Decision Analytics Journal 6 (2023) 100164

Contents lists available at ScienceDirect

Decision Analytics Journal

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

An outliers detection and elimination framework in classification task of data mining

Ch. Sanjeev Kumar Dash [a,*], Ajit Kumar Behera [a], Satchidananda Dehuri [b], Ashish Ghosh [c]

[a] Department of Computer Science & Engineering, Silicon Institute of Technology, Bhubaneswar 751024, Odisha, India
[b] Department of Computer Science, Fakir Mohan University, Balasore 756019, Odisha, India
[c] Machine Intelligence Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata 700108, India

ARTICLE INFO

Keywords:
Radial basis function neural networks
Outlier detection
Winsorizing
Classification
Principal component analysis
Teaching–learning-based optimization

ABSTRACT

An outlier is a datum that is far from other data points in which it occurs. It can have a considerable impact on the output. Therefore, removing or resolving it before the analysis is essential to prevent skewing. Outliers in a survey sampling can have a significant outcome on statistical results. The goal of discovering outliers in data mining is to find a pattern in data that does not conform to expected behavior. In this paper, we have proposed a framework in which a popular statistical approach termed Inter-Quartile Range (IQR) is used to detect outliers in data and deal with them by Winsorizing method. A radial basis function network trained by teaching a learning-based optimization model is developed using the preprocessed dataset under this framework. A few standard University of California Irvine (UCI)datasets are employed to measure the framework's effectiveness. The outcome of the experiments shows that our proposed framework can be a viable alternative tool for the classification task of data mining where prior outliers preprocessing is necessary.

1. Introduction

In machine learning (ML), classification [1] task of data mining is treated as a supervised learning. To build a classification model via machine learning, an accurate training dataset is needed, but assigning raw data for the appropriate class is prone to error. In the classification process, one of the critical elements corresponding to the data set is an outlier that might impact on the performance of the classifier. In real-world data, outliers are common [2–5]. Outliers can be found in any statistical data, as many variables are being recorded. An outlier is a value that differs extensively from the remaining part of the data. The outliers can occur due to a mistake during data collection, measurement errors, data processing errors, sampling errors, or variance in our data. During the generation, collecting, processing, and analysis of data, outliers can appear from a variety of sources and hide in a variety of dimensions. Although outliers cause serious problems in statistical analyses, they may help to capture valuable information that is part of our study area. Detecting and eliminating outliers is the act of removing individual data vectors from a larger set of data, which is critical in nearly any quantitative field such as ML. The quality of the data is just as crucial as the quality of the classification model in machine learning and any quantitative discipline. As a result, we must identify and address them.

The outlier detection methods are broadly divided into two types: parametric and non-parametric. Probabilistic and statistical modelling are examples of parametric techniques. Linear regression and proximity-based modeling are examples of non-parametric techniques. Many strategies for detecting outliers have been presented in recent years. Inter-Quantile Range Technique has recently been employed by Durai et al. in [6] to identify outliers for smart farming. In this work, we have used an accepted statistical method Inter-Quartile Range (IQR) to detect outliers in data and deal with them using the Winsorizing method. For classification problem, it is discovered that radial basis function network (RBFN) can estimate any continuous multivariate function. The center of gravity and distribution of the networks must be revealed throughout the training method of RBFNs in order to increase their performance. This work employs a rapid input selection strategy to take away irrelevant neural inputs before building an efficient RBFNN model. Secondly, the RBFN's parameters are tuned using an empirical Teaching–Learning-Based-Optimization (TLBO) method to produce the best model.

This article is set out as follows. The related work is described in Section 2. The RBF network with outlier detection, handling outliers, principal component analysis (PCA), and TLBO is described in Section 3. Section 4 describes the proposed strategy. Section 5 presents the experimental setup, findings, and analysis. Finally, Section 6 of the article concludes with a proposal for future research.

* Corresponding author.
E-mail addresses: sanjeev_dash@yahoo.com (C.S.K. Dash), ajit_behera@hotmail.com (A.K. Behera), satchi.lapa@gmail.com (S. Dehuri), ash@isical.ac.in (A. Ghosh).

https://doi.org/10.1016/j.dajour.2023.100164
Received 17 May 2022; Received in revised form 13 January 2023; Accepted 13 January 2023
Available online 18 January 2023

2. Related work

In this section, we will study some of the works which are very much related to this work. Baykasoğlu et al. [7] looked at how well the TLBO method performed on scheduling problem. It was found that TLBO has more potential in comparison to another heuristic algorithm. Singh et al. [8] presented about how to use the TLBO algorithm to optimize DOCR(Directional Over Current Relay)relay coordination in a looping power system. They lessen the time needed for the primary and emergency relays to operate by modifying the objective function. The results were compared to the updated differential evolution algorithm's optimal values for the time dial setting and Plug setting values. Rao et al. [9] have used machine learning techniques for the process of parameter optimization works. They have used two different multi-objective problems in their work. Their findings show that the new approach superior population size, generation number as well as computation time.

Kiziloz et al. [10] have proposed TLBO as an FSS(Feature Subset Selection)mechanism. This technique does not necessitate the tuning of any parameters throughout the optimization process. Most traditional meta-heuristics algorithms require extra work to tune their parameters that can have a negative impact on their efficiency. Kumar et al. [11] suggested a new technique for simultaneous project selection with the goal of maximizing the portfolio's total expected benefit. Three meta-heuristics were built and as compared to present algorithms: TLBO, TS(Turbo Search), and hybrid TLBO-TS. The algorithms' performance is assessed using different data sets. The suggested hybrid TLBO-TS algorithm demonstrated high solution quality and good convergence rate. Naik et al. [12] integrated TLBO with the gradient descent training to arrive at the optimal weights for the FLANN model. For solving mechanical design difficulties, Rao et al. [13] proposed TLBO. On the basis of the best solution, convergence rate, and computing effort, TLBO approach has better performance in comparison to other optimization techniques. The outcomes imply that TLBO surpasses other optimization methodologies in terms of efficacy and efficiency for the mechanical design optimization challenges addressed. For discovering global solutions, Rao et al. [14] devised a massive nonlinear optimization problem. The proposed strategy is based upon the stimulus of teacher's influence on student performance in a classroom. The method's effectiveness is evaluated using a variety of benchmark problems with varying features, and the findings are compared to those obtained using alternative population-based methods. In their work, Rao et al. [15] have worked on both unconstrained and constrained activities in past. Their approach is focused on impact of a teacher's influence on a classroom's student output. The technique is examined on 25 unconstrained and 35 restricted benchmark functions with numerous properties. TLBO is examined using several constraint management strategies for restricted benchmark functions. It includes features like self-adaptive penalty, constraint, stochastic ranking. The TLBO method's performance is compared to that of existing optimization techniques, and the findings reveal that the suggested algorithm performs better. Rao et al. [16] proposed the TLBO method, as well as its elite and non-dominated multi-objective versions. To exemplify the algorithm's procedural phases, two instances of unconstrained, constrained benchmark functions, and a multi-objective constrained issue. Dash et al. [17] established a unique technique for developing a classifier with missing values and irrelevant characteristics by combining TLBO with RBF network. They have used least square estimator assigning missing values and used relief technique for evaluating relevance attributes. Guo et al. [18] recommended a new approach for handling nonlinearity in data sets using principal component analysis (PCA) and RBFNN to handle the no-Gaussian problem. The suggested RBFNN-PCA technique is a trustworthy system since it is an effective extension to the linear PCA approach.

For network intrusion detection, Aljanabi et al. [19] used ITLBO-IPJAYA (Improved Teaching–learning-based optimization algorithm, improved parallel JAYA) approach whose consequences has been as compared to those of TLBO, ITLBO, and ITLBO-JAYA. The ITLBO-IPJAYA consequences confirmed better balance and higher accuracy than ITLBO and ITLBO-JAYA algorithms. Several robust approaches and outlier detection tools have been presented by Rousseeuw et al. [20]. We cover location and scatter estimation, linear regression, principal component analysis, and classification as well as other robust techniques for univariate, and high-dimensional data. Vinutha et al. in [21] employed the interquartile Range approach to find outliers. The input range is separated into quartiles on a continuous scale, which are then analyzed to discover outliers in this approach. The collected outliers are then removed using a remove with value filter. The experiment is carried out with the help of the Weka data mining tool. Rivest et al. [22] propose approaches for reducing the influence of outliers for disaggregated levels while maintaining the aggregated values. Ben-Gal et al. [23] describe many outliers detection approaches, distinguishing between univariate and multivariate strategies, as well as parametric and nonparametric procedures.

Shao et al. [24] transformed dam monitoring data into a binary image of a scatter plot. After Otsu binarization, the gray scales of solitary points (outliers) are decreased by Gaussian blur and subsequently deleted. Then, using the Cuckoo Search (CS) method, the most connections between the pixel aggregations are obtained, automatically separating outliers of the clustered-pattern from the process line. A brand-new supervised outlier estimator is proposed by Fernández et al. in [25]. This is accomplished by coupling an outlier detector to a supervised model in such a way that the targets of the latter model oversee the selection of all the hyperparameters involved in the outlier detector. In order to quantify the weight of various indexes in multi-dimensional data and ascertain the impact of different qualities on the prediction outcomes, Yang et al. [26] first developed a new index weight measuring approach that was paired with information entropy. Then, based on the separation between the target sequence and the non-self-match, they created a new sliding window and sub-sequence measuring mechanism to determine whether the data is abnormal. After that, they created a pruning approach to further simplify the algorithm's computations.

Coelho et al. [27] have combined two different approaches for successful downtime prediction: (a) time segmentation along with feature dimension reduction and anomaly detection; and (b) machine learning classification algorithms. Li et al. [28] have concentrated on semi-supervised outlier detection using a vast amount of unlabeled data and few detected abnormalities. A distribution construction sub-module and a data augmentation sub-module are then presented to find discrete anomalies and partially identifiable group anomalies, respectively, from the task of semi-supervised outlier detection. Due to the combination of the two sub-modules, the dual multiple generative adversarial networks (Dual-MGAN) are able to recognize both discrete and partially recognized group abnormalities.

In order to build an unsupervised anomaly detection ensemble, Kandanaarachchi et al. [29] used Item Response Theory (IRT), a class of models used in educational psychometrics to evaluate student and test question characteristics. Because the latent characteristic can be used to find the hidden ground truth, IRT's computation of latent traits is well suited for anomaly identification. They create an ensemble that can minimize noisy, complex anomaly detection problems using a unique IRT mapping.

For the purpose of offering a useful and accurate way to handle real-life online scenarios, Scaranti et al. [30] have concentrated on avoiding the requirement for labeling and prior information. The cluster's structure is projected over the feature space in our approach, opening the door for a thorough study and revealing information on assault kind, intensity, and seasonality.

When a model is trained with normal data, a resampling approach is performed to the latent space vector; however, this can make the model perform badly if the data also contain aberrant data. In order

to assess the consistency of the input clip, Hao et al. [31] merged an input clip with a generated frame to create a reformed video clip, which was then fed into the discriminator built by the 3D CNN(Convolutional Neural Network).

The deep adversarial anomaly detection (DAAD) approach using task-specific characteristics was proposed by the authors in [32]. The proposed approach uses adversarial training and self-supervised learning to get around the drawbacks of inference-based approaches. The DAAD technique is quicker and more effective than GAN-based approaches. The authors in [33] proposed a novel anomaly detection approach that represents the original time series data and their amplitude data using interval information granularity representation based on the justified granularity principle to provide the appropriate representation results.

An innovative approach to creating an ensemble of one-class classifiers (OCCs) utilizing a WA(Weighted Averaging) fusion has been given in [34] in order to enhance classification performance. Comparing the suggested model method to cutting-edge techniques, it confirms its usefulness. A deep learning-based probabilistic anomaly detection technique for solar power forecasting was developed by the authors in [35]. The case study's findings under various assault scenarios demonstrated that the created probabilistic anomaly detection system could accurately and successfully identify a variety of cyberattacks.

Finding anomalous sections is made easier by a unique approach that was put out in [36] and is based on transformer and self-supervised learning. According to experimental findings, self-supervised learning can boost efficiency on a small dataset of unlabeled images. Based on the ICP-OPTICS (Independent Central Point OPTICS)algorithm, the authors of [37] suggested a brand-new semi-supervised technique for identifying anomalous continuous glucose monitoring (CGM) readings. It is possible to automatically configure the suggested detector application without any prior expertise. To demonstrate the viability and superiority of the technique, CGM systems are subjected to online anomaly detection, and greater accuracy rates are obtained.

The use of extreme gradient boosting (XGBoost) and long short-term memory-based stacked denoising autoencoders (LSTM-SDAE) as an anomaly identification and diagnostic approach for wind turbines is proposed in [38]. The outcomes demonstrate the effectiveness of the suggested approach in the identification and detection of anomalies in wind turbines.

For the purpose of detecting anomalous events in contexts with and without monitoring, the authors devised a dual stream CNN architecture [39]. Compared to current anomaly detection approaches, the suggested framework offers a better balance between accuracy and computation.

3. Background

We give an overview of RBFN, Winsorizing-based outlier detection, PCA, and TLBO in this section. These are the foundations upon which our work has been built.

3.1. RBF network classifier

A supervised learning technique [40–43] is used in the RBF network to create innovative and possibly valuable models. With certain benefits over a multilayer neural network, it is the most extensively used network in function approximation.

The RBFN is feed-forward network. The input neurons are fed in the first layer, hidden units that implement a radial activation function in the second layer, and in the third layer, implement the weighted sum hidden units to produce the output (see Fig. 1).

The second layer performs nonlinear transformation with Gaussian kernel, as expressed With inside as expressed:

$$\phi_i(x) = \exp\left(\frac{\|x-\mu_i\|^2}{2\sigma_i^2}\right) \tag{1}$$

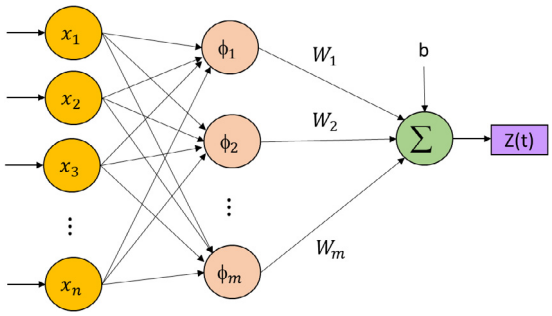

Fig. 1. Radial basis neural network.

where, μ_i, σ_i and ϕ_i are center, spread, and output for the ith neuron. Weighted connections w_i are formed by the connectivity between the middle and final layers. The network's response to the outside world is provided by the output layer, which is a summing unit.

3.2. Outlier detection

Different methods are used to detect outliers [44,45]. One of the methods is visualization which is similar to Boxplot, Histogram, Scatter Plot, IQR methods. There are two ways to handle outliers:

(i) **Handling Error Outliers:** In this method, every error should either be removed or corrected. If there is data available within significant values, then the original entry of the data points to avoid a significant loss of information through deletion. If the data will have some error, then the best method is to simply remove the entries.

(ii) **Handling Non-Error Outliers:** There are three methods are used to handle non-error outliers: *keep*, *delete*, *record*. (i) **Keep**-When keeping outliers, be aware that they can distort the results of your actual task. (ii) **Delete**- The most straight forward option is to delete any outlying observation. (iii) **Recode**- Avoid the loss of large amount of data using winsorizing method for handling outliers. The usage of winsorizing is presented at Section 4.

3.3. Principal component analysis

PCA [46] is used for reducing the dimension of big data i.e., condensing a large number of attributes into a smaller number. Consider a $n \times q$ data matrix X, in which each row signifies a separate test repetition and every q column indicates a specific attribute.

A collection of x-dimensional weights is used to specify the transformation mathematically or coefficients $W_{(k)} = (W_1, \ldots, W_q)_{(k)}$ of size one that transform the row vector $x_{(j)}$ to $s_{(j)} = (s_1, \ldots, s_l)_{(j)}$ such that $s_{m(j)} = x_{(j)} * W_{(m)}$, where j = 1, …,n and m = 1, …,l.

3.3.1. First component

In order to maximize, $w_{(1)}$ has to satisfy

$$W_{(l)} = \arg \max_{\|W\|=1} \left\{ \sum_j (s_1)^2_{(j)} \right\} = \arg \max_{\|W\|=1} \left\{ \sum_j (x_{(j)} * W)^2 \right\} \tag{2}$$

The matrix form is expressed as:

$$W_{(l)} = \arg \max_{\|W\|=1} \left\{ \|X\,W\|^2 \right\} = \arg \max_{\|W\|=1} \left\{ W^T X^T X\, W \right\} \tag{3}$$

Because (W)1 is a unit vector, it also meets the criteria

$$W_{(l)} = \arg \max \left\{ \frac{W^T X^T X W}{W^T W} \right\} \tag{4}$$

In the transformed coordinates, A score can be determined for the first main component of a data vector as: $s1_{(j)=}x_{(j)} * W_{(1)}$.

3.3.2. Further components

The mth component is computed by difference initial m-1 principal components and X.

$$\hat{X}_m = X - \sum_{t=1}^{m-1} X\, W_{(t)} W_{(t)}{}^T \tag{5}$$

$$W_{(m)} = \arg \max_{\|W\|=1} \left\{ \left\| \hat{X}_m W \right\|^2 \right\} = \arg \max \left\{ \frac{W^T \hat{X}_m^T \hat{X}_m W}{W^T W} \right\} \tag{6}$$

This, it turns out, generates the remaining eigen vectors of $X^T X$, and their corresponding eigen values giving maximum values. As a result, the weight vectors are $X^T X$ eigen vectors.

The mth principal constituent of $x_{(i)}$ is expressed as $s_{m(j)} = x_{(j)} \cdot W_{(m)}$ in converted co-ordinates, or $W_{(m)}$ is the eigenvector of $X^T X$, and the equivalent vector of the original variables $x_{(j)} * W_{(m)} W_{(m)}$.

As a result, the whole principal component decomposition of X can be written as:

$$S = X * W \tag{7}$$

where *we* is a weight matrix with columns $X^T X$.

3.4. Teaching–learning based optimization

Genetic Algorithm (GA), Evolution approach, chemical reaction optimization (CRO), and swarm intelligence (SI) are a few well-known evolutionary and metaheuristic algorithms [47–52]. All evolutionary algorithms are probabilistic, and they all use the same set of governing factors, such as population size and generation number. Every algorithm needs its own set of control parameters in addition to the standard ones. The proper adjustment of parameters influences the performance of an algorithm. Incorrect parameters selection both will increase processing attempt or produces the nearby great result. Rao et al. used this phenomenon and established the TLBO approach without algorithm-specific parameters. Only a few fundamental regulating criteria, such as population size and generation, are required for successful usage of TLBO. The TLBO method has become quite famous among optimization experts.

4. Proposed method

The proposed approach is divided into three stages. In the first phase, outlier is detected using Inter Quartile Range (IQR) method and dealt them by Winsorizing method. In the second phase, PCA is a dimensionality reduction technique that generate lower-dimensional statistics at the same time as retaining as a lot variety as possible, every statistics factor is projected onto most effective the primary few fundamental components. In the third phase, we are focusing on the learning of the RBFN classification model. Here, teaching–learning optimization is engaged to reveal the center and spread of the RBFN.

4.1. Outlier detection using inter quartile range (IQR)

In this method, we calculate the Inter Quartile Range [42,53] which tells us the variation in the data. Any value which lies outside the range of (25th Percentile–1.5x **Inter Quartile Range**) to (75th Percentile + 1.5x **Inter Quartile Range**)are noted as outliers where IQR is defined as 75th Percentile – 25th Percentile (see Fig. 2).

After detecting the outlier, winsorizing technique is used to reduce the effects of outliers in the data set. With winsorizing, each of the variables' distribution, any value of a variable above or below a percentile k is substituted with the value of the kth percentile itself. In our project, we used winsorizing to recode all the outliers detected by the IQR method. All the value s less than (25th percentile–1.5*IQR) and more than (75th percentile + 1.5*IQR) are substituted to (25th percentile–1.5*IQR) and (75th percentile + 1.5 *IQR) respectively.

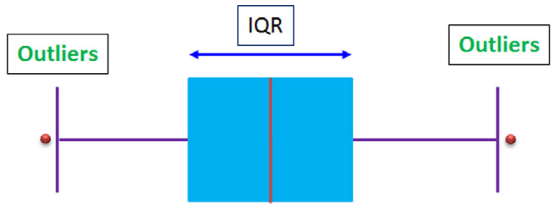

Fig. 2. IQR method for outlier detection.

4.2. Dimensionality reduction using PCA

The PCA [46] is used to reduce dimensionality via way of means of projecting every data point onto most effective the primary few principal components, ensuing in lower-dimensional data with as tons variance as feasible.

Using Eq. (7), $x_{(j)}$ is a data vector that transforms an initial space of x variables into a new space that is uncorrelated across the dataset. The truncated transformation is obtained by selecting the first M principal components, which are obtained by utilizing the first M eigenvectors:

$$S_M = A(WT_M) \tag{8}$$

Where T_M now contains n rows and M columns. In other words, PCA discovers a linear transformation $s = (WT)_M^T x, x \in R^Q, s \in R^M$ in which the columns of the $Q \times M$ matrix $(WT)^M$ serve as an orthogonal basis for the decorrelated M features. The variation in the original data is maximized while the overall squared reconstruction error is minimized with this score matrix, out of all the converted data matrices with just L columns. $\left\| S(WT)^T - S_M(WT_M^T) \right\|_2^2 \|A - A\|_2^2$.

4.3. Learning of RBFN classifier

In the third phase, we are classifying data set using RBFN [54,55]. The challenge of selecting an appropriate quantity of basis functions is a significant concern for RBFN. Because RBFNs are mostly determined by the kernel's center and width, we simply encode the centers and widths for the stochastic search using TLBO. Here, teaching–learning optimization is applied to get the center and spread of the RBFN. First, we will go over the basics of TLBO, and then we will move on to the TLBO + RBFN network, which is designed to improve classification accuracy.

4.3.1. Teaching–learning based optimization algorithm

TLBO's operation is split into two phases: (i) Teacher phase and (ii) Learner phase. The process of the above phase is descried below.

(i) Teacher Phase In this phase, a teacher uses his or her talents to try to enhance the class's mean outcome in the subject he or she teaches. Assume there are 'm' subjects, 'n' learners, and $N_{j,i}$ the average outcome of the learners in a given subject 'j' at any iteration i. The best overall result is: $A_{total1=pbest1,1}$.

The effect of the best learner k-best might be considered the sum of all the topics attained in the total population of learners. However, because a teacher is typically thought of as a highly educated somebody who instructs students in order for them to get higher results, the algorithm considers the best learner identified as the instructor. It is determined the difference between the current mean performance for each topic and the teacher's equivalent result for each subject as follows:

$$Difference1\_Mean1_{m,p,1} = r_i(A_{m,pbest1} - T_F N_{m,1}) \tag{9}$$

where $A_{m,pbest1}$ is the best learner. The mean value is determined by the teaching factor T_F, while r_i is a random number between 0 and 1. The value of T_F is calculated as shown in Eq. (10).

$$T_F = round[1 + rand(0,1)\{2-1\}] \tag{10}$$

Center				Width				Bias
c_1	c_2		c_{Kmax}	σ_1	σ_2		σ_{kmax}	b

Fig. 3. Structure of the individual.

After performing a series of experiments, it is discovered that the method performs best when T_F is between [1,2]. However, the algorithm performs considerably better when the value of T_F is either 1 or 2, therefore to simplify the procedure, the teaching factor should be either 1 or 2, based on the rounding up criteria provided by Eq. (10). Based on the $Difference1\_Mean1_{m,p,l}$, during the teacher phase, the existing solution is modified using the formula below:

$$A'_{m,p,l} = A_{m,p,l} + Difference1\_Mean1_{m,p,l} \tag{11}$$

where $A'_{m,p,l}$ is the updated value of $A_{m,p,l}$. If $A'_{m,p,l}$ delivers a better function value, it is acceptable. At the end of the teacher phase, all valid function values are kept and used to start the learner phase, which is reliant on the teacher phase.

(ii) The learner phase: A learner connects with other learners at random in order to improve his or her knowledge. If another student has greater knowledge than the learner, the learner learns new things. The learning processes of this period are described below:

Choose two learners P and Q randomly as a result $A'_{total1=P,l} \neq A'_{total1=Q,l}$ (where, $A'_{total1=P,l}$ and $A'_{total1=Q,l}$ are the updated function values of $A_{total1=P,l}$ and $A_{total1=Q,l}$ of P and Q, respectively).

$$A''_{m,P,l} = A'_{m,P,l} + r_i(A'_{m,P,l} - A'_{m,Q,l}),\ If\ A^l_{total1-P,l} < A'_{total1-Q,l} \tag{12}$$

$$A''_{m,P,l} = A'_{m,P,l} + r_l(A'_{m,Q,l} - A'_{m,P,l}),\ If\ A^l_{total1-Q,l} < A'_{total1-P,l} \tag{13}$$

$X''_{m,p,l}$ is accepted if it offers a higher feature value.

The Eqs. (12) and (13) are used for minimization problems. For maximization problems, Eqs. (14) and (15) are used.

$$A''_{m,R,i} = A'_{m,P,l} + r_l(A'_{m,P,l} - A'_{m,Q,l}),\ If\ A^l_{total1-Q,l} < A'_{total1-P,l} \tag{14}$$

$$A''_{m,P,l} = A'_{m,P,l} + r_l(A'_{m,Q,l} - A'_{m,P,l}),\ If\ A^l_{total1-P,l} < A'_{total1-Q,l} \tag{15}$$

Although TLBO may be used to construct centers, widths, and weights that join kernel and output nodes at the same time, we will only focus on developing centers and spreads here. This guarantees that a TLBO individual is effectively represented. If all of these parameters are stored, the length of the individual becomes too long, and the search space becomes too large, resulting in a very slow convergence rate. Because the RBFNs' performance is mostly determined by the kernel's center and width, we simply encode the centers and widths into an individual for stochastic search.

Assuming the maximum number of kernel is set to K_{max}, the structure of the individual is represented in Fig. 3.

Putting it another way, each individual is made up of three parts: center, width, and bias. The individual is $2K_{max} + 1$.

Eq. (16) defines the fitness function that is used to steer the search process.

$$f(x) = \frac{1}{N}\sum_{i=1}^{N}(t_n - \hat{\Phi}(\vec{x}_i))^2 \tag{16}$$

where N is the total number of individuals, t_n is the actual output and $\hat{\Phi}(\vec{x}_i)$ is the expected output. The process of calculating weights in the second phase of learning reduces to solving a simple linear solution after the centers and widths are fixed. The pseudo inverse approach is used to determine a setoff ideal weight in this study.

Because the centers, widths, and bias are calculated from the training data, those can be expressed as:

$$\begin{aligned} &Y = (WT) * \Phi \\ &\Rightarrow WT = (\Phi^T\Phi)^{-1}\Phi^T Y \end{aligned} \tag{17}$$

The overall description of the proposed process is depicted in Fig. 4.

Table 1
Description of datasets.

	Diabetes	Forest Cover	Satellite	Arrhythmia
#Classes	2	7	6	16
#Instances	768	581 012	6435	452
#Attributes	8	12	36	279
Percentage of outliers	268(35%)	2747(0.9%)	2036(32%)	66(15%)

Table 2
Classification accuracy using RBFN + TLBO, Logistic regression and decision Tree.

Datasets	Diabetes	Forest Cover	Satellite	Arrhythmia
RBFN	97.5	**94.46**	**90.28**	**63.46**
RBFN with TLBO	98.25	**95.85**	**91.38**	**64.84**
Logistic regression	77.75	64.54	80.00	60.44
Decision tree	**98.75**	93.15	87.72	62.63

Table 3
Number of attributes dropped for different data sets.

	Diabetes	Forest Cover	Satellite	Arrhythmia
#Attributes dropped using PCA	0	6	24	122

Table 4
Classification accuracy using RBFN + TLBO, Logistic regression and Decision tree using PCA.

Model	Diabetes	Forest Cover	Satellite	Arrhythmia
RBFN + PCA	**97.47**	**94.46**	**91.36**	**65.4**
RBFN + TLBO + PCA	**98.25**	**95.87**	**92.54**	**67.03**
Logistic regression + PCA	78.75	68.80	86.79	65.93
Decision tree + PCA	98.00	92.35	89.36	58.24

5. Experimental study

We briefly outline the datasets and settings that must be set in the experimental investigation in Section 5.1. The results and analysis are presented in Section 5.2.

5.1. Description of dataset

The datasets for this study got here from the UCI machine learning repository [56]. The detail information about the datasets is given in Table 1. Every dataset in our study is split into two mutually exclusive parts: a training set and a testing set. Table 1 shows the parameter values utilized to validate our proposed method. For experiment, we have used MATLAB 2018a.

5.2. Results and analysis

Table 2 offer an overview of the experimental effects without PCA. We have divided the results of the classification model into three types based on empirical validation. From the table, it is found that RBFN with TLBO gives better results than logistic regression and decision tree [39] for Forest Cover, Satellite and Arrhythmia data sets while decision tree has better accuracy result for Diabetes data set.

Table 3 summarizes the details of the number of attributes dropped for different datasets using PCA.

Table 4 presents a summary of the PCA-based experimental results. It is found that RBFN with TLBO gives better accuracy results than other models for all three datasets.

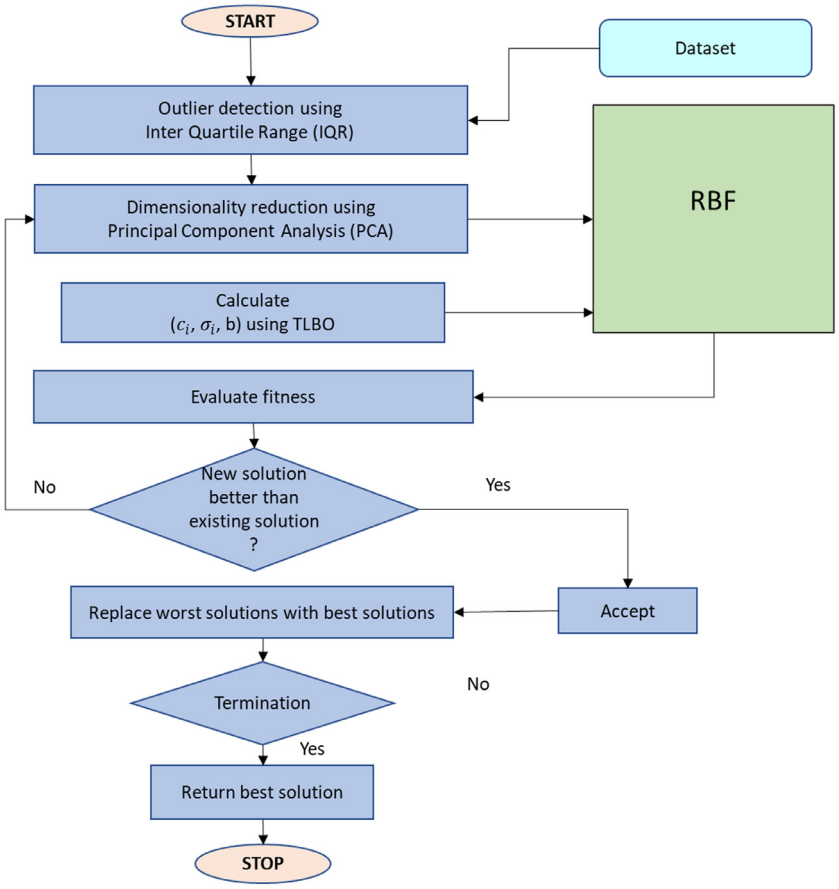

Fig. 4. Flow chat of proposed model.

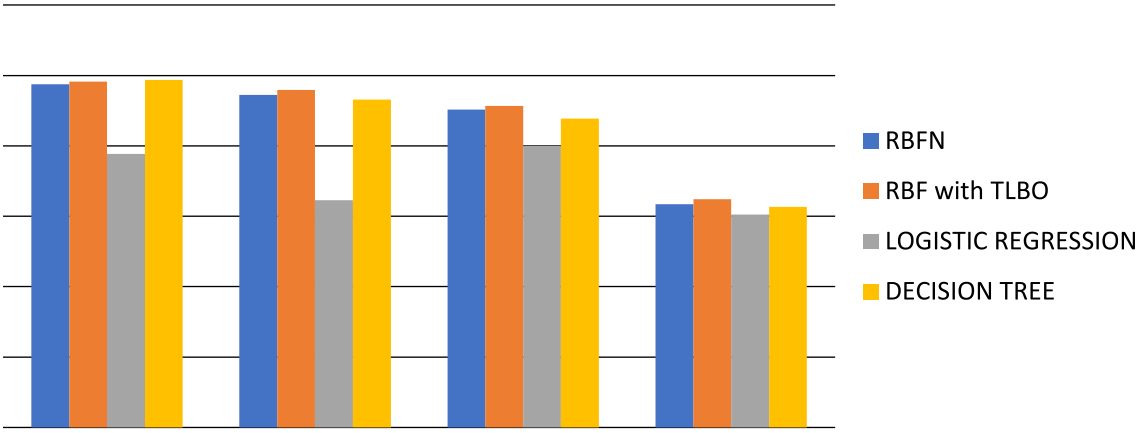

Fig. 5. Classification accuracy of RBFN, RBFN + TLBO, logistic regression and decision tree without PCA.

The comparative study of different models for different datasets without PCA is shown in Fig. 5. Also, the classification accuracy of different models for different datasets with PCA is shown in Fig. 6.

Figs. 7–9 shown the ROC curve. For each learning algorithm on Diabetes dataset, 10 sets of results (one for each of the 10-fold cross-validation partition) were stored. The original data was recorded as a confusion matrix, and the decision criteria for each of the 10 test partitions were modified (to produce the ROC curves). A variety of measures were collected from the raw data in order to evaluate the effectiveness of different learning algorithms on the data set. From Fig. 7, it is noted that the true area under ROC curve is found to be 0.98 using Decision tree. Fig. 8 also shows the value is 0.98 for the proposed RBFN + TLBO algorithm. From Fig. 8 the area under ROC curve is found to be 0.71 for Logistic regression for the Diabetes dataset.

6. Conclusion

In this paper, a framework for data classification has been proposed which works in three phases. In the initial phase, preprocessing task like outlier detection and handling of outlier is carried out. A popular statistical method called Inter Quartile Range (IQR)is used to identify outliers in data. Subsequently the Winsorizing technique is used for normalizing the outlier data. In second phase, PCA is used for dimensionality reduction. The third phase consists of building a TLBO

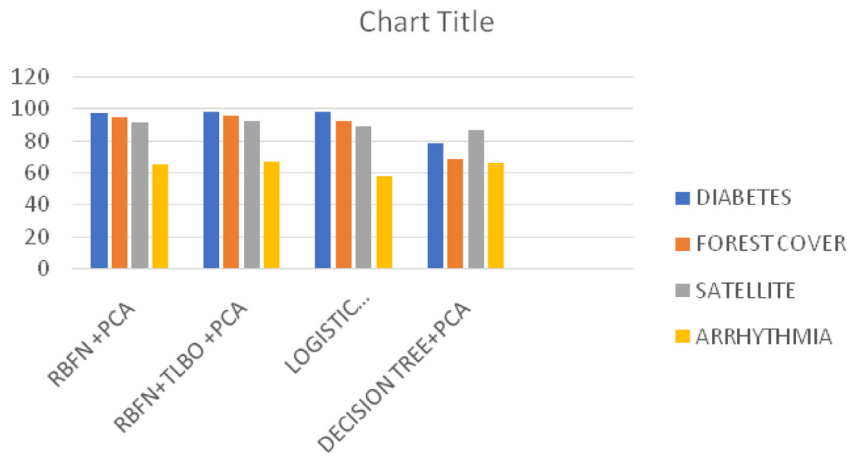

Fig. 6. Classification accuracy of RBFN, RBFN + TLBO, logistic regression, and decision tree with PCA.

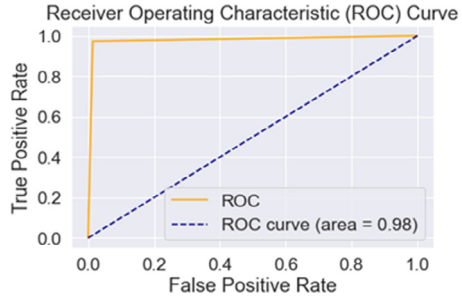

Fig. 7. ROC curve for Diabetes dataset using decision tree without using PCA.

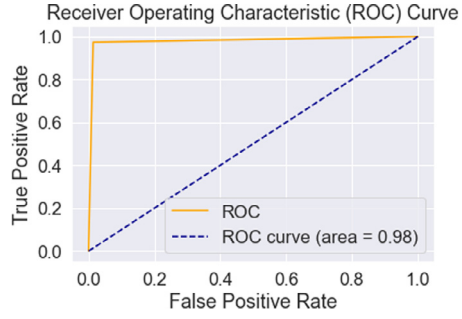

Fig. 8. ROC curve for Diabetes dataset using RBF + TLBO without using PCA.

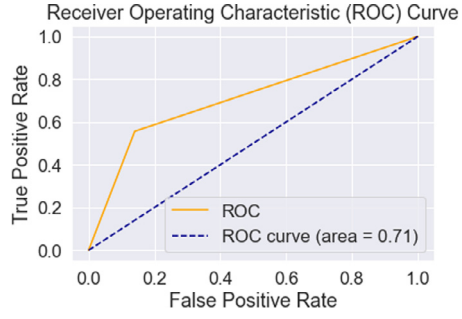

Fig. 9. ROC curve for Diabetes dataset using Logistic regression without using PCA.

trained RBFN based classifier. Finding the best possible topology of RBFN, TLBO can be chosen in searching the best possible parameter set. After cautiously training, the classifier was tested in all datasets. Experimental results reveal that the proposed approach has better accuracy than other competitive approaches. Future research projects include big data applications and a more parametric study of TLBO in accordance with the natural teaching–learning process.

Data availability

Data will be made available on request.

Acknowledgments

Dr. Satchidananda Dehuri, Professor of Computer Science (Erstwhile P. G. Department of Information and Communication Technology), Fakir Mohan University would like to thank SERB, Govt. of India for financial support under Teachers' Associateship for Research Excellence (TARE) fellowship vide file no. TAR/2021/000065 for the period 2021–2024.

References

[1] S. Haykin, Neural Networks and Learning Machines, third ed., Pearson Education, Inc., Upper Saddle River, New Jersey, 2008.
[2] V. Hodge, J. Austin, A survey of outlier detection methodologies, Artif. Intell. Rev. 22 (2) (2004) 85–126.
[3] C.C. Aggarwal, P.S. Yu, Outlier detection for high dimensional data, ACM SIGMOD Rec. 30 (2) (2001) 37–46.
[4] V. Hodge, J. Austin, A survey of outlier detection methodologies, Artif. Intell. Rev. 22 (2) (2004) 85–126.
[5] R.R. Wilcox, Outlier Detection. Encyclopedia of Statistics in Behavioral Science, John Wiley & Sons, Ltd, 2005.
[6] S.K.S. Durai, M.D. Shamili, Smart farming using machine learning and deep learning techniques, Decis. Anal. J. 3 (2022) 100041.
[7] A. Baykasoğlu, A. Hamzadayi, S.Y. Köse, Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases, Inform. Sci. 276 (2014) 204–218.
[8] M. Singh, B.K. Panigrahi, A.R. Abhyankar, Optimal coordination of directional over-current relays using teaching learning-based optimization (TLBO) algorithm, Int. J. Electr. Power Energy Syst. 50 (2013) 33–41.
[9] R.V. Rao, V.D. Kalyankar, Parameters Optimization of Advanced Machining Processes using TLBO Algorithm 20, EPPM, Singapore, 2011, pp. 21–31.
[10] H.E. Kiziloz, A. Deniz, T. Dokeroglu, A. Cosar, Novel multiobjective TLBO algorithms for the feature subset selection problem, Neurocomputing 306 (2018) 94–107.
[11] M. Kumar, M.L. Mittal, G. Soni, D. Joshi, A hybrid TLBO-TS algorithm for integrated selection and scheduling of projects, Comput. Ind. Eng. 119 (2018) 121–130.
[12] B. Naik, J. Nayak, H.S. Behera, A TLBO based gradient descent learning-functional link higher order ANN: An efficient model for learning from non-linear data, J. King Saud Univ.-Comput. Inf. Sci. 30 (1) (2018) 120–139.
[13] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315.
[14] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: an optimization method for continuous non-linear large-scale problems, Inform. Sci. 183 (1) (2012) 1–15.

[15] R.V. Rao, V.J. Savsani, J. Balic, Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems, Eng. Optim. 44 (12) (2012) 1447–1462.
[16] R.V. Rao, Teaching-learning-based optimization algorithm, in: Teaching Learning Based Optimization Algorithm, Springer, Cham, 2016, pp. 9–39.
[17] C.S.K. Dash, A.K. Behera, S. Dehuri, S.B. Cho, Building a novel classifier based on teaching learning based optimization and radial basis function neural networks for non-imputed database with irrelevant features, Appl. Comput. Inform. 18 (1/2) (2020) 151–162.
[18] Y. Guo, K. Li, Z. Yang, J. Deng, D.M. Laverty, A novel radial basis function neural network principal component analysis scheme for PMU-based wide-area power system monitoring, Electr. Power Syst. Res. 127 (2015) 197–205.
[19] M. Aljanabi, M.A. Ismail, V. Mezhuyev, Improved TLBO-jaya algorithm for subset feature selection and parameter optimisation in intrusion detection system, Complexity 2020 (2020) 5287684.
[20] P.J. Rousseeuw, M. Hubert, Robust statistics for outlier detection, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 1 (1) (2011) 73–79.
[21] H.P. Vinutha, B. Poornima, B.M. Sagar, Detection of outliers using interquartile range technique from intrusion dataset, in: Information and Decision Sciences, Springer, Singapore, 2018, pp. 511–518.
[22] L.P. Rivest, M. Hidiroglou, Outlier treatment for disaggregated estimates, in: Proceedings of the Section on Survey Research Methods, American Statistical Association, 2004, pp. 4248–4256.
[23] I. Ben-Gal, Outlier detection, in: Data Mining and Knowledge Discovery Handbook, Springer, Boston, MA, 2005, pp. 131–146.
[24] C. Shao, S. Zheng, C. Gu, Y. Hu, X. Qin, A novel outlier detection method for monitoring data in dam engineering, Expert Syst. Appl. 193 (2022) 116476.
[25] Á. Fernández, J. Bella, J.R. Dorronsoro, Supervised outlier detection for classification and regression, Neurocomputing 486 (2022) 77–92.
[26] Y. Yang, C. Fan, L. Chen, H. Xiong, IPMOD: An efficient outlier detection model for high-dimensional medical data streams, Expert Syst. Appl. 191 (2022) 116212.
[27] D. Coelho, D. Costa, E.M. Rocha, D. Almeida, J.P. Santos, Predictive maintenance on sensorized stamping presses by time series segmentation, anomaly detection, and classification algorithms, Procedia Comput. Sci. 200 (2022) 1184–1193.
[28] X. Du, J. Yu, Z. Chu, L. Jin, J. Chen, Graph autoencoder-based unsupervised outlier detection, Inform. Sci. 608 (2022) 532–550.
[29] S. Kandanaarachchi, Unsupervised anomaly detection ensembles using item response theory, Inform. Sci. 587 (2022) 142–163.
[30] G.F. Scaranti, L.F. Carvalho, S.B. Junior, J. Lloret, M.L. Proença Jr., Unsupervised online anomaly detection in software defined network environments, Expert Syst. Appl. 191 (2022) 116225.
[31] Y. Hao, J. Li, N. Wang, X. Wang, X. Gao, Spatiotemporal consistency-enhanced network for video anomaly detection, Pattern Recognit. 121 (2022) 108232.
[32] X. Zhang, J. Mu, X. Zhang, H. Liu, L. Zong, Y. Li, Deep anomaly detection with self-supervised learning and adversarial training, Pattern Recognit. 121 (2022) 108234.
[33] Y. Zhou, H. Ren, Z. Li, W. Pedrycz, Anomaly detection based on a granular Markov model, Expert Syst. Appl. 187 (2022) 115744.
[34] S. Fatemifar, M. Awais, A. Akbari, J. Kittler, Developing a generic framework for anomaly detection, Pattern Recognit. 124 (2022) 108500.
[35] M. Sun, L. He, J. Zhang, Deep learning-based probabilistic anomaly detection for solar forecasting under cyberattacks, Int. J. Electr. Power Energy Syst. 137 (2022) 107752.
[36] Z. Lin, H. Wang, S. Li, Pavement anomaly detection based on transformer and self-supervised learning, Autom. Constr. 143 (2022) 104544.
[37] Y. Zhao, H. Li, X. Yu, N. Ma, T. Yang, J. Zhou, An independent central point OPTICS clustering algorithm for semi-supervised outlier detection of continuous glucose measurements, Biomed. Signal Process. Control 71 (2022) 103196.
[38] C. Zhang, D. Hu, T. Yang, Anomaly detection and diagnosis for wind turbines using long short-term memory-based stacked denoising autoencoders and XGBoost, Reliab. Eng. Syst. Saf. 222 (2022) 108445.
[39] W. Ullah, T. Hussain, Z.A. Khan, U. Haroon, S.W. Baik, Intelligent dual stream CNN and echo state network for anomaly detection, Knowl.-Based Syst. 253 (2022) 109456.
[40] Y. Kuvvetli, M. Deveci, T. Paksoy, H. Garg, A predictive analytics model for COVID-19 pandemic using artificial neural networks, Decis. Anal. J. 1 (2021) 100007.
[41] P. Chhajer, M. Shah, A. Kshirsagar, The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction, Decis. Anal. J. 2 (2022) 100015.
[42] M. Seyedan, F. Mafakheri, C. Wang, Cluster-based demand forecasting using Bayesian model averaging: An ensemble learning approach, Decis. Anal. J. 3 (2022) 100033.
[43] M. Bansal, A. Goyal, A. Choudhary, A comparative analysis of K-nearest neighbour, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning, Decis. Anal. J. (2022) 100071.
[44] H. Wang, M.J. Bah, M. Hammad, Progress in outlier detection techniques: A survey, IEEE Access 7 (2019) 107964-108000.
[45] H.P. Kriegel, P. Kröger, A. Zimek, Outlier detection techniques, Tutor. KDD 10 (2010) 1–76.
[46] https://en.wikipedia.org/wiki/Principal_component_analysis.
[47] D.R. Carvalho, A.A. Freitas, A hybrid decision tree/genetic algorithm method for data mining, Inform. Sci. 163 (1–3) (2004) 13–35.
[48] D.E. Goldberg, Genetic Algorithm in Search Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[49] S. Forerest, Genetic algorithms: principles of natural selection applied to computation, Science 261 (1993) 872–888.
[50] Michalewicz. Z., Genetic Algorithm + Data Structure = Evolution Programs, Springer Verlag, New York, 1996.
[51] A.K. Behera, M. Panda, S. Dehuri, Software reliability prediction by recurrent artificial chemical link network, Int. J. Syst. Assur. Eng. Manag. (2021) 1–14.
[52] A.K. Behera, C.S.K. Dash, M. Panda, S. Dehuri, R. Mall, A state-of-the-art neuro-swarm approach for prediction of software reliability, Int. J. Adv. Intell. Paradigms 20 (3–4) (2021) 296–322.
[53] D.L. Whaley III, The Interquartile Range: Theory and Estimation (Doctoral dissertation), East Tennessee State University, 2005.
[54] C.S.K. Dash, A.P. Dash, S. Dehuri, S.B. Cho, G.N. Wang, DE+ RBFNs based classification: A special attention to removal of inconsistency and irrelevant features, Eng. Appl. Artif. Intell. 26 (10) (2013) 2315–2326.
[55] C.S.K. Dash, A. Saran, P. Sahoo, S. Dehuri, S.B. Cho, Design of self-adaptive and equilibrium differential evolution optimized radial basis function neural network classifier for imputed database, Pattern Recognit. Lett. 80 (2016) 76–83.
[56] A. Frank, A. Asuncion, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2010, (http://archive.ics.uci.edu/ml).