An Empirical Comparison of Techniques for Handling Incomplete Data Using Decision Trees

BHEKISIPHO TWALA
Modelling and Digital Intelligence, CSIR, P O Box 395, Pretoria 0001, South Africa

---

OBJECTIVE: Increasing the awareness of how incomplete data affects learning and classification accuracy has led to increasing numbers of missing data techniques. This paper investigates the robustness and accuracy of seven popular techniques for tolerating incomplete training and test data for different patters of missing data; different proportions and mechanisms of missing data on resulting tree-based models.

METHOD: The seven missing data techniques were compared by artificially simulating different proportions, patterns, and mechanisms of missing data using twenty one complete (i.e. with no missing values) datasets obtained from the UCI repository of machine learning databases [Blake and Merz, 1998]. A 4-way repeated measures design was employed to analyze the data.

RESULTS: The simulation results suggest important differences. All methods have their strengths and weaknesses. However, listwise deletion is substantially inferior to the other six techniques while multiple imputation, which utilizes the Expectation Maximization algorithm, represents a superior approach to handling incomplete data. Decision tree single imputation and surrogate variables splitting are more severely impacted by missing values distributed among all attributes compared to when they are only on a single attribute. Otherwise, the imputation--versus model-based imputation procedures gave reasonably good results although some discrepancies remained.

CONCLUSIONS: Different techniques for addressing missing values when using decision trees can give substantially diverse results, and must be carefully considered to protect against biases and spurious findings. Multiple imputation should always be used, especially if the data contain many missing values. If few values are missing, any of the missing data techniques might be considered. The choice of technique should be guided by the proportion, pattern and mechanisms of missing data, especially the latter two. However, the use of older techniques like listwise deletion and mean or mode single imputation is no longer justifiable given the accessibility and ease of use of more advanced techniques such as multiple imputation and supervised learning imputation.

Keywords: incomplete data, machine learning, decision trees, classification accuracy

---

1. INTRODUCTION

Machine learning (ML) algorithms have proven to be of great practical value in a variety of application domains. Unfortunately, such algorithms generally operate in environments that are fraught with imperfections. One form of imperfection is incompleteness of data. This is

currently an issue faced by machine learning and statistical pattern recognition researchers who use real-world databases. One primary concern of classifier learning is prediction accuracy. Handling incomplete data (data unavailable or unobserved for any number of reasons) is an important issue for classifier learning since incomplete data in either the training data or test (unknown) data may not only impact interpretations of the data or the models created from the data but may also affect the prediction accuracy of learned classifiers. Rates of less than 1% missing data are generally considered trivial, 1-5% manageable. However, 5-15% requires sophisticated methods to handle, and more than 15% may severely impact any kind of interpretation [Pyle, 1999].

Handling incomplete data is an important issue for classifier learning since incomplete data in either the training set or test set or in both sets affect the prediction accuracy of learned classifiers. The presence of missing values can also pose serious problems for researchers. In fact, inappropriate handling of missing data in the analysis may introduce bias and can result in misleading conclusions drawn from a research study, and can also limit generalizability of the research findings. For example, if you embark on formal statistical analysis, you may miss the important feature of your data. Also, formal statistical analysis assumes some characteristics about your data such as the number of instances, number of attributes, number of classes and so on. If these assumptions are wrong, the results of statistical analysis may be quite misleading and invalid. The seriousness of this problem depends in part on the proportion of missing data. Therefore, it is important to consider how much data is missing when assessing the impact of missing data.

The two most common tasks when dealing with missing values, thus, choosing a missing data technique, are to investigate the pattern and mechanism of missingness to get an idea of the process that could have generated the missing data, and to produce sound estimates of the parameters of interest, despite the fact that the data are incomplete. In other words, the potential impact missing data can have is dependent on the pattern and mechanism leading to the nonresponse. In addition, the choice of how to deal with missing data should also be based on the percentage of data that are missing and the size of the sample.

Robustness has twofold meaning in terms of dealing with missing values when using decision trees (DTs). The toleration of missing values in training data is one, and the toleration of missing data in test data is the other. Although the problem of incomplete data has been treated adequately in various real world datasets, there are rather few published works or empirical studies concerning the task of assessing learning and classification accuracy of missing data techniques (MDTs) using supervised ML algorithms such as DTs [Breiman *et al.*, 1984; Quinlan, 1993].

There are two common solutions to the problem of incomplete data that are currently applied by researchers. The first includes omitting the instances having missing values (i.e. listwise deletion), which does not only seriously reduce the sample sizes available for analysis but also ignores the mechanism causing the missingness. The problem with a smaller sample size is that it gives greater possibility of a non-significant result, i.e., the larger the sample the greater the statistical power of the test. The second solution imputes (or estimate) missing values from the existing data. The major weakness of single imputation methods is that they underestimate uncertainty and so yield invalid tests and confidence intervals, since the estimated values are derived from the ones actually present [Little and Rubin, 1987].

To the best of our knowledge, very little work has been published in the ML or data mining literature that looks at the effect of MDTs on predictive accuracy, taking into account different missing data patterns, missing data proportions and missing data mechanisms. In other research areas, missing data has usually been inadequately handled.

The following section briefly discusses missing data patterns and mechanisms that lead to the introduction of missing values in datasets. Section 3 presents details of seven MDTs that are used in this paper. Section 4 empirically evaluates the robustness and accuracy of the eight MDTs on twenty one machine learning domains. We close with a discussion and conclusions, and then directions for future research.

2. PATTERNS AND MECHANISMS OF MISSING DATA

The pattern simply defines which values in the data set are observed and which are missing. The three most common patterns of nonresponse in data are univariate, monotonic and arbitrary. When missing values are confined to a single variable we have a univariate pattern; monotonic pattern occurs if a subject, say, Y_j is missing then the other variables, say Y_{j+1},..., Y_p, are missing as well or when the data matrix can be divided into observed and missing parts with a "staircase" line dividing them; arbitrary patterns occur when any set of variables may be missing for any unit.

The law generating the missing values seems to be the most important task since it facilitates how the missing values could be estimated more efficiently. If data are missing completely at random (MCAR) or missing at random (MAR), we say that missingness is *ignorable*. For example, suppose that you are modelling software defects as a function of development time. If missingness is not related to the missing values of defect rate itself and also not related on the values of development time, such data is considered to be MCAR. For example, there may be no particular reason why some project managers told you their defect rates and others did not.

Furthermore, software defects may not be identified or detected due to a given specific development time. Such data are considered to be MAR. MAR essentially says that the cause of missing data (software defects) may be dependent on the observed data (development time) but must be independent of the missing value that would have been observed. It is a less restrictive model than MCAR, which says that the missing data cannot be dependent on either the observed or the missing data. MAR is also a more realistic assumption for data to meet, but not always tenable. The more relevant and related attributes one can include in statistical models, the more likely it is that the MAR assumption will be met. For data that is informatively missing (IM) or not missing at random (NMAR) then the mechanism is not only non-random and not predictable from the other variables in the dataset but cannot be ignored, i.e., we have *non ignorable* missingness [Little and Rubin, 1987; Schafer, 1997]. In contrast to the MAR condition outlined above, IM arise when the probability that defect rate is missing depends on the unobserved value of defect rate itself. For example, software project managers may be less likely to reveal projects with high defect rates. Since the pattern of IM data is not random, it is not amenable to common MDTs and there are no statistical means to alleviate the problem.

MCAR is the most restrictive of the three conditions and in practice it is usually difficult to meet this assumption. Generally you can test whether MCAR conditions can be met by comparing the distribution of the observed data between the respondents and non-respondents. In other words, data can provide evidence against MCAR. However, data cannot generally distinguish between MAR and IM without distributional assumptions, unless the mechanisms is well understood. For example, right censoring (or suspensions) is IM but is in some sense known. An item, or unit, which is removed from a reliability test prior to failure or a unit which is in the field and is still operating at the time the reliability of these units is to be determined is called a suspended item or right censored instance.

3. DECISION TREES AND MISSING DATA TECHNIQUES

DTs are a simple yet successful technique for supervised classification learning. A DT is a model of the data that encodes the distribution of the class label in terms of the predictor attributes; it is a directed, acyclic graph in a form of a tree. The root of the tree does not have any incoming edges. Every other node has exactly one incoming edge and zero or more outgoing edges. If a node n has no outgoing edges we call n a leaf node, otherwise we call n an internal node. Each leaf node is labelled with one class label; each internal node is labelled with one predictor attribute called the splitting attribute. Each edge e originating from an internal node n has a predicate q associated with it where q involves only the splitting attribute of n.

There are two ways to control the size of the tree. For a bottom-up pruning strategy, a very deep tree is constructed, and this tree is cut back to avoid overfitting the training data. For top down pruning, a stopping criterion is calculated during tree growth to inhibit further construction of parts of the tree when appropriate. In this paper we follow the bottom up strategy.

A DT can be used to predict the values of the target or class attribute based on the predictor attributes. To determine the predicted value of an unknown instance, you begin at the root node of the tree. Then decide whether to go into the left or right child node based on the value of the splitting attribute. You continue this process using the splitting attribute for successive child nodes until you reach a terminal or leaf node. The value of the target attribute shown in the leaf node is the predicted value of the target attribute.

DT induction has several advantages to other methods of classifier construction. One property that sets DTs apart from all other methods is their invariance to monotone transformations of the predictor variables. For example, replacing any subset of the predictor variables $\{x_j\}$ by (possible different) arbitrary strictly monotone functions of them $\{x_j \leftarrow m_j(x_j)\}$, gives rise to the same tree model. Thus, there is no issue of having to experiment with different possible transformations $m_j(x_j)$ for each individual predictor x_j to try to find the best ones. This invariance provides immunity to the presence of extreme values ("outliers") in the predictor variable space. In addition, DTs incorporate a pruning scheme that partially addresses the outlier (noise) removal problem.

Several methods have been proposed in the literature to treat missing data when using DTs. Missing values can cause problems at two points when using DTs; 1) when deciding on a splitting point (when growing the tree), and 2) when deciding into which daughter node each instance goes (when classifying an unknown instance). Methods for taking advantage of unlabelled classes can also be developed, although we do not deal with them in this paper, i.e., we are assuming that the class labels are not missing.

Specific DT techniques for handling incomplete data are now going to be discussed. These missing data techniques (MDTs), which we shall now call testing methods as they can handle incomplete test data, are some of the widely recognized and they are divided into three categories: ignoring and discarding data, imputation and machine learning.

3.1 Ignoring and Discarding Missing Data

Over the years, the most common approach is to simply omit any instances that have missing values, and to perform the statistical analysis on the remaining instances. This approach is

called complete-case analysis or listwise deletion (LD). One major drawback of LD is that it can drastically reduce the sample size since it can sacrifice a large amount of data leading to a severe lack of statistical power (Roth, 1994). In some cases, it can even lead to complete instance loss if many variables are involved. However, due to its simplicity and ease of use, LD is the default analysis in most statistical packages. LD is also based on the assumption that data are MCAR.

3.2 Imputation

Imputation methods involve replacing missing values with estimated ones based on information available in the dataset. Imputation methods can be divided into single and multiple imputation methods. In single imputation the missing value is replaced with only one imputed value while in multiple imputation, each missing value is replaced with a set of *M* plausible values. This step results in *M* complete datasets. An illustration of multiple imputation is given in Section 3.2.2. Most imputation procedures for missing data are single imputation. In the following section we briefly describe how each of the imputation techniques works.

3.2.1 Single Imputation Techniques

3.2.1.1 Decision Tree Imputation

The unordered attribute DTs approach, which can also be considered a supervised learning technique as it uses a DT algorithm to impute missing values, is another strategy that has been used for handling missing values in tree learning. This technique was suggested by Shapiro (1987) and followed up by Quinlan (1987). The method builds DTs to determine the missing values of each attribute, and then fills the missing values of each attribute by using its corresponding tree. Separate trees are built using a reduced training set for each attribute, i.e., restricting your analysis to only those instances that have known values. The original class is treated as another attribute, while the value of the attribute becomes the "class" to be determined. The attributes used to grow the respective trees are unordered. These trees are then used to determine the unknown values of that particular attribute.

In the classification phase (where the class attribute is not present) the tree uses, instead of the class attribute, all the attributes in the test set in alternating fashion. In other words, each of the attributes would become a "class" variable at one point in time. For an example, say, you have a dataset with four attributes (A_1, A_2, A_3, A_4) in the test set. Considering A_1 as a "class" attribute would yield $\boldsymbol{A}_1(A_2, A_3, A_4)$; considering A_2 as a "class" attribute would yield $\boldsymbol{A}_2(A_1, A_3, A_4)$; and

so on. Suppose that in the first case {i.e., $A_1(A_2, A_3, A_4)$} some of A_2 is missing. The DT is constructed using $A_2(A_1, A_3, A_4)$ and with cases that are non-missing on A_2. Notice that A_2 now becomes the class attribute, replacing A_1. The tree is then to estimate values of A_2 using (A_1, A_3, A_4) and on cases which are missing on A_2. The same procedure is followed for the other attributes. Classification and regression trees for continuous dependent variables (regression) and categorical predictor variables (classification) are built.

[Lobo and Numao, 1999; 2000] follows-up Quinlan's unordered DT single imputation (DTSI) approach but by first ordering the attributes using mutual information before growing the tree. As with Quinlan's method, only those attributes with known values and low mutual information with respect to class are included in the reduced training set. After constructing a DT for filling the missing values of an attribute, it makes sense to use the data with filled values in order to construct a DT for filling the missing values of other attributes. Ordering on the attribute's trees construction was empirically found to improve the accuracy of the DT learning algorithm while keeping the computational cost to a sustainable level [Lobo and Numao, 2000].

Note that, DTSI can also be considered a supervised learning technique as it defines the effect one set of instances (called inputs) has on another set of observations (called outputs). Two other supervised learning techniques are described in Section 3.3. Also, it is not clear what the probability generating the missingness is for both strategies. Thus, we shall assume that the data is MCAR.

3.2.1.2 Expectation Maximization

In brief, expectation maximization (EM) is an iterative procedure where a complete dataset is created by filling-in (imputing) one or more plausible values for the missing data by repeating the following steps: 1.) In the E-step, one reads in the data, one instance at a time. As each case is read in, one adds to the calculation of the sufficient statistics (sums, sums of squares, sums of cross products). If missing values are available for the instance, they contribute to these sums directly. If a variable is missing for the instance, then the best guess is used in place of the missing value. 2.) In the M-step, once all the sums have been collected, the covariance matrix can simply be calculated. This two step process continues until the change in covariance matrix from one iteration to the next becomes trivially small. Details of the EM algorithm for covariance matrices are given in [Dempster *et al.*, 1977; Little and Rubin, 1987]. EM requires that data are MAR. As mentioned earlier, the EM algorithm (and its simulation cased variants) could be utilised to impute only a single value for each missing value, which from now on we shall call EM single imputation (EMSI). The single imputations are drawn from the predictive

distribution of the missing data given the observed data and the EM estimates for the model parameters. A DT is then grown using the complete dataset. The tree obtained depends on the values imputed.

3.2.1.3 Mean or Mode Imputation

Mean or more single imputation (MMSI) is one of the most common and extremely simple method of imputation of missing values. In MMSI, whenever a value is missing for one instance on a particular attribute, the mean (for a continuous or numerical attribute) or modal value (for a nominal or categorical attribute), based on all non-missing instances, and is used in place of the missing value. Although this approach permits the inclusion of all instances in the final analysis, it leads to invalid results. Use of MMSI will lead to valid estimates of mean or modal values from the data only if the missing value are MCAR, but the estimates of the variance and covariance parameters (and hence correlations, regression coefficients, and other similar parameters) are invalid because this method underestimates the variability among missing values by replacing them with the corresponding mean or modal value. In fact, the failure to account for the uncertainty behind imputed data seems to be the general drawback for single imputation methods

3.2.2 Multiple Imputation

Multiple imputation (MI) is one of the most attractive methods for general purpose handling of missing data in multivariate analysis. Rubin (1987; 1996) described MI as a three-step process. First, the missing entries of the incomplete datasets are imputed (or filled in), not once, but *M* times (*M*=5 in Figure1). The imputed values are drawn for a distribution (that can be different for each missing entry. This step results in *M* complete datasets as shown in Table 1 whereby two pairs of attributes are modelled with A_1 having missing values and A_2 values complete or observed. Second, each of these *M* datasets can be analyzed using complete-data methods (analysis). This step results in *M* analyses. Finally, the results from the *M* complete datasets are combined, which also allows that the uncertainty regarding the imputation is taken into account (pooling or combining).

For example, replacing missing value with a set of five plausible values or imputations (as it was the case in our illustration in Figure 1 and Table 1 would result to building five DTs, and the predictions of the five trees would be averaged into a single tree, i.e., the average tree is obtained by MI. For example, using the artificial dataset in Table 1, the first DT will be grown using imputation 1 (IMP 1) data, the second DT grown using IMP 2 data, the third DT grown using IMP3 data, and so on. The results for each DT are then integrated (combined) into a final result. This approach is similar to the ensemble learning approach which is used to improve

predictive accuracy in machine learning. MI retains most of the advantages of single imputation and rectifies its major disadvantages as already discussed. There are various ways to generate imputations.

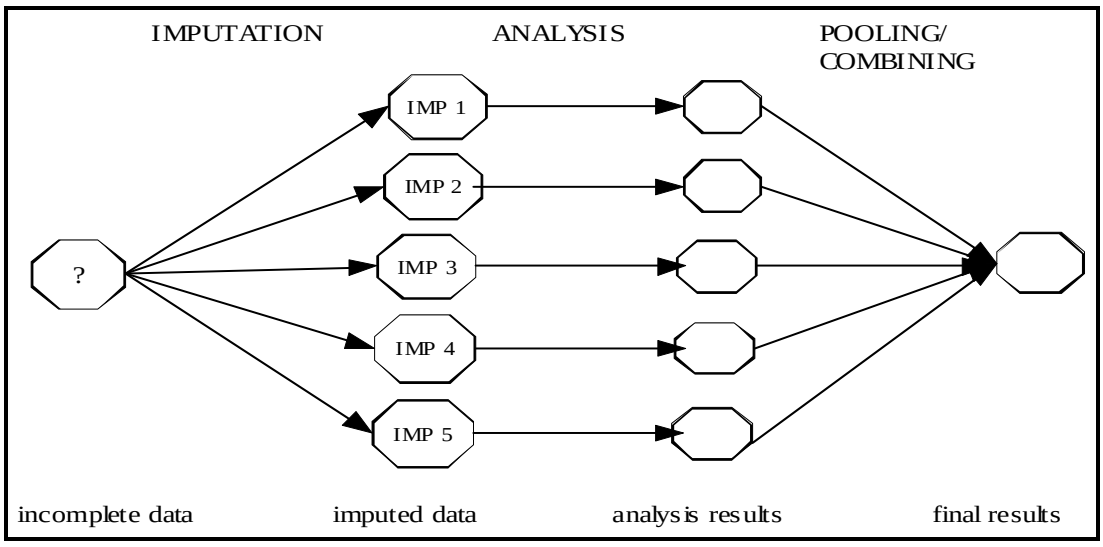

Figure 1: The three steps of multiple imputation

Table 1 Artificial dataset with missing values on attribute A_1

	Data		IMP 1		IMP 2		IMP 3		IMP 4		IMP 5	
Unit	A_1	A_2	A_1	A_2	A_1	A_2	A_1	A_2	A_1	A_2	A_1	A_2
1	1.1	3.4	1.1	3.4	1.1	3.4	1.1	3.4	1.1	3.4	1.1	3.4
2	1.5	3.9	1.5	3.9	1.5	3.9	1.5	3.9	1.5	3.9	1.5	3.9
3	0.8	2.6	0.8	2.6	0.8	2.6	0.8	2.6	0.8	2.6	0.8	2.6
4	**?**	0.8	**0.2**	0.8	**0.8**	0.8	**0.3**	0.8	**2.3**	0.8	**1.0**	0.8
5	**?**	2.0	**1.7**	2.0	**2.4**	2.0	**1.8**	2.0	**3.5**	2.0	**1.7**	2.0

Schafer (1997) has written a set of general purpose programs for MI of continuous multivariate data (NORM), multivariate categorical data (CAT), mixed categorical and continuous (MIX), and multivariate panel or clustered data (PNA). These programs were initially created as functions operating within the statistical languages S and SPLUS [SPLUS, 2003]. NORM includes an EM algorithm for maximum likelihood estimation of means, variance and covariances. NORM also adds regression-prediction variability by using a Bayesian procedure known as data augmentation [Tanner and Wong, 1987] to iterate between random imputations

under a specified set of parameter values and random draws from the posterior distribution of the parameters (given the observed and imputed data). These two steps are iterated long enough for the results to be reliable for multiple imputed datasets (Schafer, 1997). The goal is to have the iterates converge to their stationary distribution and then to simulate an approximately independent draw of the missing values. The algorithm is based on the assumptions that the data come from a multivariate normal distribution and are MAR.

Although not absolutely necessary, it is almost always a good idea to run the expectation maximization (EM) algorithm [Dempster *et al.*, 1977] before attempting to generate MIs. The parameter estimates from EM provide convenient starting values for data augmentation (DA). Moreover, the convergence behaviour of EM provides useful information on the likely convergence behaviour of DA. Therefore, EM estimates of the parameters are computed and then recorded the number of iterations required, say t. Then, a single run of DA algorithm of length tM using the EM estimates as starting values is performed, where M is the number of imputations required. The convergence of the EM algorithm is linear and is determined by the fraction of missing information. Thus, when the fraction of missing information is large, convergence will be very slow due to the number of iterations required. However, for small missing value proportions convergence is obtained much more rapidly with less strenuous convergence criteria. This is the approach we follow in this paper, which we shall now call EMMI.

3.3 Supervised Learning Techniques

Supervised learning (SL) is a machine learning technique for learning a function from training data. The training data consist of pairs of input objects (typically vectors), and desired outputs. The output of the function can be a continuous value (called regression), or can predict a class label of the input object (called classification). Supervised learning techniques have been successfully used to handling incomplete data. The SL techniques investigated in this paper involve the use of DTs [Breiman *et al.*, 1984; Quinlan, 1993]. These non-parametric techniques deal with missing values during the training (learning) or testing (classification) process. A well-known benefit of nonparametric methods is their ability to achieve estimation optimality for any input distribution as more data are observed, a property that no model with a parametric assumption can have. In addition, tree-based models do not make any assumptions on the distributional form of data and do not require a structured specification of the model, and thus not influenced by data transformation, nor are they influenced by outliers [Breiman *et al.*, 1984].

3.3.1 Fractional Cases

Quinlan (1993) borrows the probabilistic approach by Cestnik *et al.*, (1987) by "fractioning" cases or instances based on a *priori* probability of each value determined from the cases at that node that have specified values. Quinlan starts by penalising the information gain measure by the proportion of unknown cases and then splits these cases to both subnodes of the tree.

The learning phase requires that the relative frequencies from the training set be observed. Each case of, say, class *C* with an unknown attribute value *A* is substituted. The next step is to distribute the unknown examples according to the proportion of occurrences in the known instances, treating an incomplete observation as if it falls down all subsequent nodes. For example, if an internal node *t* has ten known examples (six examples with t_L and four with t_R), then we would say the probability of t_L = 0.6, and the probability of t_R is 0.4. Hence, a fraction of 0.6 of instance *x* is distributed down the branch for t_L and a fraction 0.4 of instance *x* to t_R. This is carried out throughout the tree construction process. The evaluation measure is weighted with the fraction of known values to take into account that the information gained from that attribute will not always be available (but only in those cases where the attribute value is known). During training, instance counts used to calculate the evaluation heuristic include the fractional counts of instances with missing values. Instances with multiple missing values can be fractioned multiple times into numerous smaller and smaller "portions".

For classification, Quinlan (1993)'s technique is to explore all branches below the node in question and then take into account that some branches are more probable than others. Quinlan further borrows Cestnik *et al.*'s strategy of summing the weights of the instance fragments classified in different ways at the leaf nodes of the tree and then choosing the class with the highest probability or the most probable classification. Basically, when a test attribute has been selected, the cases with known values are divided on the branches corresponding to these values. The cases with missing values are, in a way, passed down all branches, but with a weight that corresponds to the relative frequency of the value assigned to a branch. Both strategies for handling missing attribute values are used for the C4.5 system.

Despite its strengths, the fractional cases technique can be quite a slow, computationally intensive process because the probability calculation for the several DT branches must be done simultaneously. For example, if, say, *k* branches of the tree are all taken into account in the calculation, then the central processing unit (CPU) time spent is *k* times the individual branch calculation.

3.3.2 Surrogate Variable Splitting

Surrogate variable splitting (SVS) has been used for the classification and regression tress (CART) system and further pursued by Therneau and Atkinson (1997) in recursive partitioning and regression tress (RPART). CART handles missing values in the database by substituting "surrogate splitters". Surrogate splitters are predictor variables that are not as good at splitting a group as the primary splitter but which yield similar splitting results; they mimic the splits produced by the primary splitter; the second does second best, and so on. The surrogate splitter contains information that is typically similar to that which would be found in the primary splitter. The surrogates are used for tree nodes when there are values missing. The surrogate splitter contains information that is typically similar to what would be found in the primary splitter. Both values for the dependent variable (response) and at least one of the independent attributes take part in the modelling. The surrogate variable used is the one that has the highest correlation with the original attribute (observed variable most similar to the missing variable or a variable other than the optimal one that best predicts the optimal split). The surrogates are ranked. Any observation missing on the split variable is then classified using the first surrogate variable, or if missing that, the second is used, and so on. The CART system only handles missing values in the testing case but RPART handles them on both the training and testing cases.

The idea of surrogate splits solves the problem of missing values by identifying the nodes where masking or disguise (when one attribute hides the importance of another attribute) of specific attributes occurs. This is as a result of its ability to making use of all the available data by involving all the attributes when there is any observation missing the split attribute [Loh and Vanichsetakul, 1988]. By using surrogates, CART handles each instance individually, providing a far more accurate analysis. Also, other incomplete data techniques treat all instances with missing values as if the instances all had the same unknown value; with that technique all such "missings" are assigned to the same bin. For SVS, each instance is processed using data specific to that instance; and this allows instances with different data patterns to be handled differently, which results in a better characterisation of the data (Breiman *et al.*, 1984). However, practical difficulties can affect the way surrogate splitting is implemented. For example, SVS ignores the quantity of missing values. Like, a variable taking a unique value for exactly one case in each class and missing on all other cases yields the largest decrease in impurity (Wei-Yin, 2001). In addition, the idea of surrogate splitting is reasonable if high correlations among the predictor variables exist. Since the “problem” attribute (the attribute with missing values) is crucially dependent on the surrogate attribute in terms of a high correlation, when the correlation between the “problem” attribute and the surrogate is low, surrogate splitting becomes very

clumsy and unsatisfactory. In other words, the method is highly dependent on the magnitude of the correlation between the original attribute and its surrogate.

4. RELATED WORK

Significant advances have been made in the past few decades regarding methodologies which handle responses to problems and biases which can be caused by incomplete data. Unfortunately, these methodologies are often not available to many researchers for a variety of reasons (for example, lack of familiarity, computational challenges) and researchers often resort to ad-hoc approaches to handling incomplete data, ones which may ultimately do more harm than good [Little and Rubin, 1987; Schafer and Graham, 2002].

Several researchers have examined various techniques to solve the problem of incomplete data. One popular approach already discussed in Section 3.1 is LD.

Another common way uses imputation (estimation) approaches that fill in a missing value with an efficient single replacement value, such as the mean, mode, hot deck (using data from other observations in the sample at hand), and approaches that take advantage of multivariate regression and *k*-nearest neighbour models. Another technique for treating incomplete data is to model the distribution of incomplete data and estimate the missing values based on certain parameters. Specific results are discussed below.

Lakshminarayan *et al.* (1999) performed a simulation study comparing two ML methods for missing data imputation. Their results show that for the single imputation task, the supervised learning algorithm C4.5 [Quinlan, 1993], which utilizes the fractional cases (FC) strategy, performed better than Autoclass [Cheeseman *et al.*, 1988], a strategy based on unsupervised Bayesian probability. For the MI task, both methods perform comparably.

Lobo and Numao (1999; 2000) evaluated the accuracy performance of decision tree learning from data whose missing values were filled using the DTSI method (with ordered attributes), the majority method and the probabilistic method. For incomplete training and test data, the DTSI method outperformed both the probabilistic and the majority methods with the majority method giving the worst performance. For incomplete test data, no method performed better than the other.

Kalousis and Hilario (2000) evaluated seven classification algorithms with respect to missing values: two rule inducers (C5.0-rules and Ripper), one nearest neighbour method, one orthogonal (C5.0-tree), one oblique decision tree algorithm, a naïve Bayes algorithm and a linear discriminant. Various patterns and mechanisms of missingness (MCAR and MAR) in

current complete datasets were simulated. Their results indicate that naïve Bayes (NB) is most resilient to missing values while the k-nearest neighbour single imputation (*k*NNSI) and FC are more sensitive to missing values. Their results further show that for a given proportion of missing values, the distribution of missing values among attributes is at least as important as the mechanism of missingness.

Another comparative study of LD, MMSI, similar response pattern imputation (SRPI) and full information maximum likelihood (FIML) in the context of software cost estimation was carried out by Myvreit *et al.* (2001). The simulation study was carried out using 176 projects. Their results show FIML performing well for missing completely at random (MCAR) data. Also, LD, MMSI and SRPI are shown to yield biased results for other missing data mechanisms other than MCAR.

Fujikawa and Ho (2002) evaluated theoretically several methods of dealing with missing values. The methods evaluated were MMSI, linear regression, standard deviation method, *k*NNSI, DTSI, auto-associative neural network, LD, lazy decision tree, FC and SVS. *k*NNSI and DTSI showed good results. In terms of computation cost, MMSI and FC were found to be reasonably good.

Batista and Monard (2003) investigated the effects of four methods of handling missing data at different proportions of missing values. There methods investigated were *k*NNSI, MMSI, and internal algorithms used by FC and CN2 to treat missing data. Missing values were artificially simulated in different rates and attributes into the datasets. *k*NNSI imputation showed a superior performance compared with MMSI when missing values were in one attribute. However, both methods compared favourably when missing values were in more than one attribute. Otherwise, FC achieved a performance as good as *k*NNSI.

Song and Shepperd (2004) evaluated *k*NNSI and class mean imputation (CMI) for different patterns and mechanisms of missing data. Their results show *k*NNSI slightly outperforming CMI with the missing data mechanisms having no impact on either of the two imputation methods.

The use of multinomial logistic regression imputation (MLRI) for handling missing categorical values on a datataset on 166 projects of the ISBSG multi-organizational software database was proposed by Sentas *et al.* [2004]. Their proposed procedure was compared with LD, MMSI, expectation-maximization single imputation (EMSI) and regression-based single imputation (RBSI). Their results showed LD and MMSI as efficient when the percentage of missing values is small while RBSI and MLRI were shown to outperform LD and MMSI as the amount of missing

values increased. Overall, MLRI gave the best results, especially for MCAR and IM data. For MAR data, MLRI compared favourably with RBSI.

Twala *et al.* (2005) evaluates the impact of seven MDTs (LD, EMSI, *k*NNSI, MMSI, EMMI, FC and SVS) on 8 industrial datasets by artificially simulating three different proportions, two patterns and three mechanisms of missing data. Their results show EMMI achieving the highest accuracy rates with other notably good performances by methods such as FC and EMSI. The worst performance was by LD. Their results further show MCAR data as easier to deal with compared with IM data. Twala (2005) further found missing values as more damaging when they are in the test sample than in the training sample. An ensemble of missing incorporated in attributes (MIA) and EMMI approach was shown to improve prediction accuracy when dealing with incomplete data using DTs [Twala *et al.*, 2008].

According to the above studies, among the single imputation techniques, the results are not so clear, especially for small amounts of missing data. However, the performance of each technique differs with increases in the amount of missing data. Also, despite the fact that the LD procedure involves an efficiency cost due to the elimination of a large amount of valuable data, most researchers have continued to use it due to its simplicity and ease of use. There are other problems caused by using the LD technique. For example, elimination of instances with missing information decreases the error degrees of freedom in statistical tests such as the student *t* distribution. This decrease leads to reduced statistical power (i.e. the ability of a statistical test to discover a relationship in a dataset) and larger standard errors. Other researchers have shown that randomly deleting 10% of the data from each attribute in a matrix of five attributes can easily result in eliminating 59% of instances from analysis [Kim and Curry, 1977].

Furthermore, MI, which overcomes limitations of single imputation seem not to have been widely adopted by researchers even though it has been shown to be flexible and software for creating multiple imputations is available and some downloadable free of charge from the Methodology Centre website at the Penn State University *http://methodology.psu.edu/*. A great deal of past research on the effectiveness of MDTs to overcome missing data problems has utilized data that is missing randomly. Finally, results from previous studies suggest that results achieved using simulated data are very sensitive to the MAR assumption. Hence, if there is a reason to believe that if the MAR assumption does not hold, alternative methods should be used.

5. EXPERIMENTS

5.1. Experimental Set-Up

The objective of this paper is to investigate the robustness and accuracy of methods for tolerating incomplete data using tree-based models. This section describes experiments that were carried out in order to compare the performance of the different approaches previously proposed for handling missing values in both the training set and test (unseen) set. The effects of different proportions of missing values when building or learning the tree (training) and when classifying new instances (testing) are further examined, experimentally. Finally, the impact of the nature of different missing data mechanisms on the classification accuracy of resulting trees is examined. A combination of small and large datasets, with a mixture of both nominal and numerical attribute variables, was used for these tasks. All datasets have no missing values. The main reason for using datasets with no missing values is to have total control over the missing data in each dataset.

To perform the experiment each dataset was split randomly into 5 parts (Part I, Part II, Part III, Part IV, Part V) of equal (or approximately equal) size. 5-fold cross validation was used for the experiment. For each fold, four of the parts of the instances in each category were placed in the training set, and the remaining one was placed in the corresponding test as shown in Table 2. The same splits of the data were used for all the methods for handling incomplete data.

Table 2 Partitioning of dataset to training and test sets

	Training Set	*Test Set*
Fold 1	Part II + Part III + Part IV + Part V	Part I
Fold 2	Part I + Part III + Part IV + Part V	Part II
Fold 3	Part I + Part II + Part IV + Part V	Part III
Fold 4	Part I + Part II + Part III + Part V	Part IV
Fold 5	Part I + Part II + Part III + Part IV	Part V

Since the distribution of missing values among attributes and the missing data mechanism were two of the most important dimensions of this study, three suites of data were created corresponding to MCAR, MAR and IM. In order to simulate missing values on attributes, the original datasets are run using a random generator (for MCAR) and a quintile attribute-pair approach (for both MAR and IM, respectively). Both of these procedures have the same percentage of missing values as their parameters. These two approaches were also run to get

datasets with four levels of proportion of missingness *p*, i.e., 0%, 15%, 30% and 50% missing values. The experiment consists of having *p*% of data missing from both the training and test sets. This was carried out for each dataset and 5-fold cross validation was used.

The missing data mechanisms were constructed by generating a missing value template (1 = present, 0 = missing) for each attribute and multiplying that attribute by a missing value template vector. Our assumption is that the instances are independent selections.

For each dataset, two suites were created. First, missing values were simulated on only one attribute. Second, missing values were introduced on all the attribute variables. For the second suite, the missingness was evenly distributed across all the attributes. This was the case for the three missing data mechanisms, which from now on shall be called MCAR*univa*, MAR*univa*, IM*univa* (for the first suite) and MCAR*unifo*, MAR*unifo*, IM*unifo* (for the second suite). These procedures are described below.

For MCAR, each vector in the template (values of 1's for non-missing and 0's for missing) was generated using a random number generator utilising the Bernoulli distribution. The missing value template is then multiplied by the attribute of interest, thereby causing missing values to appear as zeros in the modified data.

Simulating MAR values was more challenging. The idea is to condition the generation of missing values based upon the distribution of the observed values. Attributes of a dataset are separated into pairs, say, (A_X, A_Y), where A_Y is the attribute into which missing values are introduced and A_X is the attribute on the distribution of which the missing values of A_Y is conditioned, i.e., $P(A_Y = \text{miss} \mid A_X = \text{observed})$. Pair selection of attributes was based on high correlations among the attributes. Since we want to keep the percentage of missing values at the same level overall, we had to alter the percentage of missing values of the individual attributes. More precisely, given that in each pair of attributes, one of the two attributes has no missing values, the other must have 2*k*% missing values in order to get an average of *k*% missing values over the whole dataset. For example, having 10% of missing values on two attributes is equivalent to having 5% of missing values on each attribute. Thus, for each of the A_X attributes its 2*k* quintile was estimated. Then all the instances were examined and whenever the A_X attribute has a value lower than the 2*k* quintile a missing value on A_Y is imputed with probability 0, and 1 otherwise. More formally, $P(A_Y = \text{miss} \mid A_X < 2k) = 0$ or $P(A_Y = \text{miss} \mid A_X > 2k) = 1$. This technique generates a missing value template which is then multiplied with A_Y. Once again, the attribute chosen to have missing values was the one highly correlated with the class variable. Here, the same levels

of missing values are kept. For multi-attributes, different pairs of attributes were used to generate the missingness. Each attribute is paired with the one it is highly correlated to. For example, to generate missingness in half of the attributes for a dataset with, say, 12 attributes (i.e., $A_1, \ldots A_{12}$), the pairs (A_1, A_2), (A_3, A_4) and (A_5, A_6) could be utilised. We assume that A_1 is highly correlated with A_2; A_3 highly correlated with A_4, and so on. For the (A_1, A_2) pairing, A_1 is used to generate a missing value template of zeros and ones utilizing the quintile approach. The template is then used to "knock off" values (i.e., generating missingness) in A_2, and vice versa.

In contrast to the MAR situation outlined above where data missingness is explainable by other measured variables in a study, IM data arise due to the data missingness mechanism being explainable, and only explainable by the very variable(s) on which the data are missing. For conditions with data IM, a procedure identical to MAR was implemented. However, for the former, the missing values template was created using the same attribute variable for which values are deleted in different proportions.

For consistency, missing values were generated on the same attributes for each of the three missing data mechanisms. This was done for each dataset. For split selection, the impurity approach was used. For pruning, a combination of 10-fold cross validation cost complexity pruning and 1 Standard Error (1-SE) rule (Breiman *et al.* 1984) to determine the optimal value for the complexity parameter was used. The same splitting and pruning rules when growing the tree were carried out for each of the twenty one datasets.

It was reasoned that the condition with no missing data should be used as a baseline and what should be analysed is not the error rate itself but the increase or excess error induced by the combination of conditions under consideration. Therefore, for each combination of method for handling incomplete data, the number of attributes with missing values (or missing data patterns), the proportion of missing values, the error rate at the control level (i.e., when there were no missing values in the data) was subtracted from the error when there were missing values. This would be the justification for the use of differences in error rates analysed in some of the experimental results.

Note that the modelling of the missingness was carried out after the training-testing split for each of the 5 iterations of cross validation. In other words, missingness was injected after the splitting of the data into training and test sets for each fold.

All statistical tests were conducted using the MINITAB statistical software program (MINITAB, 2002). Analyses of variance, using the general linear model (GLM) procedure [Kirk, 1982] were

used to examine the main effects and their respective interactions. This was done using a 4-way repeated measures designs (where each effect was tested against its interaction with datasets). The fixed effect factors were the: missing data techniques; number of attributes with missing values (missing data patterns); missing data proportions; and missing data mechanisms.

A 1% level of significance was used because of the many number of effects. The twenty one datasets used were used to estimate the smoothed error. Results were averaged across five folds of the cross-validation process before carrying out the statistical analysis. The averaging was done as a reduction in error variance benefit.

5.2. Datasets

This section describes the twenty one datasets that were used in the experiments to explore the impact of missing values on the classification accuracy of resulting DTs. All twenty one datasets were obtained from the Machine Learning Repository maintained by the Department of Information and Computer Science at the University of California at Irvine. They are summarized in Table 3.

The selected twenty one datasets cover a comprehensive range for each of the following characteristics:

- the size of datasets, expressed in terms of the number of instances ranges between 57 and 20000
- the number of attributes ranges between 4 and 6
- the number of classes ranges between 2 and 26
- the number of the type of attributes (numerical or nominal or both)

The first eight involve datasets with only two classes and the last thirteen involve datasets with more than two classes.

In general, the datasets were selected in order to assure reasonable comprehensiveness of the results.

Table 3 Datasets used for the experiments

Dataset	Instances	Attributes		Classes
		Ordered	*Nominal*	
Two classes:				
german	1000	7	13	2
glass (g2)	163	9	0	2
heart-statlog	270	13	0	2
ionosphere	351	31	1	2
kr-vs-kp	3196	0	36	2
labor	57	8	8	2
pima-indians	768	8	0	2
sonar	208	60	0	2
More than two				
balance scale	625	4	0	3
iris	150	4	0	3
waveform	5000	40	0	3
lymphography	148	3	15	4
vehicle	846	18	0	4
anneal	898	6	32	5
glass	214	9	0	6
satimage	6435	36	0	6
image	2310	19	0	7
zoo	101	1	15	7
LED 24	1500	0	24	10
vowel	990	10	3	11
letter	20000	16	0	26

5.3 Missing Data Programs and Codes

Programs and codes that were used for the methods are briefly described below:

No software or code was used for LD. Instead, all instances with missing values on that particular attribute were manually excluded or dropped, and the analysis was applied only to the complete instances.

The DTSI method uses a decision tree for estimating the missing values of an attribute and then uses the data with filled values to construct a decision tree for estimating or filling in the missing values of other attributes. This method makes sense when building a decision tree with incomplete data, the class variable (which plays a major role in the estimation process) is always present. For classification purposes (where the class variable is not present), first, imputation

for one attribute (the attribute highly correlated with class) was done using the mean (for numerical attributes) or mode (for categorical attributes), and then the attribute was used to impute missing values of the other attributes using the decision tree single imputation technique. In other words, two single imputation techniques were used to handle incomplete test data. An S-PLUS code that was used to estimate missing attribute values using a decision tree for both incomplete training and test data was developed.

S-PLUS code was also developed for the MMSI approach. The code was developed in such a way that it replaced the missing data for a given attribute by the mean (for numerical or quantitative attribute) or mode (for nominal or qualitative attribute) of all known values of that attribute.

There are many implementations of MI. Schafer's (1997) set of algorithms (headed by the NORM program) that use iterative Bayesian simulation to generate imputations was an excellent option. NORM was used for datasets with only continuous attributes. A program called MIX written was used for mixed categorical and continuous data. MIX is an extension of the well-known general location model. It combines a log-linear model for the categorical variables with a multivariate normal regression for the continuous ones. For strictly categorical data, CAT was used by following these steps: 1) You apply a model to the instances with observed Y, 2) You draw a vector from model parameter distribution, 3) You compute the predicted probabilities for each instance with missing Y under the new model, and 4) You create imputations by drawing a category using the predicted probabilities. All three programs are available as S-PLUS routines. Schafer (1997), and Schafer and Olsen (1998) gives details of the general location model and other models that could be used for imputation tasks. We use the completed datasets from iterations $2t$, $4t$, ..., $2Mt$. In our experiments we used MI with M=5, and averaged the predictions of the 5 resulting trees. Note that since MI bears a close resemblance to the EM algorithm, M=1 leads to single imputation, hence, EMSI.

Due to the limit of the dynamic memory in S-PLUS for Windows [S-PLUS, 2003] when using the EM approach, all the big datasets (i.e., datasets with more than 5000 instances) were partitioned into subsets, and S-PLUS run on one subset at a time. Our partitioning strategy was to put variables with high correlations with close scales (for continuous attributes) into the same subset. This strategy made the convergence criteria in the iterative methods easier to set up and very likely to produce more accurate results. The number of attributes in each subset depended on the number of instances and the number of free parameters to be estimated in the model, which included cell probabilities, cell means and variance-covariances. The number of attributes in each subset was determined in such a way that the size of the data matrix and the dynamic memory requirement was under the S-PLUS limitation and the number of instances was large relative to the number of free parameters.

Separate results from each subset were then averaged to produce an approximate EM-based method which are substituted for (and continue to call) EM in our investigation.

The DT learner C4.5 was used as a representative of the FC or probabilistic technique for handling missing attribute values in both the training and test samples. This technique is probabilistic in the sense that it constructs a model of the missing values, which depends only on the prior distribution of the attribute values for each attribute tested in a node of the tree. The main idea behind the technique is to assign probability distributions at each node of the tree. These probabilities are estimated based on the observed frequencies of the attribute values among the training instances at that particular node.

For the SVS method, a recursive partitioning (RPART) routine, which implements within S-PLUS many of the ideas found in the CART book and programs of Breiman *et al.* (1984) was used for both training and testing DTs. This programme, which handles both incomplete training and test data, is by Therneau and Atkinson (1997).

To measure the performance of methods, the training set/test set methodology is employed. For each run, each dataset is split randomly into 80% training and 20% testing, with different percentages of missing data (0%, 15%, 30%, and 50%) in the covariates for both the training and testing sets. A classifier was built on the training data and the predicted accuracy is measured by the smoothed error rate of the tree.

Trees on complete training data were grown using the *Tree* function in S-PLUS [Becker *et al.*, 1988, Venables and Ripley, 1994]. The function uses the *GINI* index of impurity [Breiman *et al.*, 1984] as a splitting rule and cross validation cost-complexity pruning as pruning rule. As indicated earlier, the accuracy of the tree, in the form of a smoothed error rate, was predicted using the test data.

5.4. Experimental Results

Experimental results on the effects of current methods for handling both incomplete training and test data on predictive accuracy on resulting DTs for different proportions, patterns are and mechanisms of missing data are summarised in Figure 2.

The error rates of each method of the introduced missing values are averaged over the 21 datasets. Also, all the error rates increased over the complete data case formed by taking differences. From these experiments the following results are observed.

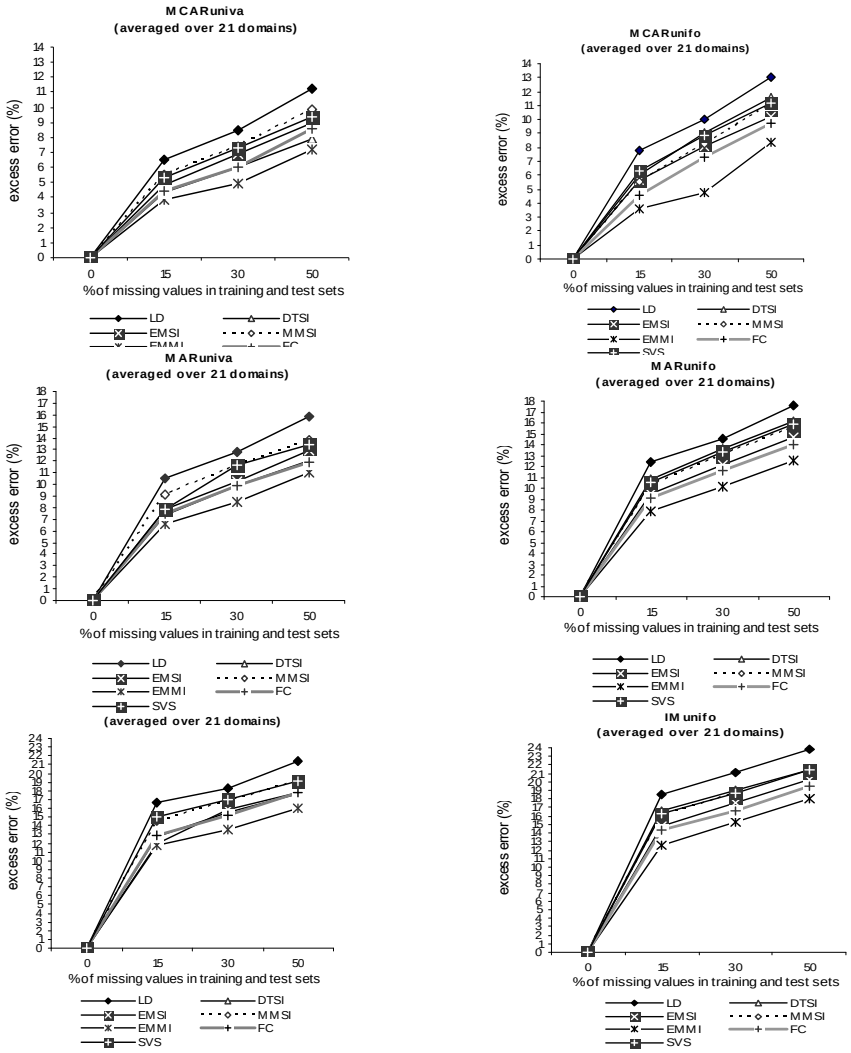

Fig. 2. Effects of missing values in training and test data on the excess error for methods over the 21 domains. A) MCAR*univa*, B) MCAR*unifo*, C) MAR*univa*, D) MAR*unifo*, E) IM*univa*, F) IM*unifo*

For MCAR*univa* data, EMMI has on average the best accuracy while LD exhibits one of the biggest increases in error (Figure 2A). From Figure 2B, most of the methods achieve slightly bigger error rate increases for MCAR*unifo* data compared with MCAR*univa* data. In the MAR*univa* suite, the behaviour of methods is not very different from the one observed in the MCAR*univa* case (Figure 2C). From Figure 2D, the performance of methods for MAR*unifo* data is very similar to the one observed for MCAR*unifo* data. Figure 2E shows bigger increases in error rates for all the methods for IM*univa* data compared with MCAR*univa* and MAR*univa* data, individually. The performance of all the methods, on average, worsens for IM*unifo* data. Once again, EMMI proves to be the best method at all levels of missing with LD exhibiting the worst performance with an excess error rate of 23.9% at the 50% level (Figure 2F).

Main Effects

All the main effects (training and testing methods, number of attributes with missing values, missing data proportions and missing data mechanisms) were found to be significant at the 1% level.

From Figure 3, EMMI represents a superior approach to missing data while LD is substantially inferior to the other techniques. The second best method is FC, closely followed by EMSI, DTSI, MMS and SVS, respectively. However, there appears no clear 'winner' among the single imputation techniques (DTSI, EMSI, MMSI) and the supervised learning approach (SVS) in terms of classification accuracy.

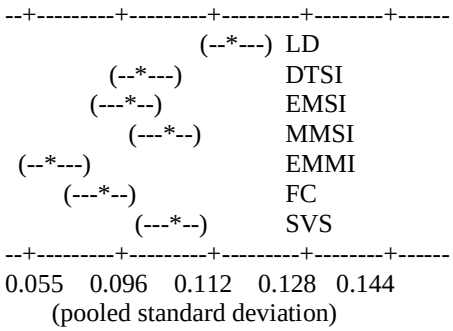

Fig. 3. Comparison for missing data methods: confidence intervals of mean error rates (*)

From Figure 4, it appears that missing values have a greater effect when they are distributed among all the attributes (arbitrary pattern) compared with when missing values are on a single attribute variable (univariate pattern).

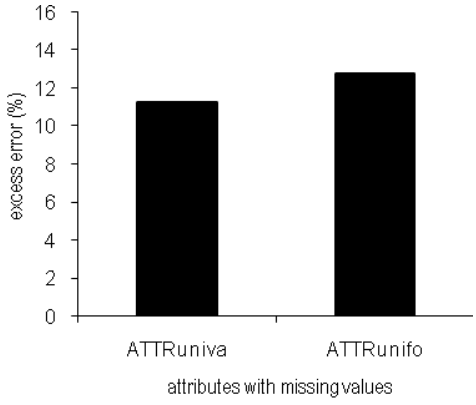

Fig. 3. Overall means for missing data patterns

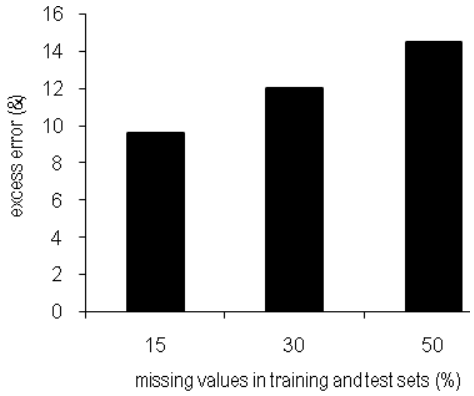

Fig. 4. Overall means for missing data proportions

The results for proportion of missing values in both the training and test sets show increases in missing data proportions being associated with increases in error rates (Fig. 5). In fact, the error rate increase when 50% of values are missing in both the training and test sets is about one and a half times as big as the error rate increase when 15% of values are missing on both sets.

From the results presented in Figure 6, IM values entail more serious deterioration in predictive accuracy compared with randomly missing data (i.e. MCAR or MAR data). Overall, MCAR data have a lesser impact on classification accuracy with an error rate increase difference of about 4% (when compared with MAR data) and a much bigger difference of about 10% (when compared with IM data).

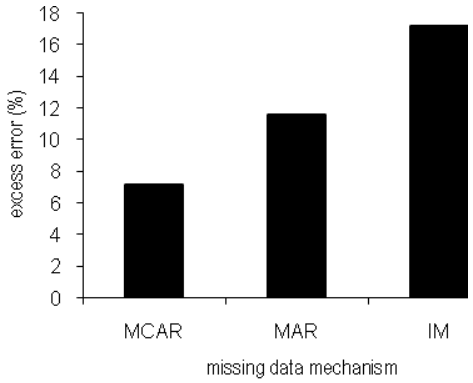

Fig. 5. Overall means for missing data mechanisms

Interaction effects

The interaction effect between methods for handling incomplete training and test data and the number of attributes with missing values is displayed in Figure 7. From the figure, it follows that all methods perform differently from each other with bigger error rate increases observed when missing values are in all the attributes compared with when they are on a single attribute variable. A severe impact of having missing values in only one attribute and having missing values in all the attributes is observed for DTSI, LD and SVS, while for the remaining methods the impact is not clear.

From Figure 8, the performance by methods do not differ much at lower levels of missing values but vary noticeably as the amount of missing values increases.

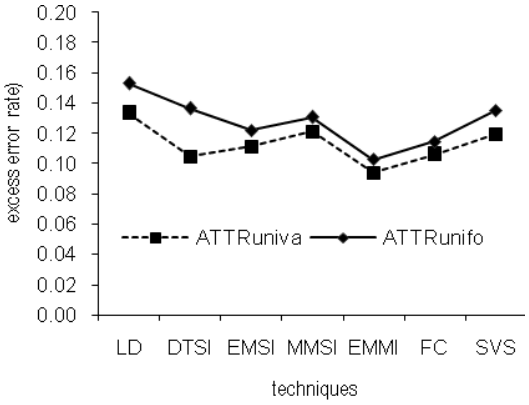

Fig. 7. Interaction between missing data techniques and missing data pattern

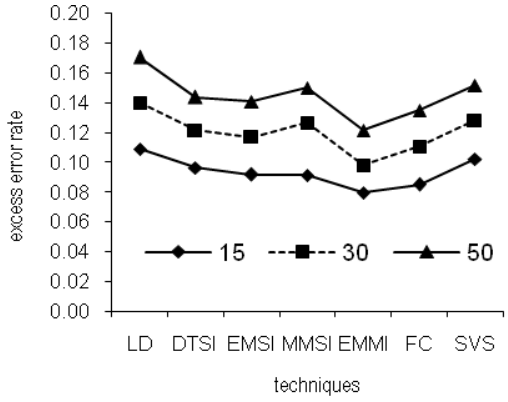

Fig. 8. Interaction between methods and proportion of missing values

As observed earlier, all the methods are more severely impacted by IM data compared with MCAR or MAR data (Figure 9). In addition, all the methods are more effective for dealing with MCAR data.

The results of the interaction between the number of attributes with missing values and missing data proportions show increases in missing data proportions being associated with slightly greater increases in excess error rate (Figure 10). Once again, missing values appear to have more impact for an arbitrary missing data pattern compared with a univariate pattern.

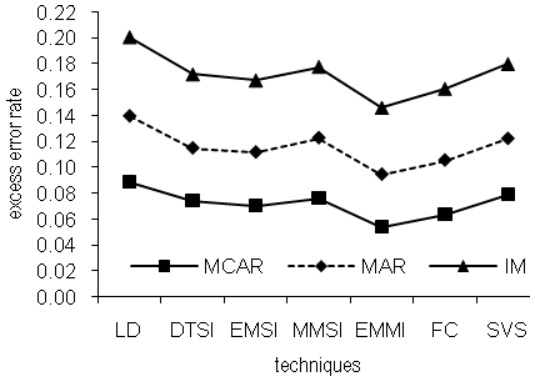

Fig. 9. Interaction between missing data techiniques and missing data mechanisms

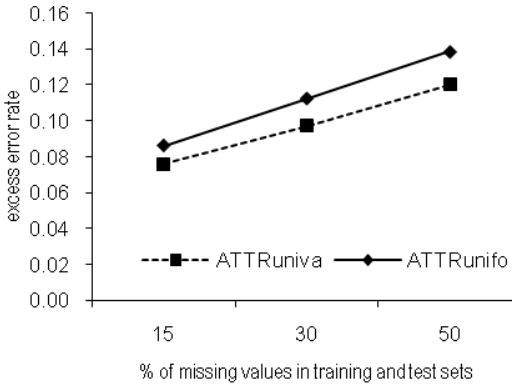

Fig. 10. Interaction between missing data patterns and proportion of missing data

5. DISCUSSION AND CONCLUSIONS

The research questions asked which MDTs yielded the least amount of average error when using tree-based models. To date, there have been very few studies examining the effects of MDTs for different patterns, proportions and mechanisms of missing data. Each of the techniques

reviewed in this paper has strengths and limitations. However, the use of imputations has become common in many fields in the last few years. It is due to the increase in incomplete data in research, and to the advanced methodology for imputations.

The selection of a particular MDT can have important consequences on the data analysis and subsequent interpretation of findings in models of supervised learning, especially if the amount of missing data is large. The impact of missing data can not only lead to bias in results (especially parameter estimates derived from the model) but to loss of information and thus to a decrease in statistical power of hypothesis tests. This may also result in misleading conclusions drawn from a research study and limit generalizability of the research findings.

The results of the simulation study show that the proportion of missing data, the missing data mechanism, the pattern of missing values, and the design of database characteristics (especially the type of attributes) all have effects on the performance of any MDT.

The effects of missing data have been found to adversely affect DT learning and classification performance, and this effect is positively correlated with the increasing fractions of missing data.

Another point of discussion is the significance of having missing values in only one attribute, on the one hand, and allowing missing values in all the attributes, on the other hand. The idea was to see the impact of pattern over mechanism, or vice versa, at both lower levels and higher levels of missing values. Our results show the impact on the performance of methods being caused by the pattern and mechanism of missing values, especially at lower levels of missingness. However, as the proportion of missing values increases the major determining factor on the performance of methods is how the missing values are distributed among attributes. All methods yield lower accuracy rates when missing values are distributed among all the attributes (MCAR*unifo*, MAR*unifo* and IM*unifo*) compared with when missing values are on a single attribute (MCAR*univa*, MAR*univa* and IM*univa*).

The worse performance achieved by methods are for IM data, followed by MAR and MCAR data, respectively. This is in accordance with statistical theory which considers MCAR easier to deal with and IM data as very complex to deal with since it requires assumptions that cannot be validated from the data at hand [Little and Rubin, 1987]. In addition, in many settings the MAR assumption is more reasonable than the MCAR. In fact, an MAR method is valid if data are MCAR or MAR, but MCAR methods are valid only if data are MCAR. This could have attributed to the superior performance of EMMI (an MAR method) and the substantially inferior performance of LD (an MCAR method).

The results also show that the performance of methods depends on whether missing values are in the test or training set or in both the training and test sets. Training methods appear to achieve superior performances compared with testing methods. An explanation for such behaviour will be given later in the section.

With this experimental set-up, it is easy to say with conviction that from the eight current techniques investigated that EMMI is the overall best method for handling both incomplete training and test data. However, there are competitors like FC and DTSI which performed reasonably well. One important advantage of FC over DTSI is that it can handle missing values in both the training and test sets while DTSI struggles as a technique for handling incomplete test data and when all attributes have missing values. The heavily dependence of DTSI on strong correlations among attributes might have attributed to its poor performance as correlations among attributes for some of the datasets were not strong. However, DTSI performs better when missing values are on a single attribute – a very serious restriction. The results also indicate that LD is the worst method for handling incomplete data. In general, it can be seen that model-based methods have better performance than ad hoc methods. Furthermore, probabilistic methods seem to outperform non-probabilistic methods.

There are several dimensions on which learning methods of handling incomplete data using tree-based models can be compared. Also, combinations of methods for handling incomplete data while varying the number of attributes with missing values were not tried. However, prediction accuracy rates of estimation methods like the EMMI were very impressive. The improvement in accuracy of EMMI over single imputation methods (CCSI, DTSI, EMSI and MMSI) and other methods (LD, SVS and FC) could be as a result of a reduction in variance resulting from averaging the number of trees like is done in bagging (Breiman, 1996). Even though EMMI emerges as the overall best of the eight techniques, it has come under fire by critics claiming that proper imputations, necessary for valid inferences, are difficult to produce, especially in data where multiple factors are deficient (Schafer, 1997), and even then EMMI is biased in some cases [Robins and Wang, 2000]. Another argument against EMMI is that it is much more difficult to implement than some of the techniques mentioned. One potential problem that was encountered in this research is convergence of the EM algorithm [Wu, 1983], especially for big datasets and datasets with more than 30 attribute variables.

The results also show that the performance of methods depends on whether missing values are in the test or training set or in both the training and test sets. When looking at the overall performance of methods, training methods appear to achieve superior performances compared with testing methods. However, in terms of relative performance, they seem to be about the same.

Each dataset might have more or less its own favourite techniques for processing incomplete data. However, most of our dataset results are similar to one another. Several factors contribute here: the methods used; the different types of datasets; the distribution of missing values among attributes, the magnitude of noise, distortion and source of missingness in each dataset.

There was evidence from our results that how well a method performs depends on the dataset one is analysing. In fact, all methods were able to handle datasets with numerical attributes better compared with datasets with only nominal or a mixture of both nominal and numerical attributes.

For small datasets all the techniques seemed to work well with the estimation methods, especially EMMI which performed better than both the other estimation methods and machine learning methods. One the other hand, LD gave the worst performance for small datasets. This seems to be logical since when using LD you tend to lose a lot of information, especially at higher levels of missing values. However, for bigger datasets, LD was equally effective compared with methods like CCSI and MMSI and outperformed them in other situations. The effectiveness of LD probably stems from the fact that deletion techniques result in data matrices that mirror the true data structure. When data are systematically missing from a study, the imputation techniques create a "reproduced" data matrix with a structure somewhat different from that of the true data matrix.

Following EMMI as the second best overall method for handling both incomplete training and test data is FC. In general it can be seen that probabilistic methods have better performance than non probabilistic methods. However, one important disadvantage of FC, just like EMMI, is that it takes a long time processing (especially big trees) due to the way it handles missing values. Due to its reliance on the number of branches to do the calculation simultaneously, if K branches do the calculation, then the CPU time spent is K times the individual branch calculation.

Of all the single imputation methods, EMSI seems to perform the best for datasets with numerical attributes. However, despite its strengths, EMSI suffers (like all the single imputation methods) from biased and sometimes inefficient estimates. DTSI and MMSI achieved good results when missing values were on categorical attributes. For DTSI, this was the case when the missing values were in only one attribute while MMSI gave good results generally and in some cases was a good method for datasets with mixed attributes. Overall, the differences in results between single imputation methods are relatively small. The similarity of results begs an important question: when and why should we choose one single imputation method over the other?

Like DTSI, for all the datasets where the correlations among attributes were found to be quite high, SVS (which relies heavily on strong concordance between a primary splitter and its surrogate(s)), achieved good results. However, SVS also struggles when missing values are distributed among all the attributes. In fact, for a few datasets SVS collapsed completely when an instance was missing all the surrogates. However, some strategies when simulating the missingness among the attributes were used when the technique collapsed.

Despite good performances by some of the single imputation strategies, we will still strongly recommend avoiding *ad hoc* approaches such as LD, CCSI and MMSI when dealing with missing data. Such approaches not only do they give unpredictable results they are also not underpinned by statistical theory.

Another point of discussion is why missing values are more damaging when they are in the test sample than training sample. If you have a lot of training data then missing values do not make much impact on the parameter estimates but missing data in the test set refers to only individual cases. That is, the training data yield statistical summaries, but the test data are concerned with individuals. In other words, missing data will tend to cancel each other out when training the model. On a new test case, the investigator must still suffer accuracy affects though, inevitably. What is really happening here is that the increased error in test cases is to be expected and the significantly reduced error when training is a pleasant surprise and this is due to the averaging (ensemble) of individual results.

Furthermore, it is worthwhile mentioning that the performance of some methods could have been slightly affected by other factors like errors in some datasets. For example, the Pima Indians diabetes database had quite a number of observations with "zero" values, which are most likely to indicate missing values although the data was described as being complete. Nonetheless, the prediction that the impact of certain types of missing data mechanisms on both the testing and training cases should differ by dataset, by mechanism and by the proportion of missing values is confirmed.

Some experimental results from Section 4.3 support previous findings in the literature and other results extend the literature. The relative superiority of model-based methods over ad hoc methods is consistent with past results [Kim and Curry, 1977; Little and Rubin, 1987; Rubin, 1987].

Overall, the performance of each MDT under more complex forms of systematic missingness is unknown and likely to be problematic [Little and Rubin, 1987]. Systematic missingness in this simulation was always based on the variables that were in the model rather than unmeasured variables or combinations of variables. In addition, it is impossible, in practice, to demonstrate

whether data are MAR versus IM, because the values of the missing data are not available for comparison. IM is still a problem for the methods reviewed here.

While this study confirms statistical insight on the importance of the pattern of missingness, it also reveals that the distribution of unknown values among the predictors plays an equally (if not more) decisive role, depending on the percentage of missing data. In a subsequent study we plan to distribute missing values over as many attributes as possible (other than being restricted to one).

So far, we have restricted our experiments to only tree-based models. It would be interesting to carry out a comparative study of tree based models with other (non-tree) methods which can handle missing values. Furthermore, it would be possible to explore the different patterns and levels of missing values.

This paper also addresses the problem of tolerating missing values in both the training and test sets. Breiman *et al.*, (1984) argue that missing values have more impact when they occur in the test set. An interesting topic would be to assess the impact of missing values when they only occur in either the training or test set.

We leave the above issues to be investigated in the future.

In sum, this paper provides the beginnings of a better understanding of the relative strengths and weaknesses of MDTs and using DTs as their component classifier. It is hoped that it will motivate future theoretical and empirical investigations into incomplete data and DTs, and perhaps reassure those who are uneasy regarding the use of imputed data in prediction.

ACKNOWLEDGEMENTS

The work was funded by the Council of Science and Industrial Research (CSIR) under project MDSARR1. The author would like to thank Wray Buntine, David Hand and Chris Jones for helpful discussions, and to the anonymous reviewers for their useful comments.

REFERENCES

ARBUCKLE, J.L. AND & WOTHKE, W. (1999). *Amos 4.0 user's guide*. Chicago, IL: Smallwaters.

BATISTA, G. AND MONARD, M.C. (2003). An Analysis of Four Missing Data Treatment Methods for Supervised Learning, *Applied Artificial Intelligence,* **17**, 519-533.

BECKER, R., CHAMBERS, J., AND WILKS, A. (1988). *The New S Language*. Wadsworth International Group.

BLAKE, C.L. and MERZ, C.J. (1998). UCI Repository of machine learning databases. University of California, Department of Information and Computer Science, Irvine, CA. <http:/www.ics.uci.edu/mlearn/MLRepository.html>

BREIMAN, L., FRIEDMAN, J., OLSHEN, R., AND STONE, C. (1984). *Classification and Regression Trees*, Wadsworth.

BREIMAN, L. (1996). Bagging predictors. *Machine Learning*, **26** (2), 123-140.

CARTWRIGHT, M., SHEPPERD, M.J., AND SONG, Q. (2003). Dealing with Missing Software Project Data. In *Proceedings of the 9th International Symposium on Software Metrics 2003*, 154-165.

CESTNIK, B., KONONENKO, I. AND BRATKO, I. (1987). Assistant 86: a knowledge-elicitation tool for sophisticated users. In I. Bratko and N. Lavrac, editors, *European Working Session on Learning* – EWSL87. Sigma Press, Wilmslow, England, 1987.

CHEESEMAN, P., KELLY, J., SELF, M., STUTZ, J., TAYLOR, W., AND FREEMAN, D. (1988). Bayesian Classification. In *Proceedings of American Association of Artificial Intelligence (AAAI)*, Morgan Kaufmann Publishers: San Meteo, CA, 607-611.

CHEN, J. AND SHAO, J. (2000). Nearest Neighbour Imputation for Survey Data. *Journal of Official Statistics*, **16**, 2, 113-131.

DEMPSTER, A.P., LAIRD, N.M., AND RUBIN, D.B. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, **39**, 1-38.

DIETTERICH, T.G. (2000). Ensemble Methods in Machine Learning. Proceedings of the 1st International Workshop on Multiple Classifier Systems, **1857**, 1-15.

EL-EMAM, K. AND BIRK, A. (2000). Validating the ISO/IEC 15504 Measures of Software Development Process Capability. *Journal of Systems and Software*, **51** (2), 119-149.

FUJIKAWA, Y., HO, T.B. (2002). Cluster-based Algorithms for Filling Missing Values, 6th Pacific-Asia Conf. Knowledge Discovery and Data Mining, Taiwan, 6-9 May, Lecture Notes in Artificial Intelligence 2336, Springer Verlag, 549-554.

GRAHAM, J.W. AND HOFER, S.M. (1993). *EMCOV.EXE user's guide* (unpublished manuscript). University Park, PA: Pennsylvania State University Department of Bahavioral Health.

JÖNSSON, P. AND WOHLIN, C. (2004). An Evaluation of k-Nearest Neighbour Imputation Using Likert Data. In *Proceedings of the 10th International Symposium on Software Metrics*, 108-118, September 11-17, 2004.

KALOUSIS, A. AND HILARIO, M. (2000). Supervised knowledge discovery from incomplete data. In *Proceedings of the 2nd International Conference on Data Mining 2000*. Cambridge, England. July 2000.WIT Press.

KALTON, G. (1983). Compensating for Missing Survey Data, Michigan.

KIM, J.-O. AND CURRY, J. (1977). The treatment of missing data in multivariate analysis. *Sociological Methods and Research*, **6**, 215-240.

KIRK, R.E. (1982). *Experimental design* (2nd Ed.). Monterey, CA: Brooks, Cole Publishing Company.

LAKSHMINARAYAN, K., HARP, S.A., AND SAMAD, T. (1999). Imputation of Missing Data in Industrial Databases. *Applied Intelligence*, **11**, 259-275.

LITTLE, R.J.A. AND RUBIN, D.B. (1987). *Statistical Analysis with missing data*. New York: Wiley.

LOBO, O.O. AND NUMAO, M. (1999). Ordered Estimation of Missing Values. In *Proc. of the 3rd Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, 1574, 1999, 274-278.

LOBO, O.O. AND NUMAO, M. (2000). On the Applicability of a Machine Learning Method for Estimating Missing Values. *IMLC 2000*, Palo Alto, California.

LOH, W.-Y. and VANICHSETAKUL, N. (1988). Tree-structured classification via Generalised Discriminant Analysis. *Journal of the American Statistical Association*, **83**, 715-728.

MINITAB. (2002). MINITAB *Statistical Software for Windows 9.0*. MINITAB, Inc., PA, USA.

MULTIPLE IMPUTATION SOFTWARE. Available from <http:/www.stat.psu.edu/jls/misoftwa.html, http:/methcenter.psu.edu/EMCOV.html>

MYRTVEIT, I., STENSRUD, E., AND OLSSON, U. (2001). Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods. *IEEE Transactions on Software Engineering*, **27** (11), 1999-1013.

PYLE, D. (1999). *Data Preparation for Data Mining*. Morgan Kauffman, San Francisco.

QUINLAN, J.R. (1987). Simplifying decision trees, *International Journal of Man - Machine Studies*, **27**, 221-234.

QUINLAN, JR. (1993). *C.4.5: Programs for machine learning*. Los Altos, California: Morgan Kauffman Publishers, INC.

ROBINS, D.B. AND WANG, N. (2000). Inference for imputation estimators. *Biometrika*, **87**, 113-124.

ROTH, P.L. (1994). Missing data: A conceptual overview for applied psychologists. *Personnel Psychology*, **47**, 537-560.

RUBIN, D.B. (1987). *Multiple Imputation for Nonresponse in Surveys*. New York: John Wiley and Sons.

RUBIN, D.B. (1996). Multiple Imputation After 18+ Years. *Journal of the American Statistical Association*, **91**, 473-489.

SCHAFER, J.L. (1997). *Analysis of Incomplete Multivariate Data*. Chapman and Hall, London.

SCHAFER, J.L. AND OLSEN, M.K. (1998). Multiple Imputation for multivariate missing data problems: a data analyst's perspective. *Multivariate Behavioral Research*, **33** (4): 545-571.

SCHAFER, J.L. AND GRAHAM, J.W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, **7** (2), 147-177.

SHAPIRO, A. (1987). *Structured Induction in Expert Systems*. Addison Wesley, London.

SENTAS, P., LEFTERIS, A., AND STAMELOS, I. (2004). Multiple Logistic Regression as Imputation Method applied on Software Effort prediction. In *Proceedings of the 10th International Symposium on Software Metrics*, Chicago, 14-16 September 2004.

STRIKE, K., EL-EMAM, K.E., AND MADHAVJI, N. (2001). Software Cost Estimation with Incomplete Data. *IEEE Transaction on Software Engineering*, **27** (10), 890-908.

SONG, Q. AND SHEPPERD, M. (2004). A Short Note on Safest Default Missingness Mechanism Assumptions. In *Empirical Software Engineering*. (Accepted for publication in 2004).

S-PLUS. (2003). *S-PLUS 6.2 for Windows*. MathSoft, Inc., Seattle, Washington, USA.

TANNER, M.A. AND WONG, W.H. (1987). The Calculation of Posterior Distributions by Data Augmentation (with discussion). *Journal of the American Statistical Association*, **82**, 528-550

THERNEAU, T.M., AND ATKINSON, E.J. (1997). An Introduction to Recursive Partitioning Using the RPART Routines. *Technical Report*, Mayo Foundation.

TWALA, B., JONES, M.C. AND HAND, D.J. (2008). Good methods for coping with missing data in decision trees. *Pattern Recognition Letters*, 29, 950-956.

TWALA, B. (2005). *Effective Techniques for Handling Incomplete Data Using Decision Trees*. Unpublished PhD thesis, Open University, Milton Keynes, UK

TWALA, B., CARTWRIGHT, M., AND SHEPPERD, M. (2005). Comparison of Various Methods for Handling Incomplete Data in Software Engineering Databases. *4th International Symposium on Empirical Software Engineering*, Noosa Heads, Australia, November 2005.

VENABLES, W.N. AND RIPLEY, B.D. (1994). *Modern Applied Statistics with S-PLUS*. New York: Springer.

WU, C.F.J. (1983). On the convergence of the EM algorithm. *The Annals of Statistics*, **11**, 95-103.