International Journal of Innovative Computing, Information and Control
Volume **8**, Number **7(A)**, July **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 4705–4717

A NOVEL HYBRID APPROACH TO ESTIMATING MISSING VALUES IN DATABASES USING K-NEAREST NEIGHBORS AND NEURAL NETWORKS

İbrahim Berkan Aydilek and Ahmet Arslan

Department of Computer Engineering
Selçuk University
Konya 42075, Turkey
{ Berkan; Ahmetarslan }@selcuk.edu.tr

Received February 2011; revised July 2011

Abstract. *Missing values in datasets and databases can be estimated via statistics, machine learning and artificial intelligence methods. This paper uses a novel hybrid neural network and weighted nearest neighbors to estimate missing values and provides good results with high performance. In this work, four different characteristic datasets were used and missing values were estimated. Error ratio, correlation coefficient, prediction accuracy were calculated between actual and estimated values and the results were compared with basic neural network-genetic algorithm estimation method.*

Keywords: Missing data, Missing value, K-nearest neighbors, Neural networks, Imputation, Auto-associative neural networks, Genetic algorithm

1. Introduction.

Biological or medical experimentation results, chemical analysis results, meteorology, microarray gene monitoring technology or survey databases or datasets contain missing values. Missing values can occur due to a sensor faults, not reacting experiments, not recovering some situation of work, measuring a faulty value, transferring data to digital system problem or participants skip some survey question [4,5].

Data quality is a major criterion for machine learning, data mining, information and knowledge [13]. Data cleaning aims to produce quality data; thus, missing values are a part of information and knowledge [14]. Deleting, ignoring records containing missing values or imputation mean, mode or the most common value instead of missing value significantly bias datasets and make datasets noisy and low quality [6,16]. Most computational intelligence techniques such as neural networks, support vector machines, and decision trees are predictive models that take discovered data as inputs and estimate an output class. This model fails if one or more inputs are missing. Consequently, they cannot be used for decision-making purposes if the data variables are not complete [15].

2. Literature Review.

In recent years, many applications and methods of missing value estimation have been developed by researchers.

Troyanskaya et al. presented missing value estimation methods for DNA Microarrays with weighted k-NN [1]. Thompson studied the contractive nature of auto-encoders, that is, the practice of missing sensor restoration by using neural network [2]. Qiao et al. also presented a missing data estimator based on a neural network coupled with particle swarm optimization [3] and used it in tracking the dynamics of a plant in the presence of missing sensor measurements. Abdella et al. proposed the use of genetic algorithms and neural networks to approximate missing data in databases [4]. Mohamed et al. offered an approach, which is both agent-based and neural network-based, to estimating missing data in databases [5]. Nelwamondo et al. worked on a comparison of neural network and expectation maximization techniques [7]. Mohamed et al. also proposed hybrid method estimating missing data using neural network techniques, principal component analysis and genetic algorithms [9]. Nelwamondo et al. worked on a dynamic programming approach to missing data estimation using neural networks [10], which is a novel technique for missing data estimation using a combination of dynamic programming, neural networks and genetic algorithms (GA) on suitable subsets of the input data. Blend et al. provided a comparative study of data imputation techniques and their impact [11]. Their paper comprehensively compares auto-associative neural networks, neuro-fuzzy systems and the hybrid combinations methods with hot-deck imputation. Hlalele et al. studied the imputation of missing data using pca, neuro-fuzzy and genetic algorithms [12]. The presented model was tested on the South African HIV sero-prevalence dataset. The results indicated an average increase in accuracy. Patil et al. focused on multiple imputations of missing data with genetic algorithm-based techniques [13]. The proposed imputation algorithm was based on the genetic algorithm, which used a pool of solutions to estimate missing values, and whose fitness function was decision tree classification accuracy. Ramírez et al. also proposed a missing value imputation on missing completely at random data using multilayer perceptrons [14]. Their study focused on a methodological framework for the development of an automated data imputation model based on artificial neural networks. Several architectures and learning algorithms for the multilayer perceptron were tested and compared with three classic imputation procedures: mean/mode imputation, regression and hot-deck. Nelwamondo et al. introduced techniques for handling missing data for applications in online condition monitoring systems, in which missing data are often attributed to sensor failure in online condition monitoring systems. Hybrid genetic algorithms and fast simulated annealing are used to predict the missing values [16].

2.1. **Missing data.** Missing data are classified into three categories [6,7]:

- Missing Completely at Random (MCAR) – The missing value has no relation on any other variable.
- Missing at Random (MAR) – The missing value has relation to other variables. The missing value can be estimated by viewing the other variables.
- Missing Not at Random (MNAT) – The missing value has relation to other missing values and so missing data estimation cannot be performed by using the existing variables.

In this paper we assume that the data is MAR, which implies that the missing values are deducible in some complex manner from the remaining data [7]. In Table 1, we see a section of a dataset with missing values. In this paper, we aim to estimate missing values with artificial neural network and k-nearest neighbors. We will use the notations; 'X1, X2, X3, X4, X5, X6'; 'Y1, Y2, Y3, Y4, Y5'; 'X2, X5, X6' and 'X1, X3, X4' which refer to records; attributes; records without missing values; that is, 'complete' rows; records with missing values; that is, 'incomplete' rows, respectively.

2.2. **Neural networks.** A neural networks (NN) objective acts like human nervous system and target the learning process. NN contain input, hidden and output layers. Neurons construct a layer and every neuron connects the next layer's neurons with weight values. After training the optimization of weight values, a trained neural network (NN) can be accepted as a specialist in the category of trained information it has been given to learn. This specialist system can then be used to provide predictions for new situations. Neural networks (NN) have been widely implemented to pattern recognition, signal processing, time series prediction, non-linear control, identification problems, and so on. The class

Table 1. A subset of dataset with missing values

X/Y	***Y1***	***Y2***	***Y3***	***Y4***	***Y5***
X1	*0,113524*	*0,084785*	*?*	*0,625473*	*0,06385*
X2	*0,112537*	*0,138211*	*0,15942*	*0,625473*	*0,068545*
X3	*0,110563*	*?*	*0,144928*	*0,624212*	*0,083568*
X4	*0,110563*	*0,170732*	*0,146998*	*0,623581*	*?*
X5	*0,108588*	*0,129501*	*0,144928*	*0,624212*	*0,076056*
X6	*0,108588*	*0,082462*	*0,112836*	*0,626103*	*0,015023*

of networks consists of multiple layers of computational units interconnected in a feed-forward way [18]. Neural network classification, which is supervised, has been proved to be a practical approach with lots of success stories in several classification tasks [17].

2.2.1. *Auto-associative neural networks.* An auto-associative Neural Network also known as auto-encoder is a relevant neural network; it is trained to recall its inputs [2,19]. A set of existing inputs, the model in Figure 1 predicts these inputs as outputs which have the equal number of output and input nodes [2,19]. This means that the difference between input and output can be used an optimization problem's fitness function and also can be used in missing data estimation methods [2-5,7,8,10-12,15,19].

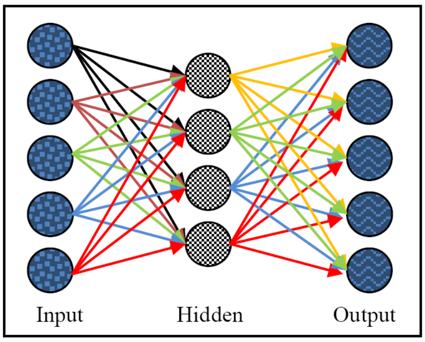


Figure 1. Auto-associative neural networks: Input ≈ Output

2.3. **Genetic algorithm (GA).**

In 1975, John Holland considered the genetic algorithm (GA) as a robust computational model for optimization. The basic concept of GA is choosing better species from the parent generation and randomly interchanging gene information in order to produce a better generation [20]. The aim of GAs is that they mimic the principle of natural selection to guide the search in the problem space so as to find a "better" solution [21]. After several generations, the inappropriate genes are eliminated and more appropriate genes are produced. In general, GA is applicable to a wide range of optimization problems. There are two important issues in searching strategies for optimization problems: exploiting the best solution and exploring the search space. GA makes a balance between the exploitation and exploration of the search space [22]. Therefore, in recent years, GA has been used to estimate missing data problems [23].

Basically GA pseudo code:

```
Step 1: Generate initial population;
Step 2: Calculate fitness function and select the best population;
Step 3: Generate new population with crossovers and mutations;
Step 4: Go Step 2 until the termination criteria are met.
```

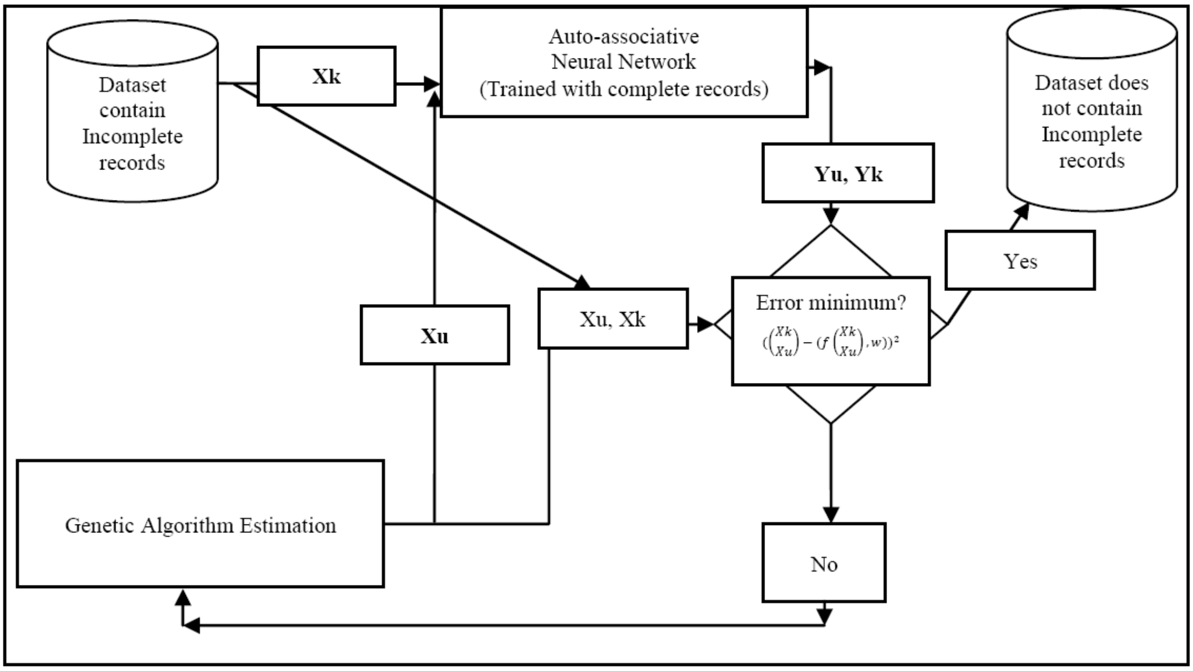

Dataset contain Incomplete records
Xk
Auto-associative Neural Network (Trained with complete records)
Dataset does not contain Incomplete records
Yu, Yk
Yes
Xu
Xu, Xk
Error minimum?
$((\begin{smallmatrix}Xk\\Xu\end{smallmatrix}) - (f(\begin{smallmatrix}Xk\\Xu\end{smallmatrix}), w))^2$
Genetic Algorithm Estimation
No

Figure 2. Missing value estimator using auto-associative neural network and genetic algorithm (NN-GA)

2.4. **Missing data estimate by auto-associative neural network and genetic algorithm.** Abdella and Marwala used the NN-GA model in Figure 2 to estimate missing values. The Auto-associative Neural Network consists of input (X), output (Y) and weights (w).

Mathematically the neural network can be written as (1):

$$Y = f(X, w) \tag{1}$$

Since the network is trained to predict its own input vector, the input vector X will approximately be equal to output vector Y (2):

$$X \approx Y \tag{2}$$

In reality, the input vector X and output vector Y will not always be perfectly the same; hence, we will have an error function expressed as the difference between the input and output vector. Thus, the error (3) can be formulated as:

$$e = X - Y \tag{3}$$

Y is formulated as (1) and error (e) can be written (4) as:

$$e = X - f(X, w) \tag{4}$$

It is preferred that the minimum error (e) must be non-negative. Hence, the error function (5) can be rewritten as the square of the equation:

$$e = (X - f(X, w))^2 \tag{5}$$

Missing data occurs when some of the input vectors (X) values are not available. Thus, the input vector (X) can categorize elements in to X known values represented by (Xk) and X unknown values represented by (Xu). The rewritten equation in terms of Xk and Xu error (6) is;

$$e = \left(\begin{pmatrix} Xk \\ Xu \end{pmatrix} - \left(f \begin{pmatrix} Xk \\ Xu \end{pmatrix}, w \right) \right)^2 \tag{6}$$

The equation was supplied to the GA as fitness function (6). Thus, the final error function was minimized using the genetic algorithm. This is the fundamental equation required to relate the problem to the GA and to promote successful data estimation.

2.5. **K-nearest neighbors (KNN).** Missing values are estimated by combining the columns of K-nearest records chosen based on a given similarity measure [1]. Thus, KNN predictions are based on the sensitive supposition that objects close in distance are potentially similar. Both the measure to use for computing similarities between records and the number of nearest neighbors (K) must be determined [1,24].

Euclidean distance d_{ik} (7), calculates the distance between the nearest (K) complete records x_i and incomplete record y_i:

$$d_{ik} = \sqrt{\sum_{j=1}^{n} (x_{kj} - y_{ij}^2)} \tag{7}$$

where j is the element in the ith row and jth column of the missing indicator in complete record. The missing entry j of complete record i is then estimated by the weighted average of the column values of the neighbors (K) most similar records, rows:

$$w_{ik} = \frac{1/d_{ik}}{\sum_{k=1}^{K} 1/d_{ik}} \tag{8}$$

where w_{ik} is the weight for the kth neighbor record of complete data i normalized by the sum of the inverse weighted Euclidean distance for all K neighbors [24]:

$$y_{ij} = \sum_{k=1}^{K} w_{ik} x_{kj} \tag{9}$$

Missing value y_{ij} is estimated from most similar (K) neighboring values and (K) neighbors weighted distance average (9). For the best accuracy of estimation (K) number of nearest neighbors must be determined. (K) neighbor size can change depending on dataset records, attribute counts and missing ratio. In this paper we aim to generate a fitness function to determine the best optimal (K) number of nearest neighbors.

3. **Proposed Method.**

3.1. **Data.** We used the four datasets in Table 2, frequently used in related works in literature. The first dataset was the data of a 120-MW power plant in France 19, under normal operating conditions. This dataset comprises five inputs. Sampling of the data was done every 1228.8 s and a total of 200 instances were recorded [3,5,7,10,15]. The second one, named Ogawa data, was initially composed of $N = 827$ records and $n = 8$ attributes, column conditions about the phosphate accumulation and the polyphosphate metabolism of the yeast Saccharomyces cerevisiae [25]. The third dataset comes from a study of the cell cycle regulated genes in Saccharomyces cerevisiae and consists of time series cDNA microarray data. The data set called cdc28 contains data from a cdc28-based synchronization [26]. And, the fourth dataset represents a test set-up of an industrial winding process. The full dataset has 2500 instances, and 5 inputs sampled every 0.1 s [15].

TABLE 2. The used datasets

Dataset Name	***Records***	***Attributes***
Power plant	*200*	*5*
Ogawa	*827*	*8*
Cdc28	*1370*	*17*
Winding	*2500*	*5*

3.2. **The proposed method for estimating missing values with auto-associative neural network (NN) and k-nearest neighbors (KNN).** We can estimate missing values with weighted k-NN where k is the number of neighbors. To achieve the best accuracy, we should find optimal (K). Auto-associative Neural Network error function is $\boldsymbol{error = (X - Y)^2}$; X is input, and Y is output of Auto-associative neural network. Thus, we use fitness function where minimum error gives optimal (K) size. For example, a typical dataset can be divided into two sections: Dataset [complete], Dataset [incomplete] rows. It can be described that incomplete record is a row of the dataset in which one or more attributes are missing values; on the other hand, complete record is a row of the dataset in which all attributes are known not missing. Figure 3 illustrates the proposed method.

Proposed Method NN-KNN imputation pseudo code:

```
1. Train Auto-associative Neural Network with Dataset [complete], Input $(X)$ ≈ Output $(Y)$;
2. From $k = 1$ to Dataset [complete] rows size, estimate missing values with k-NN $(X)$;
   a. Get Auto-associative Neural Network output $(Y)$ corresponding k-NN $(X)$ estimation results and calculate error $e = (Y - (k - NN(X)))^2$;
   b. Find optimal k neighbors number corresponding minimum error $(e)$.
3. Impute missing Dataset [incomplete] records with optimal observed k neighbors number.
```

Motivation of the practical use of the theoretical results obtained can be addressed with this practical example in Figure 4.

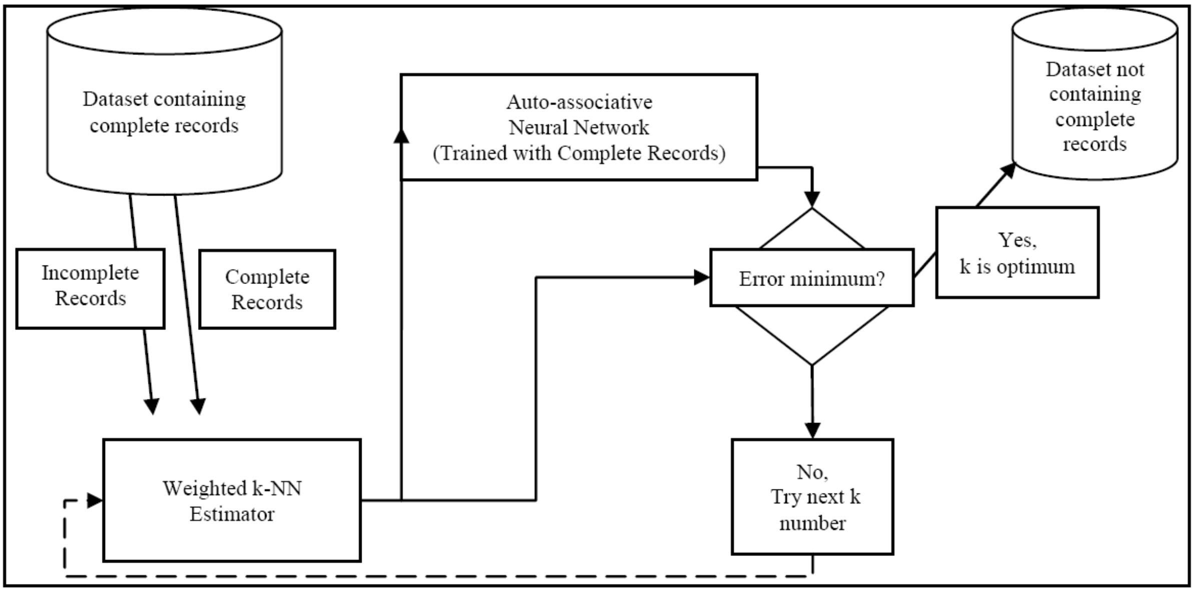

Dataset containing complete records
Incomplete Records
Complete Records
Weighted k-NN Estimator
Auto-associative Neural Network (Trained with Complete Records)
Error minimum?
Yes, k is optimum
Dataset not containing complete records
No, Try next k number

FIGURE 3. Proposed method NN-KNN imputation

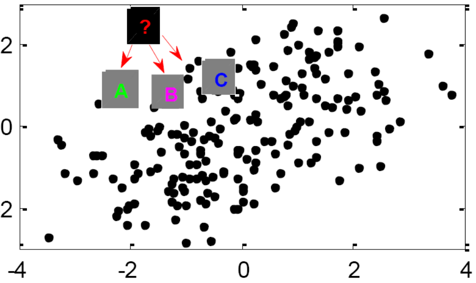


FIGURE 4. An example of practical use of theoretical results

We can assume that '?', 'A', 'B', 'C' are records of the dataset. '?' demonstrates record containing a missing value; 'A', 'B', 'C' are the nearest three neighbors, values are $x_{Ai} = 3$, $x_{Bi} = 2$, $x_{Ci} = 4$ and euclidean distances d_{ik} (7), between '?' and 'A', 'B' , 'C' are, $d_{?\text{-}A} = 13.43$, $d_{?\text{-}B} = 9.43$, $d_{?\text{-}C} = 10.13$, and computation of weights (8) are $w_{?\text{-}A} = 0.2792$, $w_{?\text{-}B} = 0.3798$, $w_{?\text{-}C} = 0.3536$. Estimation '?' missing value (9) is $y_{?i} = 3 * 0.2792 + 2 * 0.3798 + 4 * 0.3536$, '?' $= 3.0116$. This value can change depending on k neighbor size. The trained neural network also predict a result for '?'; thus, both proximate prediction result is the best imputation and gives optimal k neighbor size for the dataset used.

4. Experimental Implementation.

4.1. **Data implementation.** The attributes of some records were artificially deleted to check performance and compare the results of our proposed method and those of works in the literature.

TABLE 3. Implemented datasets

Dataset Name	*Records*	*Attributes*	*Records with Missing Values*	*Records with Complete Values*
Power plant	*200*	*5*	*20*	*180*
Ogawa	*827*	*8*	*82*	*745*
Cdc28	*1370*	*17*	*137*	*1233*
Winding	*2500*	*5*	*250*	*2250*

All datasets were transformed using a min-max normalization (10) to $\{0, 1\}$ before use, to ensure that they fall within the active range of the activation function of the neural network.

$$x_{i,norm} = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}} \tag{10}$$

4.2. **Auto-associative neural network implementation.** We used Matlab R2009b version 7.9, which has neural network toolbox and a Matlab code was written. The neural network has one hidden layer with the same number of neurons of input and output. 'Trainscg' is the network training function that updates weight and bias values according to the scaled conjugate gradient method, as transfer functions hyperbolic tangent sigmoid transfer function (tansig) for hidden layer and linear transfer function (purelin) were used for output layer. Dataset records [complete] were divided into 60% train, 20% validate 20% test data, and the network was trained until Performance goal = 0.0001 was met.

4.3. **Genetic algorithm implementation.** Similarly, we used MatlabR2009b version 7.9, which has genetic algorithm toolbox and Matlab code was written. The genetic algorithm used a population size of 20 and 40 generations, a crossover fraction of 60% and a mutation fraction of 3%.

4.4. **Proposed method ANN-KNN imputation.** We used MatlabR2009b version 7.9 and Matlab code was written based in Section 3.2 new proposed method pseudo code and flow diagram in Figure 3.

4.5. **Performance evaluation.** The efficiency of the missing data estimation system is evaluated using the Root Mean Standard Error (RMSE), the Correlation Coefficient (r) and the relative prediction accuracy (A). The Root Mean Standard Error measures the error between the real values and the estimated values and can refer to the capability of prediction [1,4,9,15,24]. It is given by (11), where x_i is the real value, $x^\wedge$ is the estimated value and n is the number of missing values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x_i^\wedge)^2}{n}} \tag{11}$$

The Correlation Coefficient (r) measures the linear similarity between the estimated and real data [4,9,15,24] r ranges between -1 and 1, where its absolute value relates to the strength of the relationship while the sign of r indicates the direction of the relationship. Hence, a value close to 1 indicates a strong predictive capability. The formula is given by (12), where mx is the mean of the data.

$$r = \frac{\sum_{i=1}^{n}(x_i - mx_i)(x_i^\wedge - mx_i^\wedge)}{\left[\sum_{i=1}^{n}(x_i - mx_i)^2 \sum_{i=1}^{n}(x_i^\wedge - mx_i^\wedge)^2\right]^{1/2}} \tag{12}$$

The relative prediction accuracy (A) is a measure of how many estimations are made within a certain tolerance [4,9,15]; tolerance is set to 10% as done by Nelwamondo et al. The accuracy is given by (13), where n is the total number of predications, n_T is the number of correct predications within certain tolerance.

$$A = \frac{n_T}{n} \times 100 \tag{13}$$

5. **Experimental Results and Discussion.** The comparisons of the Root Mean Standard Errors, Correlation Coefficients and Relative prediction accuracies between NN-GA, NN-KNN methods are illustrated in Figure 5.

Figure 5 shows the actual value versus estimated with different methods, power plant dataset has 20 missing values (Table 4), and estimation NN-KNN shows better estimation accuracy rather than NN-GA.

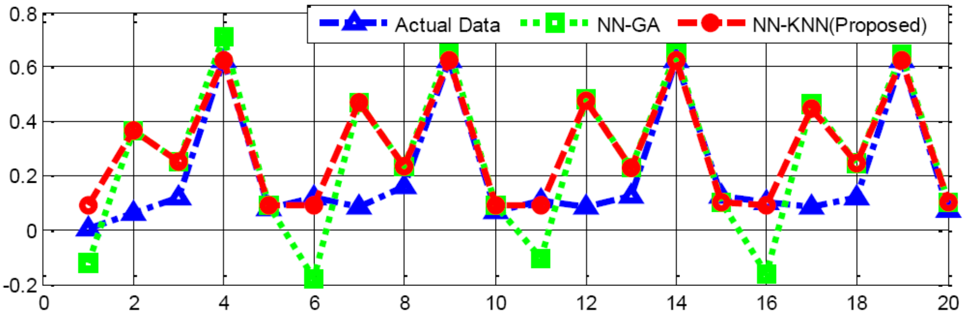


FIGURE 5. Power plant dataset real vs. estimated values

The conditions imposed to develop the main results can be tracked in Figure 5, artificially deleted values to be estimated are actual data in Table 4. NN-GA method was used to estimate missing values, but as it is seen in Figure 5, some of values (1, 4, 6, 9, 11, 14, 16, 19) have less imputation accuracy because some of them may be outliers, noisy or the neural network may not have trained enough. A problem with GAs is; the genes which are highly fit, but not optimal may rapidly come to dominate the population causing it to converge on a local minimum. Once the population has converged, the ability of the GA to search for better solutions is effectively eliminated: the crossover of almost identical chromosomes produces little that is new. Another GA deficiency is that using only mutation to explore entirely new ground results in a slow, random search [27]. On the other hand, one of the advantages of the proposed KNN hybrid method used with neural network is that some noise reduction techniques that work only for KNN can be effective in improving the accuracy of the computations. It provides weighted mean on noisy or deficient of neural network training output values which can be seen in Figure 5 (1, 4, 6, 9, 11, 14, 16 and 19).

Table 4. Power plant dataset real vs. estimated values

	Actual	*NN-GA*	*NN-KNN (Proposed)*		*Actual*	*NN-GA*	*NN-KNN (Proposed)*
1	*0.006*	*− 0.120*	*0.092*	*11*	*0.109*	*− 0.108*	*0.093*
2	*0.062*	*0.368*	*0.367*	*12*	*0.082*	*0.482*	*0.475*
3	*0.121*	*0.250*	*0.250*	*13*	*0.123*	*0.229*	*0.229*
4	*0.627*	*0.712*	*0.626*	*14*	*0.624*	*0.651*	*0.625*
5	*0.077*	*0.090*	*0.090*	*15*	*0.124*	*0.099*	*0.099*
6	*0.118*	*− 0.178*	*0.093*	*16*	*0.103*	*− 0.160*	*0.093*
7	*0.085*	*0.471*	*0.471*	*17*	*0.083*	*0.464*	*0.446*
8	*0.159*	*0.234*	*0.233*	*18*	*0.121*	*0.244*	*0.244*
9	*0.624*	*0.666*	*0.625*	*19*	*0.622*	*0.645*	*0.626*
10	*0.070*	*0.092*	*0.092*	*20*	*0.072*	*0.100*	*0.100*

Figure 6 shows Root Mean Standard Errors (RMSE) comparison of two methods on four datasets. RMSE indicates error of predictions, and a low error value indicates a better performance.

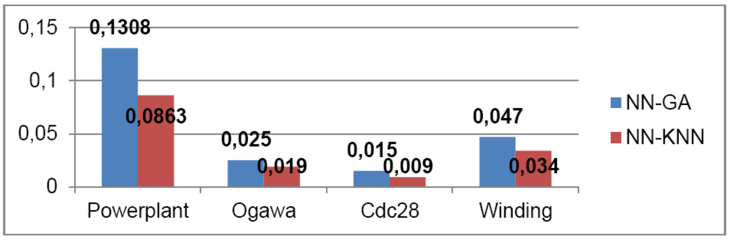


Figure 6. Root mean standard errors (RMSE)

Figure 7 shows correlation coefficients (r) of two methods on four datasets. The correlation calculates a similarity between actual data and predicted data; a high value of correlation indicates a better performance.

Figure 8 shows relative prediction accuracies (A) of two methods on four datasets. (A) is the calculation of actual data versus predicted data number with a tolerance of 10%, a high value of accuracy indicates a better performance.

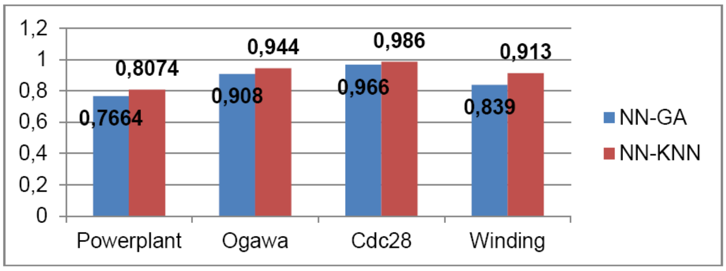


Figure 7. Correlation coefficients (r)

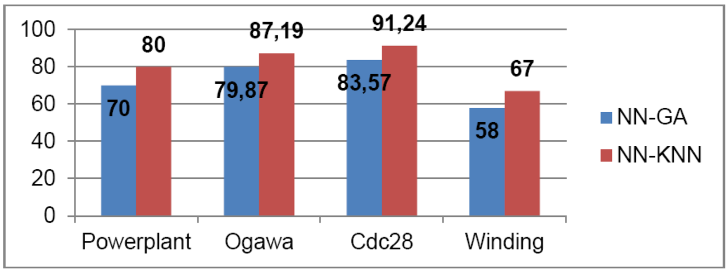


Figure 8. Relative prediction accuracies (A)

The examples used to demonstrate the main results in Figure 6, Figure 7, Figure 8 indicate that Ogawa, Cdc28 microarray datasets produce better results than industrial time series datasets power plant and winding, it is quite clear in this paper that microarray datasets are more suitable data for neural network training progress and KNN is very sensitive to irrelevant or redundant features because all features contribute to the similarity, and thus, to the imputation. This can be ameliorated by careful feature selection or feature weighting [28]. Both GAs and neural nets are adaptive; they can learn and deal with highly nonlinear models and are robust, ”weak” random search methods. They do not need gradient information or smooth functions. In both cases their flexibility is also a drawback, since they have to be carefully structured and coded and are fairly dataset specific [29]. One of advantages of KNN is that it is well suited for multi-modal classes as its classification decision is based on a small neighborhood of similar objects i.e., the major class. So, even if the target class is multi-modal i.e., consists of objects whose independent variables have different characteristics for different subsets, it can still lead to good accuracy. There are some noise reduction techniques working only for KNN that can be effective in improving the accuracy of the imputation [28].

As shown in Figure 9, some records of dataset may have more than one missing values. We tested our method with records which have 1, 2 or 3 missing values in a record. The empirical results showed that the proposed NN-KNN method still has stable good results with low RMS error ratio versus NN-GA.

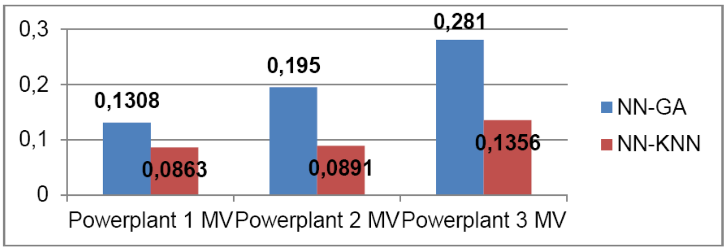
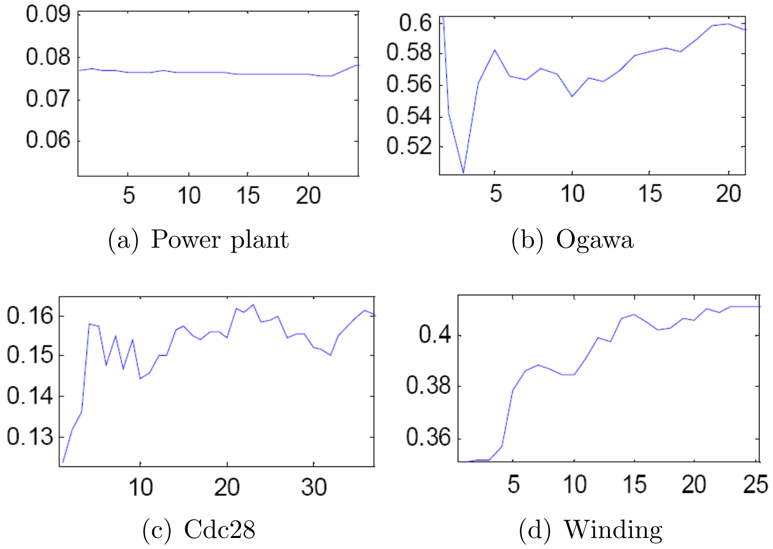


Figure 9. RMS error ratios on different missing values in a power plant row

Unique features of the approaches proposed and main advantages of the results over others are shown in Figure 10. Base model KNN imputation has different k value for each dataset and cannot able to determine the best k value. K size changes depending on dataset rows, attributes size and type of data; for example, time series or non-time series.



(a) Power plant (b) Ogawa

(c) Cdc28 (d) Winding

FIGURE 10. Basic model KNN imputation k-neighbor size (axis X) and RMS error ratio (axis Y)

NN-GA imputation fails in some outliers data and genetic algorithm and this may risk finding a suboptimal solution unless the whole solution space is searched. This is a local minimization problem. On the other hand, the NN-KNN approach proposed decides best k value suitable on data and uses KNN weighted mean result. The proposed method produces much conservative data estimation on outliers. At the same time there are some deficiencies of the proposed method; neural network training is a significant subject choosing neural network type and training until what performance criteria are met, must be elaborated before imputation. To make further improvement in imputation accuracy, feature selections; for example, principal component analysis methods can be applied before neural network training. Another deficiency is computation time; in this paper, we worked on different k neighbor sizes corresponding to each missing value to obtain more sensitive results, but this extends computation time. For further improvement, a unique k size can be obtained for whole missing values.

6. **Conclusion.** This paper has proposed a new approach which utilizes machine learning and artificial intelligence systems. The results showed that k-nearest neighbors estimation method adds significant advantages on neural network base model. A number of tests carried out on different characteristics of the four datasets showed that more efficient results depend on neural network training performance; therefore, with more complete training data, NN-KNN estimations are more sensitive and exact. This paper investigates the estimation of missing data through novel techniques. The estimation system involves an auto-associative model to predict the input data, coupled with the k-nearest neighbors to approximate the missing data. The results show that the hybrid NN-KNN can be applied when the record has more than one missing value in a row. We have studied and compared the neural network and KNN approach with the neural network and GA

combination approach for missing data approximation. In both approaches, an auto-associative neural network was trained to predict its own input space. In basic model GA was used to approximate the missing data. On the other hand, the KNN algorithm was implemented for the same problem. The results show that NN-KNN method is able to produce better imputation accuracy. The findings also show that the method seems to perform better in cases where there is dependency among the variables.

REFERENCES

[1] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R. B. Altman, Missing value estimation methods for DNA microarrays, *Bioinformatics*, vol.17, pp.520-525, 2001.

[2] B. B. Thompson, R. J. Marks and M. A. El-Sharkawi, On the contractive nature of autoencoders: Application to sensor restoration, *Proc. of the International Joint Conference on Neural Networks*, vol.4, pp.3011-3016, 2003.

[3] W. Qiao, Z. Gao and R. G. Harley, Continuous on-line identification of nonlinear plants in power systems with missing sensor measurements, *Proc. of Intnl. Joint Conf. on Neural Networks*, pp.1729-1734, 2005.

[4] M. I. Abdella and T. Marwala, The use of genetic algorithms and neural networks to approximate missing data in database, *Computing and Informatics*, vol.24, pp.1001-1013, 2006.

[5] S. Mohamed and T. Marwala, Neural network based techniques for estimating missing data in databases, *The 16th Annual Symposium of the Patten Recognition Association of South Africa*, Langebaan, pp.27-32, 2005.

[6] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, Wiley, New York, 1987.

[7] F. V. Nelwamondo, S. Mohamed and T. Marwala, Missing data: A comparison of neural network and expectation maximization techniques, *Current Science*, vol.93, no.11, 2007.

[8] F. Nelwamondo and T. Marwala, Fuzzy ARTMAP and neural network approach to online processing of inputs with missing values, *SAIEE Africa Research Journal*, vol.93, no.11, pp.1514-1521, 2007.

[9] A. K. Mohamed, F. V. Nelwamondo and T. Marwala, Estimating missing data using neural network techniques, principal component analysis and genetic algorithms, *PRASA Proceedings*, 2007.

[10] F. V. Nelwamondo, D. Golding and T. Marwala, A dynamic programming approach to missing data estimation using neural networks, *Information Sciences*, 2009.

[11] D. Blend and T. Marwala, Comparison of data imputation techniques and their impact, *Scientific Commons*, 2008.

[12] N. Hlalele, F. Nelwamondo and T. Marwala, Imputation of missing data using PCA, neuro-fuzzy and genetic algorithms, *Advances in Neuro-Information Processing, Lecture Notes in Computer Science*, vol.5507, pp.485-492, 2009.

[13] D. V. Patil, Multiple imputation of missing data with genetic algorithm based techniques, *IJCA Special Issue on "Evolutionary Computation for Optimization Techniques"*, 2010.

[14] E.-L. Silva-Ramírez, R. Pino-Mejías, M. López-Coello and M.-D. Cubiles-de-la-Vega, Missing value imputation on missing completely at random data using multilayer perceptrons, *Neural Networks*, 2011.

[15] T. Marwala, Computational intelligence for missing data imputation, estimation, and management: Knowledge optimization techniques, *Information Science Reference*, USA, 2009.

[16] F. V. Nelwamondo and T. Marwala, Techniques for handling missing data: Applications to online condition monitoring, *International Journal of Innovative Computing, Information and Control*, vol.4, no.6, pp.1507-1526, 2008.

[17] Z. Chen, B. Yang, Y. Chen, L. Wang, H. Wang, A. Abraham and C. Grosan, Improving neural network classification using further division of recognition space, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.301-310, 2009.

[18] S. Gao, J. Zhang, X. Wang and Z. Tang, Multi-layer neural network learning algorithm based on random pattern search method, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.489-502, 2009.

[19] B. L. Betechuoh, T. Marwala and T. Tettey, Autoencoder networks for HIV classification, *Current Science*, vol.91, no.11, pp.1467-1473, 2006.

[20] J.-F. Chang, A performance comparison between genetic algorithms and particle swarm optimization applied in constructing equity portfolios, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.5069-5079, 2009.

[21] Y. Li, Y. Yang, L. Zhou and R. Zhu, Observations on using problem-specific genetic algorithm for multiprocessor real-time task scheduling, *International Journal of Innovative Computing, Information and Control*, vol.5, no.9, pp.2531-2540, 2009.

[22] M. Braik, A. Sheta and A. Arieqat, A comparison between GAs and PSO in training ANN to model the TE chemical process reactor, *AISB Convention Communication, Interaction and Social Intelligence*, 2008.

[23] K. Hengpraphrom, S. N. Wichian and P. Meesad, Missing value imputation using genetic algorithm, *The 3rd International Symposium on Intelligent Informatics the 1st International Symposium on Information and Knowledge Management*, 2010.

[24] L. P. Bras and J. C. Menezes, Improving cluster-based missing value estimation of DNA microarray data, *Biomolecular Engineering*, vol.24, pp.273-282, 2007.

[25] N. Ogawa, J. DeRisi and P. O. Brown, New components of a system for phosphate accumulation and polyphosphate metabolism in Saccharomyces cerevisiae revealed by genomic expression analysis, *Mol. Biol. Cell*, vol.11, no.12, pp.4309-4321, 2000.

[26] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein and B. Futcher, Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization, *Molecular Biology of the Cell*, vol.9, no.12, pp.3273-3297, 1998.

[27] D. E. Goldberg, Sizing populations for serial and parallel genetic algorithms, *Proc. of the 3rd International Conference on Genetic Algorithms*, pp.70-79, Morgan Kaufmann, 1989.

[28] P. Cunningham and S. J. Delany, k-nearest neighbour classifiers, *University College Dublin Technical Report UCD-CSI-2007-4*, 2007.

[29] F. Busetti, *Genetic Algorithms Overview*, 2001.