Building a novel classifier based on teaching learning based optimization and radial basis function neural networks for non-imputed database with irrelevant features

Ch. Sanjeev Kumar Dash and Ajit Kumar Behera
Silicon Institute of Technology, Bhubaneswar, India
Satchidananda Dehuri
Department of Information and Communication Technology, Fakir Mohan University, Balasore, India, and
Sung-Bae Cho
Soft Computing Laboratory, Department of Computer Science, Yonsei University, Seoul, Republic of Korea

Abstract
This work presents a novel approach by considering teaching learning based optimization (TLBO) and radial basis function neural networks (RBFNs) for building a classifier for the databases with missing values and irrelevant features. The least square estimator and relief algorithm have been used for imputing the database and evaluating the relevance of features, respectively. The preprocessed dataset is used for developing a classifier based on TLBO trained RBFNs for generating a concise and meaningful description for each class that can be used to classify subsequent instances with no known class label. The method is evaluated extensively through a few bench-mark datasets obtained from UCI repository. The experimental results confirm that our approach can be a promising tool towards constructing a classifier from the databases with missing values and irrelevant attributes.

Keywords Pattern recognition, Imputation, Classification, Radial basis function neural networks, Teaching learning based optimization, k-Nearest neighbor

Paper type Original Article
1. Introduction
The occurrence of missing values and irrelevant features in real data are not uncommon, whereas data mining algorithms are designed for quality data [1]. Hence, building a classifier for the dataset consist of missing values and many irrelevant attributes leads to non-useful results [2]. Therefore, to derive novel and useful results for the decision maker, the process of imputing and identifying missing values and relevant features, respectively are highly recommended. Since decades ago these two problems are treated as the problem of importance in object detection & recognition (pattern recognition) [11] and data mining [3] in general and ECG signals diagnosis [13], power flow calculation [14], simulation and control of dynamic system [15], magnetic modeling [16], identification and classification of plant leaf diseases [17], discrimination of low and full fat Yogurts [19,20,22] in specific.

There are several approaches to impute missing values of which we concentrate on least square estimation method [2,3]. A large variety of feature selection techniques have been developed under the umbrella of filter, wrapper, and embedded methods with a goal to select relevant subset of features [4]. In this work a filter style approach known as “Relief” method is used for selecting a subset of attributes that preserves the relevant information found in the entire set of attributes [5]. After the task of imputation of missing values and selection of the relevant set of features, we develop a classifier based on TLBO and RBFNs by inheriting their best features [6,7,8]. RBFN one of the members of artificial neural networks (ANNs) [21,22] has good generalization, simple structure and strong tolerance to noise which ignited us to consider here as a suitable method of classification. Many methods have been developed for training RBFNs [12,17,18], however, to the best of our knowledge, training RBFNs using TLBO is new. TLBO is a population based optimization algorithm motivated by a teacher on the output of learners within a classroom environment, where learners first obtain knowledge from teacher and subsequently from classmates. Moreover, a new improved TLBO (iTLBO) has been proposed to train the RBFNs.

In a nutshell, this work undergoes three different phases like imputation of missing values by least square estimation approach, feature selection through Relief, and classification by iTLBO trained RBFNs in pipeline.

2. Background
The background of this research work like missing values imputation, feature selection, RBFNs, and TLBO are discussed here.

2.1 Imputation of missing values and feature selection
The problem of classification is basically the foundation of dividing the feature space into sections, one section for each category of inputs. Classifiers are usually, designed with labeled data, which is sometime referred to as supervised classification. In general, classification with missing data and irrelevant features focuses on three distinct tasks: handling missing values [1] (i.e., imputing values), feature selection, and pattern classification. Let \( D = \{x_{ij}\}_{N \times d} \), where \( i = 1, 2, \ldots, N \) and \( j = 1,2, \ldots,d \), is the dataset containing \( N \) samples and \( d \) features. In \( D \), each sample is assigned a class label from the set \( C = \{c_1, c_2, \ldots, c_M\} \), where \(|C| = M\). Let each \( x_{ij} \) be represented as a tuple \((x_{ij}, y_{ij})\), in which \( y_{ij} \) can take only two values either 0 or 1. If the value of \( v_{ij} = 0 \), then its associated \( x_{ij} \) value is missing, otherwise present. Input data has quantitative and qualitative variables. Quantitative or continuous data is measured on a numerical scale. Non-numerical (i.e., colors, names, opinions) is called qualitative data, which can be discrete or categorical. The
Overall goal of handling missing value is to map the value of \( y_{ij} \) from 0 to 1 by substituting an appropriate value of \( x_{ij} \) with less bias.

Alongside feature selection problem is defined as to select a subset of features from the given set of features, thereby the dataset is mapped from \((x_{ij})_{N \times d}\) to \((x_{ij})_{N \times k}\), where \( k \ll d \).

With this intention, filter method is selecting the most relevant features, however, a predefined quality measure is necessary to establish the level of relevance of the features. Filter method is not able to identify correlation among the features simultaneously. Unlike filter, wrapper is able to address correlation among features because it uses the performance of the classifier to optimize the subset. This also led towards problem of intractability. Moreover, this method has the additional cost of reconstructing the classifier with modified feature subset. Hence to avoid these issues, a filter-like algorithm known as Relief method is employed here.

### 2.2 Radial basis function networks

The RBF network [8] is a topology having three layers: an input, a hidden, and a linear output layer (see Figure 1). The input can be modeled as \( n \)-dimensional input vector. The hidden layer implements a radial activation function and that carry out a non-linear transformation from the input space to the hidden space. The center and width are two parameters associated for each hidden node. Usually, a nonlinear transformation from input to the hidden space is made based on Gaussian kernel as described in Eq. (1).

\[
\varphi_i(x) = \exp \left( -\frac{||x - \mu_i||^2}{2\sigma_i^2} \right),
\]

where \( \| \cdot \| \) represents Euclidean norm, \( \mu_i, \sigma_i \), and \( \varphi_i \) are center, spread, and the output of \( i \)th hidden unit, respectively. The interconnection between the hidden and output layer are made through a weighted connections \( w_i \). The output layer, a summation unit, supplies the response of the network to the outside world.

The radial basis function is so named because the value of the function is same for all points which are at the same distance from the center.

![Figure 1. Architecture of RBFN.](image)
In literature, radial basis function networks [6] have many extensive uses, including classification, time series prediction, function approximation, etc. Training RBF networks is normally faster than training multi-layer perceptron (MLP) networks. Training of RBF network [9,11] involves two steps: (1) the kernel parameters of the hidden neurons are determined by an unsupervised method or heuristic method; (2) The weights of the output-layer are determined by pseudo-inverse method.

2.3 Teaching learning based optimization
Teaching learning based optimization is one of the population based nature inspired algorithms introduced by Rao et al. [6,9]. This is inspired purely from the natural phenomena of teaching-learning process that motivated by a teacher on the output of learners within a classroom environment, where learners first obtain knowledge from teacher and subsequently from classmates. In the first phase, a teacher imparts knowledge directly to his/her students. In practice, the possibility of a teacher’s teaching being successful, is distributed under Gaussian law. Overall, how much knowledge is transferred to a student depends not only on his/her teacher but also interactions among the students through peer learning. A basic algorithm of TLBO is presented below.

3. Proposed method
Our integrated approach is undergoing three phases in pipeline. In first phase, the missing values are imputed by least square estimator, in second phase the relief algorithm is used for feature selection and finally our improved TLBO based RBFN is used for building the classifiers for the preprocessed database. Figure 2 is illustrating our approach.

3.1 Missing value imputation using least-square estimator
In this phase, we estimate the missing value from D by formulating a matrix A, where all the attribute values are known. In the least-square problem, the output of a model is given by the linearly parameterized expression,

\[ y = \phi_1 f_1(u) + \phi_2 f_2(u) + \ldots + \phi_n f_n(u). \]

If the target system has q outputs, expressed as \( y = [y_1, \ldots, y_q]^T \) with \( q > 1 \) then we have a set of linear equations in matrix form, \( A\Theta + E = Y \), where \( A \) is an \( m \times n \) matrix as given below:

\[
A = \begin{bmatrix}
    f_1(u_1) & \ldots & f_n(u_1) \\
    \vdots & & \vdots \\
    f_1(u_m) & \ldots & f_n(u_m)
\end{bmatrix}
\]

and \( \Theta \) is an \( n \times q \) unknown parameter matrix:

\[
\Theta = \begin{bmatrix}
    \theta_{11} & \ldots & \theta_{1q} \\
    \vdots & & \vdots \\
    \theta_{n1} & \ldots & \theta_{nq}
\end{bmatrix}
\]

\[ Y = \begin{bmatrix}
    y_{11} & \ldots & y_{1q} \\
    \vdots & & \vdots \\
    y_{m1} & \ldots & y_{mq}
\end{bmatrix}
\]

is an \( m \times q \) output matrix with \( y_{ij} \) denoting the \( j \)th output value in the \( i \)th data pairs.

\[
\Theta = \left( A^T A \right)^{-1} A^T Y
\]

After getting the value of \( \Theta \), we will continue imputing all the values in the data set D.
Algorithm TLBO
BEGIN

Set generation = 0;
Initialize Population (P); // |P| is the size of the population
Evaluate (P)
DO

BS = Select Best Solution from (P); // Archive of best solution.
FOR i = 1 to |P|

T = Round (1+r);
// T & r are teaching factor and random number, respectively.
Xmean = Mean Vector from (P);
Xteacher = Select Best Solution from (P);
Xnew = Xi + r. (Xteacher - (T. Xmean));
IF
Xnew is better than Xi then Xi = Xnew;
ENDIF

Xi = random (P); Xi <> Xi;
IF
Xi is better than Xj then Xinew = Xi + r. (Xj - Xi)
ELSE
Xinew = Xi + r. (Xj - Xi);
ENDIF
Evaluate (Xinew);
IF
Xinew is better than Xi then Xi = Xinew;
ENDIF
END FOR
P = Replace worst solution of (P) by (BS);
P = Remove duplicate individuals from (P);
generation++;
WHILE (generation <> Max-generation);
PRINT best result of P;
END

Figure 2. Pictorial Representation of our Approach (LSEI: Least Square Estimation for Imputation).
3.2 Relief algorithm for feature selection

In this second phase of our work, we discuss Relief algorithm inspired by instance based learning. It is a filter method algorithm for individual feature selection. It calculates a proxy statistics for each feature that can be used to estimate the feature quality or relevance to the target concept. The pseudocode of this method is given below.

**Algorithm Relief** (D, No.Feature, No.Sample, Threshold)

BEGIN

T=\{

Initialize all feature weights, W=0;

FOR i=1 to No.Sample

    Randomly choose an instance \( x \) from D;
    Find its near-Hit and near-Miss;

    FOR j=1 to No.Feature

        \[ W_j = W_j - \| x_j - \text{nearHit}_j \| + \| x_j - \text{nearMiss}_j \| \]

    END FOR

END FOR

FOR j=1 to No.Feature

    IF \( W_j \) is above the Threshold THEN

        Append feature \( F_j \) to T;

    END IF

END FOR

Return T.

END

3.3 Improved TLBO based RBFN

In third phase, we are building a RBFN classifier which is trained by TLBO and improved TLBO. First we will provide a detailed introduction to the improved TLBO and then the improved TLBO + RBFN network is developed with the aim of achieving better classification accuracy.

3.3.1 Improved TLBO (iTLBO). In the canonical TLBO, during the learning phase the learner is exposed to the entire population of the class. However, it has been realized that if the learner is restricted with a peer team instead of all individuals of the population then he/she can raise his/her level of acquiring knowledge. With this idea, we are introducing a neighborhood structure of learners as peer learners group for making a learner to learn. Hence, in the learner phase, we have adopted a square topology as peer learners group for a learner. That means a student will not only acquire knowledge from the best of all individuals (i.e., teacher) but also he/she improves his/her standard from his neighborhood of fellow learners. In that context, the learner phase of TLBO has been modified as given below.
Here the nearest_neighbor() will find out a group of peer learners for a learner. The size of the neighborhood can be treated as a parameter for learner phase. Alongside, we have also made the teaching parameter (T_T) adaptive by considering the individual fitness value and population diversity. Recall that the teaching factor decides the value of mean to be changed. In the canonical TLBO, the value of T_T is either 1 or 2 thereby learners learn nothing from the teacher or learn all the things from the learner. But in real practice, the value of T_T may be between 1 and 2 include both. Hence to make this idea fruitful, the fitness variable is selected as inputs to choose T_T. BS is containing the global best solution denoted as X_S^k found so far i.e. up to kth iteration, which is just a position for one individual, corresponding to the best fitness F_S^k. So the global best solution fitness differentials between kth and k-1th can be defined as:

$$\Delta F^k = \left| F_S^k - F_{SB}^k \right|.$$ \hspace{1cm} (4)

Now, we can give definition for function of convergence speed as follows:

$$C_s = \frac{\Delta F^k}{\Delta},$$ \hspace{1cm} (5)

where $\Delta = \max\{\Delta F^1, \Delta F^2, \ldots, \Delta F^k\}$ Eq. (5) can calculate the convergence speed, which is less than or equal to 1.

In evolution process of TLBO, population diversity is a major factor. For computing the diversity of the population, standard deviation of the individual fitness values of population can be used. In this paper, we present a new strategy for calculating population position diversity by fitness value. The population position diversity can be obtained by using deviation ideology approach defined in Eqs. (6)–(8).

$$F_{avg}^k = \frac{1}{|P|} \sum_{i=1}^{|P|} F^k(i),$$ \hspace{1cm} (6)

$$\Delta_2 = \max\left\{ \left| F^k(1) - F_{avg}^k \right|, \ldots, \left| F^k(|P|) - F_{avg}^k \right| \right\},$$ \hspace{1cm} (7)

$$\sigma^2 = \frac{1}{|P|} \sum_{i=1}^{|P|} \left( \frac{F^k(i) - F_{avg}^k}{\Delta_2} \right)^2,$$ \hspace{1cm} (8)
where \( F_{\text{avg}}^k \) is the average population fitness for current \( k \)th iteration; \( F^k(i) \) stands for \( i \)th individual fitness; \( \Delta_2 \) stands for normalization factor; \( |P| \) is the population size, \( \sigma^2 \) represents the population diversity. It is evident that larger the \( \sigma^2 \), the larger is the population diversity.

To improve adaptive teaching factor \( T_F \), we use the index \( C_S \) for representing the convergence speed with respect to the best solution fitness found so far in current iteration, and the index \( \sigma^2 \) to represent diversity with regards to the population fitness deviation. Hence we can compute \( T_F \) by \( C_S \) and \( \sigma^2 \) adaptively, as follows:

\[
T_F = \alpha \cdot C_S + \beta \cdot \sigma^2 + 1,
\]

where \( \alpha, \beta \) are factors; \( \sigma^2 \) and \( C_S \) are less than or equal to 1 and greater than 0, so \( 1 \leq T_F \leq \alpha + \beta + 1 \), Rao et al. [6] suggested that the value of \( T_F \) can be either 1 or 2. Hence, we set \( \alpha + \beta + 1 \leq 2 \). The proposed method of adaptive teaching factor \( (T_F) \) is applied for better local searching ability that improves the accuracy and convergence speed.

3.3.2 iTLBO + RBFN. This section describes the iTLBO + RBFN which can adjust the network parameters during the training process. In the initialization stage, let the position of the \( i \)th individual be represented as shown in Figure 3. RBFNs mainly depend on center and width of the kernel in addition to weights and bias. However, here, we just encode the centers, widths, and bias into an individual for stochastic search using iTLBO.

Suppose the maximum number of kernel nodes is set to \( K_{\text{max}} \), then the structure of the individual is represented as follows (c.f., Figure 3):

In other words, each individual has three constituent parts such as center, width, and bias. The length of the individual is \( 2K_{\text{max}} + 1 \).

The fitness function which is used to guide the search process is defined in equation (10).

\[
f(x) = \frac{1}{N} \sum_{i=1}^{N} (t_i - \Phi(x_i))^2
\]

where, \( N \) is the total number of training instances, \( t_i \) is the actual output and \( \Phi(x_i) \) is the estimated output of RBFNs. Initially, the centers, widths, and bias are computed using training vectors, the weight is computed using pseudo-inverse method.

\[
Y = W \Phi
\]

\[
\Rightarrow W = (\Phi^T \Phi)^{-1} \Phi^T Y
\]

Algorithm for iTLBO+RBFN

BEGIN

Impute missing value of the dataset using least square estimator according to equation (3);

Select relevant feature from the dataset using RELIEF algorithm;

Using training Instances compute the performance of iTLBO+RBFN;

END
4. Experimental study
In the experimental study, we start with a brief description of the datasets, their characteristics about missing information and parameters used for simulation. Then we display results obtained by two different methods like TLBO + RBFN and iTLBO + RBFN along with detail analysis.

4.1 Description of datasets and parameters
The datasets used in this work were obtained from the UCI machine learning repository [10]. Seven datasets have been chosen to validate the proposed method i.e., iTLBO + RBFN. The details about the seven datasets are given in Table 1. The algorithmic parameters like population size, number of iterations, etc are fixed based on empirical analysis as follows.

The size of the population is equal to 100, number of iterations fixed at 300, size of the neighborhood is restricted with 10% of population size, and the value of $T_F$ has been adapted as per suggestions given in sub-section 3.3.1 with $\alpha$ value from (0, 1) and $\beta$ value from (0, 1). The parameters of multi-layer perceptron (MLP) along with training algorithms and Simple Logistic are defined as prescribed in [3].

4.2 Results and analysis
The average results of the experiment obtained from 10 fold cross validation of 30 independent runs are given in Tables 2–7.

From Table 2 it is found that for 7 different datasets iTLBO + RBFN gives better accuracy than TLBO + RBFN, MLP, and Simple Logistic. To support the above results of TLBO + RBFN, statistical analysis based on the measures derived through confusion matrix is presented in Tables 3 and 4.
From the Statistical analysis it can be observed that the calculated Kappa-values for TLBO + RBFN with feature selection are much better than TLBO + RBFN without feature selection.

### Table 3.
**Further Details Analysis of TLBO + RBFN (Table 2).**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>Kappa Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hepatitis</td>
<td>88.5</td>
<td>19</td>
<td>88.3</td>
<td>88.5</td>
<td>88.2</td>
<td>0.7194</td>
</tr>
<tr>
<td>House-votes</td>
<td>96.8</td>
<td>2</td>
<td>96.9</td>
<td>96.8</td>
<td>96.6</td>
<td>0.9318</td>
</tr>
<tr>
<td>Mammographic</td>
<td>81.5</td>
<td>19</td>
<td>81.6</td>
<td>81.5</td>
<td>81.4</td>
<td>0.6272</td>
</tr>
<tr>
<td>Horse Colic</td>
<td>82.6</td>
<td>23</td>
<td>82.5</td>
<td>82.6</td>
<td>82.3</td>
<td>0.6515</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>95.1</td>
<td>4</td>
<td>95.2</td>
<td>95.1</td>
<td>95.2</td>
<td>0.9015</td>
</tr>
<tr>
<td>Diabetes</td>
<td>71.1</td>
<td>37.6</td>
<td>70.4</td>
<td>71.1</td>
<td>0.7</td>
<td>0.3525</td>
</tr>
<tr>
<td>Post-operative</td>
<td>66.7</td>
<td>66.7</td>
<td>44.4</td>
<td>66.7</td>
<td>53.3</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 4.
**Further Details Analysis of iTLBO + RBFN (Table 2).**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>Kappa Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hepatitis</td>
<td>91.0</td>
<td>65.3</td>
<td>89.4</td>
<td>91.0</td>
<td>89.9</td>
<td>0.684</td>
</tr>
<tr>
<td>House-votes</td>
<td>98.6</td>
<td>17</td>
<td>98.6</td>
<td>98.6</td>
<td>98.6</td>
<td>0.9714</td>
</tr>
<tr>
<td>Mammographic</td>
<td>83.8</td>
<td>16.8</td>
<td>83.9</td>
<td>83.8</td>
<td>83.7</td>
<td>0.6728</td>
</tr>
<tr>
<td>Horse Colic</td>
<td>82.1</td>
<td>20.2</td>
<td>82.1</td>
<td>82.1</td>
<td>82.1</td>
<td>0.6163</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>97.7</td>
<td>3.7</td>
<td>97.7</td>
<td>97.7</td>
<td>97.7</td>
<td>0.9093</td>
</tr>
<tr>
<td>Diabetes</td>
<td>74.5</td>
<td>33.7</td>
<td>74.1</td>
<td>74.5</td>
<td>73.6</td>
<td>0.4292</td>
</tr>
<tr>
<td>Post-operative</td>
<td>75.6</td>
<td>75.6</td>
<td>57.1</td>
<td>75.6</td>
<td>65.0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 5.
**Classification Accuracy of iTLBO + RBFN Using Least Square Imputation with Feature Selection.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No of Feature Removed</th>
<th>TLBO + RBFN</th>
<th>MLP</th>
<th>Simple Logistic</th>
<th>iTLBO + RBFN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hepatitis</td>
<td>6</td>
<td>89.4615</td>
<td>89.7436</td>
<td>82.0513</td>
<td>93.5897</td>
</tr>
<tr>
<td>House-votes</td>
<td>5</td>
<td>98.6239</td>
<td>96.789</td>
<td>93.0876</td>
<td>99.0826</td>
</tr>
<tr>
<td>Mammographic</td>
<td>2</td>
<td>78.1705</td>
<td>81.4969</td>
<td>81.3546</td>
<td>83.7838</td>
</tr>
<tr>
<td>Horse Colic</td>
<td>7</td>
<td>85.8696</td>
<td>80.4348</td>
<td>81.0362</td>
<td>87.5</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>3</td>
<td>95.1429</td>
<td>93.7104</td>
<td>92.8571</td>
<td>99.1429</td>
</tr>
<tr>
<td>Diabetes</td>
<td>2</td>
<td>73.4375</td>
<td>69.7129</td>
<td>70.7291</td>
<td>81.7708</td>
</tr>
<tr>
<td>Post-operative</td>
<td>2</td>
<td>66.6667</td>
<td>63.4144</td>
<td>62.5434</td>
<td>75.5556</td>
</tr>
</tbody>
</table>

### Table 6.
**Further Details Analysis of TLBO + RBFN (Table 5).**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>Kappa Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hepatitis</td>
<td>89.7</td>
<td>78.3</td>
<td>86.7</td>
<td>89.7</td>
<td>87.8</td>
<td>0.1545</td>
</tr>
<tr>
<td>House-votes</td>
<td>98.6</td>
<td>13</td>
<td>98.6</td>
<td>98.6</td>
<td>98.6</td>
<td>0.9705</td>
</tr>
<tr>
<td>Mammographic</td>
<td>78.3</td>
<td>27.3</td>
<td>77.9</td>
<td>78.3</td>
<td>77.9</td>
<td>0.5218</td>
</tr>
<tr>
<td>Horse Colic</td>
<td>85.9</td>
<td>18</td>
<td>85.8</td>
<td>85.9</td>
<td>85.7</td>
<td>0.6917</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>95.1</td>
<td>4</td>
<td>95.2</td>
<td>95.1</td>
<td>95.2</td>
<td>0.9015</td>
</tr>
<tr>
<td>Diabetes</td>
<td>73.4</td>
<td>34.5</td>
<td>72.8</td>
<td>73.4</td>
<td>72.7</td>
<td>0.405</td>
</tr>
<tr>
<td>Post-operative</td>
<td>66.7</td>
<td>56.7</td>
<td>62.8</td>
<td>66.7</td>
<td>61.7</td>
<td>0.1176</td>
</tr>
</tbody>
</table>

### Table 7.
**Further Details Analysis of iTLBO + RBFN (Table 5).**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>Kappa Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hepatitis</td>
<td>92.3</td>
<td>15.0</td>
<td>92.5</td>
<td>92.5</td>
<td>92.1</td>
<td>0.81</td>
</tr>
<tr>
<td>House-votes</td>
<td>99.1</td>
<td>1.0</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>0.98</td>
</tr>
<tr>
<td>Mammographic</td>
<td>83.8</td>
<td>16.0</td>
<td>83.9</td>
<td>83.8</td>
<td>83.8</td>
<td>0.67</td>
</tr>
<tr>
<td>Horse Colic</td>
<td>87.5</td>
<td>17.1</td>
<td>87.5</td>
<td>87.5</td>
<td>87.3</td>
<td>0.72</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>99.1</td>
<td>0.003</td>
<td>99.2</td>
<td>99.1</td>
<td>99.1</td>
<td>0.98</td>
</tr>
<tr>
<td>Diabetes</td>
<td>81.8</td>
<td>25.8</td>
<td>81.5</td>
<td>81.8</td>
<td>81.6</td>
<td>0.57</td>
</tr>
<tr>
<td>Post-operative</td>
<td>75.6</td>
<td>75.6</td>
<td>57.1</td>
<td>75.6</td>
<td>65.0</td>
<td>0.00</td>
</tr>
</tbody>
</table>
5. Conclusions
An integrated approach of iTLBO and RBFN has been proposed for making a classifier to classify unseen data by carefully considering the issues like missing values and dimensionality reduction. The approach undergoes three different phases before drawing any conclusions. In first phase, preprocessing task like missing value imputation is carried out by least square estimator. In second phase by Relief the relevant attributes are selected. Finally in the third phase a classifier is built by integrating iTLBO and RBFN. Determining the optimum key parametric values of RBFN, iTLBO is adopted. After careful training, the model was tested and it was noticed that in all datasets, iTLBO + RBFN is performing better than TLBO + RBFN in the case of complete dataset. Our bag of future research includes applications in big data and more parametric analysis of iTLBO in correspondence with the natural teaching-learning process.

References


**Corresponding author**
Satchidananda Dehuri can be contacted at: satchi.lapa@gmail.com

For instructions on how to order reprints of this article, please visit our website: [www.emeraldgrouppublishing.com/licensing/reprints.htm](http://www.emeraldgrouppublishing.com/licensing/reprints.htm)
Or contact us for further details: permissions@emeraldinsight.com