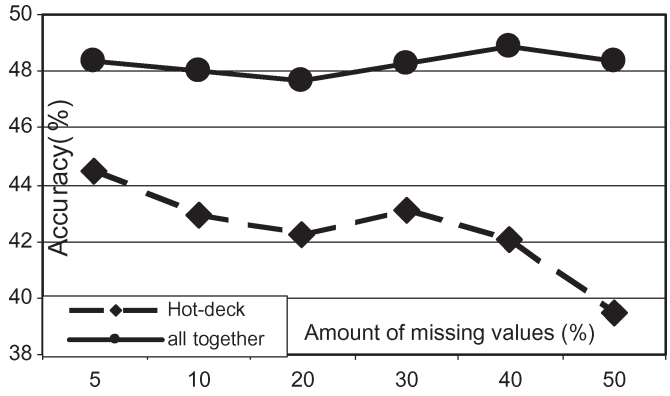


Fig. 15. Accuracy of imputation using the framework with HD and stand-alone HD.

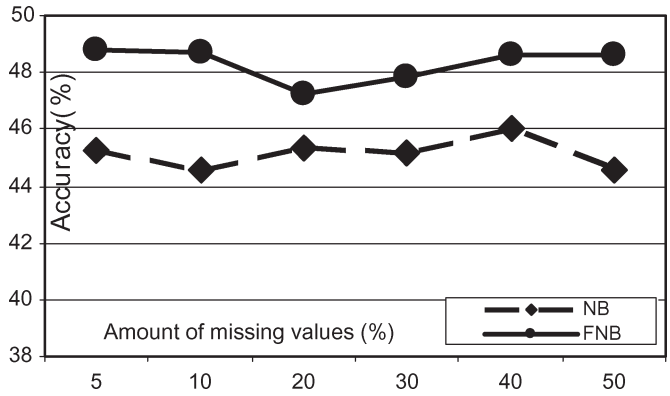


Fig. 16. Accuracy of imputation using the framework with NB ML and stand-alone NB ML method.

HD imputation method (see Fig. 15). The application of the proposed framework compensates for degradation of imputation accuracy of the base method caused by larger amounts of missing values, which is particularly valuable when dealing with sparse databases. We again note that the level of the reported improvement depends on the quality of the base method. For a high-quality method such as NB ML imputation, the improvement is relatively small, i.e., 2%–4%, whereas the accuracy of the base method is, on average, about 44.5%. In the case of HD imputation, the improvement ranges between 4% and 9%, whereas the accuracy of the base method is, on average, about 42%. We note that the accuracies of both imputation methods combined with the framework are very similar.

We stress that our recent study has shown that NB ML method has superior accuracy when compared with HD imputation [22]. At the same time, the application of framework to HD method results in imputation method that has a higher accuracy than the accuracy of stand-alone NB ML-based imputation method. This shows that the proposed framework provides a solution that helps to develop relatively simple and efficient imputation methods that are characterized by high imputation accuracy.

### C. Experimental Comparison With Other Imputation Methods

As described earlier, a representative imputation method from the three categories is chosen for the experimental part. They include data-driven methods such as SAM, Mean, and HD, model-based methods such as POLYLOGREG and LDALOGREG, and ML-based methods such as NB. These methods are compared with FNB and FHD. Therefore, in total, eight methods are compared on 15 databases. The MI methods were set to five imputation rounds. The number of rounds was established experimentally. More rounds resulted, on average, in insignificant or no improvement in accuracy but have worsened the running time.

Fig. 17 shows the average imputation accuracy using the eight imputations and for all considered levels of missing values, over 14 databases (*syn* database is omitted, and *soy* database has not enough records to perform regression-based imputation). The results show that the best results are achieved by the FNB method. The method is consistently better considering the entire spectrum of the missing value levels. The second best is the FHD imputation, which has superior accuracy over more complex model-based imputation methods, such as POLYLOGREG and LDALOGRED, and the ML-based NB imputation, for larger amount of missing values, and similar accuracy for small amounts. The least accurate are the data-driven imputation methods, such as HD, Mean, and SAM. We note that while the HD imputation has a poor performance, applying the framework results in improving the accuracy to be superior to, or at least as accurate as, the accuracy of the advanced model-based methods. We also note that the accuracy of some imputation methods, such as LDALOGREG, POLYLOGREG, and SAM, deteriorates with the increasing amount of missing data, whereas the methods that utilize the framework perform with the same level of accuracy. The experiments clearly demonstrate the effectiveness of the proposed framework, which can be applied to any simple imputation method.

In the scatter plot shown in Fig. 18, the accuracy of the FNB imputation method is compared with the accuracy of all methods that do not operate within the proposed framework. The shown values are the average imputation accuracy for each of the 14 databases, over the six levels of missing values. The $y$-axis position is the accuracy of FNB, whereas the $x$-axis is the accuracy of other imputation methods. Therefore, points above the diagonal line correspond to databases for which FNB achieves better average imputation accuracy. Visual inspection confirms that the FNB imputation method performs better than other imputation methods on the significant majority of the databases. Similarly, a scatter plot of Fig. 19 compares FHD method with other methods that do not utilize the proposed framework. Again, since the majority of the points are located above the diagonal line, we conclude that the FHD method, on average, performs better than other imputation methods.

In the nutshell, experimental results indicate that the application of the proposed framework results in the improvement of imputation accuracy when compared with the accuracy of the stand-alone base imputation method and other state-of-the-art single-imputation and MI methods. Applying the framework to simple imputation methods such as HD results in an imputation method that, on average, performs better than complex model-based imputation methods. We also note that the application of the framework makes the base method more robust to the larger levels of missing data.

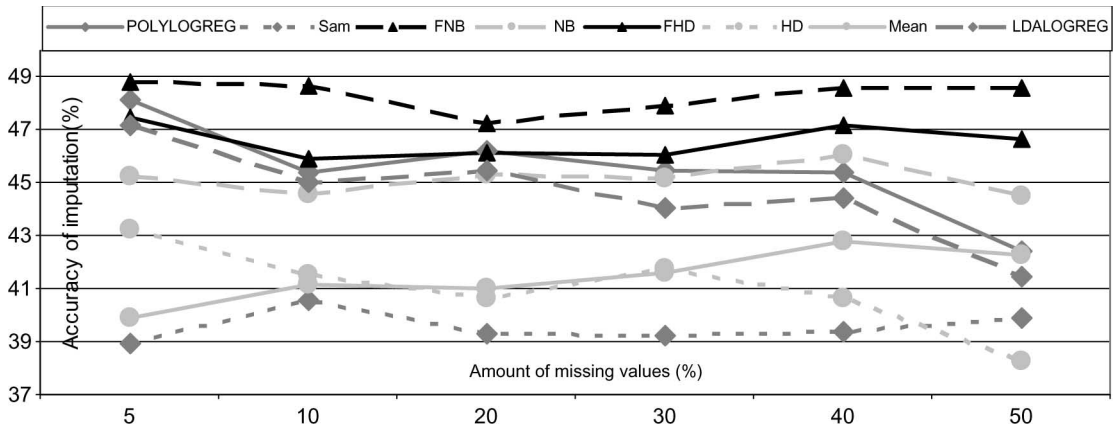


Fig. 17. Summary of imputation accuracy results for the considered eight imputation methods.

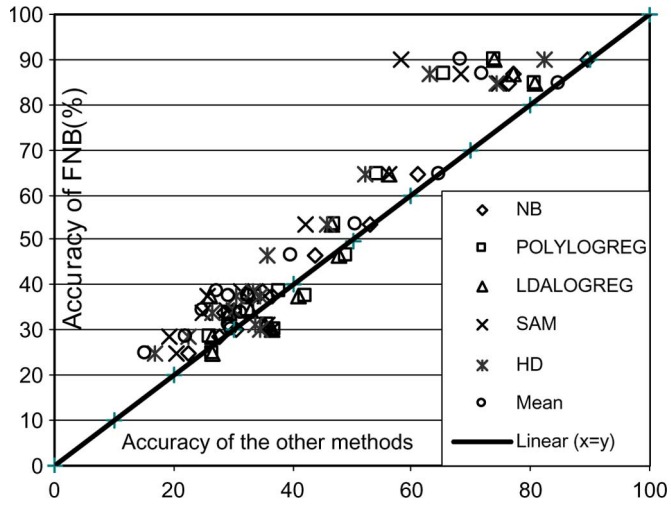


Fig. 18. Accuracy of imputation using the framework with NB against other imputation methods.

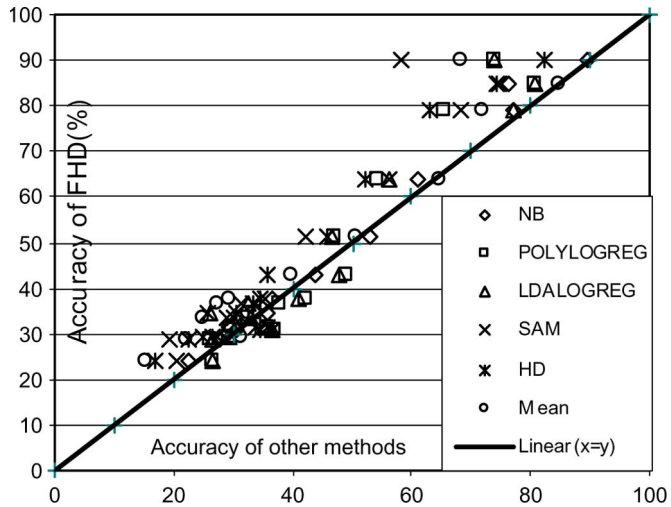


Fig. 19. Accuracy of imputation using the framework with HD against other imputation methods.

### *D. Analysis of Experimental Complexity*

The demonstrated experiments show that the application of the proposed framework results in the improvement of the imputation accuracy. However, the important question is concerned with the computational effort that becomes necessary to apply the framework. Even more importantly, we would like to investigate whether the application of the framework could worsen the computational complexity of the base imputation method. Therefore, the conducted tests are aimed at testing the computational complexity associated with the application of the proposed framework to the base imputation methods. The main goal is to experimentally assess theoretical estimate that implies linear complexity with respect to the number of records. Confirming this hypothesis implies that the application of the framework does not worsen the asymptotic complexity of the base imputation method since there is no imputation method with sublinear complexity. For this purpose, the *syn* database with 256-kB records was chosen to observe the steepness of the running time curve with the increasing size of the database. The *syn* database was used to randomly derive nine databases of different sizes, including 1, 2, 4, 8, 16, 32, 64, and 128 kB and, finally, the original database with 256-kB records. The experiments record the running time on the databases with the incrementally doubled size. Also, to investigate the effect of the level of missing values on the asymptotic complexity of the method, two levels of missing values, i.e., 10% and 60%, were randomly introduced into the databases, and the experiments were performed separately for both levels.

Fig. 20 shows the results of the run time versus the size of the database in the log–log scale for the FNB and NB imputation methods and the two levels of missing values used in the experiments (FNB 10%, FNB 60%, NB 10%, and NB 60%) and for the generated nine databases. Both linear and log–linear curves were plotted on the same figure for the reader's convenience. The curves for the FNB and NB methods align in parallel to the linear curves for both levels of missing data, which shows that the linear asymptotic complexity of the NB ML-based imputation method is preserved when applying the proposed framework. We note that the corresponding curves for the stand-alone and framework-based methods are shifted in parallel. This indicates that additional computational work, which is connected with the application of the framework, is performed, but it does not change the type of asymptotic complexity.

Similar experiment was performed with FHD and HD imputation methods, as summarized in Fig. 21. Closer analysis of the figure shows that plots for both HD and FHD are parallel

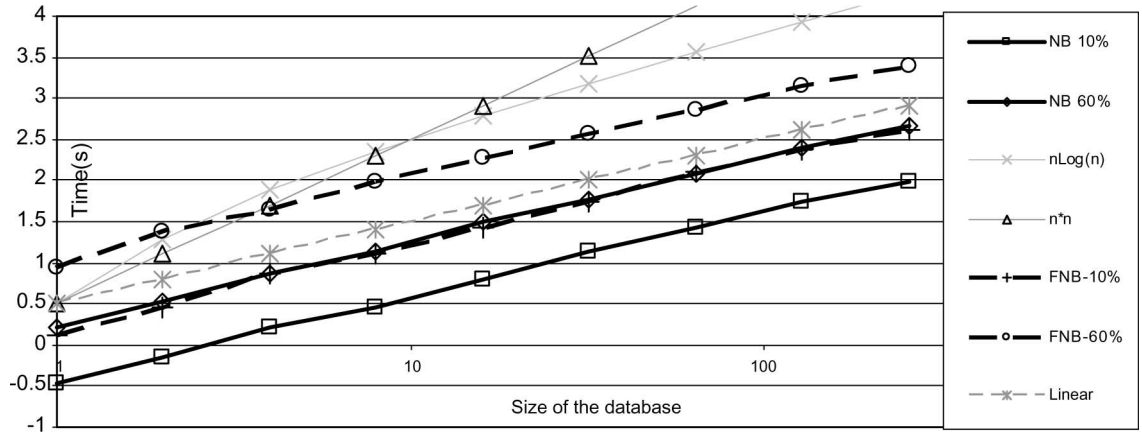


Fig. 20. Run time against the size of the database for the FNB and NB imputation methods and for 10% and 60% of missing values.

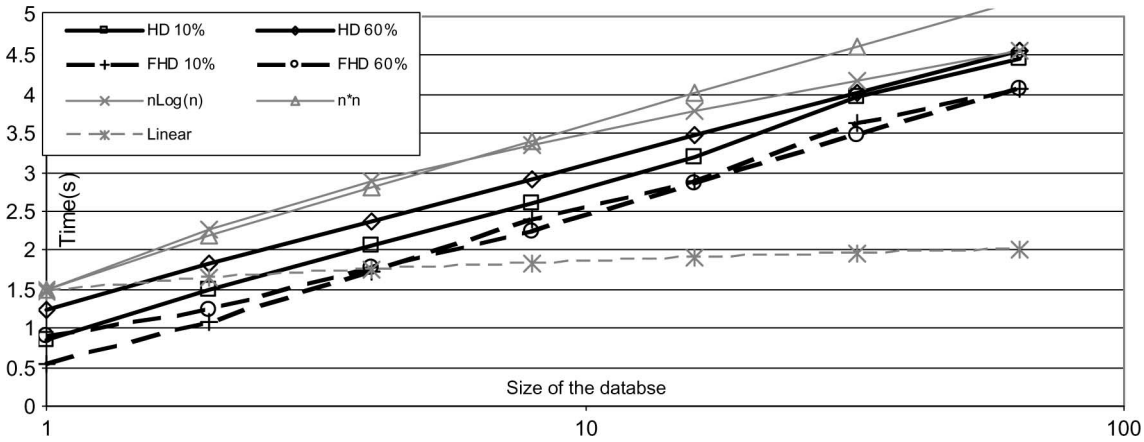


Fig. 21. Run time against the size of the database for HD imputation with and without the framework for 10% and 60% of missing values.

to the quadratic curve. This implies that the original complexity of the stand-alone HD method, i.e., quadratic with the number of data records, is preserved when the framework is applied. The corresponding curves for the stand-alone and framework-based method are shifted in parallel, which indicates identical asymptotic complexity, but the results show that the application of the framework actually shortens the running time when compared to the running time of the stand-alone method. These are the results of applying confidence intervals that filter out less probable candidates for imputed values. Thus, the search space of the HD imputation procedure to find the closest record is reduced, resulting in a shorter running time. Therefore, in case of the HD imputation, the application of the framework results not only in improving the imputation accuracy but also in lowering the running time of the method.

In short, the application of the framework does not change the asymptotic complexity of the base method; however, it results in the increasing accuracy of imputation.

The experimental complexity analysis is supplemented by the running time of the eight considered missing data imputation methods for the 14 databases and for the six levels of missing values (see Table III). The first two rows for each of the missing data levels show the results of imputation methods that use the proposed framework, i.e., FNB and FHD; the next three rows show the results for MI methods, i.e., LDALOGREG, LDALOGREG, and SAM; and the last three rows show the results for the single-imputation methods, i.e., NB, HD, and Mean. The values in boldface indicate the lowest run time for a given database for a given level of missing data.

As expected, the mean imputation is the fastest imputation method. At the same time, Fig. 17 shows that its imputation accuracy, on average, is better than the accuracy of SAM and HD methods. We note that while, in general, high amounts of missing values result in lowering the imputation accuracy, the mean imputation method is robust to the large amount of missing values [22]. Analysis of Table III reveals the following.

- The running time of the most accurate FNB method, which uses the proposed framework, is always significantly shorter than the running time of the considered MI methods, with exception of results for the *krs* database for large amounts of missing data that are a bit higher than the poorly performing SAM method.
- The application of the proposed framework to NB method results, on average, through all experiments, in 3.7 times increase of running time when compared with running time of the stand-alone method. Similarly for the HD method, 1.6 times increase is observed.
- The application of the framework to the HD imputation method may result in the decrease of the running time when compared with the running time of the stand-alone method. It can be observed for the *krs* database and for small amount of missing data for the *pro*, *bal*, *tic*, *car*, and *led* databases. This is attributed to the filtering of less probable candidate imputed values by the confidence

TABLE III
RUNNING TIME OF THE EIGHT IMPUTATION METHODS FOR THE 14 DATABASES AND THE SIX LEVELS OF MISSING VALUES

| | imputation methods | Pos | Pro | Mk1 | Mk2 | Mk3 | Bal | Tic | Cmc | Car | Spl | Krs | Led | Nrs | Krv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5% | FNB | **0.00** | 0.11 | 0.03 | 0.02 | 0.02 | **0.00** | 0.08 | 0.28 | 0.10 | 11.31 | 4.13 | 0.37 | 2.23 | 3.80 |
| | FHD | 0.02 | 0.04 | 0.33 | 0.11 | 0.15 | 0.26 | 0.74 | 2.44 | 2.39 | 616.50 | 40.56 | 45.55 | 1351.11 | 721.03 |
| | POLYLOGREG | 10.04 | 1317.95 | 12.54 | 13.42 | 13.27 | 22.97 | 99.99 | 1201.94 | 126.17 | 714.56 | 71.25 | 62079.59 | 1312.23 | 11887.99 |
| | LDALOGREG | 5.32 | 306.96 | 6.42 | 6.11 | 6.11 | 6.49 | 99.99 | 117.29 | 28.77 | 707.00 | 20.47 | 3310.00 | 301.47 | 1080.00 |
| | SAM | 3.50 | 74.19 | 3.00 | 2.98 | 3.19 | 2.89 | 54.70 | 52.01 | 13.58 | 444.55 | 8.70 | 1978.00 | 177.73 | 371.00 |
| | NB | 0.01 | 0.03 | **0.00** | **0.00** | **0.00** | **0.00** | 0.01 | 0.07 | 0.03 | 1.59 | 0.59 | 0.11 | 0.40 | 0.90 |
| | HD | 0.01 | 0.15 | 0.25 | 0.15 | 0.14 | 0.29 | 1.83 | 2.17 | 3.96 | 364.60 | 130.75 | 58.12 | 907.67 | 490 |
| | Mean | **0.00** | **0.02** | **0.00** | 0.02 | **0.00** | **0.00** | **0.00** | **0.02** | **0.00** | **0.28** | **0.14** | **0.06** | **0.13** | **0.28** |
| 10% | FNB | 0.01 | 0.20 | 0.04 | 0.04 | 0.03 | 0.03 | 0.11 | 0.36 | 0.19 | 14.95 | 5.00 | 0.48 | 2.71 | 4.54 |
| | FHD | 0.03 | 0.09 | 0.25 | 0.28 | 0.32 | 0.44 | 1.90 | 14.11 | 13.82 | 600.00 | 34.16 | 94.83 | 1525.99 | 961.00 |
| | POLYLOGREG | 8.70 | 1374.00 | 12.63 | 11.92 | 13.08 | 21.81 | 98.10 | 1149.67 | 115.94 | 699.72 | 78.55 | 50357.00 | 1294.41 | 12756.00 |
| | LDALOGREG | 5.73 | 297.73 | 6.67 | 6.47 | 6.33 | 6.64 | 98.10 | 111.45 | 30.06 | 649.39 | 21.14 | 3218.00 | 273.40 | 1087.56 |
| | SAM | 3.30 | 76.90 | 2.99 | 3.15 | 2.94 | 2.80 | 56.20 | 44.88 | 13.95 | 153.03 | 8.96 | 1821.00 | 178.00 | 359.00 |
| | NB | **0.00** | 0.04 | 0.02 | **0.00** | **0.00** | **0.01** | 0.03 | 0.11 | 0.04 | 2.03 | 0.75 | 0.16 | 0.52 | 1.50 |
| | HD | 0.03 | 0.13 | 0.28 | 0.29 | 0.26 | 0.53 | 4.08 | 9.50 | 10.19 | 427.33 | 139.00 | 104.17 | 1051.95 | 738 |
| | Mean | **0.00** | **0.00** | **0.00** | **0.00** | **0.00** | 0.02 | **0.00** | **0.03** | **0.02** | **0.31** | **0.14** | **0.06** | **0.14** | **0.30** |
| 20% | FNB | 0.01 | 0.36 | 0.05 | 0.06 | 0.05 | 0.04 | 0.17 | 0.67 | 0.25 | 19.59 | 5.88 | 0.69 | 3.47 | 6.39 |
| | FHD | 0.06 | 0.19 | 0.44 | 0.46 | 0.30 | 1.12 | 4.14 | 12.61 | 6.73 | 439.37 | 86.71 | 104.34 | 975.95 | 1325.84 |
| | POLYLOGREG | 7.97 | 1393.50 | 13.49 | 11.92 | 12.48 | 20.88 | 97.47 | 1064.78 | 112.00 | 620.36 | 59.11 | 46806.00 | 1076.13 | 11253.00 |
| | LDALOGREG | 3.92 | 270.83 | 7.03 | 6.47 | 6.27 | 6.59 | 97.47 | 101.72 | 28.97 | 599.12 | 21.51 | 1607.00 | 250.57 | 1461.99 |
| | SAM | 3.36 | 74.60 | 3.17 | 3.14 | 3.22 | 2.92 | 63.31 | 45.97 | 14.03 | 336.86 | 8.49 | 1955.00 | 192.55 | 416.95 |
| | NB | **0.00** | 0.06 | 0.02 | 0.02 | 0.02 | 0.02 | 0.06 | 0.19 | 0.08 | 2.89 | 0.99 | 0.25 | 0.89 | 3.81 |
| | HD | 0.03 | 0.09 | 0.35 | 0.34 | 0.31 | 0.75 | 6.72 | 8.08 | 6.78 | 378.25 | 140.74 | 103.64 | 1094.03 | 1136.00 |
| | Mean | **0.00** | **0.00** | **0.01** | **0.00** | **0.00** | **0.00** | **0.00** | **0.04** | **0.00** | **0.29** | **0.16** | **0.06** | **0.13** | **0.31** |
| 30% | FNB | 0.02 | 0.50 | 0.06 | 0.08 | 0.06 | 0.08 | 0.24 | 1.03 | 0.35 | 24.58 | 6.78 | 0.83 | 4.45 | 7.39 |
| | FHD | 0.07 | 0.18 | 0.48 | 0.58 | 0.36 | 0.98 | 3.51 | 11.58 | 6.68 | 135.42 | 86.22 | 102.48 | 1165 | 5942.00 |
| | POLYLOGREG | 7.18 | 1467.91 | 11.27 | 11.40 | 11.66 | 18.36 | 95.01 | 932.94 | 102.24 | 563.49 | 51.50 | 37999.71 | 969.40 | 9648.00 |
| | LDALOGREG | 3.85 | 248.51 | 7.07 | 6.40 | 6.49 | 7.07 | 95.01 | 86.05 | 29.34 | 526.98 | 21.68 | 1844.00 | 258.19 | 973.45 |
| | SAM | 3.35 | 76.83 | 2.97 | 3.42 | 2.98 | 2.84 | 58.4 | 48.90 | 19.81 | 340.07 | 8.69 | 1844.00 | 185.33 | 384.00 |
| | NB | **0.00** | 0.07 | 0.01 | 0.03 | 0.02 | 0.03 | 0.08 | 0.27 | 0.11 | 3.72 | 1.19 | 0.31 | 1.21 | 4.06 |
| | HD | 0.02 | 0.11 | 0.32 | 0.33 | 0.28 | 0.82 | 2.84 | 5.20 | 5.53 | 170.95 | 131.11 | 78.40 | 1165.00 | 1585.00 |
| | Mean | **0.00** | **0.00** | **0.00** | **0.00** | **0.00** | **0.02** | **0.00** | **0.03** | **0.00** | **0.29** | **0.18** | **0.07** | **0.14** | **0.31** |
| 40% | FNB | 0.02 | 0.62 | 0.07 | 0.08 | 0.06 | 0.11 | 0.33 | 1.26 | 0.41 | 32.41 | 8.86 | 1.01 | 5.93 | 9.44 |
| | FHD | 0.05 | 0.18 | 0.47 | 0.59 | 0.47 | 0.97 | 3.08 | 18.64 | 6.21 | 303.77 | 98.52 | 96.01 | 1057.00 | 5521.00 |
| | POLYLOGREG | 6.34 | 1456.96 | 11.06 | 10.35 | 11.55 | 16.39 | 82.48 | 862.94 | 88.72 | 426.67 | 47.97 | 32124.69 | 852.56 | 8022.00 |
| | LDALOGREG | 3.89 | 232.92 | 7.02 | 6.61 | 6.64 | 7.47 | 82.48 | 58.81 | 30.46 | 474.25 | 24.08 | 1593.00 | 241.77 | 907.92 |
| | SAM | 3.17 | 76.69 | 3.11 | 3.25 | 3.09 | 2.98 | 54.75 | 59.44 | 14.19 | 324.00 | 8.61 | 1593.00 | 172.81 | 321.49 |
| | NB | **0.00** | 0.09 | 0.04 | 0.03 | 0.03 | 0.03 | 0.09 | 0.30 | 0.14 | 4.58 | 1.61 | 0.35 | 1.54 | 5.16 |
| | HD | 0.02 | 0.12 | 0.26 | 0.25 | 0.26 | 0.55 | 1.7 | 3.26 | 4.99 | 176.22 | 287.75 | 56.05 | 1057.00 | 1203.00 |
| | Mean | **0.00** | **0.01** | **0.00** | **0.00** | **0.00** | **0.00** | **0.00** | **0.02** | **0.02** | **0.31** | **0.16** | **0.07** | **0.16** | **0.32** |
| 50% | FNB | 0.02 | 0.75 | 0.11 | 0.10 | 0.09 | 0.17 | 0.40 | 1.54 | 0.52 | 37.05 | 9.42 | 1.14 | 3.75 | 12.11 |
| | FHD | 0.04 | 0.19 | 0.62 | 0.47 | 0.46 | 0.95 | 2.91 | 17.31 | 6.41 | 195.25 | 81.88 | 97.13 | 566.00 | 5923.00 |
| | POLYLOGREG | 6.61 | 1210.97 | 10.29 | 10.64 | 10.79 | 15.63 | 85.50 | 749 | 96.15 | 383.88 | 45.47 | 28000.00 | 1085.77 | 6069.00 |
| | LDALOGREG | 3.10 | 236.85 | 7.18 | 6.76 | 6.54 | 7.61 | 85.50 | 51.68 | 30.05 | 380.00 | 24.36 | 1689.00 | 263.15 | 858.40 |
| | SAM | 3.19 | 69.02 | 3.08 | 3.39 | 3.06 | 2.88 | 60.59 | 52.70 | 14.58 | 343.75 | 8.83 | 1689.00 | 176.84 | 318.65 |
| | NB | **0.00** | 0.11 | 0.03 | 0.03 | 0.03 | 0.04 | 0.13 | 0.42 | 0.16 | 5.17 | 1.58 | 0.42 | 0.83 | 7.34 |
| | HD | 0.02 | 0.13 | 0.24 | 0.25 | 0.17 | 0.39 | 1.87 | 3.61 | 3.46 | 170.52 | 354.00 | 41.84 | 566.00 | 791.00 |
| | Mean | **0.00** | **0.01** | **0.00** | **0.00** | **0.00** | **0.00** | **0.00** | **0.03** | **0.02** | **0.32** | **0.18** | **0.06** | **0.14** | **0.34** |

intervals, which results in a shorter time to find the closest record. Consequently, computational time may be reduced.

- The regression-based MI method, i.e., POLYLOGREG, is characterized by the longest running time. Its running time is six orders of magnitude slower than the running time of the fastest single mean imputation method and five orders of magnitude slower than the running time of the most accurate FNB single-imputation method (see results for the *led* database). Although the experiments were performed using the same hardware and different software packages (the POLYLOGREG, LDALOGREG, and SAM methods were executed using MICE package [28], whereas the remaining methods were implemented in C++ by the authors), which may result in some minor distortion of running time results, the significance of the difference cannot be disputed.

To summarize, the experiments demonstrate that each module of the proposed framework, i.e., confidence intervals, mean pre-imputation, and boosting, improves the imputation accuracy of the base imputation method. The proposed framework can be successfully used to improve the imputation accuracy of any base method, which can generate weights representing the quality of each imputed value to perform boosting. In practice, almost all existing imputation methods satisfy this requirement. This paper demonstrates how to apply the framework with two imputation methods: HD and NB ML-based imputation. The use of the framework results, on average, in the significant gain

of imputation accuracy when compared with the accuracy of the base method. The results show that a poor-quality single-imputation method such as HD can be improved with the use of the framework to match the quality of the advanced MI methods. The results also show that the NB ML-based imputation combined with the proposed framework achieved the best imputation accuracy. It performed with a higher imputation accuracy and in a lower running time than any of the considered model-based and ML-based single and MI methods. Finally, we have shown, both theoretically and experimentally, that the proposed framework exhibits linear asymptotic complexity, and therefore, its application does not worsen the asymptotic computational complexity of the base method.

## V. Conclusion

Most of the real-world industrial and research databases have a shortcoming of containing missing values. In this paper, a novel framework that aims to improve the accuracy of the existing imputation methods is proposed. The new framework consists of three modules, namely mean pre-imputation, confidence intervals, and boosting, and can be applied to many of the existing imputation methods, including data driven, model based, and ML based. The framework is characterized by a number of advantages. Its application to an imputation method results, on average, in a significant improvement of imputation accuracy while, at the same time, maintaining the same asymptotic computational complexity. For some imputation methods such as hot deck, the application of the framework may even result in lowering the running time, whereas, in general, the computational cost of applying the framework is relatively low.

To demonstrate the advantages of the proposed framework, it was used with two imputations methods: NB ML-based imputation method and HD data-driven imputation method. The two aforementioned imputation methods were experimentally tested on 15 databases and compared with six other popular imputation methods, including single-imputation mean and hot-deck methods, MI random sample, regression, and LDA methods, and ML-based single-imputation NB method. The results show that a significant improvement of imputation accuracy can be achieved by applying the proposed framework and that the accuracy of the framework-based methods was, on average, the highest among the considered methods. We stress that combining the proposed framework with a simple and low-quality single-imputation method, such as hot deck, has resulted in a method that was characterized by the level of the imputation accuracy that is comparable to the accuracy of some advanced MI methods. At the same time, the application of the framework to a better-quality single-imputation method, such as the ML-based NB method, resulted in the imputation accuracy that was superior with respect to the accuracy of other single-imputation and MI techniques. We have also shown a linear complexity of the framework and thus emphasized that it does not change the asymptotic complexity of the associated imputation method.

Finally, as mentioned in Section I, some databases may include large amounts of missing data, i.e., more than 50%. In this case, previous results suggest that unsupervised imputation methods may provide more accurate imputation [22]. Supervised methods build data models, which quality is dependent on the quality of the complete data, to perform imputation, whereas unsupervised methods directly use the complete data. As a result, unsupervised methods seem to be more robust to large quantities of missing data. Our future work will concentrate on investigating the quality of imputation methods for databases with more than 50% of missing data. This will naturally focus our attention on various techniques of unsupervised imputation.

## References


[1] E. Acuna and C. Rodriguez, "The treatment of missing values and its effect in the classifier accuracy," in *Classification, Clustering and Data Mining Applications*, D. Banks, L. House, F. R. McMorris, P. Arabie, and W. Gaul, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 639–648.

[2] C. Alzola and F. Harrell, *An Introduction of S-Plus and the Hmisc and Design Libraries*, 1999. [Online]. Available: http://www.med.virginia.edu/medicine/clinical/hes

[3] J. Barnard and X. L. Meng, "Applications of multiple imputation in medical studies: From AIDS to NHANES," *Stat. Methods Med. Res.*, vol. 8, no. 1, pp. 17–36, Jan. 1999.

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

[5] C. L. Blake and C. J. Merz, *UCI Repository of Machine Learning Databases*. Irvine, CA: Univ. California at Irvine, Dept. Inf. and Comput. Sci.,1998. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[6] J. P. L. Brand, "Development, implementation and evaluation of multiple imputation strategies for the statistical analysis of incomplete datasets," Ph.D. dissertation, Erasmus Univ., Rotterdam, The Netherlands, 1999.

[7] P. Brazdil, J. Gama, and R. Henery, "Characterizing the applicability of classification algorithms using meta level learning," in *Proc. ECML*, 1994, pp. 83–102.

[8] S. F. Buck, "A method of estimation of missing values in multivariate data suitable for use with an electronic computer," *J. R. Stat. Soc.*, vol. B22, no. 2, pp. 302–306, 1960.

[9] S. Buuren and C. G. M. Oudshoorn, *Flexible Multivariate Imputation by MICE*. Leiden, The Netherlands: TNO Preventie en Gezondheid, 1999. TNO/VGZ/PG 99.054.

[10] S. Buuren and C. G. M. Oudshoorn, *Flexible Multivariate Imputation by MICE*. Leiden, The Netherlands: TNO Preventie en Gezondheid, 1999. TNO/VGZ/PG 99.045.

[11] S. V. Buuren, E. M. Mulligen, and J. P. L. Brand, "Routine multiple imputation in statistical databases," in *Proc. 7th Int. Work. Conf. Sci. and Statistical Database Manage.,* J. C. French and H. Hinterberger, Eds., Los Alamitos, CA, 1994, pp. 74–78.

[12] G. Casella and E. L. George, "Explaining the Gibbs sampler," *Amer. Stat.*, vol. 46, no. 3, pp. 167–174, Aug. 1992.

[13] K. Chan, T. W. Lee, and T. J. Sejnowski, "Variational Bayesian learning of ICA with missing data," *Neural Comput.*, vol. 15, no. 8, pp. 1991–2011, Aug. 2003.

[14] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman, "Autoclass: A Bayesian classification system," in *Proc. 5th Int. Workshop Mach. Learn.*, Ann Arbor, MI, 1988, pp. 54–64.

[15] K. J. Cios and L. A. Kurgan, "Hybrid inductive machine learning: An overview of CLIP algorithms," in *New Learning Paradigms in Soft Computing*, L. C. Jain and J. Kacprzyk, Eds. New York: Springer-Verlag, 2001, pp. 276–322.

[16] K. J. Cios and L. A. Kurgan, "CLIP4: Hybrid inductive machine learning algorithm that generates inequality rules," *Inf. Sci.*, vol. 163, no. 1–3, pp. 37–83, 2004.

[17] K. J. Cios and G. Moore, "Uniqueness of medical data mining," *Artif. Intell. Med.*, vol. 26, no. 1/2, pp. 1–24, Sep./Oct. 2002.

[18] K. J. Cios, W. Pedrycz, and R. Swiniarski, *Data Mining Methods for Knowledge Discovery*. Norwell, MA: Kluwer, 1998.

[19] C. Conversano and C. Capelli, "Missing data incremental imputation through tree-based methods," in *Proc. COMPSTAT,* W. Hardle and B. Ronz, Eds., 2002, pp. 455–460.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *J. R. Stat. Soc.*, vol. 82, pp. 528–550, 1978.

[21] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Hoboken, NJ: Wiley, 1977.

[22] A. Farhangfar, L. A. Kurgan, and W. Pedrycz, "Experimental analysis of methods for imputation of missing values in databases," in *Proc. Intell.*

*Comput.: Theory and Appl. II Conf., Conjunction with SPIE Defense and Security Symp. (formerly AeroSense)*, Orlando, FL, 2004, pp. 172–182.
[23] A. J. Feelders, "Handling missing data in trees: Surrogate splits or statistical imputation," in *Proc. 3rd Eur. Conf. PKDD*, 1999, pp. 329–334.
[24] B. M. Ford, *An Overview of Hot-Deck Procedures. Incomplete Data in Sample Surveys*, vol. 2. New York: Academic, 1983.
[25] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, pp. 146–148.
[26] Z. Ghahramani and M. I. Jordan, "Mixture models for learning from incomplete data," in *Computational Learning Theory and Natural Learning Systems, vol. 4, Making Learning Systems Practical*, R. Greiner, T. Petsche, and S. J. Hanson, Eds. Cambridge, MA: MIT Press, 1997, pp. 67–85.
[27] S. Hettich and S. D. Bay, *The UCI KDD Archive*. Irvine, CA: Univ. California, Dept. Inf. and Comput. Sci., 1999. [Online]. Available: http://kdd.ics.uci.edu
[28] N. J. Horton and S. R. Lipsitz, "Multiple imputation in practice: Comparison of software packages for regression models with missing variables," *Amer. Stat.*, vol. 55, no. 3, pp. 244–254, 2001.
[29] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen, "Methods for imputation of missing values in air quality data sets," *Atmos. Environ.*, vol. 38, no. 18, pp. 2895–2907, Jun. 2004.
[30] L. A. Kurgan, K. J. Cios, M. Sontag, and F. J. Accurso, "Mining the cystic fibrosis data," in *Next Generation of Data-Mining Applications*, J. Zurada and M. Kantardzic, Eds. Piscataway, NJ: IEEE Press, 2005, pp. 415–444.
[31] K. Lakshminarayan, S. A. Harp, and T. Samad, "Imputation of missing data in industrial databases," *Appl. Intell.*, vol. 11, no. 3, pp. 259–275, Nov./Dec. 1999.
[32] K.-H. Li, "Imputation using Markov chains," *J. Comput. Simul.*, vol. 30, no. 1, pp. 57–79, 1988.
[33] R. J. Little and D. B. Rubin, *Statistical Analysis With Missing Data*. Hoboken, NJ: Wiley, 1987.
[34] B. F. J. Manly, *Multivariate Statistical Methods: A Primer*, 2nd ed. London, U. K.: Chapman & Hall, 1994.
[35] D. Michie, D. J. Spiegelhalterand, and C. Taylor, *Machine Learning, Neural and Statistical Classification*. New York: Ellis Horwood, 1994.
[36] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
[37] A. Naeim, E. B. Keeler, and C. M. Mangione, "Options for handling missing data in the health utilities index mark 3," *Med. Decision Making*, vol. 25, no. 2, pp. 186–198, 2005.
[38] H. L. Oh and F. L. Scheuren, "Weighting adjustments for unit nonresponse, incomplete data in sample survey," in *Theory and Bibliographies*, vol. 2, W. G. Madow, I. Olkin, and D. B. Rubin, Eds. New York: Academic, 1983, pp. 143–183.
[39] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1992.
[40] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
[41] A. Ragel, "A preprocessing method to treat missing values in knowledge discovery in databases," *Comput. Inf. Syst.*, vol. 2, pp. 66–72, 2000.
[42] D. B. Rubin and J. L. Schafer, "Efficiently creating multiple imputation for incomplete multivariate normal data," in *Proc. Stat. Comput. Section*, 1990, pp. 83–88.
[43] D. B. Rubin, "Formalizing subjective notions about the effect of nonrespondents in sample surveys," *J. Amer. Stat. Assoc.*, vol. 72, no. 359, pp. 538–543, Sep. 1977.
[44] D. B. Rubin, "Multiple imputation after 18+ years," *J. Amer. Stat. Assoc.*, vol. 91, no. 434, pp. 473–489, Jun. 1996.
[45] D. B. Rubin, *Multiple Imputations for Nonresponse in Surveys*. New York: Wiley, 1987.
[46] D. B. Rubin, "Multiple imputations in sample surveys," in *Proc. Survey Res. Methods Section Amer. Stat. Assoc.*, 1978, pp. 20–34.
[47] M. M. Rueda, S. González, and A. Arcos, "Indirect methods of imputation of missing data based on available units," *Appl. Math. Comput.*, vol. 164, no. 1, pp. 249–261, May 2005.
[48] G. Sande, *Hot-Deck Imputation Procedures, Incomplete Data in Sample Surveys*, vol. 3. New York: Academic, 1983.
[49] W. S. Sarle, "Prediction with missing inputs," in *Proc. 4th JCIS*, 1998, vol. 2, pp. 399–402.
[50] R. E. Schapire, "A brief introduction to boosting," in *Proc. 16th Int. Joint Conf. Artif. Intell.*, 1999, pp. 1401–1406.
[51] J. L. Shafer, *Analysis of Incomplete Multivariate Data*. London, U.K.: Chapman & Hall, 1997.
[52] J. L. Shafer, "Multiple imputations: A primer," *Stat. Methods Med. Res.*, vol. 8, pp. 3–15, 1999.
[53] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.
[54] G. W. Snedecor and W. G. Cochran, *Statistical Methods*, 8th ed. Ames, IA: Iowa State Univ. Press, 1989.
[55] V. Tresp, R. Neuneier, and S. Ahmad, "Efficient methods for dealing with missing data in supervised learning," in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. Cambridge, MA: MIT Press, 1995, pp. 689–696.
[56] W. Vach, "Missing values: Statistical theory and computational practice," in *Computational Statistics*, P. Dirschedl and R. Ostermann, Eds. Heidelberg, Germany: Physica-Verlag, 1994, pp. 345–354.
[57] R. Winter and K. Auerbach, *Contents Under Pressure*, May 2004, Intelligent Enterprise. [Online]. Available: http://www.intelligententerprise.com/showArticle.jhtml?articleID=18902161
[58] W. Zhang, "Association based multiple imputation in multivariate datasets: A summary," in *Proc. 16th ICDE*, 2000, pp. 310–311.
[59] P. Zhang, "Multiple imputation: Theory and method (with discussion)," *Int. Stat. Rev.*, vol. 71, no. 3, pp. 581–592, 2003.

**Alireza Farhangfar** received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Tehran, Tehran, Iran, in 2000 and 2002, respectively, and the M.Sc. degree in computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2004. He is currently working toward the Ph.D. degree in the Computing Science Department, University of Alberta, holding three major federal and provincial scholarships for his Ph.D. program.

His research interests include machine learning, data mining and knowledge discovery, computational biology, and bioinformatics.

**Lukasz A. Kurgan** (M'02) received the M.Sc. degree (with honors) in automation and robotics from the AGH University of Science and Technology, Krakow, Poland, in 1999 and the Ph.D. degree in computer science from the University of Colorado at Boulder in 2003.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is the author or coauthor of several inductive machine learning and data mining algorithms and of more than 50 papers published in international journals and conference proceedings. His research interests include data mining and knowledge discovery, machine learning, computational biology, and bioinformatics. He currently serves as an Associate Editor of the *Neurocomputing* journal and is actively involved in organization of special issues related to computational biology and bioinformatics.

Dr. Kurgan is a member of the Association for Computing Machinery and the International Society for Computational Biology. He is a Steering Committee Member for the International Conference on Machine Learning and Applications and has been a member of numerous conference program committees in the area of data mining, machine learning, and computational intelligence. He was a recipient of the Outstanding Student Award.

**Witold Pedrycz** (M'88–SM'94–F'99) received the M.Sc. degree, the Ph.D. degree, and the D.Sc degree from the Silesian University of Technology, Gliwice, Poland, in 1977, 1980, and 1984, respectively.

He is a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also a Canada Research Chair in Computational Intelligence. He has published numerous papers in this area and is also an author of seven research monographs covering various aspects of computational intelligence and software engineering. He is actively pursuing research in computational intelligence, fuzzy modeling, knowledge discovery and data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, relational computation, and software engineering.

Dr. Pedrycz has been a member of numerous program committees of IEEE conferences in the areas of fuzzy sets and neurocomputing. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A, and the IEEE TRANSACTIONS ON FUZZY SYSTEMS.