Neural Networks 24 (2011) 121–129

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Missing value imputation on missing completely at random data using multilayer perceptrons

Esther-Lydia Silva-Ramírez [a,*], Rafael Pino-Mejías [b,c], Manuel López-Coello [a], María-Dolores Cubiles-de-la-Vega [c]

[a] Department of Computer Languages and Systems, University of Cadiz, C/Chile N 1, 11003 Cadiz, Spain
[b] Andalusian Prospective Center, Avda. Reina Mercedes s/n, 41012 Seville, Spain
[c] Department of Statistics and Operational Research, University of Seville, Avda. Reina Mercedes s/n, 41012 Seville, Spain

ARTICLE INFO

Article history:
Received 12 March 2010
Received in revised form 9 September 2010
Accepted 10 September 2010

Keywords:
Multilayer perceptron
Hot-deck model
Imputation
Mean/mode model
Missing data
Regression model

ABSTRACT

Data mining is based on data files which usually contain errors in the form of missing values. This paper focuses on a methodological framework for the development of an automated data imputation model based on artificial neural networks. Fifteen real and simulated data sets are exposed to a perturbation experiment, based on the random generation of missing values. These data set sizes range from 47 to 1389 records. A perturbation experiment was performed for each data set where the probability of missing value was set to 0.05. Several architectures and learning algorithms for the multilayer perceptron are tested and compared with three classic imputation procedures: mean/mode imputation, regression and hot-deck. The obtained results, considering different performance measures, not only suggest this approach improves the quality of a database with missing values, but also the best results are clearly obtained using the Multilayer Perceptron model in data sets with categorical variables. Three learning rules (Levenberg–Marquardt, BFGS Quasi-Newton and Conjugate Gradient Fletcher–Reeves Update) and a small number of hidden nodes are recommended.

1. Introduction

Nowadays different data collection systems exist, for example, the computer assisted personal interview (CAPI), the computer assisted telephone interview (CATI) or the web assisted personal interview (WAPI). However, there is not any system granting perfect data sets, and a certain risk of generating errors is always present. Therefore, data sets usually contain errors in the form of missing or inconsistent values. Missing values are due to the lack of response, while inconsistent values are produced when the answer is not accurately recorded.

Data cleaning is concerned with the data quality, and therefore the treatment of missing values belongs to this step of the data knowledge discovery process. A possible strategy to deal with this problem is to perform data imputation, defined as the process by which values in a data set that are missing are estimated by appropriately computed values. In other words, data imputation is capable of filling in the gaps of the data set with errors of non-response, producing a complete data set. Some classical techniques used in this imputation approach are hot-deck, the mean/mode substitution or regression models.

In this paper, data imputation is formulated as a problem of estimation of missing values using Artificial Neural Networks. Artificial Neural Networks (ANNs in what follows) are a set of nonlinear mathematical models, suitable for prediction and classification problems. Their high flexibility is characterized by a variety of theoretical properties, which convert them into universal approximators (Bishop, 1995; Ripley, 1996).

An automatic procedure to the missing value imputation based on neural networks is described in Section 4. Its performance is compared with three alternatives in Section 5, where the neural network configuration is extensively studied. The results and conclusions shown in Sections 5 and 6 reveal a clear improvement of the quality of the data sets for this machine learning approach.

We have chosen the Multilayer Perceptron (MLP) to carry out an exhaustive study of the influence of decisions such as the size of the net, the learning rule or the number of epochs. However, our approach needs a unique neural net to impute the missing values for the different variables in the data set.

Other works about data imputation consider more involved techniques, but the data imputation needs more models. For example, in Junninen, Niska, Tuppurainen, Ruuskanen, and Kolehmainen (2004), where several methods for data imputation in air quality data sets are compared, for each missing data pattern a

* Corresponding author. Tel.: +34 956 01 52 96; fax: +34 956 01 51 39.
E-mail addresses: esther.silva@uca.es (E.-L. Silva-Ramírez), rafaelp@us.es (R. Pino-Mejías), manuel.coello@uca.es (M. López-Coello), cubiles@us.es (M.-D. Cubiles-de-la-Vega).

0893-6080/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2010.09.008

multilayer perceptron network trained with backpropagation is fitted. The Self-Organized Map and MLP performed slightly better than regression and nearest neighbour, but MLP was computationally less demanding than SOM. Ssali and Marwala (2008) study the decision trees with neural networks, principal component analysis and genetic algorithms to impute missing data, separating into two architectures, one based on an auto-associative neural network and the other one on the principal component analysis. Each one is combined with a decision tree as well as the genetic algorithm optimization routine. This work shows that both architectures for missing data imputation are improved by adding decision trees. Nelwamondo, Mohamed, and Marwala (2007) study the maximum likelihood approach with the neural network and GA combination approach. An auto-associative neural network was trained to predict its own input space and GAs were used to approximate the missing data. This work shows that the imputation ability of one method over another seems highly problem-dependent.

2. Data imputation methods

We consider a data set S, disposed as data matrix where p variables have been measured in each one of n records, defining a matrix $n \times p$, but the data collecting process may have produced missing values for some cells of S. Thus, S is an approximation to the true data set T, where the $n \times p$ values would have been completely recorded. A data imputation model is defined by a set of rules and procedures to obtain an approximation T^* to T working on the available data set S.

A wide range of methods and tools for data imputation are available. Little and Rubin (1987) gave extensive methods of treatment to analyze incomplete data, many of which are intended for continuous and normally distributed data. Some methods try to make a maximum use of the available information, for example, listwise, casewise and pairwise data deletion techniques, based on the omission of all those records that contain a missing value for one or more variables, depending on the population parameters to be estimated. Other methods are proper imputation techniques as they compute appropriate values for replacing the missing data. According to Little and Rubin (1987), methods of handling missing data as listwise and pairwise data deletion and mean/mode are inferior, regression methods are somewhat better, but not as good as hot-deck or procedures based on multiple imputation. So, according to their degree of complexity, we have implemented three of these methods for our empirical comparison with ANNs: mean/mode substitution, regression imputation and hot-deck imputation.

- *Mean/mode imputation*: It is a simple method where any missing value of a quantitative variable is replaced by the mean of the observed values for that variable. So, if a variable presents several missing values for different records, all of them are imputed with the same value. If the variable is qualitative, the missing values are replaced by the mode.
  $X_j^{(obs)}$ denotes the set of records with values observed for the variable or attribute j, $X_j^{(mis)}$ denotes the set of records with missing values for the variable j, with $j = 1, \ldots, p$. Since there are variables of any kind, numerical and categorical, distinguishing between types of variables is necessary. If the variable is quantitative, missing values for that variable are imputed with the mean of observed values. On the contrary, if the variable is qualitative, missing values for that variable are imputed with the category that have the most of individuals with observed values, this is, with the mode. The set of estimated values is denoted by $\overline{X}^{(obs)}$, therefore, $\widehat{X}_j^{(mis)} = \overline{X}_j^{(obs)}$.
- *Regression models*: Given a missing value for a variable X, suppose that q variables have been observed for that record. The records where these $q + 1$ variables are available define a training set, and a regression model to predict X from the q predictors is fitted. Finally, the fitted model provides a prediction for the initial missing value of X. Three multiple regression procedures have been considered in our study: Multiple Linear Regression for a quantitative variable X, Logistic Regression when the dependent variable X is dichotomous, and Multinomial Logistic Regression is used to handle categorical variables with more than two categories.
- *Multiple Linear Regression*. A number $p > 1$ of independent variables $X_1, X_2, \ldots, X_p$ is considered, so a population model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$, is assumed where Y denotes the dependent variable or response, $X_1, X_2, \ldots, X_p$ are the independent or predictor variables, ε is a random disturbance or error whose presence represents the absence of an accurate relationship. And $\beta_0, \beta_1, \ldots, \beta_p$ are unknown coefficients or parameters that define the regression hyperplane $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$.
  If a qualitative variable is considered with c categories, $c - 1$ dummy dichotomous variables are introduced into the model:

$$z_{i1} \begin{cases} 0 & \text{if } i \notin \text{category 1} \\ 1 & \text{if } i \in \text{category 1} \end{cases}$$

$$z_{i2} \begin{cases} 0 & \text{if } i \notin \text{category 2} \\ 1 & \text{if } i \in \text{category 2} \end{cases}$$

$$\vdots$$

$$z_{i,c-1} \begin{cases} 0 & \text{if } i \notin \text{category } c-1 \\ 1 & \text{if } i \in \text{category } c-1. \end{cases}$$

  The category c is the base category. Any variable for which the category is built, defined and identified, are all individuals that have value 0 for the other $c - 1$ variables. Thus, considering these $c - 1$ new variables:

$$Y = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_k X_{ik} + \alpha_1 z_{i1} + \alpha_2 z_{i2} + \cdots + \alpha_{c-1} z_{i,c-1} + \varepsilon_i \quad i = 1, 2, \ldots, n.$$

- *Logistic Regression*. This method is applied when the dependent variable Y is dichotomous. We suppose a dependent dichotomic variable Y from the sample X, with $(X_1, \ldots, X_p) = (x_1, \ldots, x_p) = \underline{X}_i$. The random variable Y is Bernoulli whose probability parameter (its mean) is given by a function $\mu(\underline{X}_i)$. So, the mean and variance of Y depend on the value of the vector of predictors. $E[Y|X = \underline{X}_i] = \mu(\underline{X}_i)$ $V[Y|X = \underline{X}_i] = \mu(\underline{X}_i)[1 - \mu(\underline{X}_i)]$. One of the most used models to express this relationship is given by the logistic function: $\mu(\underline{X}_i, \beta) = P[Y = 1|\underline{X}_i] = \frac{e^{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}}}{1 + e^{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}}}$.
- *Multinomial Logistic Regression*. This method is a generalization or extension of the previous, which is used when the dependent variable Y is a categorical variable with more than two categories. Assuming $Y \equiv 1, \ldots, K$, the log odds ratio between categories k and K (base category) is defined as $\theta(k|\underline{X}_i) = \log \frac{P[Y=k|\underline{X}_i]}{P[Y=K|\underline{X}_i]}$ $k = 1, \ldots, K$. This model assumes $\theta(k|\underline{X}_i) = \beta_{k0} + \beta_{k1} x_{i1} + \cdots + \beta_{kp} x_{ip}$. Thus, $P[Y = k|\underline{X}_i] = \frac{e^{\theta(k|\underline{X}_i)}}{e^{\theta(1|\underline{X}_i) + \cdots + \theta(K|\underline{X}_i)}}$.
- *Hot-deck*: Hot-deck imputation estimates missing values on incomplete records using values from similar but complete records of the same data set. The nearest neighbour technique 1-NN and the Gower's general similarity coefficient (Gower, 1971) for measuring the proximity between records for mixed data types have been used.
  This is, the proximity between the receptor records and the potential donor records from the complete records set is

calculated by the Gower's general similarity coefficient and then the most similar case to the case with missing values is selected. For the sample X with qualitative and quantitative variables, with $j = 1, \ldots, p$ and G representing the Gower's similarity function, this method allows to estimate the missing value x_{ij} from the value of the variable X_j for the record from the complete records set which make maximum the Gower's general similarity coefficient with the record to impute:

$$\hat{x}_{ij} = x_{ij} | G(\underline{X}_i, \underline{X}_t) = \max_{\underline{X}_\gamma \in X_j^{(obs)}} G(\underline{X}_i, \underline{X}_\gamma).$$

Gower's general similarity coefficient s_{ij}, for a sample with n records and p variables, is defined as follows:

$$s_{ij} = \frac{\sum_{k=1}^{p} w_{ijk} s_{ijk}}{\sum_{k=1}^{p} w_{ijk}}$$

w_{ijk} denotes the number of variables which have observed values for both records:

$$w_{ijk} = \begin{cases} 1 & \text{if } X_k \text{ is known in } i, j \\ 0 & \text{otherwise} \end{cases}$$

s_{ijk} denotes the contribution provided by the kth variable, distinguishing between different types of data.
For ordinal and continuous variables the value s_{ijk} is defined as:

$$s_{ijk} = 1 - \frac{|x_{ik} - s_{jk}|}{r_k}$$

where r_k denotes the range of values for the kth variable and x_{ik} is the value of the record i for variable k.
For nominal variables the value s_{ijk} is equal to 1 if both records i and j have the same category for the variable k and it is equal to 0 otherwise:

$$s_{ijk} = \begin{cases} 1 & \text{if } x_{ik} = x_{jk} \\ 0 & \text{if } x_{ik} \neq x_{jk}. \end{cases}$$

Missing data can appear by different mechanisms or with different patterns. Little and Rubin (1987) and Rubin (1976) define three types of missing data mechanisms.

- MCAR (missing completely at random). For any individual, the probability that the value of a variable X_j is observed or missing, does not depend on any variable:

$$P[X_j = mis \mid X_1 \ldots X_p] = P[X_j = mis]. \quad (1)$$

- MAR (missing at random). For any individual, the probability that the value of a variable X_j is observed does not depend on that variable, but depends on the value of the other variables:

$$P[X_j = mis \mid X_1 \cdots X_p] = P[X_j = mis \mid X_1 \cdots X_{j-1} X_{j+1} \cdots X_p]. \quad (2)$$

- NMAR (not missing at random). For any individual, the probability that the value of a variable X_j is observed, depends on the value of that variable, being this value unknown.

Two patterns of missing data are usually distinguished: a monotone pattern, where a lack of response for the same records and variables is observed, and non-monotone pattern, where any record and variable can present a missing value. Both patterns have been considered in our study. We have introduced a MCAR mechanism in our comparison for the non-monotone pattern, while the monotone pattern has been defined by a set of randomly generated variables and records.

3. Artificial neural networks

We have considered a three layered perceptron (MLP) with the hyperbolic tangent activation function $g(u) = (e^u - e^{-u})/(e^u + e^{-u})$ in the hidden layer and the identity function as the activation function for the output layer. Denoting by H the size of the hidden layer, $\{v_{ih}, i = 0, 1, 2, \ldots, p, h = 1, 2, \ldots, H\}$, the synaptic weights for the connections between the p-sized input and the hidden layer, $\{w_{hj}, h = 0, 1, 2, \ldots, H, j = 1, 2, \ldots, q\}$, the synaptic weights for the connections between the hidden and the q-sized output layer, the outputs o_j of the neural network for p inputs $x_1, \ldots, x_p$ are

$$o_j = w_{0j} + \sum_{h=1}^{H} w_{hj} g\left(v_{0h} + \sum_{i=1}^{p} v_{ih} x_i\right), \quad j = 1, 2, \ldots, q. \quad (3)$$

In our data imputation problem each categorical variable must be codified by a vector formed by dummy variables 0–1, one for each class, so the number of inputs p is usually larger than the number of variables in the data file. The number of outputs of the MLP is equal to the number of inputs, and thus the size of the network is (p, H, p). For obtaining the imputed value of a categorical variable, the largest predicted dummy variable provides the associated category as the prediction.

The network learns by modifying the values of the synaptic weights in a supervised scheme. Given a training data set, examples of both input and output values, are repeatedly presented to the network, and thus the weights are adapted looking for the maximum possible similarity between the network responses and the actual output values. However, there are many available learning rules for the multilayer perceptron, but there is not any known procedure assuring us to obtain a global solution, and usually one of the many possible local minima is obtained at the most. Thus, we have considered several learning algorithms, as it is described in next section.

4. Empirical experiments

In most articles that appear in the literature on imputation process with ANN (Nordbotten, 1995, 1996; Norway, 1997; Sonnberger & Maine, 2000), the training process is carried out with the observed data and step by step for each single variable. That is, for each variable, networks are trained on records for which the target value is not missing, so the generated network is applied for imputing missing values. In works such as Koikkalainen (2002) and Norway (1997), they perform, in the training phase of the network, with the entire data set. In others such as Nordbotten (1998) and Laaksonen (1999), they perform the process on a single variable, so they work with a single output network.

Among these previous works, similar to that presented here, distinguish those carried out within the Euredit (2005) project, whose results were never published in their entirety, but they are available at the project website.

Our work has several and important differences with Euredit, as it is described in next paragraphs.

In Euredit (2005), a subset without missing values for the target variable is used for the neural network training, setting the variable to be imputed as target variable. They perform the following steps:

1. Setting the variable to be imputed as target variable.
2. The training data set is split into two subsets: training and test subsets.
3. The training subset, containing the observed data, is used to train the networks.
4. The test subset, containing missing values, is used to measure the efficiency of the network. The predicted values are compared with the true values.
5. The best trained network is chosen.
6. This generated network is applied to records with missing values.

However, our study has not been performed neither step by step for each single variable nor on observed data. The networks have been trained on all the records, both observed data and missing data. All variables are input variables and at the same time output variables. In this way, the number of neurons in the output layer of the MLP is equal to the number of neurons in the input layer. Therefore, the training process is carried out only once, while, in the Euredit project, it is performed and repeated for each variable defining it as target variable.

1. All variables are target variables.
2. The training data set is split into two subsets: training and test subsets.
3. The training subset, containing both observed and missing data, is used to train the networks.
4. The test subset, containing both observed and missing values, is used to measure the efficiency of the network. The predicted values are compared with the true values.
5. Networks are compared with other imputation methods, which use the same inputs (all variables) with observed and missing values.

In this paper are described two studies. In the first, several architectures for the multilayer perceptron have been performed, so results can be obtained taking into account different parameters: number of hidden neurons, number of the training epochs and training algorithms. The second has the main purpose of verifying the robustness of artificial neural networks for the imputation process in the presence of errors.

We use a supervised learning scheme, where the complete and correct data set contains the target variables, while the perturbed data set provides the inputs to the networks, as it is explained in Sections 4.2 and 4.3. Shortly, a correct data set is perturbed, obtaining a new version with missing values. The network tries to learn the correct values from the perturbed data set.

To sum up, in this study, the network training is conducted directly with the perturbed data. The model works on inputs and outputs with the same number of variables and the error rate is large. This complicates the computing process of the networks. So, the added difficulty of the number of neurons at the input and at the output should also be considered, which depends on the variable type. For a quantitative variable the neuron is only one, but for a qualitative variable the neurons are as many as the number of categories of the variable, which may influence the quality of results obtained by the network.

4.1. Data preprocessing

For each data set, several preprocessing tasks have been performed. Firstly, all variables that are constant across the data file are deleted. Secondly, the quantitative variables are normalized, computing for the quantitative variable X_i the following value for the record j:

$$z_{ij} = \frac{x_{ij} - x_{i,\min}}{x_{i,\max} - x_{i,\min}}. \tag{4}$$

Thirdly, the categorical variables are codified as we have described in the next section. Besides these transformations, each data set is randomly split into training (70%) and test (30%) sets to obtain reliable measures of the ANN models performance.

4.2. Data perturbation

We assume that the data set is complete and correct, so a perturbed data file is obtained by introducing random errors of non-response. Thus, a disturbed variable Y^d has been defined for each original variable Y. For the non-monotone pattern, we have fixed 0.05 as the probability of error. This error rate can be considered high, for example for a data set with 10 variables, approximately 40% of the perturbed data set records are incorrect. While the number of records that may have errors is not limited, we have restricted the number of missing values in a record to not more than half of the variables. In particular, the missing value mechanism is assumed to be missing completely at random (MCAR). As far as the quantitative variables have been normalized, all values of these variables are between 0 and 1, and therefore the non-response can be reflected by assigning for example the value -1 to the disturbed variable.

$$P[Y^d = -1] = 0.05 \wedge P[Y^d = Y] = 0.95. \tag{5}$$

For the monotone pattern, a set of randomly generated variables and records are set to missing value as it is more deeply described in the next section.

We have tried to use a realistic probability of missing value. Thus, we have analyzed several data sets that we have obtained from surveys and the majority of missing value rates was not greater than 5%. Nonetheless, some preliminary studies with 1%, 5% and 10% did not reveal differences between the models.

4.3. Scheme of the experiments

Given a correct data set T, the previous perturbations are performed, and therefore an associated perturbed data set T^d is also obtained. Specifically, in the experiments following a non-monotone pattern, T is randomly split into a training (70%) set T^1 and a test (30%) set T^2. Their associated perturbed versions T^{d1} and T^{d2} are defined by the same records respectively appearing in T^1 and T^2. The inputs to the MLP training are defined by the rows of T^{d1}, while the rows of T^1 are the target records. As it is explained below, $M = 357$ different parameter configurations for the MLP are considered, and a 10-fold validation procedure is followed to select the best configuration. The rows of T^{d2} are fed to the fitted MLP and the output records of the MLP, contained in T^{*2}, are compared with the true records contained in T^2. To avoid that the obtained results depend simply on the performance of a single imputation process, 15 different perturbed data sets have been generated for each data set and the imputation process has been repeated for each one. Fig. 1 displays the whole procedure for a fixed data set T.

The process for the data imputation experiments following a monotonous pattern is shown in Fig. 2. Each data set T is split into training (70%) and test (30%) sets, obtaining two files, T^1 and T^2. A 30% of variables are randomly selected, let X_d be the resulting selection, and a perturbed version T^{d2} of T^2 is defined by setting to missing all the values of all the variables in X_d. Denoting by X_o the remaining variables not included in X_d, a multilayer perceptron to predict X_d from X_o can be fitted on T^1. The rows of T^{d2} are fed to the fitted MLP and the output records of the MLP, contained in T^{*2}, are compared with the true records contained in T^2. As it is explained below, $M = 357$ different parameter configurations for the MLP are considered, so a 10-fold validation procedure is followed to select the best configuration. In the same way, to avoid that the obtained results depend simply on the performance of a single imputation process, this same perturbation pattern is repeated 50 times, thus 50 perturbed files are obtained for each data set and the imputation process has been repeated for each one. Fig. 2 displays the whole procedure for a fixed data set T.

As mentioned earlier, other related works are more limited than our approach. For example, in some papers the MLP is trained only over complete or almost complete data set (Euredit, 2005; Koikkalainen, 2002; Norway, 1997; Yoon & Lee, 1999).

Other available findings have been obtained for only one variable (Laaksonen, 1999; Nordbotten, 1998; Norway, 1997). And other developments were conducted with data sets formed

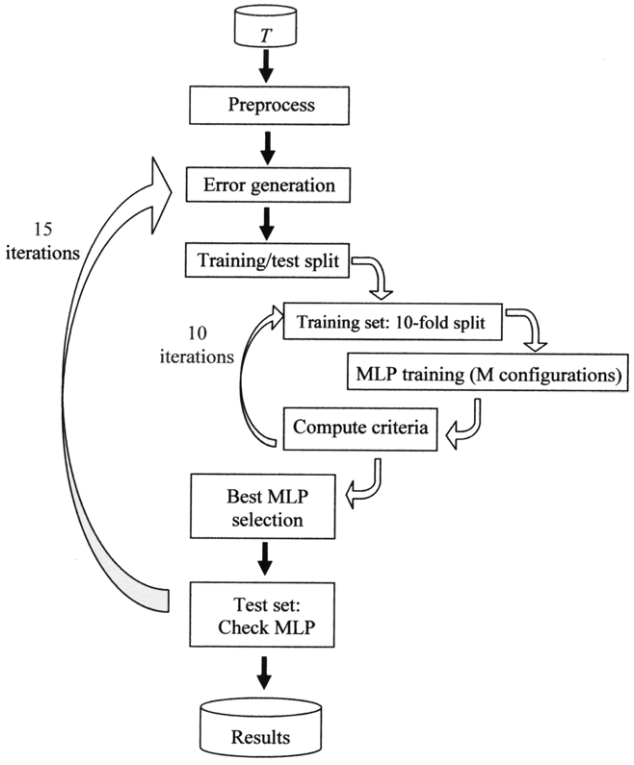

Fig. 1. Strategy used in the process of imputation following a non-monotone pattern for each data set T.

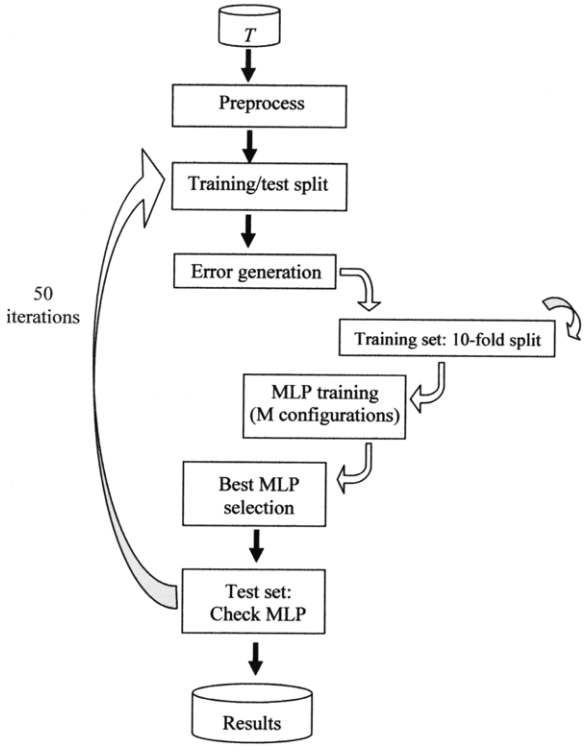

Fig. 2. Strategy used in the process of imputation following a monotone pattern for each data set T.

by either numeric or categorical variables (Laaksonen, 2002; Nordbotten, 1995).

The MLP fitting requires to take several decisions such as the learning algorithm, the random initialization of the MLP weights, the number of hidden units or the number of iterations (epochs) of the learning algorithm. Thus, different alternatives are considered, as it is discussed in the following.

Table 1
Table of learning algorithms.

Abbr.	Algorithm
GD	Gradient descent
GDM	Gradient descent with momentum
BA	Gradient descent with adaptive learning rate
BE	Resilient backpropagation (RProp)
GC	Conjugate gradient Fletcher–Reeves update
QN	BFGS quasi-Newton
LM	Levenberg–Marquardt

We have considered seven learning algorithms, as it is shown in Table 1. These seven learning algorithms are representative of the main learning algorithms.

It is well known that the random initial configuration of weights may lead to very different solutions. Thus, to reduce the associated uncertainty, each training algorithm was run 5 times, from 5 random initial weights vectors, and the minimum mean squared error of the network was selected. This is done for each one of the 15 perturbed data sets obtained according to previously described process.

Three sizes of hidden layer have been considered: 5, 10 and 15 nodes. The selection has been done from previous studies ad hoc for these experiments, where the use of larger sizes frequently led to worse results.

The same occurs with the number of iterations in the training process. It was also observed that for epochs 300 and above, the error decreased very slowly. So, the number of training epochs has been studied for: 5, 10, 15, 20, 25, 30, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275 and 300.

The total number of weights for each MLP architecture of imputation following non-monotone patterns is $(2p + 1)H + p$, where p is the number of inputs and outputs and H is the number of hidden units. For example, for the Soybean database with 35 qualitative attributes, $p = 95$ after they are codified with dummy variables. For $H = 15$ the ANN model comprises 2960 weights.

The total number of different MLP architectures which have been studied for each data set is $15 \times 10 \times 7 \times 3 \times 17 \times 5 = 267.750$, that is, 15 perturbed data sets, 10-fold validation splits, 7 learning algorithms, 3 sizes for the hidden layer, 17 values for the number of epochs and 5 different initial weight sets. $M = 7 \times 3 \times 17 = 357$ different MLP configurations are generated, and for each one of these configurations, 10 values of the GCD criterion (explained in Section 4.5) are available through the 10-fold procedure, and their mean value is computed. The minimum mean GCD guides us to the selected configuration for each perturbed data set.

For monotone patterns the total number of different MLP architectures which have been studied for each database is: $50 \times 10 \times 7 \times 3 \times 17 \times 5 = 892.500$, that is, 50 perturbed data sets, 10-fold validation splits, 7 training algorithms, 3 sizes of hidden layers, 17 values for the epochs and 5 different initial weight sets. As in the non-monotone case, $M = 7 \times 3 \times 17 = 357$ different MLP configurations are generated. For each one of these configurations, 10 values of the GCD criterion (explained in Section 4.5) are available through the 10-fold procedure, and their mean value is also computed. The minimum mean GCD guides us to the selected configuration.

As previously mentioned, one of the objectives of this study is to test the influence of the parameters in the model. So a huge number of architectures have been studied, which involves working with a large number of network weights, what has entailed a great computational effort.

The source code employed in this work for all different imputation methods has been written by the authors in Matlab 6.0, being used Neural Network Toolbox (Demuth and Beale, 1997) for the construction of the MLPs.

Table 2
Data sets used in the data imputation experiments.

Name	Size	Inputs/outputs	NQV	NCV
Cleveland	303	25	6	7
Heart	270	25	6	7
Zoo	101	31	1	15
Buhl1-300	300	11	11	0
Glass	214	9	9	0
Ionosphere	351	34	34	0
Iris	150	4	4	0
Pima	768	8	8	0
Sonar	208	60	60	0
WaveForm21	500	21	21	0
Wine	178	13	13	0
Hayes-Roth	132	15	0	4
Led7	500	14	0	7
Solar	1389	16	0	7
Soybean	47	95	0	35

4.4. Data sets

Our empirical study has been conducted on fourteen data sets. A brief description of each data set is shown in Table 2. These data sets have been selected to cover several domains such as social surveys, business, census, biology, medicine, chemistry and electronic, and they also include different types of variables: quantitative, ordinal categorical variables and nominal categorical variables.

The majority of the data sets are frequently used in the scientific community. They are available in the UCI Machine Learning Repository (Asuncion & Newman, 2007), except the simulated data set Buhl1.

The synthetic data set Buhl1 (Bühlmann, 2003), has been generated for a sample size $n = 300$, being DU the uniform discrete distribution, as follows:

$$X = (X^{(1)}, \ldots, X^{(10)}) \sim DU([0, 1]^{10}). \tag{6}$$

Table 2 contains the following columns. Name: name of the data set; Size: number of records; Inputs/outputs: total number of inputs and outputs of the MLP model (this value includes the added auxiliary variables for the qualitative attributes); NQV: number of quantitative variables; NCV: number of categorical variables.

Since our objective is not to perform classification, one or more attributes have been removed in all data sets, which corresponds to the class or classes that are used for classification. The database *Cleveland* is one processed data set contained in the *Heart Disease*, which have 76 attributes, but only 14 of them are actually used.

4.5. Evaluation criteria

To evaluate the MLP data imputation model several measures have been computed for the previously presented data sets, depending on the type of variable. For quantitative variables, the performance of the data imputation model is measured computing the average of the squared linear correlation coefficient R^2, expressed as a percentage between 0 and 100%. For each variable where n values have been set to a missing state, its R^2 is computed measuring the association between its n real values and its n imputed values.

For qualitative variables other measures are more appropriate. As a first step, Euredit (2005) suggests a Wald-type statistic W to analyze if a data imputation process preserves the marginal distribution of a qualitative variable with k categories:

$$W = (Q - S)^t [diag(Q + S) - M - M^t]^{-1} (Q - S). \tag{7}$$

Q is the $k - 1$ vector of counts for the first $k - 1$ categories of the imputed variable, S is the $k-1$ vector of actual counts for these categories and M is the square matrix of order $k-1$ corresponding to the cross classification of actual vs. imputed counts, all of them computed over a test set where the actual counts are known.

Under weak assumptions the large sample size distribution of W is chi-square with $k - 1$ degrees of freedom, and therefore a statistical test may be carried out for each categorical variable, where the p value is defined as the right tail probability of a chi-square distribution with $k - 1$ degrees of freedom computed for the observed value W.

If the hypothesis of marginal distribution preservation is accepted, a second step would be to assess the preservation of the true values of the categorical variable. Euredit (2005) proposes a method based on the statistic D defined by the error rate, that is, the incorrect imputation rate, which is computed as one minus the proportion of records where the true value is equal to the imputed value, for a fixed categorical variable. When the marginal distribution is preserved, the variance of D can be estimated by $V(D) = (1 - D)/n$, n being the number of records, allowing us to build a rule to decide if D is significantly greater than 0, based on $\varepsilon = \max\{0, D - 2\sqrt{V(D)}\}$, whose value is suggested in Euredit (2005). If $\varepsilon = 0$, there is no statistical evidence supporting the alternative hypothesis that $D > 0$, so we accept that the true values of the categorical variable are preserved, otherwise $D > \varepsilon + 2\sqrt{V(D)}$ and then the true values of the categorical variable are not accepted to be preserved.

In the case of an ordinal categorical variable, the second step studies the preservation of the true value order (Euredit, 2005). The statistic for an ordinal variable Y is now:

$$D = \frac{1}{n} \sum_{i=1}^{n} d(\hat{Y}_i, Y_i) \tag{8}$$

where d is a distance between categories of Y, suggested in Euredit (2005) as:

$$d(\hat{Y}_i, Y_i) = \frac{1}{2} \left[\frac{|\hat{Y}_i, Y_i| M^t(\hat{Y}_i, Y_i)}{\max(Y) - \min(Y)} + M^t(\hat{Y}_i, Y_i) \right]. \tag{9}$$

When the order of the true values are preserved, an estimation of $V(D)$ is again provided by $(1 - D)/n$, and therefore the previous rule can also be applied. We have computed an overall measure based on these statistics, the coefficient of preservation PR. It is defined as the categorical variable percentage of a data file which preserves the marginal distribution and the true values (for nominal variables) or the true value order (for ordinal variables):

$$PR = \frac{NCVN + NCVO}{NCV}. \tag{10}$$

Let $NCVN$ be the number of nominal categorical variables which preserve the marginal distribution and true values. We denote by $NCVO$ the number of ordinal categorical variables which preserve the marginal distribution and true value order, and NCV is the number of qualitative variables.

A global criterion of the whole data imputation process (GCD) has been computed as the mean of R^2 (when available) and PR (when available).

GCD is the measure we have used in the 10-fold validation procedure to select the best configuration of learning algorithm, number of hidden units and number of epochs, for each perturbed data set.

5. Results

In this section is shown the comparison of the different methods for missing data imputation, classic imputation procedures such as mean/mode, regression and hot-deck, and ANN-based model, on the different data sets.

To verify the accuracy prediction of the methods, it is important, in order to have comparable results, to use the same sequence of inputs. All the methods, in the same empirical series, are trained and tested with the same values. This allows to carry out more

Table 3
Mean test GCD, non-monotone patterns.

DB	Mean/mode	Regression	Hot-deck	ANN
Cleveland	70.2	63.6	77.7	**90.1**
Heart	73.9	72.4	80.0	**96.7**
Zoo	83.7	79.0	93.4	**97.1**
Buhl1-300	**96.7**	96.3	93.5	96.3
Glass	99.3	99.4	99.2	**99.9**
Ionosphere	96.9	94.7	97.7	**98.2**
Iris	96.4	**97.9**	97.3	94.3
Pima	98.9	**99.0**	98.6	98.4
Sonar	98.1	95.5	**98.7**	98.6
WaveForm21	98.6	**99.1**	98.5	98.9
Wine	98.5	**99.0**	98.7	98.5
Hayes-Roth	32.6	50.7	44.7	**80.0**
Led7	71.5	29.1	78.4	**90.7**
Solar	54.5	42.0	54.0	**57.5**
Soybean	63.3	62.0	**83.2**	82.2

Table 4
Mean test GCD, monotone patterns.

DB	Mean/mode	Regression	Hot-deck	ANN
Cleveland	71.1	69.8	75.3	**89.8**
Heart	67.0	67.4	78.2	**87.1**
Zoo	81.4	73.1	92.2	**99.2**
Buhl1-300	**94.8**	94.6	89.4	94.7
Glass	98.8	**99.4**	99.3	99.1
Ionosphere	95.1	**96.6**	96.3	96.5
Iris	98.3	**99.5**	99.3	**99.5**
Pima	99.5	**99.6**	99.3	99.5
Sonar	97.6	**99.1**	98.5	98.5
WaveForm21	96.6	**98.1**	96.5	97.8
Wine	96.3	**98.2**	97.2	97.9
Hayes-Roth	27.7	69.0	29.2	**100.0**
Led7	34.5	17.2	58.8	**83.3**
Solar	92.5	84.0	90.9	**93.5**
Soybean	63.3	61.2	80.5	**90.0**

rigorous comparisons between different imputation models. So, when a method shows less error than others, it is not because it worked on a set of data easier to impute, but because that model is actually able to impute more cases correctly than others, since all are in identical situations, trained and tested, with the same examples.

Table 3 shows the mean of the global criterion GCD, computed over the 15 randomly generated test sets, for every data imputation model and for each one of the 15 data sets, considering a non-monotone pattern scheme. Table 4 shows these same measures, computed over the 50 randomly generated test sets, for every data imputation model and for each one of the 15 data sets, considering a monotone pattern scheme.

From Table 3 we can observe high values for GCD for all models in general. MLP offers the highest GCD values in six of the seven data sets including categorical variables, although in the other data set, Soybean, MLP is the second option with values very similar to the preferred hot-deck model. Regression models offer the lowest values in six of these seven data sets.

The four models provide similar GCD values in the eight data sets with only quantitative variables. From Table 3 we can see MLP offers the highest values in two of these data sets, in four of them regression models show the best value, hot-deck is the most appropriate in one of them, and mean/mode is the best method in only one data set.

MLP is the best model in eight of the fifteen data sets, regression models are the preferred technique in four data sets, hot-deck would be selected in two data sets, while the simple mean/mode algorithm is the best model in only one data set.

These results suggest there are no differences between the four data imputation models for data sets with only quantitative variables. However, for data sets with categorical variables, MLP tends to provide better results in comparison with other procedures.

Table 4 shows that the previously observed trend is more solid when a monotone pattern scheme is considered: MLP offers the highest GDC values for all the seven data sets with categorical variables.

However, the data sets with only quantitative variables do not show a so clear winner. Although regression models offer the highest GCD value in seven of these eight data sets, the four models provide very similar results, without a clear difference.

Therefore, we can conclude that for data sets with only quantitative variables any of the presented models in this study produce good and similar results, while for data sets with categorical variables the best results are clearly obtained with the MLP model.

Table 5
Mean test values of the criteria for the ANN model, non-monotone patterns.

Data set	MSE	R^2	*PR*	GCD
Cleveland	0.04	98.9	81.3	90.1
Heart	0.03	99.0	94.3	96.7
Zoo	0.03	98.7	95.4	97.1
Buhl1-300	0.10	96.3	.	96.3
Glass	0.03	98.9	.	99.9
Ionosphere	0.04	98.2	.	98.2
Iris	0.15	94.3	.	94.3
Pima	0.03	98.4	.	98.4
Sonar	0.03	98.6	.	98.6
WaveForm21	0.02	98.9	.	98.9
Wine	0.03	98.5	.	98.5
Hayes-Roth	.	.	80.0	80.0
Led7	.	.	90.7	90.7
Solar	.	.	57.5	57.5
Soybean	.	.	82.2	82.2

Table 6
Distribution (percentages) of the 10-fold selected parameters, non-monotone patterns.

Learning training		Hidden layer size		Number of epochs	
GD	6.7	5	93.3	15	6.7
BA	6.7	15	6.7	25	13.0
BE	7.2			30	6.9
GC	17.9			125	13.3
QN	27.7			175	7.1
LM	33.8			200	13.3
				225	6.8
				250	26.2
				275	6.7

Table 5 contains the mean test values for each criterion for the MLP model in the non-monotone setting.

From Table 5 we can observe low values for MSE, while the values of the criterion R^2 tend to be very high for all the data sets including quantitative variables. The mean value of the preservation measure *PR* is greater than 80% in six of the seven data sets with categorical variables. The last column of this table shows that the global criterion GCD is greater than 80% for almost all the data sets.

Table 6 contains the distribution of the 10-fold selected learning algorithms, hidden layer size and number of epochs, computed as percentages over the $210 = 15 \times 14$ disturbed data sets for the non-monotone pattern.

Table 6 suggests that LM tends to provide the preferred learning algorithm, being one gradient descent variant, namely QN, the

Table 7
Mean test values of the criteria for the ANN model, monotone patterns.

Data set	MSE	R^2	*PR*	GCD
Cleveland	0.03	98.6	81.0	89.8
Heart	0.04	99.1	75.0	87.1
Zoo	0.02	98.3	100.0	99.2
Buhl1-300	0.08	94.7	.	94.7
Glass	0.03	99.1	.	99.1
Ionosphere	0.05	96.5	.	96.5
Iris	0.03	99.5	.	99.5
Pima	0.02	99.5	.	99.5
Sonar	0.03	98.5	.	98.5
WaveForm21	0.02	97.8	.	97.8
Wine	0.02	97.8	.	97.8
Hayes-Roth	.	.	100.0	100.0
Led7	.	.	83.3	83.3
Solar	.	.	93.5	93.5
Soybean	.	.	90.0	90.0

Table 8
Distribution (percentages) of the 10-fold selected parameters, monotone patterns.

Learning training		Hidden layer size		Number of epochs	
BA	13.3	5	67.7	5	6.2
BE	6.8	10	32.3	10	6.7
GC	20.1			15	6.3
QN	6.8			20	6.7
LM	53.0			30	12.9
				125	6.7
				150	20.6
				175	13.0
				250	14.0
				300	6.9

main alternative. The learning algorithm GDM is never selected. As regards the number of hidden nodes, in general 5 hidden units are sufficient. The number of epochs appears to be more variable, without a clear conclusion.

Table 7 contains the mean test values for each criterion for the MLP model in the monotone setting. From this table we can observe low values for MSE, while the values of the criterion R^2 tend to be very high for all the data sets. The mean value of the preservation measure *PR* is greater than 75% in the seven data sets with categorical variables, reaching in two of them the 100%. The last column of this table shows that the global criterion GCD is greater than 80% for all the data sets.

Table 8 contains the distribution of the 10-fold selected learning algorithms, hidden layer size and number of epochs, computed as percentages over the $700 = 50 \times 14$ disturbed data sets for the monotone pattern. This table suggests that LM tends to provide the preferred learning algorithm, being two gradient descent variants, BA and GC, the main alternatives. Two learning algorithms, GD and GDM, are never selected. As in the non-monotone setting, 5 hidden units are sufficient. The number of epochs is usually between 100 to 250 epochs, and we have observed that it is generally lower for data sets with quantitative variables.

Another procedure to evaluate the performance of an imputation process is to fit a prediction model on the imputed data set, comparing the results with those obtained on the correct data set. We now describe a study about the employment of a variant of *k*-NN method to the task of classification on imputed databases with an ANN-based method. These proposed methods are called ε-ball$^{k\text{-NN}}$heur (Laguía & Castro, 2008).

The tests with this classifier have been performed using the distribution of 10-fold cross-validation with each one of the databases. Each one of 10 obtained training partitions is in turn performed, other 10-fold for estimating the parameters ε and k which control the running of ε-ball$^{k\text{-NN}}$heur.

Table 9
Results obtained with ε-ball$^{k\text{-NN}}$heur.

Names	Original data (%)	Imputed data (%)
Zoo	97.03	92.10
Hayes-Roth	77.27	75.36
Led7	72.80	70.02
Monk	92.34	90.18

Table 10
Mean test GCD, non-monotone patterns.

DB	Kernel	ε	C	GCD
Heart	RBF	0.40	0.13	92.2
Zoo	RBF	0.40	0.13	96.9
Glass	SIG	0.20	0.50	99.9
Iris	SIG	0.02	0.03	99.5
Wine	SIG	0.02	0.03	91.4
Hayes-Roth	RBF	0.02	0.13	49.5
Soybean	RBF	0.42	0.50	82.1

A short extract of some data sets is shown in Table 9. The first column shows the obtained results by the classifier on original databases, while the second column shows the mean of the obtained values by the classifier for each database. The difference between both columns shows the grade in which the imputation process is able to recover errors. The obtained precision with imputed databases is 5% lower than the obtained values with original databases. In some databases it is lower than 2%. In general, the correct records are unaltered and a high percentage of errors are corrected. According to the literature, it is observed that the obtained success rate is similar to the success rate offered by traditional methods.

Support Vector Machines are a powerful learning machine model which have also been used for imputation; for example Honghai, Guoshun, Cheng, Bingru, and Yumei (2005) and Mallinson and Gammerman (2005). However, a different model for each variable to be imputed must be fitted. In Honghai et al. (2005) the superiority of the performance of SVMs on statistical methods such as mean or median is demonstrated for a given data set (SARS). Mallinson and Gammerman (2005) studied the behavior of Support Vector Machines for data imputation, obtaining satisfactory results. The main limitation of these examples is that, in both cases, complete training sets are used, which is an unrealistic condition.

We are also performing a comparison with other data mining models, such as SVM techniques. But we used a training set that contains missing data, so we simulate a more realistic situation. Some preliminary results of the experiments to evaluate the performance of SVM on our data imputation process are shown in Table 10. Nonlinear SVM with two kernels functions, Radial Basis (RBF) and Sigmoid (SIG), were considered, using their implementation in MATLAB. A perturbation experiment similar to the multilayer perceptron case was performed, and for each perturbed data set, a grid search for their parameters ε and C was conducted, also selecting the best configuration through a 10-fold cross-validation process based on the GCD criterion for non-monotone patterns. The mean values of ε and C are shown in Table 10, which suggests small values for C. RBF was always selected for data basis with categorical variables, while SIG was the resulting selection in data basis with only quantitative variables. Tables 3 and 10 suggest that for categorical variables MLP is still preferable, while for quantitative variables SVM offers similar results to the other methods.

6. Conclusions

A methodology for data imputation by means of artificial neural networks has been proposed and empirically compared with three classic methods: mean/mode imputation, regression models

and hot-deck. Fifteen real and simulated data sets have been exposed to a perturbation experiment, and several architectures and training algorithms for the multilayer perceptron have been tested. Several criteria for evaluating the imputation of non-response have been computed, and both the monotone and the non-monotone patterns have been followed to generate missing values. Moreover, the empirical study reported in this paper establishes alternative works to those presented by Euredit, and the clear differences between them have been set.

Experimental results show that for data sets with only quantitative variables any of the analyzed models provides good and similar results, although the computational cost of MLP could make the other methods preferable. However, for data sets with categorical variables the best results are clearly obtained with the MLP model. Levenberg–Marquardt learning rule tends to be selected, while simple architectures (only five hidden units) are usually preferred.

A brief study of the performance of classification models fitted on the imputed data sets reveals success rates that are close to those obtained on the correct data sets.

The proposed ANN-based imputation model offers not only automatic imputation, but also success results. A huge number of architectures have been realized with the aim to fit values of parameters, so this study has required a high computational cost. It could be thought that the classic methods are computationally less demanding, but we wish to emphasise that the ANN-based model only requires training time. Once it is trained this model is completely feasible and can be performed by anybody on any computer. Moreover, a framework is provided which serves as a starting point for further research in this issue.

Future works could include other ANN models, or different data mining techniques, and new evaluation criteria. We are nowadays analyzing Multiple Imputation schemes, based on ANN models. This approach can provide an effective improvement, particularly for quantitative variables in monotone patterns.

Another future research topic would be the treatment of missing values in data sets varying in time, for example panel surveys or econometric time series. Another important topic is the estate estimation problem for nonlinear stochastic time-delay systems with missing measurements. Suitable adaptations of our approach could be devised and compared with the existing solutions as those proposed in Liang, Wang, and Liu (2009) and Wang, Ho, Liu, and Liu (2009).

Acknowledgement

This work was partially funded by Institute of Statistics of Andalusia (Spain), grant 2007/00001428.

References

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

Bishop, C. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press.

Bühlmann, P. (2003). Bagging, subagging and bragging for improving some prediction algorithms. *Tech. rep. 113, Seminar Für Statistik*, Eidgenössische Technische Hochschule (ETH), Zürich, Switzerland.

Demuth, H., & Beale, M. (1997). Neural Network TOOLBOX for Use with Matlab. User's Guide. http://www.mathworks.com Edition. The Math Works Inc.

Euredit, (2005). Interim report on evaluation criteria for statistical editing and imputation. http://www.cs.york.ac.uk/euredit.

Gower, J. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, *27*(4), 857–871.

Honghai, F., Guoshun, C., Cheng, Y., Bingru, Y., & Yumei, C. (2005). A svm regression based approach to filling in missing values. *Knowledge-Based Intelligent Information and Engineering Systems*, *3683*, 581–587.

Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J., & Kolehmainen, M. (2004). Methods for imputation of missing values in air quality data sets. *Atmospheric Environment*, *38*(18), 2895–2907.

Koikkalainen, P. (2002). Neural networks for editing and imputation. In *Dataclean 2002 conference*.

Laaksonen, S. (1999). How to find the best imputation technique? Tests with various methods. In *International conference on survey nonresponse*.

Laaksonen, S. (2002). Traditional and new techniques for imputation. *Journal of Statistics in Transition*, *5*(6), 1013–1035.

Laguía, M., & Castro, J. (2008). Local distance-based classification. *Knowledge-Based Systems*, *21*(7), 692–703.

Liang, J., Wang, Z., & Liu, X. (2009). State estimation for coupled uncertain stochastic networks with missing measurements and time-varying delays: the discrete-time case. *IEEE Transactions on Neural Networks*, *20*(5), 781–793.

Little, R., & Rubin, D. (1987). *Statistical analysis with missing data*. New York: John Wiley and Sons.

Mallinson, H., & Gammerman, A. (2005). Imputation using support vector machines. *Tech. rep*.

Nelwamondo, F., Mohamed, S., & Marwala, T. (2007). Missing data: a comparison of neural network and expectation maximization techniques. *Current Science*, *93*(11), 1514–1521.

Nordbotten, S. (1995). Editing statistical records by neural networks. *Journal of Official Statistics*, *11*(4), 391–411.

Nordbotten, S. (1996). Neural network imputation applied to the norwegian 1990 population census data. *Journal of Official Statistics*, *12*(4), 385–401.

Nordbotten, S. (1998). New methods of editing and imputation. In Institute, W. D. I. S. (Ed.), *Agriculture statistics 2000*. The Haag, Netherlands.

Norway, S. (1997). Data editing with artificial neural networks. In *Working paper* N° 10, *UN/ECE work session on statistical data editing. Conference of european statisticians*.

Ripley, B. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.

Rubin, D. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Sonnberger, H., & Maine, N. (2000). Editing and imputation in Eurostat. In *Working paper* N° 21, *UN/ECE work session on statistical data editing. Conference of european statisticians*.

Ssali, G., & Marwala, T. (2008). Computational intelligence and decision trees for missing data estimation In *International joint conference on neural networks*. Los Alamitos, California, USA.

Wang, Z., Ho, D., Liu, Y., & Liu, X. (2009). Robust *h* [infinity] control for a class of nonlinear discrete time-delay stochastic systems with missing measurements. *Automatica*, *45*(3), 684–691.

Yoon, S., & Lee, S. (1999). Training algorithm with incomplete data for feed-forward neural networks. *Neural Processing Letters*, *10*(3), 171–179.