International Journal of Software Engineering
and Knowledge Engineering
Vol. 17, No. 1 (2007) 1–26

PERFORMANCE EVALUATION OF IMPUTATION METHODS FOR INCOMPLETE DATASETS

SUMANTH YENDURI

730 East Beach Blvd, Department of Computer Science,
University of Southern Mississippi, Long Beach, MS 39560, USA
Sumanth.Yenduri@usm.edu

S. S. IYENGAR

298 Coates Hall, Department of Computer Science,
Louisiana State University, Tower Drive, Baton Rouge, LA 70803, USA
iyengar@bit.csc.lsu.edu

Received 17 January 2004
Revised 28 February 2006
Accepted 4 April 2006

In this study, we compare the performance of four different imputation strategies ranging from the commonly used Listwise Deletion to model based approaches such as the Maximum Likelihood on enhancing completeness in incomplete software project data sets. We evaluate the impact of each of these methods by implementing them on six different real-time software project data sets which are classified into different categories based on their inherent properties. The reliability of the constructed data sets using these techniques are further tested by building prediction models using stepwise regression. The experimental results are noted and the findings are finally discussed.

Keywords: Hot-deck; maximum likelihood; imputation.

1. Introduction

The problem of missing or incomplete data is common in many data bases [1] and is more severe in data collected through on-site surveys [2]. Little attention has been given to this problem in the field of Software Engineering. Significant amounts of missing or incomplete data are frequently found in data sets utilized by the effort/cost/time prediction models used in the current software industry. By knowing these estimates early in the software project life cycle, project managers can manage and exploit resources efficiently in order to meet the cost/time constraints. Traditional approaches ignore all the missing data and provide estimates based on the residual complete information. Thus, the estimates tend to be biased. To date, most companies rely on their historical database of past project data sets to predict estimates for future projects. Like other data sets, software project data sets also contain significant amounts of missing/incomplete data. Missing data create difficulty in scientific research as the statistical data analysis techniques used are not designed for them. Hence missingness causes conceptual and computational difficulties [3].

What are missing values and how are they caused?

Missing values within a data set are values due to lack of response or erroneous response. They include all the answers such as "null", "don't know", "unanswered", and so forth. The reasons for missing data are numerous. To begin with, data collection is a very painstaking (in terms of both effort and time) and costly process. The cost in collecting, reporting and maintaining data is not trivial [4, 5]. The estimates for collecting and storing data would amount from 5–10% of the total software project cost [6]. "Wild values" are another reason for missing values. A value is called a wild value when we know for sure that the value is not correct. For example, a categorical variable having a numerical value or an interval scaled variable having an alphabetic value. Punching errors or the recorder's ignorance may be the reasons for this. The most common remedy in practice for wild values is to enter "nothing" in place of the wild value, thereby creating more missing data. Not only these, but unanswered checklists/questionnaires, skipped questions, inefficient data collection may contribute to missingness in data sets.

The impact of missing values on data analysis!

Statistical methods presume that every case has information on all the variables to be included in the analysis. Hence missing data reduce the statistical power. Power represents the validity of the statistical inferences drawn from the data set. The inferences may represent relativity between variables, measures of dispersion or anything else. Further, estimates calculated from these unreliable data sets could be biased. Currently, companies ignore all the missing information and rely on the remaining complete information in order to provide estimates. This means that the companies are using lesser information to make predictions for future projects. Without accurate estimates, it would be a daunting task to manage software projects. Time and money wastage would be direct results of inaccurate estimates.

How to encounter the "missing data" problem?

The reasons for the cause of missing data reconfirm to us that it is inevitable to have data sets with missing data. Obviously, we know the difficulties caused by missing data. Various disciplines have employed the use of "Missing Data Techniques" (MDTs) or "Data Imputation Algorithms" in order to reconstruct the missing data within a data set. These procedures seem to be a promising approach to counter the problem. Imputing data means filling out probable values for the missing data. Imputation examines the range of probable values for each variable and calculates many predicted values randomly. An analyst will end up with numerous credible data sets by using these methods. The results often produce more accurate estimates. Numerous procedures are found in the literature [3] but few software engineering researchers have employed them in their analysis. Initial research has shown that there have been better prediction accuracies when relatively simple data imputation methods were applied to the software project data sets instead of the traditional practices of ignoring missing data [1, 7, 8].

The goal of this study is to analyze numerous data sets using statistical tools under various patterns of censorship and mechanisms governing missingness and data imputation. We try to show the effects of incomplete data on useful experimental analyses, how incomplete data can and probably should be dealt with, and how experiments can actually benefit from imputing data. We elaborate some potential benefits in imputing data. We intend to answer the following questions to the best of our knowledge: Does incomplete data effect predictions? When will these incomplete data models fail? and How can these prediction accuracies be improved?

Our primary aim was to investigate if accuracies of the estimates improved when completeness of a data set is enhanced using imputation techniques. We tried to maximize the response in the data set for the same [1, 3]. We test four different imputation procedures (Listwise Deletion (LD), Ten Hot-Deck (HD) Variants, and Full Information Maximum Likelihood (FIML) Approaches) on six real-time software project data sets in order to study their impact under different conditions. The most common approach, LD was used in order to compare if other imputation methods performed better [3]. We used MI to test if simple imputation techniques gave better prediction accuracies. We used HD variants because of their broad usage and proven performance [27–30]. Finally, we used FIML [7, 25] in order to investigate their robustness under different conditions. The results show that we found a reasonable improvement in the prediction accuracies. We discuss the related research in the next section. Our review focuses on usage of imputation methods in the discipline of software engineering. In the third section, we make a note about the different methods available, the background about missing mechanisms, a description about the prediction model used and finally discuss the methods implemented in this study. In the fourth section, we describe the data sets used for the analysis and provide a classification scheme for these data sets based on different parameters such as size, missing mechanism, percentage of missing data etc. In the next section, we list our experimental results and further discuss the performance of these methods. Finally we elaborate on our findings about the usage of these methods under different circumstances.

2. Literature Review

Schafer and Graham [9] said that until 1970s missing data values were handled by editing. The foundation work [10] on handling incomplete data was done by Rubin in 1976. Since then, many researchers in different disciplines employed these missing data techniques. The work was later summarized by Little and Rubin in 1987 [3] where the traditional methods were grouped into four categories: listwise deletion, imputation-based procedures, weighting procedures and model-based procedures. Cox and Folsom [11] in the late 70s performed simulations on different MDTs and reported that hot-deck imputations performed better than listwise deletion. In 1983 [12], Kaiser showed the performance of hot-deck methods were inversely proportional to the rate of missing data in the data set. Numerous studies [2, 14–19] found application of data imputation methods performed better than the listwise deletion method or pairwise deletion. El Emam and others used MDTs to fill in missing values and argued hot-deck imputation performed better than simple imputation methods [20]. We cannot say if the particular hot-deck is appropriate as important information was not provided. Summary of result statistics have not been listed. Neither the amount of data missing and in what variables data are missing or missingness mechanism is provided. "Don't know" responses were treated as missing values in their study. Finally, their results indicate that all techniques did well. But, they recommend LD to be a reasonable choice.

Kevin Strike *et al.* in 2001 [1] explored using MDTs for dealing with the problem of missing values in historical data sets when building software cost estimation models. They investigated listwise deletion, mean imputation and hot-deck imputation methods to fill the missing data. This was the first research implementation (to our knowledge) of MDTs to software engineering projects data sets in recent times. Only 3 methods were used and missingness was simulated based only on 3 productivity factors out of 15. The excluded factors may have had correlation with the 3 factors used thus affecting the performance of imputation in the hot-deck methods used. Though the data set was sizeable, only one dataset was used in the experiment. The results showed promise but the authors claim for application of more techniques on a number of data sets to determine which techniques would produce maximum prediction accuracy.

Ingunn Myrtveit *et al.* in 2001 [7] evaluated four missing data techniques in the context of software cost modeling: listwise deletion (LD), mean imputation (MI), similar response pattern imputation (SRPI), and full information maximum likelihood (FIML). It is the first time both sample-based and model-based methods were used for data imputation and compared at the same time. Their evaluation suggests that FIML is the appropriate imputation strategy when the data are not missing completely at random (MAR) but there must be sufficient data for this technique. They only consider the removal of cases and of course would be better to remove features too. They concluded that unlike FIML, prediction models constructed on LD, MI and SRPI data sets will be biased unless the data are MCAR. A superficial analysis of their results suggests the best model was derived when no data was imputed. It may have been the result of their analysis procedure. Little evidence was provided about the better performance of SRPI over MI. Their results were inconclusive. They too experimented on only one data set (sizeable) but were limited to ERP projects. The data set lacked diversity of projects which makes us question the applicability of their results to a multitude of software project data sets available. Their results can be further justified only by applying FIML to more number and variety of data sets.

In April 2003, Song and Shepperd [21] experimented with Multiple Imputation techniques for solving the problem of missing data in software project data sets. They investigated if a simple bootstrap based on a k-Nearest Neighbor method could solve the issue. They used two data sets each having cases around 20. They could not conclude if the Multiple Imputation methods were always useful for small sized software project data sets because of the low percentage of missing data.

In May 2004, Song *et al.* [22] analyzed the small sized nature of the software data sets as an important characteristic and explored using simple methods of imputation for them. They proposed a class mean imputation (CMI) method based on k-Nearest Neighbor hot deck imputation method to impute both continuous and categorical missing data in small data sets. They used an incremental approach to increase the variance. To evaluate their imputation method, they used data sets with 50 and 100 observations from a larger industrial set with varying missing data percentages. They simulated by taking into consideration both MCAR (Missing Completely At Random) and MAR (Missing At Random) mechanisms. Their result suggests their new method performed well but could be used to impute missing values in small sized software data sets only. Furthermore, there method needs to be tested on different data sets to replicate their findings.

3. Background

Table 1 depicts the various imputation strategies used by researchers from various fields. Based on the literature, the Data Imputation methods can be roughly grouped into four categories [3]: Methods Based on Complete Information, Weighting Methods, Methods Based on Imputation, Model-Based Methods. More generally, all the methods can be categorized as Random Imputation Methods and Deterministic Imputation Methods. The former methods draw imputation values randomly either from observed data or from a predicted distribution whereas the latter determine only one possible value for each missing observation.

3.1. *Ignorable and non-ignorable missing mechanisms*

Handling missing data is dependent upon how the data are missing. It is imperative to methodically categorize the data. Missing data mechanisms are classified by Rubin [3] as Ignorable and Non-Ignorable (NI). Often researchers assume that the missingness is Ignorable. Furthermore, Ignorable missing data mechanism is classified into Missing Completely at Random (MCAR) and Missing at Random (MAR).

3.1.1. *Ignorable missing data mechanisms* (*MAR, MCAR*)

The data are *Missing at Random* (MAR) means that the probability that the missing observations may be dependent on Y_o but not on Y_m (where Y represents our

Table 1. Data imputation methods.

Methods Based on Complete Information	*Listwise Deletion/Complete Case Analysis*	
	Pairwise Deletion/Avaialble Case Analysis	
Weighting Methods	*Weighting Cell Adjustments*	
Imputation Methods	*Estimation Methods (Unconditional/Conditional Mean Imputation etc.)*	
	Substitution Methods	
	Hot Deck Imputation Methods	*Adjustment Cells* *Nearest Neighbor Hood Approach* *Ex: k-NN Approach, SRPI*
	Cold Deck Imputation Methods *Composite Methods Ex: Regression Based Hot Deck Method etc.*	
Model-Based Methods	*Regression Based Imputation Methods* *Stochastic Regression Imputation Methods* *Multiple Imputation Methods* *Maximum Likelihood Approaches such as Expectation Maximization* *Algorithm, Full Information Maximum Likelihood Approach*	
Other Modern Methods	*Principal Components Analysis* *Clustering Techniques* *Neural Networks*	

data set in matrix form. Y_o represents the observed values in Y and Y_m represents the missing values in Y)

$$P(Y|Y_m, \delta) = P(Y|Y_o, \delta)\,, \tag{1}$$

conditional on a set of predictor variables δ. It means that missingness is not related to the missing values but may be related to the observed values of other variables in the data set. Cases with incomplete data differ from cases with complete data, but the missing pattern is predictable from other variables rather than being due to the specific variable on which the data are missing. For example, incompetent programmers may not want to answer all the questions on the productivity factor documents in order to hide their performance. The reason for missing data is because an external effect. MAR depends on the data and the model [23].

The data are *Missing Completely at Random* (MCAR) means the probability that the missing observations are not dependent on Y_o or Y_m.

$$P(Y|Y_m) = P(Y|Y_o) \tag{2}$$

It means the missingness is not dependent upon the values of any of the other variables in the data set (missing or observed). Cases with complete data are indifferent from cases with incomplete data. For example, suppose a personnel shuffles unadjusted productivity factor documents and arbitrarily discards some of them. If the observed values were a random sample of the complete data set, complete case analysis would give the same result similar to that of a complete data set.

This is a special case of MAR. It is more restricted. This mechanism is very easy to deal with but unfortunately data are seldom MCAR. This situation arises because the data were missing by design. The data can be tested for this condition (SYSTAT and SPSS MVA have implemented this feature). No such tests are available for the MAR condition. If the parameters of the data model and the missing parameters are different, then the missing data mechanism is Ignorable.

3.1.2. *Non-ignorable missing data mechanism* (*NI*)

Nonignorable (NI) means the probability that the missing observations may be dependent on Y_m but not on Y_o. Missingness is related to Y_m, it is non-random and it cannot be predicted from other variables of the data set. This situation arises because the missing pattern can be explained but it can only be explained by the variables where data are missing. For instance, the personnel responsible for answering the questionnaires using online forms are more likely to fill in information about their productivity factors. Suppose we cannot predict which personnel use online forms. Under such conditions, the missing mechanism is Non-Ignorable. This is the most difficult condition to deal with.

Ignorability is a judgment made by the data analyst and it depends both on the missing data mechanism as well as the data. In practice it is usually difficult to meet the MCAR assumption. MAR is an assumption that is more often used. Schafer and Graham [9] state: "When missingness is beyond the researcher's control, its distribution is unknown and MAR is only an assumption. In general, there is no way to test whether MAR holds in a data set, except by obtaining follow-up data from nonrespondents or by imposing an unverifiable model." Rubin [10] suggested that when dealing with real data, the data analyst should explicitly consider the process that causes missing data. For example, we might look at survey sampling containing missing data, where only a few variables are observed for all units in the population and a few survey variables are "missing" for units that are not given importance. The mechanism causing missing data would then be the process of variable collection. If variables are given importance in such a way, the mechanism is under the control of the data analyst and may be assumed "ignorable" [2].

3.2. *Patterns of missing data*

Let X_1 to X_k be the variables represented in a matrix form. If all the values are observed and if X_k has p values completely observed, then we say that the data are missing in univariate pattern (Fig. 1(a)). If X_1 to X_k are ordered in such a way that if X_j is missing for a unit, then $X_{j+1}, \ldots, X_k$ are missing for that unit too. Such a pattern is called monotonous pattern (Fig. 1(b)). Finally if the values are

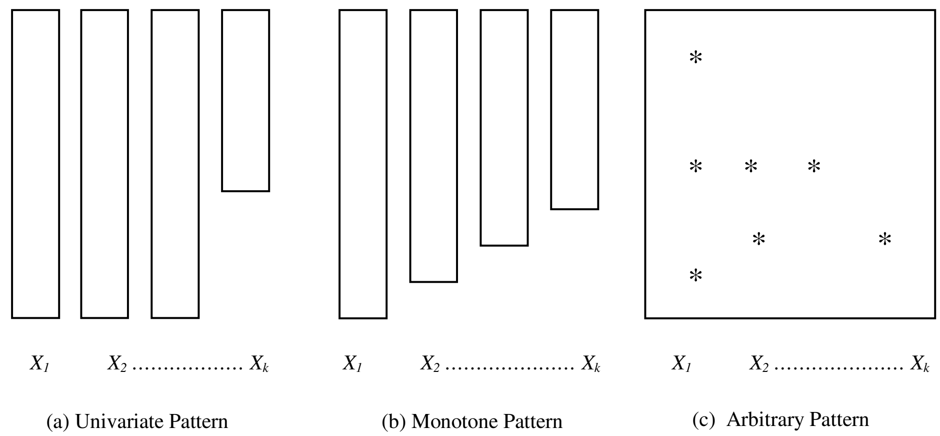


Fig. 1. Patterns of missing data.

missing in a haphazard fashion in which any variable may be missing for any unit, then we say that the data are missing in arbitrary pattern Fig. 1(c).

3.3. *Stepwise regression model*

Using the above described imputation methods, individual complete data sets were generated. To study the impact of these methods, the data sets were evaluated using prediction models. A significant step in the construction of a prediction model is the selection of independent variables. We used the Forward Entry Stepwise Regression Model-Building Procedure. To begin with, an initial model is identified. It always includes the regression intercept. Next "iterative stepping" is performed. That is changing the model repetitively by adding or removing a predictor/independent variable, which is based on the "stepping constraints (tests)". Finally the termination procedure is initiated when stepping cannot be done any more or if the maximum number of steps has been reached.

Initially, among all the independent variables, one variable is selected to enter the model. The independent variable that minimizes the residual sum of squared deviations and has a regression coefficient significantly different from zero is selected. Let $X_1, X_2, \ldots, X_p$ be the independent variables and $\beta_1, \beta_2, \ldots, \beta_p$ be the regression coefficients associated with the variables respectively (Y is the dependent variable). Then the hypothesis H: $\beta_i = 0$ is rejected in order to enter the variable X_i into the model. After the selection of the first variable, we select the second variable X_j from the remaining set such that the residual sum of squared deviations for the second selected variable combined with that of X_i is minimum and the partial correlation coefficient β_j of the second variable is significantly different from zero. The hypothesis H: $\beta_j = 0$ is rejected in order to enter the variable X_j into the model. Once X_j is entered, a test is performed to see if the first variable X_i should be included given that X_j is present in the model. If $H\colon \beta_i = 0$ is rejected both the variables remain or else X_i is removed. Thus the iterative process continues until the stepping criterion fails or if the maximum number of steps is reached.

3.4. *Methods implemented*

3.4.1. *Listwise deletion*

In List wise deletion any case/row with one or more missing values in the data set is deleted. Only complete cases are used for further analysis.

3.4.2. *Mean imputation*

Mean Imputation (MI) works by taking into account the available observations for that particular variable and fills missing values with the mean of the available observations.

3.4.3. *Hot-deck methods*

It involves filling missing value with another value drawn from other complete cases (donors) in the data set. Basically hot-deck imputation selects a recorded value that best suits the missing value and replaces it.

3.4.3.1. *Sequential hot-decking*

The procedure starts sequentially from the beginning (the first case) of the data set. The closest preceding complete case was used as a donor to impute the missing values.

3.4.3.2. *Random hot-decking*

Here for each incomplete case, a donor was selected from the complete set randomly.

3.4.3.3. *Simple response pattern imputation (SRPI)*

A matching set of variables represented by M is determined by analyzing the data set. For each incomplete case, all cases with complete values with respect to the missing values in the incomplete case were considered donors. The similarity was measured using the Euclidean distance [7]. The complete case with smallest value would be the donor.

3.4.3.4. *k-nearest neighbor method*

The missing values are replaced by the values of a "Nearest Neighbor" which is similar to the incomplete case. The method works by finding "k" most similar/nearest complete cases to the incomplete case where the similarity is measured by a distance. The value of "k" was set to 2. Two most similar/nearest cases were selected to impute the values in the incomplete case. All qualitative variables were dummy coded. Seven different distance metrics were used to form seven different complete data sets. The method was implemented in the following way [1]:

The data set was divided into two sets, the cases with missing values (Incomplete Set) and the complete cases (Complete Set). Let x_i be the vector of all the variables measured for the ith case in the incomplete set and x_{ij} would be the value for the jth variable measured on ith case. y_k be the vector for all the variables measured for the kth case in the complete set, and y_{kj} be the value for the jth variable measured on kth case.

The following distance parameters were calculated to different complete data sets:

(a) Euclidean distance

It measures the distance between two points represented by a n by p matrix. In our case n is the number of variables and p is the number of cases in our data set.

$$\text{Euclidean}_{ki}(d) = \sqrt{\sum_{j=1}^{n}(y_{kj} - x_{ij})^2} \tag{3}$$

(b) Manhattan distance

It is the sum of the absolute differences between two points.

$$\text{Manhattan}_{ki}(d) = \sum_{j=1}^{n}|y_{kj} - x_{ij}| \tag{4}$$

(c) Mahalanobis distance

Maholanobis distance is given by:

$$\text{Mahalanobis}_{ki}(d^2) = (y_k - x_i)C^{-1}(y_k - x_i)' \tag{5}$$

where i is the missing case, k is the complete case and C is the covariance matrix.

(d) Correlation distance

The correlation coefficient (r) is a measure of linear relationships between two samples/vectors. "r" is given by

$$r = \frac{n\sum_{j=1}^{n} y_{kj}x_{ij} - \left(\sum_{j=1}^{n} y_{kj}\right)\left(\sum_{j=1}^{n} x_{ij}\right)}{\sqrt{\left[n\sum_{j=1}^{n} y_{kj}^2 - \left(\sum_{j=1}^{n} y_{kj}\right)^2\right]\left[n\sum_{j=1}^{n} x_{ij}^2 - \left(\sum_{j=1}^{n} x_{ij}\right)^2\right]}} \tag{6}$$

Similarity (S) between two vectors, $(S) = (r+1)/2$.

(e) Cosine distance

The cosine similarity function between two vectors CS_{ki} [24] (Ochini Coefficient) measures the cosine of the angle in between them. The similarity is measured by cosine of the angle. CS_{ki} is given by

$$\mathrm{CS}_{ki} = \frac{\sum_{j=1}^{n} y_{kj} x_{ij}}{\sqrt{\sum_{j=1}^{n} y_{kj}^2 \sum_{j=1}^{n} x_{ij}^2}} \tag{7}$$

(f) Squared chord distance

The distance metric is given by

$$\mathrm{SCD}_{ki} = \sum_{j=1}^{n} \left(\sqrt{y_{kj}} - \sqrt{x_{ij}}\right)^2 \tag{8}$$

For the last distance metric, it may be necessary to have non-negative values in the data set. It is noted that the values be shifted to non-negative (or positive) values before calculating these distances.

(g) Combination method

We devised a combination of two distance measures for each incomplete case. One metric represented the categorical variables and the other represented the quantitative variables respectively. *Hamming distance* was calculated which included only the dummy coded categorical variables.

The Hamming distance between two sets of binary digits is the number of corresponding binary digit positions that differ given by

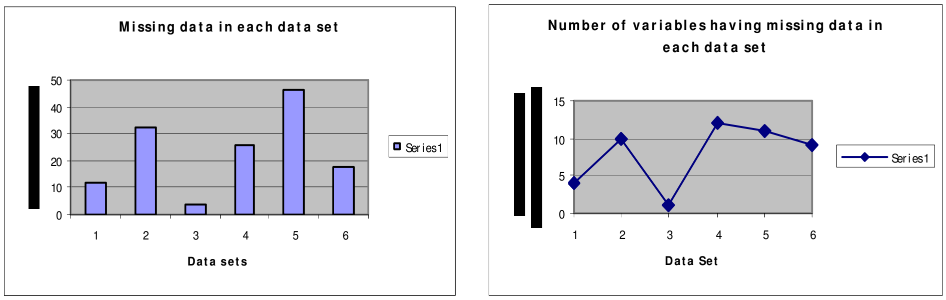
$$\mathrm{HD}_{ki} = \#(y_k \neq x_i) \tag{9}$$

The Cosine distance was computed for the quantitative variables. Both metrics were added and the cases with the first two smallest distances were selected as donors. All values were standardized using z-score for SRPI and k-NN methods.

3.4.3.5. *Maximum likelihood approach*

Maximum likelihood estimation begins with an expression known as a likelihood function. The likelihood of a sample is the probability of obtaining that particular sample of data given the chosen probability model. It contains the unknown parameters. Those values of the parameters that maximize the sample likelihood are known as the maximum likelihood estimates [31–35].

We used the Raw Maximum Likelihood Function (Full Information Maximum Likelihood). It uses all the available data to generate a vector of means and a covariance matrix among the variables that is superior to the ones produced by other methods. The FIML estimator maximizes the likelihood function which is



(a) (b)

Fig. 2. (a) Represents the percentage of missing data in each of the data sets and (b) represents the number of variables having missing data in each of the data sets.

the sum of m case wise likelihood functions. A likelihood function is calculated for each individual that measures the discrepancy between the observed data for the jth case and the current parameter estimates. The following function is maximized with the assumption that the data come from a multivariate normality distribution [3, 25]:

$$\log L_j = K_j - \frac{1}{2}\log|\Omega| - \frac{1}{2}(x_j - \mu_j)'\Omega_j^{-1}(x_j - \mu_j) \tag{10}$$

where x_j is the vector of the whole data for the case j,
μ_j is the vector of mean estimates for variables observed for case j,
K_j is a constant that depends on the number of complete values for case j,
the determinant and inverse of $\underline{\Omega}_j$ depend on variables that are observed for case j.

4. Dataset Description

We acquired six software project data sets in the past one year period from six different companies nationally and internationally. We obtained three small sized software project data sets, two medium sized and one large sized data set. Details about the characteristics of each of the data set are explained in Table 2.

4.1. *Classification scheme*

We have classified the software project data sets based on missing mechanisms and the characteristics unique to them. Using our classification scheme, each data set can be classified and by using this classification, appropriate imputation strategy can be selected. We classify software project data sets based on 4 parameters, namely, the size of the data set, the missing mechanism of the data, the percentage of missing data and finally the missing pattern of the data. The classification process proceeds in the same order. That is first a data set's size is determined. The attributes for size are small, medium and large. Here small indicates data set representing

Table 2. The real-time data sets used in the experimental analysis.

Data set	Size	Project type	Completion time (years)	Missing mechanism	% of missing data	Missing values	No. of variables	No. of cases	No. of categorial variables	No. of continuous variables	No. of variables having missing values	Values on dependent variable (Y)
D1	S	Medical	5	MAR	12	A	9	21	4	5	4	NM
D2	S	Customer service	4	MAR	32	M	12	29	3	9	10	NM
D3	S	Web focus	2	MCAR	4	U	8	17	4	4	1	NM
D4	M	Bank	6	MAR	26	A	22	42	10	12	12	NM
D5	M	Customer service	9	MAR	46	A	15	67	6	9	11	M
D6	L	Network management	10	NI	18	A	23	103	8	15	9	NM

Size (S – small, M – Medium, L – Large)
Missing pattern (U – Univariate, M – Monotonous, A – Arbitrary)
% of missing data is rounded values
Values on dependent variable (Y – Effort expended for completing the project in person hours) (M – Missing, NM – Not missing)

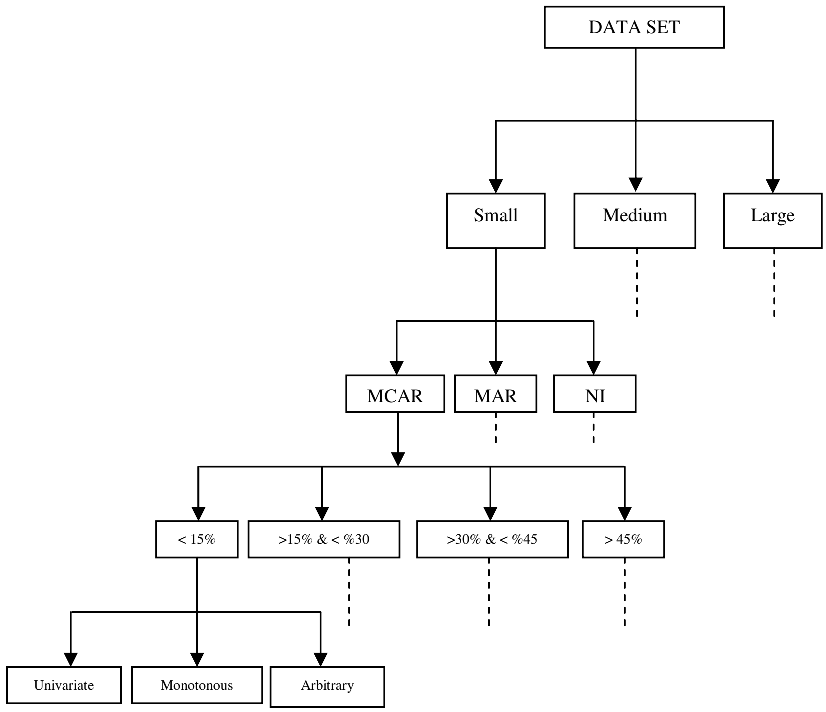


Fig. 3. Classification procedure of a dataset.

less than 30 cases, medium represents greater than 30 but less than 100 cases and large indicates greater than or equal to 100 cases. Each data set is classified as a small/medium/large sized data set. Software project data sets are generally small or medium sized. The next step involves determining the mechanism in which the data are missing within the data set. The data set is then sub-classified based on whether the missing mechanism is Ignorable or Non-Ignorable. The missingness mechanism is often assumed to be Ignorable but some times it may be the other way too. Next, the percentage of missing data is determined. The data set is selected into one of the 4 subclasses here. That is $< 15\%$ of missing data, $> 15\%$ and $< 30\%$ of missing data, $> 30\%$ and $< 45\%$ of missing data and $> 45\%$ of missing data. On general consensus, data sets having missing data greater than 45% are not imputed due to various reasons [1, 3]. Finally, they are sub classified based upon the pattern of missing data i.e., univariate, monotonous or arbitrary.

The missing pattern is more often arbitrary in software project data sets. The classification process is depicted by Fig. 3.

5. Experimental Results

We used the following measures of goodness of fit and accuracy.

Adjusted R-squared (Regression Correlation Coefficient)

It is the square of the correlation coefficient between the dependent variable and the estimate of it produced by the regressors. It is defined as the ratio of explained (regression) variation of the dependent variable to total variation. It has a value between 0 and 1 and if the value is close to 0, it means a poor model. When there are a large number of independent variables, R^2 may become large, simply because some variables chance variations "explain" small parts of the variance of the dependent variable. It is therefore essential to adjust the value of R^2 as the number of independent variables increases. In the case of a few independent variables, R^2 and adjusted R^2 will be close. In the case of a large number of independent variables, adjusted R^2 is noticeably lower. R-squared was used to assess the overall goodness of fit. Though it may not be the ideal way to compare models, it still is useful to confirm that the models converge.

Mean Magnitude of Relative Error

MMRE is the de facto standard in software engineering for assessing prediction systems. It has a clear appeal as an evaluative criterion and can be easily interpreted. The impact of the imputation methods are then determined using Mean Magnitude of Relative Error. These statistics are calculated from the model built using the predicted data sets. The Magnitude of Relative Error is defined as

$$\mathrm{MRE}_i = (|\text{Actual Effort}_i \text{ - Estimated Effort}_i|)/\text{Actual Effort}_i$$

where "i" is the observed case.

This is estimated for all predicted observations and the mean of all these values gives us Mean Magnitude of Relative Error (MMRE).

Prediction at Level 1 (Pred(l))

Pred(l) = p/n where p is the number of cases having relative error less than or equal to l and n is the total number of cases. It is a complementary criterion to MMRE.

Tables 3–8 represent the performance statistics of the methods for each of the six datasets.

6. Performance Evaluation

We applied four missing data techniques to each of the six different data sets accumulated. The methods include Listwise Deletion (LD), Mean Imputation (MI), ten variants of Hot-Deck (HD) Imputation and Full Information Maximum Likelihood Approach (FIML). The results show that we found a reasonable improvement in the prediction accuracies. The indicators measured express accuracy as well as goodness of fit. The results of our experiment have shown that there was significant improvement in accuracy as well as fitting. The adjusted-R squared is a measure of goodness of fit and MMRE indicates accuracy. We now elaborate on the impact

Table 3

Data set 1	Adj R^2	MMRE	Pred (25%)
LD	0.32	165%	21%
MI	0.41	109%	19%
Sequential hot-deck	0.43	74%	37%
Random hot-deck	0.46	89%	23%
SRPI	0.69	55%	46%
*Euclidean	0.72	61%	52%
*Manhattan	0.84	63%	41%
*Maholanobis	0.59	67%	39%
*Correlation	0.64	56%	47%
*Cosine	0.56	59%	54%
*Squared-chord	0.71	50%	38%
*Combination method	0.79	41%	59%
FIML	0.8	42%	61%

Table 4

Data set 2	Adj R^2	MMRE	Pred (25%)
LD	0.4	94%	18%
MI	0.21	102%	9%
Sequential hot-deck	0.11	114%	6%
Random hot-deck	0.61	63%	33%
SRPI	0.6	57%	34%
*Euclidean	0.69	61%	41%
*Manhattan	0.71	53%	44%
*Maholanobis	0.68	50%	49%
*Correlation	0.7	52%	47%
*Cosine	0.61	53%	40%
*Squared-chord	0.66	67%	39%
*Combination method	0.7	44%	48%
FIML	0.72	46%	40%

Table 5

Data set 3	Adj R^2	MMRE	Pred (25%)
LD	0.79	36%	58%
MI	0.43	71%	15%
Sequential hot-deck	0.5	55%	21%
Random hot-deck	0.78	35%	52%
SRPI	0.88	31%	61%
*Euclidean	0.9	30%	64%
*Manhattan	0.89	32%	65%
*Maholanobis	0.8	37%	60%
*Correlation	0.91	28%	71%
*Cosine	0.78	39%	65%
*Squared-Chord	0.88	32%	61%
*Combination Method	0.9	29%	74%
FIML	0.87	32%	70%

Table 6

Data set 4	Adj R^2	MMRE	Pred (25%)
LD	0.25	89%	16%
MI	0.56	57%	24%
Sequential hot-deck	0.51	64%	31%
Random hot-deck	0.41	70%	19%
SRPI	0.52	60%	40%
*Euclidean	0.61	50%	36%
*Manhattan	0.68	34%	35%
*Maholanobis	0.58	62%	37%
*Correlation	0.55	69%	42%
*Cosine	0.5	63%	41%
*Squared-chord	0.49	73%	56%
*Combination method	0.7	32%	68%
FIML	0.6	36%	66%

of all the methods with respect to each data set taking into account their different inherent characteristics.

6.1. *Data Set* 1 (*DS*1)

Based on our classification scheme, DS1 is a small sized data set having an ignorable missing mechanism (MAR), a missing data percentage < 15% and has data missing arbitrarily. We can observe from Table 3 that LD (Adj R^2 = 0.32 and MMRE = 165%) was inferior to all other methods. The reason would be the MAR mechanism. Moreover, only 7 cases were utilized by the method. Even though the total percentage of missing data was less than 15%, the total data loss was approximately 56% as the data set had only 7 complete cases. The Adj R^2 = 0.32

Table 7

Data set 5	Adj R^2	MMRE	Pred (25%)
LD	0.1	1125%	4%
MI	0.29	486%	9%
Sequential hot-deck	0.16	986%	6%
Random hot-deck	0.35	211%	12%
SRPI	0.36	105%	16%
*Euclidean	0.4	89%	19%
*Manhattan	0.44	90%	21%
*Maholanobis	0.41	80%	20%
*Correlation	0.32	96%	18%
*Cosine	0.36	98%	23%
*Squared-chord	0.38	103%	14%
*Combination method	0.4	85%	22%
FIML	0.52	55%	46%

Table 8

Data set 6	Adj R^2	MMRE	Pred (25%)
LD	0.21	218%	6%
MI	0.35	109%	11%
Sequential hot-deck	0.4	87%	15%
Random hot-deck	0.41	84%	14%
SRPI	0.5	68%	13%
*Euclidean	0.52	63%	21%
*Manhattan	0.58	70%	23%
*Maholanobis	0.54	66%	24%
*Correlation	0.52	60%	29%
*Cosine	0.5	64%	31%
*Squared-chord	0.58	65%	24%
*Combination method	0.59	57%	30%
FIML	0.67	48%	56%

shows us the poor model built and the MMRE = 165% shows the bias in the estimates. The performance of LD deteriorates as the number of cases with missing values increase. This converse of the above statement is not necessarily true as other factors could influence its performance. MI performed slightly better than LD but again the MAR condition accounted for its poor performance. Among the HD variants Sequential HD and Random HD performed inferior to the others (though they performed better than LD and MI). SRPI had a good Adj $R^2 = 0.69$ value and a better accuracy (MMRE = 55%). Within the k-NN HD variants, excluding Manhattan Distance Metric (Adj $R^2 = 0.84$ and MMRE = 63%) and Combination Method (Adj $R^2 = 0.79$ and MMRE = 41%), all of them performed more or less the same but with a better Adj R^2 and MMRE values than previous methods. Though the goodness of fit of the Manhattan Distance Metric is better than that of the Combination Method, the MMRE indicator shows that the Combination Method was much more accurate. The overall performance of the HD variants was better under MAR conditions. Finally, FIML (Adj $R^2 = 0.8$ and MMRE = 42%) performed well showing flexibility with small sized data sets.

6.2. *Data Set* 2 (*DS*2)

DS2 is a small sized data set having an ignorable missing mechanism (MAR), a missing data percentage $>30\%$ and $<45\%$ and has data missing monotonously. We can observe from Table 4 LD (Adj $R^2 = 0.4$ and MMRE = 94%) performed better than both MI and Sequential HD. The reason is due to the pattern in which the data are missing. Both MI (MMRE = 102%) and Sequential HD (MMRE = 114%) showed high biases for the same reason. Because of the missing pattern, the same value was imputed in all the missing values for each variable using MI, thus distorting the distribution and underestimating variance. In the case of Sequential

HD, the same donor was repeatedly used. Also the percentage of missing data could have played a role for the poor performance of MI. Random HD (Adj R^2 = 0.61 and MMRE = 63%) performed better in this case. SRPI (Adj R^2 = 0.6 and MMRE = 57%) performed well in spite of the monotonous pattern. Among the k-NN HD variants, Manhattan Distance Metric (Adj R^2 = 0.71 and MMRE = 53%) and Combination Method (Adj R^2 = 0.7 and MMRE = 44%) slightly outperformed others. FIML (Adj R^2 = 0.72 and MMRE = 46%) had the best fit and accuracy for DS2.

6.3. *Data Set* 3 (*DS*3)

DS1 is a small sized data set having an ignorable missing mechanism (MCAR), a missing data percentage < 15% and has univariate missing data pattern. From Table 5, we can see that LD (Adj R^2 = 0.79 and MMRE = 36%) performed very well under MCAR conditions. Under MCAR conditions, almost all the other methods performed exceedingly well except for MI (Adj R^2 = 0.43 and MMRE = 71%) and Sequential HD (Adj R^2 = 0.5 and MMRE = 55%). Again, the pattern of the missing values accounted for their underperformance. Euclidean Distance Metric (Adj R^2 = 0.9 and MMRE = 30%), Correlation Distance Metric (Adj R^2 = 0.91 and MMRE = 28%) and the Combination Method (Adj R^2 = 0.9 and MMRE = 29%) performed slightly better than the remaining methods giving the best fits and accuracies. FIML (Adj R^2 = 0.87 and MMRE = 32%) too did well.

6.4. *Data Set* 4 (*DS*4)

DS4 is a medium sized data set having an ignorable missing mechanism (MAR), a missing data percentage > 15% and < 30% and has data missing arbitrarily. From Table 6, we can notice LD (Adj R^2 = 0.25 and MMRE = 89%) performed badly because only 9 cases were complete out of the total 42 cases in DS4. A total data loss of 79% was accounted for while using LD. MI (Adj R^2 = 0.56 and MMRE = 57%), Sequential HD (Adj R^2 = 0.51 and MMRE = 64%) were almost similar. Though the missing data percentage was high, MI and Sequential HD performed relatively well. SRPI and k-NN methods performed better than the LD, MI, Sequential HD or Random HD. Of these, Manhattan Distance Metric (Adj R^2 = 0.68 and MMRE = 34%) and Combination Method (Adj R^2 = 0.7 and MMRE = 32%) had the best fits and accuracies. Both of them performed better than FIML (Adj R^2 = 0.6 and MMRE = 36%). Overall, most of the HD variants performed similar to or better than FIML.

6.5. *Data Set* 5 (*DS*5)

DS5 is a medium sized data set having an ignorable missing mechanism (MAR), a missing data percentage > 45% and has data missing arbitrarily. Looking at Table 7 we can see that all the methods other than FIML (Adj R^2 = 0.52 and MMRE =

55%) performed badly. No other method gave a reasonable accuracy. None of them had a reasonable goodness of fit. LD (Adj $R^2 = 0.1$ and MMRE = 1125%) performed the worst of all. The HD variants performed more or less the same. The reason for such a performance by all the methods is because of the high percentage of missing data. With a huge amount of data missing, none of the methods could lessen bias.

6.6. *Data Set 6* (*DS*6)

DS6 is a large sized data set having a non-ignorable missing mechanism (NI), a missing data percentage $> 15\%$ and $< 30\%$ and has data missing arbitrarily. We can notice from Table 8 that neither LD (Adj $R^2 = 0.21$ and MMRE = 218%) nor MI (Adj $R^2 = 0.35$ and MMRE = 109%) did well under NI conditions. Sequential HD (Adj $R^2 = 0.4$ and MMRE = 87%) and Random HD (Adj $R^2 = 0.41$ and MMRE = 84%) were slightly better than the previous two but both of them underperformed as well. SRPI and all k-NN methods had Adj R^2 values around 0.5 to 0.6 and MMRE values between 55–70%. Manhattan Distance Metric (Adj $R^2 = 0.58$ and MMRE = 70%), Squared-Chord Distance Metric (Adj $R^2 = 0.58$ and MMRE = 65%) and Combination Distance Metric (Adj $R^2 = 0.59$ and MMRE = 57%) had better accuracies among them. It was FIML (Adj $R^2 = 0.67$ and MMRE = 48%) that was most resilient to bias under non-ignorable missing mechanism conditions. FIML had the least bias and best estimates of all the methods under NI conditions.

Figure 4 shows the performances of each of the methods on the six data sets. Each graph corresponds to each imputation method. Every graph shows the Mean Magnitude of Relative Error of that method with respect to all the datasets. Figure 5 depicts the goodness of fit characteristics for each data set on all the methods implemented on it. One can compare the accuracy of the model built when each method was implemented on the data set.

7. Comparison with Previous Works and Recommendations

We agree with Kevin Strike *et al.* [1] and Myrtveit *et al.* [7] that LD be used only when the missing mechanism is MCAR. We also agree in saying that overall HD methods have lesser bias when compared to LD. But we disagree with Kevin Strike *et al.* [1] in not finding the difference among the HD variants. In our case, Manhattan Distance Metric and Combination Method outperformed the rest. For low percentages of missing data Roth [2] recommended HD methods and our results strongly concur the same. Our results were opposed to that stated by Emam *et al.* [20] that LD was a reasonable choice at most times. We also state that LDs performance decreases as the percentage of missing data increases and that LD has to be used only when the missing percentage is small. Song *et al.* [22] also come up with a hot-deck variant which yielded similar results.

Kaiser [12] said the performance of HD variants decreases with an increase in missing values and our results agree with this finding. All MDTs deteriorate as the percentage of missingness grows and it is almost inappropriate to apply any of them

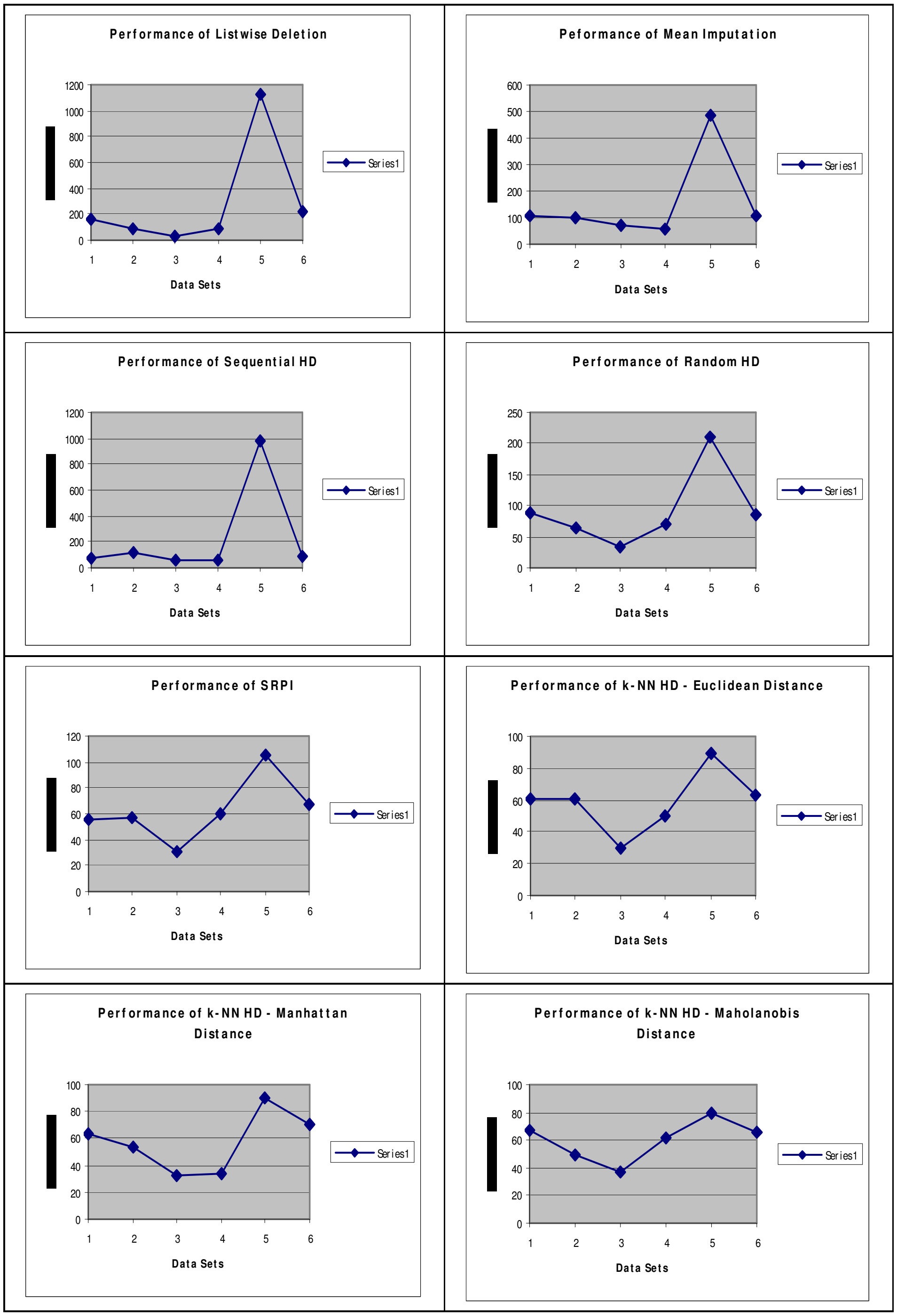


Fig. 4. Performances of each of the Imputation Methods wrt the 6 data sets.

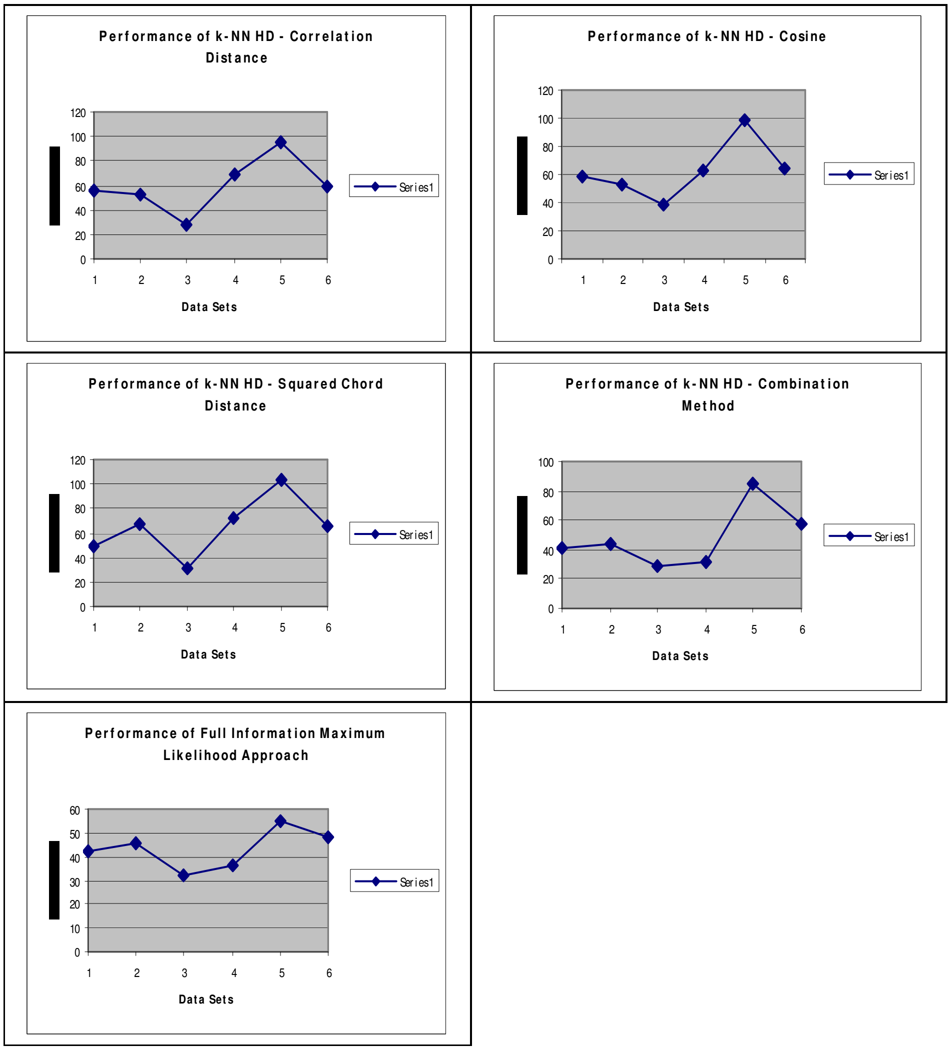


Fig. 4. (*Cont'd*)

when the missing percentage is greater than 50. Raymond *et al.* [16] found that when data are missing at random, MI performed better than LD. In our results, we found in two instances that LD outperformed MI. The missing mechanism and the missing pattern together would attribute to the performance of LD over MI. When compared to MI, HD variants were less susceptible to univariate and monotonous missing patterns. Lee *et al.* [19] said LD was preferable over MI when using polychoric correlation but we assumed a regression model. The studies by Cox *et al.* [11] and Ford [17] also state that HD methods reduce bias when compared to LD. Kromey *et al.* [14] stated that sometimes LD was more reasonable than MI, Pairwise Deletion, Simple Regression Imputation and Multiple Imputation and we observed this too.

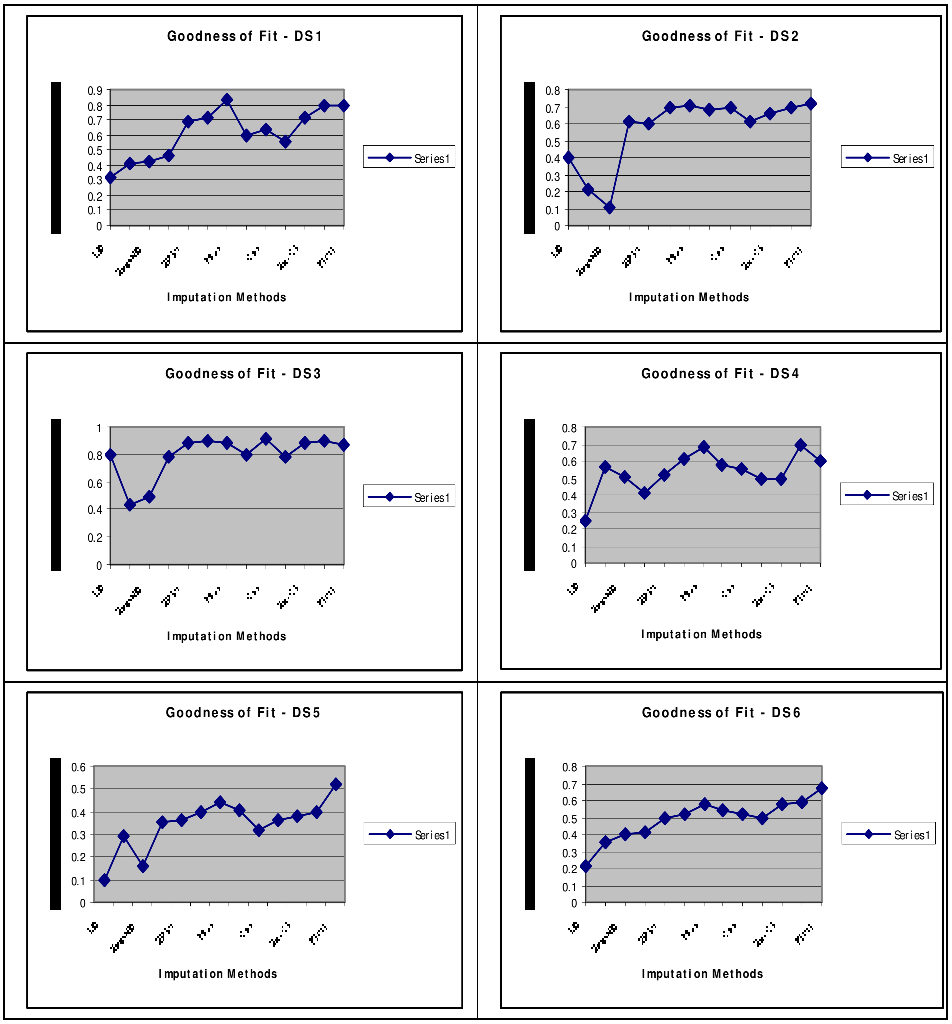


Fig. 5. Goodness of Fit Measures for each of the data sets using the Imputation Methods.

Brown *et al.* [15] found SRPI to have lesser bias than LD, PD, MI and HD imputation. Our results also show that SRPI performed better than LD, MI, Random and Sequential HD methods. But other HD variants (k-NN methods) did perform better than SRPI. Roth [22] and Myrtveit *et al.* [7] advocated the use of maximum likelihood estimation when the data are missing at random and our results denote the same particularly when the missing mechanism was NI. Though FIML showed good overall performance, we suggest not using it when the data sets are small. Browne *et al.* [18] found FIML to be superior to LD, PD and MI and our results assert the same. We now list our recommendations based on our experimental results:

- After reviewing the results, we can say that all the methods performed better than LD. Only in 2 instances did LD perform better than MI and Sequential HD. In both these instances MI and Sequential HD did not perform well because of the pattern in which data were missing. Also, whenever the data set had few complete cases, LD underperformed (DS1, DS4, and DS5). When missing data are not confined to a small percentage of cases, LD performed badly. The performance of LD deteriorates as the number of cases with missing values increase. Also, LD underperformed when the missing mechanisms were MAR (DS2) and NI (DS6). LD performed only when the missing mechanism is MCAR. We agree with Kevin Strike *et al.* [1] that
- MI and Sequential HD did not perform well when the missing patterns were monotonous (DS2) and univariate (DS3). The reason is obvious. The same value/donor was used to impute the missing values in both cases thus distorting the underlying distribution. The pattern in which the data are missing play an important role while using these methods. Even when the pattern is arbitrary, these methods may not perform well if less number of variables contributes towards large number of missing values. Moreover, we found that MI and Sequential HD may not be least biased under MAR or NI conditions (DS1 and DS6). Random HD performed slightly better than Sequential HD in most cases but did not yield reasonable fits. We suggest using MI or Sequential HD only under MCAR conditions and when the percentage of missing data is less than 5%.
- SRPI along with other k-NN HD methods performed more or less the same. Overall, the Manhattan Distance Metric and the Combination Method yielded the best results among all of them. Both of them outperformed FIML in a few instances (DS3 and DS4). It may be due to the reason HD variants work well with smaller data sets. All the methods performed well under MCAR and MAR conditions but yielded biased results under NI conditions (DS6). Their performance did not rely on the size of the data set or the missing pattern. We recommend using HD variants (particularly Manhattan and Combination Methods) when the data sets are relatively small (< 50 cases) and the missing mechanism is not NI.
- FIML performed similar to Manhattan Distance Metric and the Combination Method except the one instance under NI conditions (DS6). FIML gave least biased estimates under NI conditions. FIML works well for larger data sets and even under NI conditions. Though it may be computationally demanding, we recommend using FIML under NI conditions in particular.
- None of the methods excluding FIML performed even reasonably well when a high percentage of data was missing (DS5). FIML may perform reasonably in such situations but we are not thoroughly convinced. In our case, it did reasonably well though. In general, the performance of all techniques degrades as the missing percentage increases. We recommend not imputing when the data set has missing percentage above 50 (unless otherwise we know for sure the missing mechanism is MCAR). Imputation should be used only when necessary but not to make the data set look good by making it complete.

8. Conclusions

In this paper, we applied four missing data techniques (LD, MI, ten variants of HD and FIML) to six different real-time data sets and evaluated the performance of each of the techniques. We studied the effects of the characteristics of the data set such as size, percentage of data missing, missing data pattern, and missing mechanisms would have on the choice of imputation. Our goal was to find out whether imputation strategies could improve the prediction accuracies and decrease bias.

Our experimental results showed we succeeded in decreasing bias. The HD variants and FIML outperformed the traditional approaches. We suggest that researchers not use LD when the data are not MCAR and when missing values are present in a major number of cases but we recommend using MI only when none of the variables singly contribute to a major number of missing values. Also caution should be taken when using MI if the data are missing at random. On the other hand, HD variants performed well in our analysis. We recommend using variants of HD under MAR assumption. We also suggest using FIML under NI conditions but more testing is needed to confirm its performance. One limitation of our study though is we implemented only four imputation methods. There exist other methods which need to be tested in order to evaluate their performances.

Based on our results, we are sure that we have made a point about the validity of the inferences drawn using traditional approaches. There are only a few references in the literature related to such exploration [1, 7, 20, 26]. Most of them suggest techniques that preserve the integrity of a data set by using different statistical approaches to fill in probable values. Our results are encouraging and we recommend researchers to carry further research using other variants of HD methods, Multiple Imputation Methods and Likelihood approaches on larger number of data sets. Furthermore, we encourage analysts to devise hybrid imputation algorithms for better results.

References

1. K. Strike, K. E. Emam, and N. Madhavji, Software Cost Estimation with Incomplete Data, ERB-1071 NRC, http://wwwsel.iit.nrc.ca/∼elemam/documents/1071.pdf, also to appear in *IEEE Trans. Software Eng.*
2. P. Roth, Missing data: A conceptual review for applied psychologists, *Personnel Psychology* **47** (1994) 537–560.
3. R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data* (John Wiley, New York, 2002).
4. M. Berry and M. F. Vanderbroek, A targeted assessment of the software measurement process, in *Proc. IEEE Seventh Int. Software Metrics Symp.*, 2001, pp. 222–235.
5. B. W. Boehm, *Software Engineering Economics* (Prentice Hall, 1981).
6. T. DeMarco, *Controlling Software Projects: Management, Measurement, and Estimates* (Prentice-Hall, New York, 1982).
7. I. Myrtveit, E. Stensrud and U. H. Olsson, Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods, *IEEE*

Trans. on Software Engineering **27**(11) (2001) 999–1013.

8. M. Cartwright and M. J. Shepperd, Predicting with sparse data, *IEEE Trans. on Software Engineering* **27**(11) (2001) 1014–1022.
9. J. L. Schafer and J. W. Graham, Missing data: Our view of the state of the art, *Psychological Methods* **7**(2) (2002) 147–177.
10. D. B. Rubin, Inference and missing data, *Biometrika* **63** (1976) 581–592.
11. B. Cox and R. Folsom, An empirical investigation of alternate item nonresponse adjustments, in *Proc. Section on Survey Research Methods*, 1978, pp. 219–223.
12. J. Kaiser, Thel effectiveness of hot-deck procedures in small samples, in *Proc. Ann. Meeting of the Am. Statistical Assoc.*, 1983.
13. D. J. Mundform and A. Whitcomb, Imputing missing values: The effect on the accuracy of classification, *Multiple Linear Regression Viewpoints* **25** (1998) 13–19.
14. J. Kromrey and C. Hines, Nonrandomly missing data in multiple regression: An empirical comparison of common missing-data treatments, *Educational and Psychological Measurement* **54**(3) (1994) 573–593.
15. R. L. Brown, Efficacy of the indirect approach for estimating structural equation models with missing data: A comparison of five methods, *Structural Equation Modeling* **1**(4) (1994) 287–316.
16. M. Raymond and D. Roberts, A comparison of methods for treating incomplete data in selection research, *Education and Psychological Measurement* **47** (1987) 13–26.
17. B. Ford, Missing Data Procedures: A Comparative Study, in *Proc. Social Statistics Section*, 1976, pp. 324–329.
18. C. H. Browne, Asymptotic comparison of missing data procedures for estimating factor loadings, *Psychometrika* **48**(2) (1983) 269–291.
19. S. Y. Lee and Y.-M. Chiu, Analysis of multivariate polychoric correlation models with incomplete data, *British J. Math. and Statistical Psychology*, **43** (1990) 145–154.
20. K. E. Emam and A. Birk, Validating the ISO/IEC 15504 measure of software requirements analysis process capability, *IEEE Trans. Software Eng.* **26**(6) (2000) 541–566.
21. Q. Song and M. Shepperd, A Short Note on Using Multiple Imputation Techniques for Very Small Data Sets, Technical Report, Empirical Software Engineering Research Group, Bournemouth University, UK, April 2003.
22. Qinbao Song, Martin Shepperd, Michelle Cartwright and Bheki Twala, A New Imputation Method for Small Software Project Data Sets, Technical Report, Empirical Software Engineering Research Group, Bournemouth University, UK, May 2004.
23. J. L. Schafer, *Analysis of Incomplete Multivariate Data*, Chapman and Hall, Boca Raton, 1997.
24. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer* (Addison-Wesley, 1989).
25. P. D. Allison, *Missing Data, Quantitative Applications in the Social Sciences*, Vol. 136 (SAGE Publications, 2002).
26. M. Shepperd and M. Cartwright, Dealing with Missing Software Project Data, November 2002, Technical Report, Empirical Software Engineering Research Group, Bournemouth University, UK.
27. M. Colledge, J. Johnson, R. Pare, and I. Sande, Large scale imputation of survey data, in *Proc. Section on Survey Research Methods*, 1978, pp. 431–436.
28. O. Troyanskaya, M. Cantor, G. Sherlock *et al.*, Missing value estimation methods for DNA microarrays, *Bioinformatics* **17** (2001) 520–525.

29. B. Ford, An overview of hot-deck procedures, incomplete data in sample surveys, theory and bibliographies, Vol. 2, W. Madow, I. Olkin and D. Rubin (eds.) (Academic Press, 1983).
30. I. Sande, Hot-Deck Imputation Procedures, in *Proc. Symp. Incomplete Data in Sample Surveys*, Vol. 3, eds. W. Madow and I. Olkin, 1983.
31. T. W. Anderson, Maximum likelihood estimates for multivariate normal distributions when some observations are missing, *J. Am. Statistical Assoc.* **52** (1957) 200–203.
32. J. Anderson and D. W. Gerbing, The effects of sampling error on convergence, improper solutions, and goodness-of-fit indices for maximum likelihood confirmatory factor analysis, *Psychometrika* **49** (1984) 155–173.
33. M. C. Neal, *Mx: Statistical Modeling*, 2nd edn., 1994.
34. J. L. Arbuckle, Full information estimation in the presence of incomplete data, *Advanced Structural Equation Modeling*, eds. G. A. Marcoulides and R. E. Schumacker(Lawrence Erlbaum, Mahwah, NJ, 1996), pp. 243–277.
35. C. K. Enders, A primer on maximum likelihood algorithms available for use with missing data, *Structural Equation Modeling* **8** (2001) 128–141.