Feature Data-driven Reinforced Fuzzy Radial Basis Function Neural Network Classifier With the Aid of Preprocessing Techniques and Particle Swarm Optimization

Sang-Beom Park
The University of Suwon

Sung-Kwun Oh (ohsk@suwon.ac.kr)
University of Suwon https://orcid.org/0000-0001-6798-8955

Witold Pedrycz
University of Alberta

Research Article

Keywords: reinforced fuzzy radial basis function neural networks (FRBFNN), preprocessing techniques, particle swarm optimization (PSO), practical application, black plastic wastes

Posted Date: January 11th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1201087/v1

Feature Data-driven Reinforced Fuzzy Radial Basis Function Neural Network Classifier with the Aid of Preprocessing Techniques and Particle Swarm Optimization

Sang-Beom Park[1], Sung-Kwun Oh[1,2], and Witold Pedrycz[3,4,5]

[1.] School of Electrical & Electronic Engineering, The University of Suwon, 17 Wauan-gil, Bongdam-eup, Hwaseong-si, Gyeonggi-do, 18323, South Korea

[2.] Research Center for Big Data and Artificial Intelligence, Linyi University, Linyi, 276005, China

[3.] Department of Electrical & Computer Engineering, University of Alberta, Edmonton T6R 2V4 AB, Canada

[4.] Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah, 21589, Saudi Arabia

[5.] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Abstract

In this study, reinforced fuzzy radial basis function neural networks (FRBFNN) classifier driven by feature extracted data completed with the aid of effectively preprocessing techniques and evolutionary optimization, and its comprehensive design methodology are introduced. An Overall structure of the reinforced FRBFNN comprises the preprocessing part, the premise part and the consequence part of fuzzy rules of the network. In the preprocessing part, four types of preprocessing algorithms such as principal component analysis (PCA), linear discriminant analysis (LDA), combination of PCA and LDA (Hybrid PCA) and fuzzy transform (FT) are considered. To extract feature data suitable to characterize signal data, the feature extraction of information data is carried out through the dimensionality reduction done by the preprocessing technique, and then the reduced data are used as the input to the FRBFNN classifier. In the premise part of fuzzy rules of the network, the number of fuzzy rules is determined according to the number of clusters by fuzzy c-means (FCM) clustering. The fitness values of individual fuzzy rules are obtained based on data distribution. In the consequence part of fuzzy rules of the network, the parameters of connection weights located between the hidden layer and the output layer of FRBFNN classifier are estimated by means of the least square estimation (LSE). Particle swarm optimization (PSO) is exploited for structural as well as parametric optimization in the FRBFNN classifier. The parameters to be optimized by PSO are related to six factors such as the determination of whether to use data preprocessing, the type of data preprocessing technique, the number of input variables reduced by the preprocessing technique, fuzzification coefficient (FC) and the number of fuzzy rules used in fuzzy c-means (FCM) clustering, and the type of connection weights. By using diverse benchmark dataset obtained from UCI repository, the classification performance of the reinforced FRBFNN classifier was evaluated. Through a variety of classification algorithms existed in the Weka data mining software (Weka), the classification performance of the reinforced FRBFNN classifier was compared as well. The superiority of the proposed classifier is demonstrated through Friedman test. Furthermore, we assessed the classification performance of the reinforced FRBFNN classifier applied to black plastic wastes spectral data acquired from Raman and Laser induced breakdown spectroscopy (LIBS) equipment for the practical application of the material sorting system of the black plastic wastes.

Keywords: reinforced fuzzy radial basis function neural networks (FRBFNN), preprocessing techniques, particle swarm optimization (PSO), practical application, black plastic wastes

1. Introduction

In the pattern recognition, the paradigm of neural network classifiers has proved many advantages when it comes to generalization performance, robustness, and criteria of learning abilities (Khan et al. 2018; Yazdi et al. 2012; Oong TH et al. 2011; Jain et al. 2000; Kwan and Cai 1994;). Especially, among a variety of neural network classifiers, multilayer perceptron (MLP) classifiers are widely utilized. However, the MLPs classifier needs lots of parameters to be estimated and the number of iterations required to learn the networks is quite high. Radial basis function neural networks (RBFNN) emerged as a sound alternative to the MLPs in order to supplement such problems (Zhao et al. 2019; David et al. 2009; Apostolikas and Tzafestas 2003; Park et al. 2010). RBFNN is an artificial network applied to the structure of the neurons in order to implement the human's intelligence. RBFNN consists of three layers - input layer, hidden layer, and output layer (Hussain et al. 2021; Tolba and Abu-Rezq 2000). When compared structurally with MLPs, RBFNN have only one hidden layer. Since the structure of the RBFNN is simple, RBFNN show the rapid convergence in terms of the learning procedure of parameters (Alexandridis et al. 2003). Despite this advantage, the RBFNN are not free from limitations. discriminant functions generated by RBFNN exhibit a relatively simple geometry (Oh et al. 2016). To overcome this structural limitation, polynomial-based RBFNN (pRBFNN) with connection weights extended to diverse types of polynomials is introduced. pRBFNN has advantage of making more complicated nonlinear discriminant functions than RBFNN (Er et al. 2002; Oh et al. 2011).

In this study, we design the evolutionarily optimized FRBFNN classifier reinforced with the aid preprocessing technique. The evolutionarily optimized FRBFNN classifier is an algorithm that applied PSO to the RBFNN classifier based on FCM clustering. The PSO can help in the improvement of classification performance by optimizing the parameters of FRBFNN classifier. In the optimization process, six parameters are considered, i.e. the determination of whether to use of data preprocessing, the type of data preprocessing technique, the number of input variables reduced by the preprocessing technique, the number of fuzzy rules, fuzzification coefficient (*FC*) and the type of the connection weights. Data preprocessing technique plays a vital role when extracting the feature vectors of the data in the preprocessing part of the evolutionarily optimized FRBFNN classifier. Principal component analysis (PCA), linear discriminant analysis (LDA), combination of PCA and LDA (hybrid PCA), and fuzzy transform (FT) are used as preprocessing techniques for extraction of the feature vectors of the data such as benchmark datasets. In the hidden layer of the evolutionarily optimized FRBFNN classifier, FCM clustering is used instead of Gaussian functions. In case of Gaussian function, their centroid points and the widths have to be optimized. In contrast, when using FCM algorithm, we do not require the estimation of these parameters. By using FCM algorithm, a lower number of parameters can be optimized. In addition, FCM clustering has an advantage from the viewpoint of the reflection of the characteristics of data because FCM clustering computes the fitness values of the individual clusters by considering the distribution of the data. In the output layer of the evolutionarily optimized FRBFNN classifier, we compute the fuzzy outputs by multiplying fitness values calculated through the FCM clustering and connection weights estimated with the aid of LSE. Through fuzzy inference, final outputs are obtained.

The key issues of the proposed FRBFNN classifier can be briefly outlined as follows:

a) Navigating some parameters to realize the preferable classification performance of feature data-driven FRBFNN classifier using evolutionary optimization: Some factors affecting the classification performance of the FRBFNN classifier (determination of whether to use of data preprocessing, data preprocessing technique type, the number of input variables reduced by the preprocessing technique, the number of fuzzy rules, fuzzification coefficient, and the polynomial type) are optimized by PSO and then we could find the factors that shows the preferred classification performance for each process data type.

b) Demonstration of superiority of the evolutionarily optimized FRBFNN classifier through statistical analysis: Classification performance is compared and analysed by using a variety of classification algorithms existing in the Weka such as Naïve Bayes, SVM and also demonstrate the performance of the evolutionarily optimized FRBFNN classifier through Friedman test which is one of the statistical analysis methods.

c) Practical application of the proposed classifier for sorting black plastic wastes: The classification performance is evaluated by applying to practical sorting system for the classification of black plastic wastes with the aid of the evolutionary optimized FRBFNN classifier proposed in this study; the classification performance is compared with various classification algorithms.

In the sequel, the main contribution of this study is to construct feature data-driven FRBFNN classifier oriented to data characteristics through the extraction of feature vectors based on diverse data-driven preprocessing techniques and structural optimization. The superiority of the proposed classifier is demonstrated through practical application of black plastic wastes sorting system.

This paper is organized as follows. First, in Section 2, we explain data preprocessing techniques for data feature extraction. In Section 3, we introduce the whole structure of the data preprocessing techniques-based FRBFNN classifier realized with the aid of FCM clustering. In Section 4, a comprehensive concept of both the PSO and the architecture of the evolutionarily optimized FRBFNN classifier are illustrated. In Section 5, by using machine learning datasets, classification performance of the proposed classifier is evaluated and compared with Weka classification algorithms through Friedman test, In addition, practical application and its analysis are conducted by the spectral data of black plastic wastes obtained from Raman and LIBS equipment. Finally, concluding remarks are drawn in Section 6.

2. Preprocessing techniques for extraction of feature vector

In this section, we explain the preprocessing techniques used for extracting the feature vectors based on the characteristic of data.

2.1 Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA)

PCA is a preprocessing technique being used in statistical pattern recognition and signal processing to reduce data dimensionality and extract features (Zhao et al 2019; Artoni et al 2018; Yeung and Ruzzo 2001; Kim et al. 2002). PCA is an unsupervised learning algorithm, which exploits only information of the input data (Costa et al. 2017; Romero et al. 2015; Roh et al. 2019). By using PCA, the procedure extracting the feature vectors of the input data is as follows:

[Step 1] Composition of the training data $\mathbf{\Gamma}$ in order to obtain the feature vectors.

$$\mathbf{\Gamma} = [\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \dots, \mathbf{\Gamma}_m]^T \qquad \mathbf{\Gamma}_m = [\gamma_1, \gamma_2, \cdots, \gamma_n] \tag{1}$$

m and n denote the number of data patterns and the number of input variables, respectively. A matrix size of the training data $\mathbf{\Gamma}$ is $(m \times n)$

[Step 2] Calculation of the mean $\mathbf{A}$ and covariance matrix $\mathbf{Cov}$ of the training data $\mathbf{\Gamma}$.

$$\mathbf{A} = \frac{1}{m}\sum_{p=1}^{m} \mathbf{\Gamma}_p \quad \mathbf{A} = [\alpha_1, \alpha_2, \cdots, \alpha_n] \qquad \mathbf{Cov} = \frac{1}{m-1}\sum_{p=1}^{m} (\mathbf{\Gamma}_p - \mathbf{A})(\mathbf{\Gamma}_p - \mathbf{A})^T \tag{2}$$

[Step 3] Computation of the eigenvalues λ and the corresponding eigenvectors $\mathbf{W}$ of the covariance matrix $\mathbf{Cov}$.

$$\mathbf{Cov} \cdot w_j = \lambda_j w_j \qquad \mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n]^T \qquad \mathbf{w}_n = [w_1, w_2, \cdots, w_n] \tag{3}$$

[Step 4] Extraction of the feature vectors from $\mathbf{W}$.

$$\widehat{\mathbf{W}} = [\widehat{\mathbf{w}}_1, \widehat{\mathbf{w}}_2, \cdots, \widehat{\mathbf{w}}_n]^T \qquad \widehat{\mathbf{w}}_n = [\widehat{w}_1, \widehat{w}_2, \cdots, \widehat{w}_k] \tag{4}$$

k stands for the number of extracted features. A matrix size of the eigenvectors $\widehat{\mathbf{W}}$ is $(n \times k)$.

[Step 5] Reduction of the input variables of the training data into $\mathbf{X}$ in the following form

$$\mathbf{X} = \mathbf{\Gamma}^{(m \times n)} \widehat{\mathbf{W}}^{(n \times k)} \tag{5}$$

LDA is one of the feature extraction methods like PCA. The objective of LDA is to maximize the ratio of the between-class scatter of the projected samples and the within-class scatter of the projected samples (Wang et al. 2016a; Ye et al. 2016b). LDA is a supervised learning algorithm, which uses the information of input data and output data (Mahmoudi and Duman 2015a; Li et al. 2015b). The procedure acquiring the feature vectors by using LDA is given as follows.

[Step 1] Computation of the mean $\boldsymbol{\mu}$ and the mean of within-class $\boldsymbol{\mu}_W$.

$$\boldsymbol{\mu} = \frac{1}{m}\sum_{p=1}^{m}\boldsymbol{\Gamma}_p \qquad \boldsymbol{\mu}_W^h = \frac{1}{m_h}\sum_{q=1}^{m_h}\boldsymbol{\Gamma}_q^h \tag{6}$$

m_h denotes the number of data patterns belong to h^{th}class.

[Step 2] Calculation of the between-class scatter $\mathbf{S}_B$ and the within-class scatter $\mathbf{S}_W$.

$$\mathbf{S}_B = \sum_{h=1}^{class} m_h(\boldsymbol{\mu}_W^h - \boldsymbol{\mu})^T(\boldsymbol{\mu}_W^h - \boldsymbol{\mu}) \qquad \mathbf{S}_W = \sum_{h=1}^{class}\sum_{q=1}^{m_h}\left(\boldsymbol{\Gamma}_q^h - \boldsymbol{\mu}_W^h\right)^T\left(\boldsymbol{\Gamma}_q^h - \boldsymbol{\mu}_W^h\right) \tag{7}$$

$class$ stands for the number of class labels.

[Step 3] Computation of the vectors $\mathbf{V}$ in the form

$$\mathbf{V} = \text{argmax}_{\mathbf{V}}\left|\frac{\mathbf{V}^T\mathbf{S}_B\mathbf{V}}{\mathbf{V}^T\mathbf{S}_W\mathbf{V}}\right| \qquad \mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n]^T \qquad \mathbf{v}_n = [v_1, v_2, \cdots, v_n] \tag{8}$$

[Step 4] Extraction of the feature vectors from $\mathbf{V}$.

$$\hat{\mathbf{V}} = [\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \cdots, \hat{\mathbf{v}}_n]^T \qquad \hat{\mathbf{v}}_n = [\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_k] \tag{9}$$

[Step 5] Transformation of the training data into $\mathbf{X}$.

$$\mathbf{X} = \boldsymbol{\Gamma}^{(m\times n)}\hat{\mathbf{V}}^{(n\times k)} \tag{10}$$

2.2 Hybrid PCA: combination of PCA and LDA

Hybrid PCA is a preprocessing method combined with PCA and LDA. The preprocessing technique combined with both PCA and LDA is considered to supplement each disadvantage related to PCA and LDA. PCA has a disadvantage that cannot use the information about class label of the data. The shortcoming of LDA is that the number of the features extracted from LDA is restricted. Therefore, the hybrid PCA has been introduced to complement the problems of both PCA and LDA. The hybrid PCA is conducted as follows.

[Step 1] Acquisition of the eigenvectors of the PCA $\mathbf{W}$ by exploiting (1) ~ (4)

[Step 2] Acquirement of the between-class scatter $\mathbf{S}_B$ and the within-class scatter $\mathbf{S}_W$ by utilizing equation (6) ~ (7).

[Step 3] Computation of the vectors $\mathbf{H}$ in the form

$$\mathbf{H} = \text{argmax}_{\mathbf{H}}\left|\frac{\mathbf{H}^T\mathbf{W}^T\mathbf{S}_B\mathbf{W}\mathbf{H}}{\mathbf{H}^T\mathbf{W}^T\mathbf{S}_W\mathbf{W}\mathbf{H}}\right| \qquad \mathbf{H} = [\mathbf{h}_1, \mathbf{h}, \cdots, \mathbf{h}_k]^T \qquad \mathbf{h}_k = [h_1, h_2, \cdots, h_k] \tag{11}$$

[Step 4] Calculation of the new vectors $\mathbf{P}$ by multiplying the $\mathbf{W}$ and $\mathbf{H}$.

$$\mathbf{P} = \mathbf{W}^{(n\times k)}\mathbf{H}^{(k\times k)} \qquad \mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n]^T \qquad \mathbf{p}_n = [p_1, p_2, \cdots, p_k] \tag{12}$$

[Step 5] Transformation of the training data into $\mathbf{X}$.

$$\mathbf{X} = \mathbf{\Gamma}^{(m\times n)}\mathbf{P}^{(n\times k)} \tag{13}$$

2.3 Fuzzy transform: fuzzy set -based preprocessing technique

Fuzzy transform was proposed for the first time by Perfilieva (Manchanda and Sharma 2018; Perfilieva 2006). Fuzzy transform based on fuzzy sets theory is a preprocessing technique where each input variable is divided into several fuzzy spaces and the representative value is computed for each fuzzy space. In order to compute the membership values, diverse functions are considered such as triangular shape, trapezoidal shape etc. In this study, we use triangular membership functions. The method of reducing the dimensionality by using fuzzy transform is given as follows (Patanè 2021; Perfilieva et al. 2008).

[Step 1] Fuzzy partitioning by using membership functions.

[Step 2] Calculation of the values of each membership function.

$$\mathbf{U}_1 = \begin{cases} 1 - \dfrac{(x - x_1)}{h_i}, & x \in [x_1, x_2] \\ 0 & otherwise \end{cases} \quad \mathbf{U}_i = \begin{cases} \dfrac{(x - x_{i-1})}{h_{i-1}}, & x \in [x_{i-1}, x_i] \\ 1 - \dfrac{(x - x_1)}{h_i}, & x \in [x_i, x_{i+1}] \\ 0 & otherwise \end{cases} \quad \mathbf{U}_k = \begin{cases} \dfrac{(x - x_{k-1})}{h_k}, & x \in [x_{k-1}, x_k] \\ 0 & otherwise \end{cases} \tag{14}$$

$\mathbf{U}_i$ represents the membership function values of the i^{th}membership function.

[Step 3] Computation of the values $\mathbf{F}$ in the form

$$\mathbf{F}_i = \frac{\sum_{p=1}^{m} f(\mathbf{x}_p)\mathbf{U}_i(\mathbf{x}_p)}{\sum_{p=1}^{m} \mathbf{U}_i(\mathbf{x}_p)} \tag{15}$$

3. Structure of fuzzy radial basis function neural networks (FRBFNN) classifier based on preprocessing algorithm

In this section, the overall structure of preprocessing algorithm-based FRBFNN classifier is discussed.

3.1 Structure of the preprocessing technique-based FRBFNN classifier

An overall structure of the classifier used in this study is shown in Fig 1.

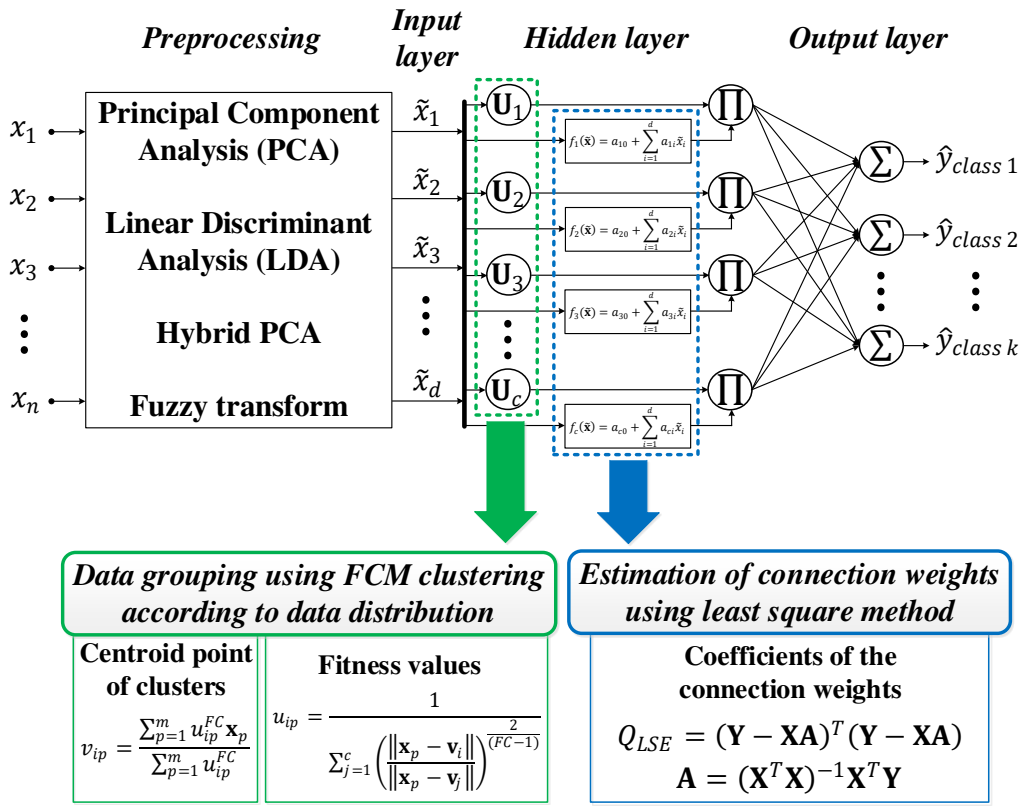


Fig. 1. Design procedure and overall structure of the preprocessing technique-based FRBFNN classifier

The preprocessing techniques-based FRBFNN classifier is composed of four parts involving preprocessing, input layer, hidden layer and output layer. In the preprocessing of the FRBFNN classifier, dimensionality reduction for transforming the *n*-dimensional data to the *d*-dimensional data is carried out by preprocessing techniques. The data produced through the dimensionality reduction are used as the inputs of the FRBFNN classifier. In the hidden layer of the FRBFNN classifier, the fitness values are computed by using fuzzy c-means (FCM) clustering and the least square estimation (LSE) is used to calculate the coefficients of the connection weights. In the output layer of the FRBFNN classifier, the output values of the classifier are calculated through fuzzy inference. Fuzzy inference is the process of the mapping from a given input data to an output data by using membership functions, logical operator and *If-Then* rules (Wang et al. 2015a; Riza et al. 2015b; Chang et al. 2016). The whole structure of the preprocessing techniques-based FRBFNN classifier can be described by using the *If-Then* rules as follows.

Table 1. Fuzzy rules for each class

(1) 1st class
R_1^1: *If* $\tilde{x}_1$ is T_{11} and $\tilde{x}_2$ is T_{12} $\cdots$ $\tilde{x}_d$ is T_{1d}, *Then* $f_1^1(\tilde{\mathbf{x}})$ is $a_{10}^1 + a_{11}^1\tilde{x}_1 + a_{12}^1\tilde{x}_2 + \cdots + a_{1d}^1\tilde{x}_d$
R_2^1: *If* $\tilde{x}_1$ is T_{21} and $\tilde{x}_2$ is T_{22} $\cdots$ $\tilde{x}_d$ is T_{2d}, *Then* $f_2^1(\tilde{\mathbf{x}})$ is $a_{20}^1 + a_{21}^1\tilde{x}_1 + a_{22}^1\tilde{x}_2 + \cdots + a_{2d}^1\tilde{x}_d$
$\vdots$
R_c^1: *If* $\tilde{x}_1$ is T_{c1} and $\tilde{x}_2$ is T_{c2} $\cdots$ $\tilde{x}_d$ is T_{cd}, *Then* $f_c^1(\tilde{\mathbf{x}})$ is $a_{c0}^1 + a_{c1}^1\tilde{x}_1 + a_{c2}^1\tilde{x}_2 + \cdots + a_{cd}^1\tilde{x}_d$
(2) 2nd class
R_1^2: *If* $\tilde{x}_1$ is T_{11} and $\tilde{x}_2$ is T_{12} $\cdots$ $\tilde{x}_d$ is T_{1d}, *Then* $f_1^2(\tilde{\mathbf{x}})$ is $a_{10}^2 + a_{11}^2\tilde{x}_1 + a_{12}^2\tilde{x}_2 + \cdots + a_{1d}^2\tilde{x}_d$
R_2^2: *If* $\tilde{x}_1$ is T_{21} and $\tilde{x}_2$ is T_{22} $\cdots$ $\tilde{x}_d$ is T_{2d}, *Then* $f_2^2(\tilde{\mathbf{x}})$ is $a_{20}^2 + a_{21}^2\tilde{x}_1 + a_{22}^2\tilde{x}_2 + \cdots + a_{2d}^2\tilde{x}_d$
$\vdots$
R_c^2: *If* $\tilde{x}_1$ is T_{c1} and $\tilde{x}_2$ is T_{c2} $\cdots$ $\tilde{x}_d$ is T_{cd}, *Then* $f_c^2(\tilde{\mathbf{x}})$ is $a_{c0}^2 + a_{c1}^2\tilde{x}_1 + a_{c2}^2\tilde{x}_2 + \cdots + a_{cd}^2\tilde{x}_d$
$\vdots$
(k) k^{th} class
R_1^k: *If* $\tilde{x}_1$ is T_{11} and $\tilde{x}_2$ is T_{12} $\cdots$ $\tilde{x}_d$ is T_{1d}, *Then* $f_1^k(\tilde{\mathbf{x}})$ is $a_{10}^k + a_{11}^k\tilde{x}_1 + a_{12}^k\tilde{x}_2 + \cdots + a_{1d}^k\tilde{x}_d$

R_2^k: $If\ \tilde{x}_1$ is T_{21} and $\tilde{x}_2$ is $\mathrm{T}_{22}\ \cdots\ \tilde{x}_d$ is $\mathrm{T}_{2d}, Then\ f_2^k(\tilde{\mathbf{x}})$ is $a_{20}^k + a_{21}^k\tilde{x}_1 + a_{22}^k\tilde{x}_2 + \cdots + a_{2d}^k\tilde{x}_d$
$\vdots$
R_c^k: $If\ \tilde{x}_1$ is T_{c1} and $\tilde{x}_2$ is $\mathrm{T}_{c2}\ \cdots\ \tilde{x}_d$ is $\mathrm{T}_{cd}, Then\ f_c^k(\tilde{\mathbf{x}})$ is $a_{c0}^k + a_{c1}^k\tilde{x}_1 + a_{c2}^k\tilde{x}_2 + \cdots + a_{cd}^k\tilde{x}_d$

In *If-Then* rule of Table 1, T stands for membership function and a denotes the parameter of connection weights.

3.2 Premise part of the fuzzy rules: FCM clustering for data grouping

FCM clustering is a kind of unsupervised learning method, and a technique to analyze the data being used in diverse fields such as machine learning, pattern recognition, data mining and image analysis (Hu et al. 2015; Khosravi et al. 2018). The FCM clustering is used in data analysis in order to find the centroid point and the degree of belonging of each cluster (Zhou and Yang 2016; Zhang et al. 2017; Anter and Ali 2020). The FCM clustering algorithm is composed of the following steps.

[Step 1] Determination of the number of clusters and initialization of membership matrix $\mathbf{U}^{\langle r\rangle}$.

$$\mathbf{U}^{\langle r\rangle} = \left\{ u_{ip} \in [0,1], \quad \sum_{i=1}^{c} u_{ip} = 1 \ \ \forall p, \quad 0 < \sum_{p=1}^{m} u_{ip} < m \ \ \forall i \right\} \tag{16}$$

[Step 2] Computation of the centroid point of each cluster.

$$v_{ip} = \frac{\sum_{p=1}^{m} u_{ip}^{FC}\mathbf{x}_p}{\sum_{p=1}^{m} u_{ip}^{FC}} \tag{17}$$

FC represents fuzzification coefficient.

[Step 3] Updating the membership matrix using Euclidean distance function.

$$\mathbf{U}^{\langle r+1\rangle} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ip}^{\langle r\rangle}}{d_{jp}^{\langle r\rangle}}\right)^{\frac{2}{(FC-1)}}} \qquad d_{ip} = \left\|\mathbf{x}_p - \mathbf{v}_i\right\| \tag{18}$$

[Step 4] Checking a termination criterion. If

$$\left\|\mathbf{U}^{\langle r+1\rangle} - \mathbf{U}^{\langle r\rangle}\right\| \le \delta \ \ (tolerance\ \ level) \tag{19}$$

Stop; otherwise set $r = r + 1$ and return to **[Step 2]**.

3.3 Consequence part of fuzzy rules: estimation of connection weights

In the consequence part of the fuzzy rules, the coefficients (parameters) of the connection weights are estimated by using LSE. LSE is a global learning algorithm that minimizes an overall squared error between output of the model and the target output (Qiu et al. 2017; Oh et al. 2013). Mean squared error (MSE) is used as the objective function of LSE.

$$Q_{LSE} = \sum_{k=1}^{class}\sum_{p=1}^{m}\left[y_p^k - \sum_{i=1}^{c} u_{ip} f_i(\mathbf{x}_p)\right]^2 = (\mathbf{Y}^k - \mathbf{X}\mathbf{A}^k)^T(\mathbf{Y}^k - \mathbf{X}\mathbf{A}^k) \tag{20}$$

The result values of the coefficients of the connection weights are determined in the form.

$$\mathbf{A}^k = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}^k \tag{21}$$

$$\mathbf{X} = \begin{bmatrix} u_{11} & \cdots & u_{1c} & u_{11}x_{11} & \cdots & u_{1c}x_{11} & \cdots & u_{11}x_{1n} & \cdots & u_{1c}x_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mc} & u_{m1}x_{m1} & \cdots & u_{mc}x_{m1} & \cdots & u_{m1}x_{mn} & \cdots & u_{mc}x_{mn} \end{bmatrix}$$

$$\mathbf{Y}^k = [y_1^k,\ y_2^k,\ \cdots,\ y_m^k]^T \qquad \mathbf{A}^k = [a_{10}^k,\ \cdots,\ a_{c0}^k,\ a_{11}^k,\ \cdots,\ a_{c1}^k,\ \cdots,\ a_{1n}^k,\ \cdots,\ a_{cn}^k]^T$$

The coefficients of polynomial functions are dealt with as the connection weights of the consequence part of fuzzy rule. Several polynomial types used as connection weights are given as shown in Table 2.

Table 2. Taxonomy of polynomial function used for connection weights

Constant	: $f_i(\mathbf{x}) = a_{i0}$
Linear	: $f_i(\mathbf{x}) = a_{i0} + \sum_{q=1}^{n} a_{iq}x_q$
Quadratic	: $f_i(\mathbf{x}) = a_{i0} + \sum_{q=1}^{n} a_{iq}x_q + \sum_{q=1}^{n}\sum_{s=1}^{n} a_{iqs}x_q x_s$

In the case of the quadratic function, it could lead to a curse of dimensionality when dealing with the problems of high dimensionality. So, the following modified quadratic function is used instead of the original quadratic function.

Modified quadratic: $f_i(\mathbf{x}) = a_{i0} + \sum_{q=1}^{n} a_{iq}x_q + \sum_{s=1}^{n} a_{iqs}x_s^2$

4. Optimization of FRBFNN classifier

PSO is an algorithm that is based on the social behavioral patterns of crowded animals such as birds and fishes and it was proposed by Kennedy and Eberhart (Grassi S and Pareschi 2021; Zhang et al. 2020; Shanthi et al 2018; Kennedy and Eberhart 1995). This optimization algorithm is one of heuristic search techniques, which belongs to the category of swarm intelligence methods. Unlike other heuristics, PSO is endowed with a flexible and well-balanced mechanism to enhance the global and local exploration abilities. Due to its efficiency in exploring global solutions, PSO has been applied to many problems such as classification (Zemmal et al. 2020), feature selection (El-Kenawy and Eid 2020), and stochastic optimization (Zhang et al. 2016). Unlike several optimization algorithms such as Genetic Algorithm (GA) or Differential Evolution (DE), the PSO has some advantages (Jordehi et al. 2015; Deng et al. 2019):

a) Unlike other search algorithms, the PSO algorithm can perform global optimization problems for large and complex functions like evolutionary computations.

b) PSO algorithm generates the optimal solutions in a short period time and represents the stable convergence characteristics than any other optimization algorithms.

c) Since the search of the optimal solutions does not depend on the initial values, PSO algorithm enables to converge the optimal solutions from anywhere points in the search space.

In general, PSO can be performed as the following sequence of steps:

[Step 1] Random generation of swarm **S**, particle's position **p**, and its velocity **v**.

$$\begin{aligned} \mathbf{S}(gen) &= [\mathbf{p}_1(gen), \mathbf{p}_2(gen), \cdots, \mathbf{p}_a(gen)] \qquad \mathbf{p}_a(t) = [p_1(gen), p_2(gen), \cdots, p_b(gen)]^T \\ \mathbf{V}(gen) &= [\mathbf{v}_1(gen), \mathbf{v}_2(gen), \cdots, \mathbf{v}_a(gen)] \qquad \mathbf{v}_a(gen) = [v_1(gen), v_2(gen), \cdots, v_b(gen)]^T \end{aligned} \tag{22}$$

In equation (22), a represents the size of swarm, b stands for the number of parameters to be optimized.

gen stands for the number of current generation performing the PSO.

[Step 2] Adjustment of the inertia weight w.

$$w(gen) = w_{max} - \frac{w_{max} - w_{min}}{gen_{max}} \times gen \tag{23}$$

In (23), w_{max} and w_{min} represents the maximum and minimum value of the inertia weight respectively. $gen_{\max}$ denotes the maximum number of generations.

[Step 3] Updating the particles. By using the values of $\mathbf{pbest}(gen)$ and $\mathbf{gbest}(gen)$, the velocity $\mathbf{v}$ of the ith particle is adjusted.

$$\mathbf{v}(gen+1) = w(gen)\mathbf{v}(gen) + c_1 r_1[\mathbf{pbest}(gen) - \mathbf{p}(gen)] + c_2 r_2[\mathbf{gbest}(gen) - \mathbf{p}(gen)] \tag{24}$$

c_1 and c_2 are positive constants, called acceleration constants. r_1 and r_2 are random values drawn from between 0 and 1.

[Step 4] Updating the position of each particle by exploiting the updated velocity.

$$\mathbf{p}(gen+1) = \mathbf{p}(gen) + \mathbf{v}(gen+1) \tag{25}$$

[Step 5] Evaluation of the updated particles using the objective function and comparison of their performance in terms of $\mathbf{pbest}(gen)$ and $\mathbf{gbest}(gen)$.

[Step 6] If the termination criterion has been not satisfied, the sequence **[Step 2]** ~ **[Step 5]** is repeated.

Figure 2 depicts overall flow of the proposed classifier and Figure 3 shows the algorithmic framework of the proposed classifier in this study

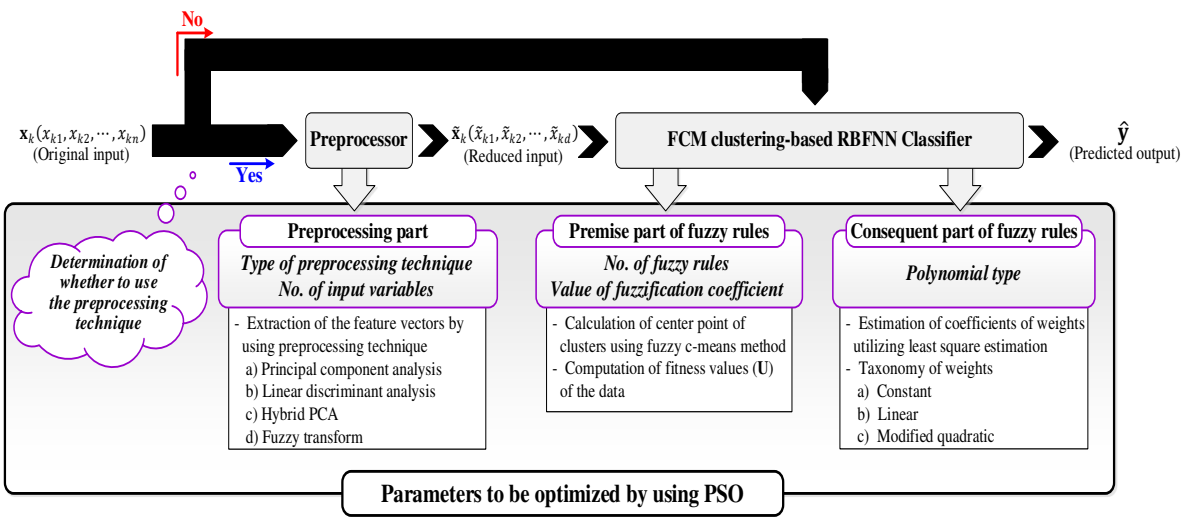


Fig. 2. Design procedure of the proposed classifier

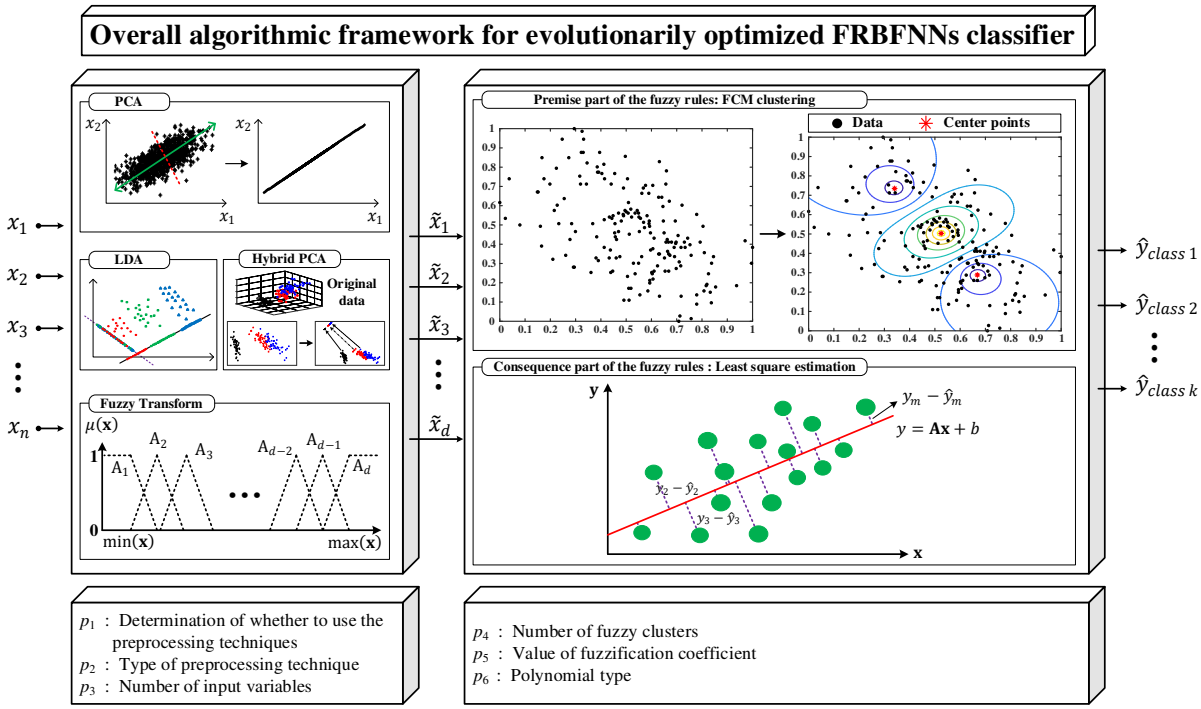


Fig. 3. Overall algorithmic framework for evolutionarily optimized FRBFNN classifier

The parameters to be optimized by using PSO are concerned with determination of whether to use preprocessing technique, type of the preprocessing technique, the number of input variables extracted from the preprocessing technique, the number of fuzzy rules, the value of fuzzification coefficient, and the type of polynomial with connection weights. The flow of the evolutionarily optimized FRBFNN classifier is explained as below.

i) Structure of particles: Decision of parameters to be optimized.

ii) Generation of swarm: Generation of various particles using velocity and inertia weights.

iii) Selection of parameters: Decision of parameters by the particle. The parameters that can be selected are as follows.

$$p_1 \begin{cases} \text{don't use} & if\ \ round(p_1) = 0 \\ \text{use} & if\ \ round(p_1) = 1 \end{cases}$$

$$p_2 \begin{cases} \text{PCA} & if\ \ round(p_2) = 1 \\ \text{LDA} & if\ \ round(p_2) = 2 \\ \text{Hybrid PCA} & if\ \ round(p_2) = 3 \\ \text{Fuzzy Transform} & if\ \ round(p_2) = 4 \end{cases}$$

p_3 : The number of feature vectors extracted from the preprocessing technique

$$p_4 \begin{cases} 2 & if\ \ round(p_4) = 2 \\ 3 & if\ \ round(p_4) = 3 \\ 4 & if\ \ round(p_4) = 4 \\ 5 & if\ \ round(p_4) = 5 \end{cases}$$

p_5 : The value of FC

$$p_6 \begin{cases} \text{Constant} & if\ \ round(p_6) = 1 \\ \text{Linear} & if\ \ round(p_6) = 2 \\ \text{Modified quadratic} & if\ \ round(p_6) = 3 \end{cases}$$

iv) Construction of network: Construction of network based on the selected parameters.

5. Experimental studies

The experiments performed in this study are evaluated and compared by using several benchmark datasets and black plastic wastes datasets for practical application and the final experimental results were carried out by using 5-fold cross validation (5-fcv) mode.

5.1 Machine learning datasets

The parameter setting conditions for experiments are shown in Table 3. Table 4 represents the several benchmark datasets used in the experiment. Benchmark datasets used in this study were obtained from UCI machine learning laboratory website (http://archive.ics.uci.edu/ml).

Table 3. Parameters setting for initial condition for experiments

Parameters		Values
Ratio of split data		Training : Validation : Testing = 3 : 1 : 1
Cross validation		5-fold cross validation
No. of generations		100
Swarm size		200
No. of particles		6
V_{max}		20% of the search space
Inertia weight		$w_{\max} = 0.9,\ w_{\min} = 0.4$
c_1, c_2		2.0
$r_1,\ r_2$		[0 1]
Objective function of the PSO		$\mathrm{CR_{TR}} * (1-\theta) + \mathrm{CR_{VA}} * \theta$ $(\theta = 0.25, 0.5, 0.75, 1.0)$
Search space by using PSO	Determination of using of preprocessing technique	0 (No), 1 (Yes)
	Type of preprocessing technique	PCA, LDA, Hybrid PCA, Fuzzy transform
	No. of reduced input variables	[2 No. of input variables-1]
	No. of fuzzy rules	[2 5]
	Fuzzification coefficient (*FC*)	[1.1 3.0]
	Polynomial type	Constant, Linear, Modified quadratic

Table 4. Information of benchmark datasets used for experiments

Data name (*Abb.*)	No. of input variables	No. of classes (No. of data patterns per class)	No. of data patterns
Balance (Bl)	4	3 (288 / 49 / 288)	625
Banknote (Ba)	4	2 (762 / 610)	1,372
Iris (Ir)	4	3 (50 / 50/ 50)	150
Hayes (Ha)	5	3 (51 / 51 / 30)	132
Liver (Li)	6	2 (145 / 200)	345
Seeds (Sd)	7	3 (70 / 70 / 70)	210
Pima (Pm)	8	2 (268 / 500)	768
Glass (Gl)	9	6 (70 / 76 / 17 / 13 / 9 / 29)	214
Heart (Hr)	13	2 (120 / 150)	270
Wine (Wi)	13	3 (59 / 71 / 48)	178
Australian (Au)	14	2 (383/ 307)	690
Leaf (Le)	15	30	340
Zoo (Zo)	16	7 (41 / 20 / 5 / 13 / 4 / 8 / 10)	101
Vehicle (Ve)	18	4 (199 / 217 / 218 / 212)	846
German (Ge)	24	2 (700 / 300)	1,000
Forest (Fo)	27	4 (195 / 86 / 159 / 83)	523
WDBC (WD)	30	2 (212 / 357)	569

Ionosphere (Io)	34	2 (126 / 225)	351
Sonar (So)	60	2 (97 / 111)	208
Libras (Lb)	90	15 (24)	360
Hillvalley (Hv)	100	2 (612 / 600)	1,212
Urban Land Cover (ULC)	147	9 (59 / 122 / 36 / 116 / 112 / 29 / 59 / 36 / 106)	675

Abb. : Abbreviation

The classification performance obtained using benchmark datasets is shown in Table 5.

In Table 5, the factors considered for obtaining classification performance are given through six parameters such as the determination of whether to use of data preprocessing, the type of data preprocessing technique, the number of input variables (extracted features) reduced by the preprocessing technique, the number of fuzzy rules (fuzzy clusters), fuzzification coefficient (FC) and the type of the connection weights based on polynomial type.

Table 5. Classification performance of the evolutionarily optimized FRBFNN classifier

Data type	θ	Preprocessing type (No. of features)	C	*FC*	Polynomial type	Training (Mean±Std)	Validation (Mean±Std)	Testing (Mean±Std)
Balance	0.75	Hybrid PCA (3)	5	3.000	Modified Quadratic	96.05±0.95	96.00±1.69	94.88±0.91
Banknote	0.25	-	3	2.270	Modified Quadratic	100.00	100.00	100.00
Iris	1.0	Hybrid PCA (2)	4	2.003	Linear	98.00±0.93	100.00	98.66±1.82
Hayes	1.0	-	2	1.113	Modified Quadratic	85.94±4.42	85.38±4.21	75.13±12.28
Liver	0.5	-	5	1.608	Linear	77.68±2.14	79.71±3.55	72.75±2.15
Seeds	0.75	Hybrid PCA (5)	4	2.367	Modified Quadratic	99.36±0.87	99.52±1.06	95.24±2.91
Pima	1.0	-	3	1.440	Linear	77.59±2.10	81.43±1.64	77.34±2.99
Glass	1.0	Hybrid PCA (6)	4	2.860	Linear	77.52±2.39	76.75±3.26	69.17±7.90
Heart	1.0	LDA (1)	4	2.064	Modified Quadratic	85.31±0.91	90.74±3.46	85.55±3.79
Wine	1.0	LDA (2)	4	1.268	Linear	99.81±0.42	100.00	99.44±1.24
Australian	0.75	LDA (1)	4	1.913	Linear	87.00±1.07	90.87±3.18	86.37±3.81
Leaf	1.0	LDA (3)	5	1.958	Linear	95.78±1.85	83.82±4.28	79.41±2.32
Zoo	0.75	LDA (6)	3	2.313	Linear	100.00	100.00	99.00±2.23
Vehicle	1.0	-	2	2.268	Modified Quadratic	88.61±0.98	86.86±1.53	82.15±2.40
German	0.75	-	3	2.558	Linear	81.03±0.81	80.20±2.36	75.50±3.37
Forest	1.0	Hybrid PCA (5)	4	2.334	Modified Quadratic	91.60±1.05	93.84±1.61	89.86±2.41
WDBC	0.5	LDA (1)	4	2.280	Linear	97.77±0.39	98.59±1.18	96.34±1.32
Ionosphere	0.75	PCA (11)	3	2.603	Modified Quadratic	97.63±0.58	95.14±2.39	90.89±3.95
Sonar	0.75	FT (13)	2	1.412	Modified Quadratic	94.89±1.44	91.22±4.08	86.06±0.98
Libras	0.25	Hybrid PCA (15)	4	2.489	Modified Quadratic	99.72±0.41	86.66±4.34	83.33±1.70
Hillvalley	0.75	FT (13)	5	2.099	Modified Quadratic	96.89±2.12	90.82±3.50	92.16±2.44
ULC	1.0	LDA (8)	2	1.598	Linear	93.68±0.51	76.00±2.65	72.74±3.60

θ : Weighting parameter used in the objective function of PSO
Preprocessing type : Preprocessing technique for feature extraction
No. of features : Number of extracted features by using the related preprocessing technique
C : Number of fuzzy clusters
FC : Fuzzification coefficient used in FCM clustering
Polynomial type : Polynomial type (L: Linear, M.Q: Modified Quadratic)
Training : Average classification performance and standard deviation of training data
Validation : Average classification performance and standard deviation of validation data
Testing : Average classification performance and standard deviation of testing data

In the case of 16 datasets among 22 datasets (about 72%), the preprocessing algorithm is used. As shown in Table 5, the selection of suitable preprocessing technique is very critical and also leads to preferred performance results when the number of input variables gets more and more increasing. Figure 4 shows the design procedure for constructing the structure of the FRBFNN classifier with better classification performance through evolutionary optimization by using 'Sonar' data as one example.

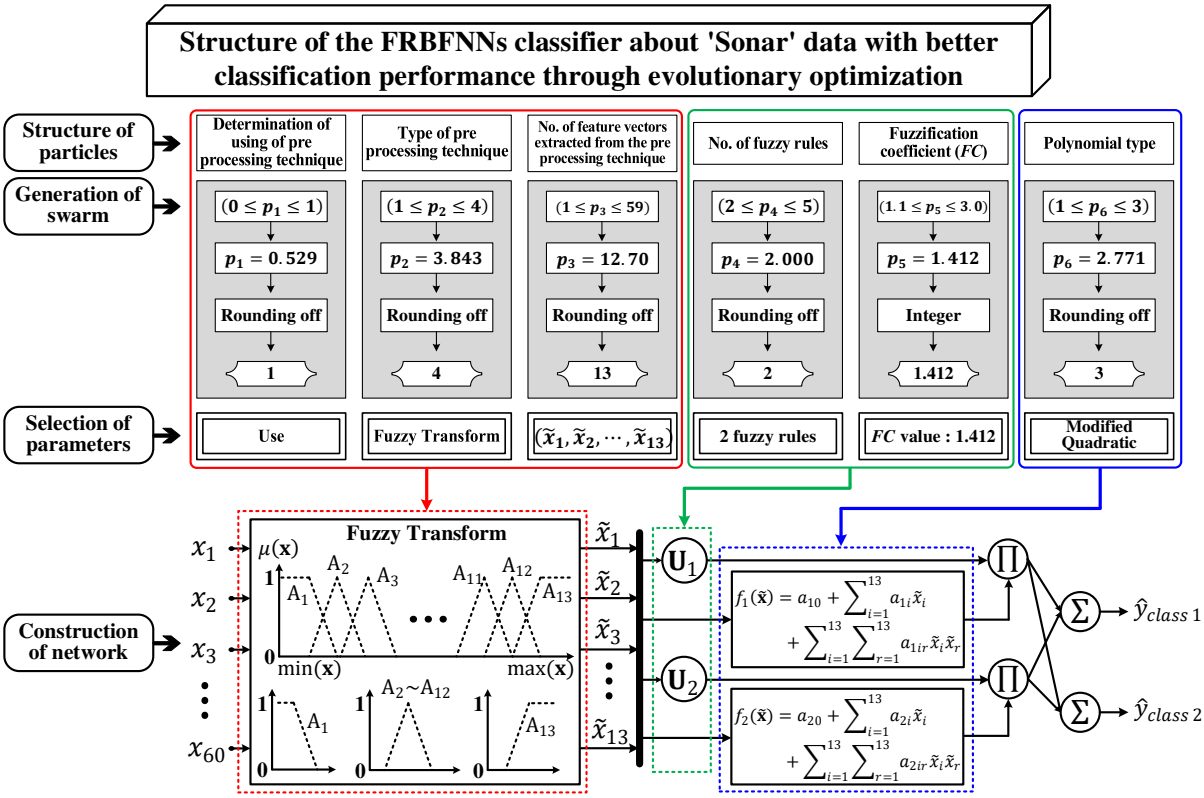


Fig. 4. Design procedure for constructing the structure of the FRBFNN classifier by 'Sonar' data for obtaining preferred classification performance through evolutionary optimization

In order to obtain better classification performance for each data, the parameters of the FRBFNN classifier needed for optimized structural design are adjusted through evolutionary optimization. From the viewpoint of 'Sonar' data, the preferred classification performance achieved through evolutionary optimization is 86.06% as shown in Table 5. To reach the classification performance of 86.06%, the parameters optimized from evolutionary optimization are given as the following.

p_1 (Determination of using of preprocessing technique) = Use
p_2 (Selection of preprocessing technique) = Fuzzy Transform
p_3 (No. of feature vectors extracted from the preprocessing technique) = 13
p_4 (No. of fuzzy rules) = 2
p_5 (Fuzzification coefficient) = 1.412
p_6 (Polynomial type) = Modified Quadratic

Additionally, in order to demonstrate the superiority of the evolutionary optimized radial basis function neural network classifier which is proposed in this study, we compare the classification performance with Weka data mining software (Weka). Weka is a software developed by Waikato University in New Zealand and provides various algorithms used in classification or regression problems.

Table 6. Comparison of classification performance of WEKA and the evolutionarily optimized FRBFNN classifier

Data (*Abb.*)	Naive Bayes	MLPs	SVM	*k*-NN	PART	J48	Random tree	FRBFNN1	FRBFNN2	Proposed
Bl	91.52±0.91(6)	91.84±1.82(5)	86.56±0.67(10)	88.00±1.60(9)	***95.52±1.45***(1)	94.88±2.08(2.5)	94.40±1.87(4)	91.20±0.80(8)	91.36±1.54(7)	94.88±0.91(2.5)
Ba	84.01±2.12(10)	99.93±0.16(4)	98.03±0.55(9)	99.85±0.19(5)	98.32±0.19(7)	98.61±0.60(6)	98.10±0.79(8)	***100.00±0.00***(2)	***100.00±0.00***(2)	***100.00±0.00***(2)
Ir	97.33±2.79(5.5)	96.66±3.33(9)	***98.66±1.82***(2)	97.33±2.79(5.5)	97.33±2.79(5.5)	97.33±2.79(5.5)	***98.66±1.82***(2)	96.66±2.35(9)	96.66±2.35(9)	***98.66±1.82***(2)
Ha	71.54±15.53(7)	69.23±8.60(8)	53.84±12.169()	35.38±11.01(10)	74.61±10.74(2)	76.92±9.42(2)	***78.46±10.74***(1)	74.19±4.54(6)	74.24±8.31(5)	75.13±12.283()
Li	55.94±4.87(10)	69.57±1.77(4)	57.39±1.29(9)	60.87±2.29(8)	65.51±4.95(6)	64.35±4.97(6)	63.48±7.76(7)	73.04±4.53(2)	***74.20±4.02***(1)	72.75±2.15(3)
Sd	***97.62±1.68***(1)	95.71±4.58(5)	96.67±1.30(2)	96.19±2.13(3)	95.24±4.45(9.5)	93.33±6.16(9.5)	93.33±4.58(9.5)	94.76±1.99(8)	96.19±2.13(4)	95.24±2.91(6.5)
Pm	76.36±3.36(4)	74.41±2.88(8)	75.97±4.33(5)	74.54±2.69(7)	74.80±5.74(9)	72.73±3.28(9)	71.30±2.45(10)	***77.86±3.62***(1)	76.69±2.16(3)	77.34±2.99(2)
Gl	52.09±7.09(10)	66.05±8.48(5)	53.02±5.79(9)	68.84±4.82(2)	62.79±5.93(7)	60.00±6.02(7)	57.67±7.05(8)	67.79±5.58(4)	68.24±6.61(3)	***69.17±7.90***(1)

Hr	84.07±2.81(3)	83.70±3.56(4)	84.07±2.11(2)	81.85±4.79(9)	83.33±3.49(5.5)	83.33±3.46(5.5)	75.93±3.21(10)	82.22±4.46(8)	82.59±6.22(7)	***85.55±3.79***(1)
Wi	97.22±3.40(6)	97.77±1.24(4.5)	98.33±1.52(2.5)	97.77±2.23(4.5)	95.00±4.97(9.5)	95.00±4.97(9.5)	98.33±2.48(2.5)	97.20±2.78(7)	96.06±3.18(8)	***99.44±1.24***(1)
Au	86.23±2.23(3.5)	86.23±2.23(3.5)	85.94±2.32(6)	85.07±2.88(9)	86.23±2.23(3.5)	86.23±2.23(3.5)	81.16±2.76(10)	85.50±4.19(8)	85.65±2.42(7)	***86.37±3.81***(1)
Le	77.06±2.23(3)	***79.41±3.44***(1.5)	64.12±3.97(6)	71.18±4.94(4)	63.53±8.34(5)	65.88±5.03(5)	59.70±6.79(9)	62.06±6.85(8)	59.41±2.86(10)	***79.41±2.32***(1.5)
Zo	96.00±4.18(4)	96.00±4.18(4)	96.00±4.18(4)	95.00±5.00(7.5)	95.00±5.00(7.5)	95.00±5.00(7.5)	94.00±2.23(10)	95.00±6.42(7.5)	97.04±2.69(2)	***99.00±2.23***(1)
Ve	46.63±5.21(10)	83.31±1.06(2)	73.61±1.70(5)	70.29±1.28(8)	71.36±2.24(7)	72.90±1.05(6)	66.15±1.64(9)	***83.69±1.45***(1)	82.98±3.18(3)	82.15±2.40(4)
Ge	76.50±1.97(3)	70.40±2.22(8)	***76.90±3.07***(1)	70.00±1.76(9)	70.90±3.89(7)	73.10±2.43(6)	68.00±1.80(10)	76.80±2.51(2)	75.00±3.77(5)	75.50±3.37(4)
Fo	89.14±3.05(6)	88.95±4.13(7)	89.33±4.73(4.5)	89.33±2.96(4.5)	85.33±2.08(9)	87.05±2.90(8)	83.43±4.34(10)	***90.25±2.37***(1)	89.67±2.83(3)	89.86±2.41(2)
WD	95.96±1.00(4)	95.79±0.73(7)	95.09±1.33(9)	95.96±0.48(4)	95.96±0.48(4)	95.96±0.48(4)	94.73±1.38(10)	95.96±1.47(4)	95.42±1.91(8)	***96.84±1.32***(1)
Io	***91.14±3.09***(1)	88.57±4.28(5)	79.71±3.41(10)	85.14±4.11(8)	86.57±5.94(7)	87.43±3.41(6)	84.00±4.88(9)	88.89±3.25(4)	89.18±3.04(3)	90.89±3.95(2)
So	66.19±7.79(9)	78.09±3.10(3)	74.28±3.91(6)	83.33±1.68(2)	73.81±9.52(7)	77.14±6.85(5)	77.62±1.30(4)	70.71±3.60(8)	65.37±14.1610()	***86.06±0.98***(1)
Lb	83.33±3.25(1.5)	80.83±3.01(3)	65.28±4.71(7)	77.78±2.59(4)	73.05±3.34(6)	73.33±4.54(5)	63.89±6.44(8)	49.44±9.14(10)	55.83±5.14(9)	***83.33±1.70***(1.5)
Hv	51.40±1.86(8)	54.30±1.19(7)	59.67±10.00(4)	57.35±4.34(5)	50.41±0.00(9.5)	50.41±0.00(9.5)	56.61±1.20(6)	77.30±6.01(3)	77.56±3.58(2)	***92.16±2.44***(1)
ULC	68.89±3.22(4)	67.55±2.74(5)	69.04±1.91(3)	69.63±4.22(2)	64.00±5.60(7)	64.74±2.54(6)	59.40±4.03(9)	60.15±3.86(8)	48.89±1.96(10)	***72.74±3.60***(1)
Avg.	(5.432)	(5.113)	(5.682)	(5.909)	(6.250)	(6.091)	(7.545)	(5.432)	(5.500)	(2.045)
Diff.	(3.387)	(3.068)	(3.637)	(3.864)	(4.205)	(4.406)	(5.500)	(3.387)	(3.455)	-

FRBFNN1: Fuzzy clustering-based RBFNN without preprocessing and optimization
FRBFNN2: Fuzzy clustering-*based* RBFNN which applied optimization (except preprocessing)
Proposed: Evolutionarily optimized Fuzzy clustering-based RBFNN
Avg.: Average rank used in Friedman test for statistical analysis
Diff.: Rank difference between the proposed classifier and others

Table 6 summarizes the comparative results of classification performance obtained by the proposed classifier and other classifiers. (a) NaiveBayes (a probabilistic model using Bayes theorem), (b) Multilayer perceptrons (a neural networks using backpropagation learning method), (c) Support vector machine (binary classifier applied by margin-maximization), (d) *k*-nearest neighbors (a classifier categorized by majority vote of its neighbors), (e) PART (partial decision tree algorithm), (f) J48 (kinds of decision tree) and (g) Random tree (a tree formed by a stochastic process) are used with the aid of Weka toolkit. (h) fuzzy radial basis function neural network (a classifier without preprocessing and optimization (PSO)) and (i) fuzzy radial basis function neural network (an optimized classifier by PSO to search suitable parameters) are also used to compare the classification performance. The entries shown in boldface stand for the best classification performance among the entire classifier. As a result, we showed better performance of the proposed classifier in comparison to the results obtained by some other classifiers.

In order to carry out a fair comparison, one of the statistical methods, Friedman test is used to compare the classification performance of the Weka algorithms and the proposed classifier. The Friedman test is a non-parametric statistical evaluation method developed by Milton Friedman (Aljarah et al. 2018). This test compares observations repeated on the same subjects and to check if the measured average ranks are significantly different from the mean rank. The number inside parenthesis in table 6 denotes rank. The Friedman test applied in this study is carried out through the following steps (Chicco and Jurman 2020; Irigaray et al. 2019; Demšar 2006).

[Step 1] Determination of the rank based on the classification performance of algorithms obtained by using machine learning datasets.

[Step 2] Calculation of the average rank of each algorithm through predetermined rank.

[Step 3] Computation of the difference of the average rank between the proposed algorithm and others.

[Step 4] Statistical analysis of the proposed algorithm.

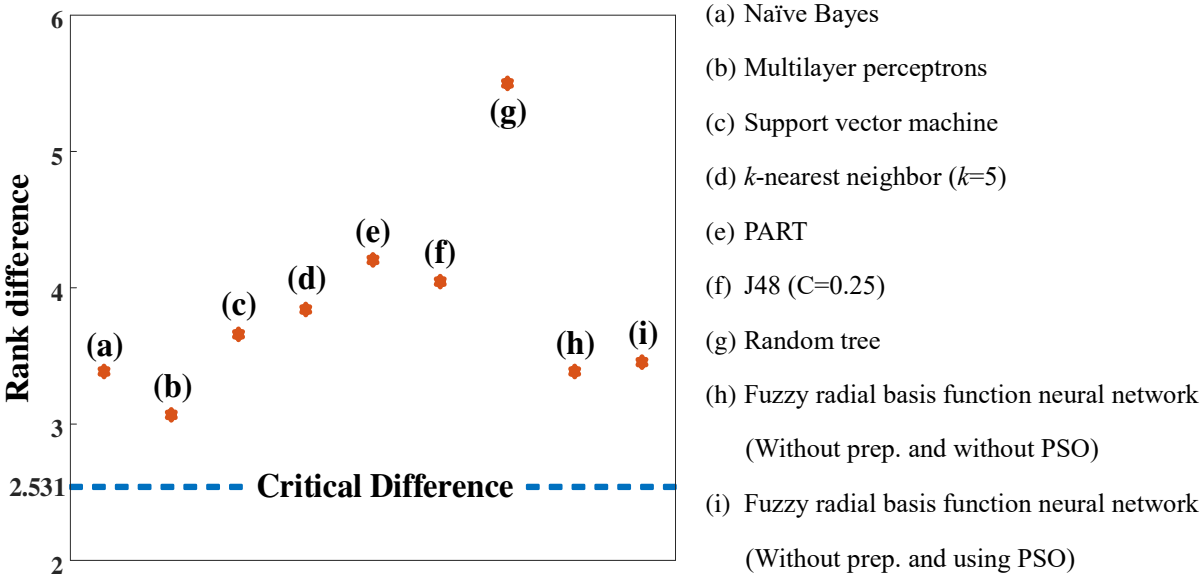
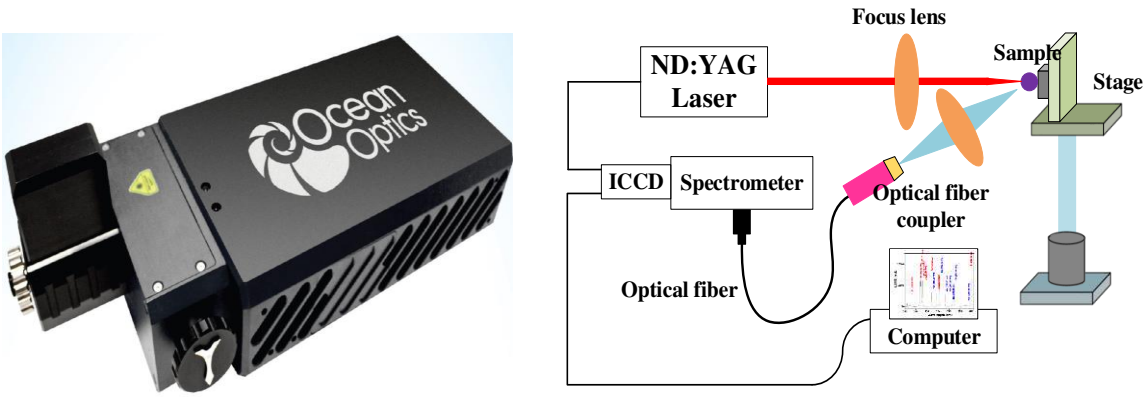


Fig. 5. Comparison of average rank difference between Weka classification algorithms and the proposed classifier

From the Friedman test based on the average rank as shown in Table 7, the null-hypothesis is rejected ($F_F = 5.162 > F(9, 189)$ for $0.05 = 1.930$). As the null-hypothesis was rejected, the post-hoc test is conducted to confirm if the evolutionarily optimized FRBFNN classifier is statistically better than other classifiers. Among several approaches, in case of this study, Bonferroni-Dunn test is applied as the post-hoc test [38]. The Bonferroni-Dunn test is a method when all classifiers are only compared to a target classifier and not between themselves. The classification performance of any two classifiers exists the difference if the corresponding average ranks are bigger than the critical difference (CD). When the p=0.05 (significance level), the value of CD is $2.531\left(2.773 * \sqrt{\frac{10*11}{6*22}}\right)$. Figure 5 describes the difference in average rank between the proposed classifier and various classifiers from the Weka. We can check that the difference of all average ranks is bigger than the value of CD in Figure 5. It means that the evolutionarily optimized FRBFNN classifier exhibits high competitiveness when it comes to the performance.

5.2 Practical application to black plastic wastes

In order to verify the black plastic wastes identification performance of the proposed classifier, we use black plastic raw spectra data obtained from Raman spectroscopy and laser induced breakdown spectroscopy (LIBS) equipment (Roh et al. 2017; Roh et al. 2018). Raman spectroscopy and LIBS equipment used to acquire the raw spectra are introduced in Fig 6.



a) Raman spectroscopy equipment produced by Ocean Optics

b) LIBS equipment produced by Applied Spectra

Fig. 6. Introduction of Raman and LIBS equipment

Individual spectra datasets obtained by using Raman and LIBS equipment are shown in Fig 7 and Fig 8, respectively.

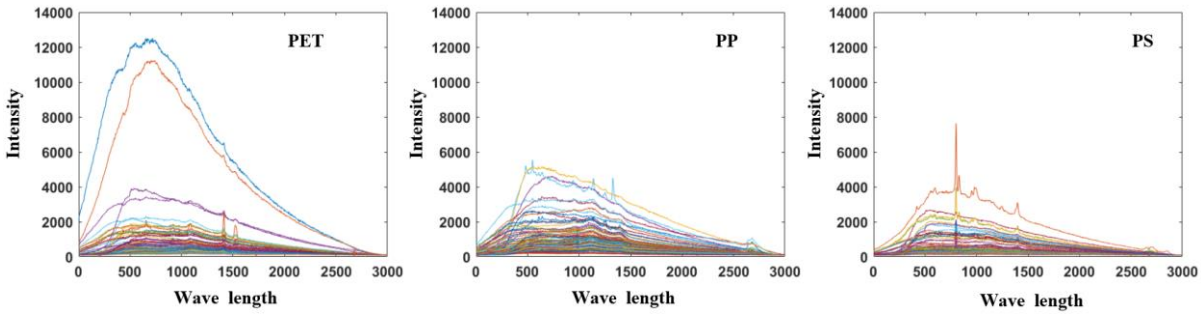


Fig. 7. Each spectra dataset of sampled black plastic wastes obtained from Raman equipment

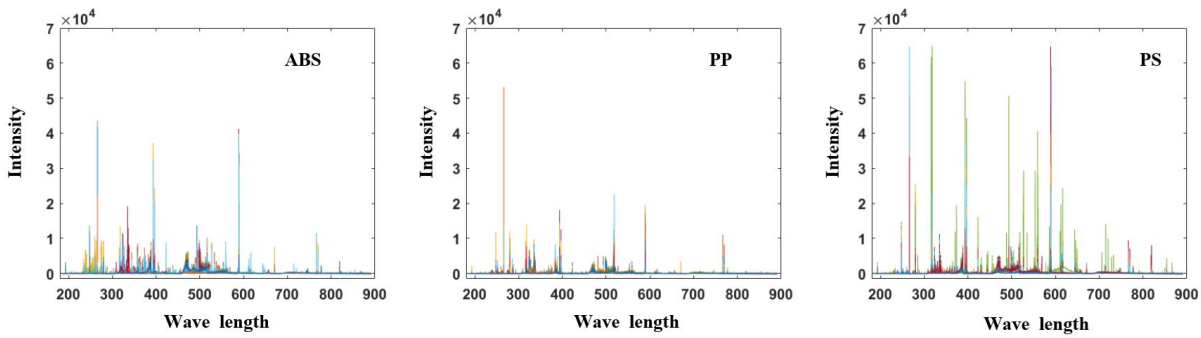


Fig. 8. Each spectra dataset of sampled black plastic wastes obtained from LIBS equipment

In the case of spectral data obtained by using Raman equipment, in order to effectively enhance the classification performance, the data were reconstructed by considering the inherent raw data as well as the chemical characteristics of each material (Bae et al. 2019). Table 7, Table 8 and Table 9 denote the detailed chemical characteristics of PET, PP and PS respectively.

Table 7. Detailed chemical characteristics of polyethylene terephthalate (PET)

Wavelength (cm^{-1})	Raman vibrational mode
702	Ring C−C stretching
808	Ring Torsion and C=O stretching
858	Ring C−C, ester CO−C
886	O−CH_2 and C−C stretch of the gauche ethylene glycol unit
998	O−CH_2 and C−C stretch of the trans ethylene glycol unit
1095	Ethylene glycol C−O and C−C stretching, C−O−C bending
1125	Ester CO−O and ethylene glycol C−C stretching
1180	Ring C−C stretching
1289	CO−C stretching
1310	Ring C−C stretching
1613	Ring C_1−C_4 stretching
1725	C=O stretching

Table 8. Detailed chemical characteristics of polypropylene (PP)

Wavelength (cm^{-1})	Assignment
252, 410	$\omega CH_2+\delta CH$
321, 458	ωCH_2
530	$\omega CH_2+\nu C-CH_3+rCH_2$
809	$\omega CH_2+\nu C-C+\nu C-CH$
841	$rCH_2+\nu C-CH_3$
900	$\nu CH_3+rCH_2+\delta CH$
941	$rCH_2+\nu C-C_{chain}$
973	$rCH_3+\nu C-C_{chain}$
998	$rCH_3+\omega CH_2+\delta CH$
1034	$\nu C-CH_3+\nu C-C+\delta CH$
1102	$\nu C-C+rCH_3+\omega CH_2+tCH+\delta CH$
1152	$\nu C-C+\nu C-CH_3+\delta CH+rCH_3$
1167	$\nu C-C+rCH_3+\delta CH$
1220	$tCH_2+\delta CH+\delta CH+\nu C-C$
1257	$\delta CH+tCH_2+rCH_3$
1296	$\omega CH_3+\delta CH$
1307	ωCH_3+tCH_2
1300	$\delta CH+tCH_2$
1360	$CH_{3symben}+\sigma CH$
1371	$\delta CH_{3sym}+CH_{2wag}$
1435	δCH_{3asym}
1457	$\delta CH_{3asym}+\delta CH_2$
2840	νCH_2
2871, 2883	νCH_{3sym}
2920	νCH_{2asym}
2957	νCH_{3asym}

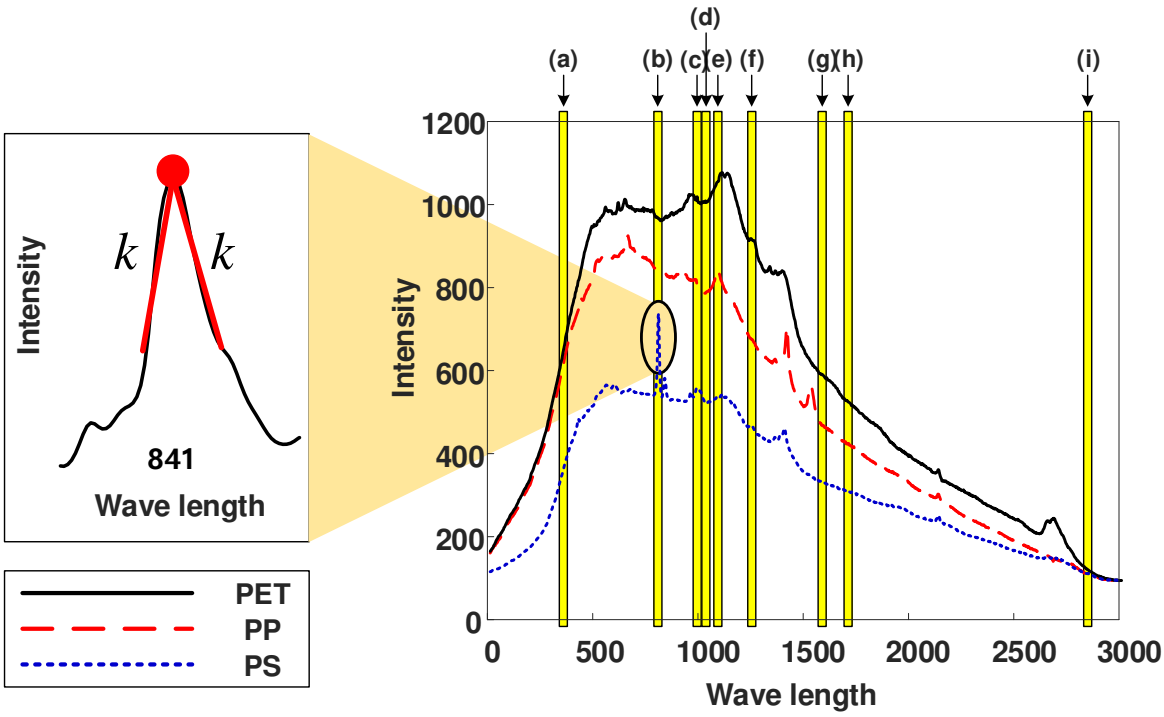
Table 9. Detailed chemical characteristics of polystyrene (PS)

Wavelength (cm^{-1})	Raman vibrational mode
621	Benzene ring vibration
785	CH_2 rocking vibration, C−C stretching vibration
1001	Benzene ring breathing mode
1032	C−H in plane bending mode
1155	C−C stretching vibration
1312	CH_2 twisting vibration, CH wagging vibration, C−C stretching vibration
1602	C=C

Based on the chemical characteristics shown in the Tables 7 - 9, we select three wavelengths of each material. Table 10 illustrates the selected Wavelengths.

Table 10. Selected wavelengths based on chemical information

Materials	Selected wavelengths		
PET	1289 (CO−C stretching)	1613 (Ring C_1-C_4 stretching)	1725 (C=O stretching)
PP	410 ($\omega CH_2+\delta CH$)	841 ($rCH_2+\nu C-CH_3$)	2871 (νCH_{3sym})
PS	1001 (Benzene ring breathing mode)	1032 (C−H in plane bending mode)	1155 (C−C stretching vibration)



(a) : 1st range of PP [410 ± k] (d) : 2nd range of PS [1032 ± k] (g) : 2nd range of PET [1613 ± k]
(b) : 2nd range of PP [841 ± k] (e) : 3rd range of PS [1155 ± k] (h) : 3rd range of PET [1725 ± k]
(c) : 1st range of PS [1001 ± k] (f) : 1st range of PET [1289 ± k] (i) : 3rd range of PP [2871 ± k]

Fig. 9. Setting the input variables of classifier by using chemical characteristics

Figure 9 depicts how the input variables of the classifier are composed which based on the selected wavelengths given in Table 10. The classifier determines whether the spot in the area around the already obtained characteristic peak position in a spectrum is real characteristic peak point. In order to identify whether the spot is a characteristic peak around the specific peak position in a spectrum, the intensity of an area around a specific peak as a wavelength range of '$2k$+1' from each characteristic is composed of the input variables.

Table 11 explains the pertinent information of spectrum datasets of black plastic wastes such as the number of input variables, the number of classes, and the number of spectra instances.

Table 11. Pertinent information of black plastic wastes datasets used in experiments

Data name	No. of input variables	No. of classes (No. of data patterns per class)	No. of instances
Raman	3,001	3 (100 / 100 / 100)	300
Raman(peak)	99		
LIBS	10,240	3 (400 / 400 / 400)	1,200

Raman : The original spectra data which not apply chemical information
Raman(peak) : The characteristic spectra data which apply chemical information
LIBS : The original spectra data obtained through laser-induced breakdown spectroscopy (LIBS) equipment

Table 12 shows the classification performance of the evolutionarily optimized FRBFNN classifier by using PSO with respect to the method of preprocessing technique, the number of extracted input variables, the number of fuzzy clusters, the values of the fuzzification coefficient, and the polynomial type.

Table 12. Classification performance of the evolutionarily optimized FRBFNN classifier by PSO

Data	θ	Preprocessing tech. (Features)	C	*FC*	Poly	Training		Validation		Testing	
						Mean	STD	Mean	STD	Mean	STD
Raman	1.0	Hybrid PCA (15)	3	2.907	Linear	98.11	0.50	97.33	1.90	92.33	1.90
Raman (peak)	1.0	Hybrid PCA (5)	4	1.981	Linear	96.33	1.08	98.33	1.67	95.67	2.24
LIBS	1.0	Hybrid PCA (45)	3	2.106	Linear	96.77	0.11	96.41	0.86	94.92	1.08

Features : No. of selected features
C : No. of clusters
FC : Fuzzification coefficient
Poly : Polynomial type

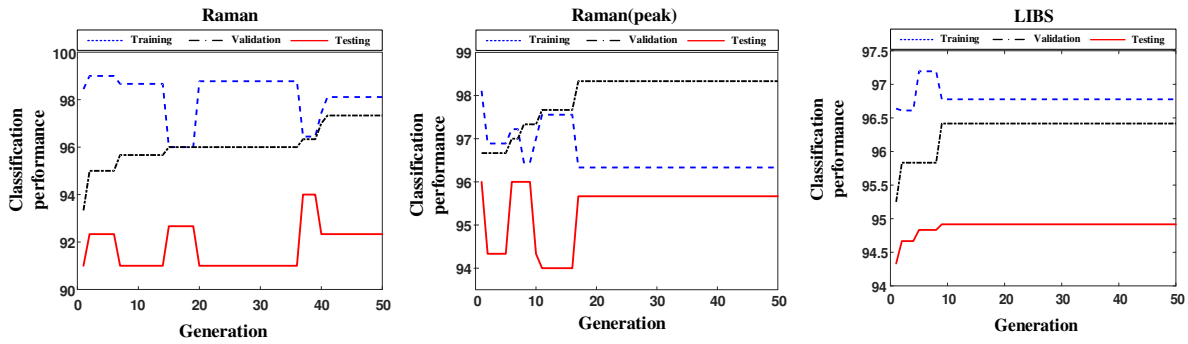


Fig. 10. Optimization process of classification performance of FRBFNN classifier according to the increase of the number of generations

Figure 10 portrays variation of the classification performance of training, validation, and testing data according to the increase of the number of generations. In the case of Raman dataset, the classification performance the training and validation data, respectively of the optimized FRBFNN classifier is 98.11% and 97.33%. 92.33% of the testing data is sorted by the proposed classifier. Raman data reconstructed by chemical characteristic shows the classification performance of 96.33%, 98.33%, and 95.67% when it comes to training, validation, and testing data. In the case of LIBS dataset, for the training data, optimized FRBFNN classifier sorts 96.77%. 96.41% of the validation data is classified by this classifier and the classification performance for the testing data is 94.92%.

By using the spectra data of black plastic wastes, we compared the classification performance with Weka algorithms. In the case of Raman and LIBS data, they consist lots of the number of input variables so dimensionality reduction technique is considered. In order to compare the classification performance between the proposed algorithm and Weka algorithms, we reduced the input variables by using the preprocessing techniques. The input variables of original Raman data, Raman data based on chemical peak points reconstructed by considering chemical characteristics, and LIBS data are reduced by preprocessing techniques. Table 13 represents the comparison of classification performance between the proposed classifier and various Weka algorithms.

Table 13. Comparison of classification performance between evolutionarily optimized FRBFNN classifier and various Weka classification algorithms combined with preprocessing techniques

Classifier	Preprocessing tech.	Raman (No. of reduced input variables)	Raman (peak) (No. of reduced input variables)	LIBS (No. of reduced input variables)
NaïveBayes	PCA	58.00±4.31 (15)	91.00±4.94 (5)	75.66±2.79 (45)
MLPs		***93.14±4.15*** (15)	94.50±2.36 (5)	92.92±0.93 (45)
RBF Classifier		80.67±6.52 (15)	89.33±1.90 (5)	81.08±8.77 (45)
SVM		78.33±13.12 (15)	90.00±4.47 (5)	89.16±2.14 (45)
k-NN (*k*=5)		90.00±6.45 (15)	91.00±2.78 (5)	93.45±2.29 (45)
PART		88.66±5.70 (15)	87.33±1.90 (5)	91.08±2.35 (45)
J48		89.33±6.41 (15)	86.10±6.48 (5)	87.67±1.57 (45)
Random tree		76.00±5.08 (15)	83.66±3.20 (5)	91.33±2.34 (45)
Proposed	Hybrid PCA	92.33±1.90 (15)	***95.67±2.24*** (5)	***94.92±1.08*** (45)

In the case of Raman data with chemical peak points, the classification performance is better when compared to that of original Raman data obtained through entire spectra range. In order to evaluate of the proposed classifier with noise, we performed the experiment by the testing data with Gaussian noise 10, 20, and 30dB, respectively and the classification performance is shown in Table 14.

Table 14. Comparison of classification performance between evolutionarily optimized FRBFNN classifier and Weka classification algorithms combined with preprocessing techniques when added white Gaussian noise

Noise / Algorithm	10dB			20dB			30dB		
	Raman	Raman (peak)	LIBS	Raman	Raman (peak)	LIBS	Raman	Raman (peak)	LIBS
NaïveBayes	46.66±9.72	73.67±5.05	74.75±3.23	59.67±7.94	88.67±5.05	76.67±4.37	55.66±4.50	88.00±1.82	78.25±4.80
MLPs	66.67±6.87	***78.00±4.31***	81.25±2.90	84.33±8.54	91.00±4.18	92.08±1.79	90.15±2.58	93.33±3.12	92.33±2.74
RBF Classifier	45.33±12.8	68.00±4.62	85.00±4.75	72.00±15.9	89.67±4.62	71.33±24.2	85.66±6.72	87.00±10.89	84.33±1.63
SVM	66.00±7.32	76.66±3.03	87.75±1.63	72.00±5.05	91.33±3.61	88.41±1.19	79.00±13.2	90.33±1.39	89.91±1.98
k-NN (*k*=5)	64.67±7.40	75.00±5.89	72.33±7.49	76.33±3.42	89.67±5.05	72.33±3.79	85.00±5.40	88.66±6.60	72.50±5.49
PART	60.33±8.45	64.00±7.51	87.75±3.83	86.33±4.77	80.00±5.00	89.25±2.09	87.00±3.20	86.00±1.90	89.00±2.29
J48	62.66±8.54	68.00±4.62	87.75±3.11	83.33±6.56	89.67±4.62	88.16±1.60	87.00±4.31	83.67±4.91	88.25±2.75
Random tree	57.33±12.1	65.66±2.52	80.75±3.27	74.33±8.46	82.66±7.87	81.16±3.27	76.33±6.81	81.66±3.91	82.00±3.23
Proposed	***71.67±5.00***	***78.00±4.47***	***91.50±1.30***	***87.67±3.84***	***95.00±3.72***	***92.68±1.99***	***91.67±2.79***	***94.66±2.17***	***94.25±1.43***

As the comparative results of the classification performance when white Gaussian noise is added to the testing data, we can see the preferred classification performance of the evolutionary optimized FRBFNN classifier proposed in this study. The experimental results of the proposed classifier show the superiority from the viewpoint of robustness.

6. Concluding Remarks

In this study, we proposed evolutionarily optimized FRBFNN classifier by using preprocessing techniques as well as PSO. The classification performance of the evolutionarily optimized FRBFNN classifier is shown through experiments by using several machine learning datasets. The conclusions drawn from the experiments are as follows.

a) 16 datasets out of the 22 (about 72%) machine learning datasets used in the experiment showed the superb classification performance when the preprocessing techniques are used. It is confirmed that the classification performance of the classifier can be visibly improved based on preprocessing techniques after the number of entire input variables and the number of patterns of the data are determined.

b) We confirmed that the classification performance can be made better when compared to that of the conventional FCM-based RBFNN classifier through the optimization of six parameters. The six parameters are given as follows; whether or not the preprocessing techniques is used, the selection of preprocessing techniques, the number of input variables extracted from preprocessing technique, the number of fuzzy rules and fuzzification coefficient (*FC*) used in FCM clustering, and the polynomial type of be using as connection weights between hidden layer and output layer.

c) Through the Friedman test one of the statistical analysis methods, we demonstrated that the proposed classifier has a higher competitiveness than various classification algorithms of Weka data mining software in terms of the obtained classification performance.

Furthermore, we evaluated the classification performance by using spectra data obtained from Raman and LIBS equipment for practical application related to material sorting of black plastic wastes. The conclusions drawn from the experiments are given as follows.

a) In the case of spectral data obtained from Raman spectroscopy, we constructed peak data considering chemical characteristics as well as entire raw data for improvement of the classification performance. As a result, we confirmed that the classification performance of spectral data reconstructed by considering chemical characteristics was improved 2% higher than the classification performance of the entire raw data.

b) When the classifier is applied to practical application, the experimental results of the proposed classifier have shown preferred classification performance against noise because data including noise contains as input of the classifier. For this reason, the classification performance was evaluated by adding 5, 10, and 20dB white Gaussian noise to testing data respectively. In the sequel, the proposed classifier was demonstrated to show the superiority from the viewpoint of robustness.

Some future studies could be focused on the comparison of multi-objective particle swarm optimization (MOPSO) with single objective PSO in terms of the effective improvement of classification performance as well as the construction of some practical application systems related to the proposed classifier.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government(MSIT) (NRF-2021R1F1A1056102)

Compliance with ethical standards

Conflict of interest Sang-Beom Park, Sung-Kwun Oh, and Witold Pedrycz declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animal performed by any of the authors.

Informed consent None.

References

Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: A review. IEEE Transactions on pattern analysis and machine intelligence 22(1):4-37

Kwan HK and Cai Y (1994) A fuzzy neural network and its application to pattern recognition. IEEE transactions on Fuzzy Systems 2(3):185-193

David de la Mata-Moya, Jarabo-Amores MP, Rosa-Zurera M, Borge JCN, López-Ferreras F (2009) Combining MLPs and RBFNNs to detect signals with unknown parameters. IEEE Transactions on Instrumentation and Measurement 58(9):2989-2995

Apostolikas G and Tzafestas S (2003) On-line RBFNN based identification of rapidly time-varying nonlinear systems with optimal structure-adaptation. Mathematics and Computers in Simulation 63(1):1-13

Park B-J, Pedrycz, W, Oh S-K (2010) Polynomial-based radial basis function neural networks (P-RBF NNs) and their application to pattern classification. Applied Intelligence 32(1):27-46

Tolba AS and Abu-Rezq AN (2000) Combined classifiers for invariant face recognition. Pattern Analysis & Applications 3(4):289-302

Alexandridis A, Sarimveis H, Bafas G (2003) A new algorithm for online structure and parameter adaptation of RBF networks. Neural Networks 16(7):1003-1017

Oh S-K, Kim W-D, Pedrycz W (2016) Design of radial basis function neural network classifier realized with the aid of data preprocessing techniques: design and analysis. International Journal of General Systems 45(4):434-454

Er MJ, Wu S, Lu J, Toh HL (2002) Face recognition with radial basis function (RBF) neural networks. IEEE transactions on neural networks 13(3):697-710

Oh S-K, Kim W-D, Pedrycz, W, Park B-J (2011) Polynomial-based radial basis function neural networks (P-RBF NNs) realized with the aid of particle swarm optimization. Fuzzy Sets and Systems 163(1):54-77

Yeung KY and Ruzzo WL (2001) Principal component analysis for clustering gene expression data. Bioinformatics 17(9):763-774

Kim KI, Jung K, Kim HJ (2002) Face recognition using kernel principal component analysis. IEEE signal processing letters 9(2):40-42

Costa NC, Hu W, Bai ZJ, Scalettar RT, Singh RR (2017) Principal component analysis for fermionic critical points. Physical Review B 96(19):195138

Romero A, Gatta C, Camps-Valls G (2015) Unsupervised deep feature extraction for remote sensing image classification. IEEE Transactions on Geoscience and Remote Sensing 54(3):1349-1362

Roh S-B, Oh S-K, Yoon J-H, Seo K (2019) Design of face recognition system based on fuzzy transform and radial basis function neural networks. Soft Computing 23(13):4969-4985

Wang S, Lu J, Gu X, Du H, Yang J (2016) Semi-supervised linear discriminant analysis for dimension reduction and classification. Pattern Recognition 57:179-189

Ye Q, Yang J, Liu F, Zhao C, Ye N, Yin T (2016) L1-norm distance linear discriminant analysis based on an effective iterative algorithm. IEEE Transactions on Circuits and Systems for Video Technology 28(1):114-129

Mahmoudi N and Duman E (2015) Detecting credit card fraud by modified Fisher discriminant analysis. Expert Systems with Applications 42(5):2510-2516

Li C-N, Shao Y-H, Deng N-Y (2015) Robust L1-norm two-dimensional linear discriminant analysis. Neural Networks 65:92-104

Perfilieva I (2006) Fuzzy transforms: Theory and applications. Fuzzy sets and systems 157(8):993-1023

Perfilieva I, Novák V, Dvořák A (2008) Fuzzy transform in the analysis of data. International Journal of Approximate Reasoning 48(1):36-46

Wang X, Liu X, Pedrycz W, Zhang L (2015) Fuzzy rule based decision trees. Pattern Recognition 48(1):50-59

Riza LS, Bergmeir CN, Herrera Triguero F, Benítez Sánchez JM (2015) frbs: Fuzzy rule-based systems for classification and regression in R. American Statistical Association

Chang L, Zhou Z, You Y, Yang L, Zhou Z (2016) Belief rule based expert system for classification problems with new rule activation and weight calculation procedures. Information Sciences 336:75-91

Hu X, Li SE, Yang Y (2015) Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles. IEEE Transactions on Transportation electrification 2(2):140-149

Khosravi A, Koury RNN, Machado L, Pabon JJG (2018) Prediction of hourly solar radiation in Abu Musa Island using machine learning algorithms. Journal of Cleaner Production 176:63-75

Zhou K and Yang S (2016) Exploring the uniform effect of FCM clustering: A data distribution perspective. Knowledge-Based Systems 96:76-83

Zhang L, Zhong W, Zhong C, Lu W, Liu X, Pedrycz W (2017) Fuzzy C-Means clustering based on dual expression between cluster prototypes and reconstructed data. International Journal of Approximate Reasoning 90:389-410

Anter AM and Ali M (2020) Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems. Soft Computing 24(3):1565-1584

Qiu X, Zhang L, Suganthan PN, Amaratunga GA (2017) Oblique random forest ensemble via Least Square Estimation for time series forecasting. Information Sciences 420:249-262

Oh S-K, Yoo S-H, Pedrycz W (2013) Design of face recognition algorithm using PCA-LDA combined for hybrid data pre-processing and polynomial-based RBF neural networks: Design and its application. Expert Systems with Applications 40(5):1451-1466

Kennedy J and Eberhart R (1995) Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks 4:1942-1948

Jordehi AR, Jasni J, Abd Wahab N, Kadir MZ, Javadi MS (2015) Enhanced leader PSO (ELPSO): a new algorithm for allocating distributed TCSC's in power systems. International Journal of Electrical Power & Energy Systems 64:771-784

Deng W, Yao R, Zhao H, Yang X, Li G (2019) A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm. Soft Computing 23(7):2445-2462

Aljarah I, Faris H, Mirjalili S (2018) Optimizing connection weights in neural networks using the whale optimization algorithm. Soft Computing 22(1):1-15

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research 7:1-30

Roh S-B, Oh S-K, Park E-K, Choi W-Z (2017) Identification of black plastics realized with the aid of Raman spectroscopy and fuzzy radial basis function neural networks classifier. Journal of Material Cycles and Waste Management 19(3):1093-1105

Roh S-B, Park S-B, Oh S-K, Park E-K, Choi W-Z (2018) Development of intelligent sorting system realized with the aid of laser-induced breakdown spectroscopy and hybrid preprocessing algorithm-based radial basis function neural networks for recycling black plastic wastes. Journal of Material Cycles and Waste Management 20(4):1934-1949

Bae J-S, Oh S-K, Pedrycz W, Fu Z (2019) Design of fuzzy radial basis function neural network classifier based on information data preprocessing for recycling black plastic wastes: comparative studies of ATR FT-IR and Raman spectroscopy. Applied Intelligence 49(3):929-949

Khan S, Naseem I, Malik MA, Togneri R, and Bennamoun M (2018) A fractional gradient descent-based rbf neural network. Circuits, Systems, and Signal Processing 37(12), 5311-5332.

Yazdi HS, Rowhanimanesh A, and Modares H (2012) A general insight into the effect of neuron structure on classification. Knowledge and Information Systems, 30(1): 135-154.

Oong TH, and Isa NAM (2011) Adaptive evolutionary artificial neural networks for pattern classification. IEEE Transactions on Neural Networks 22(11):1823-1836

Zhao Y, Pei J, and Chen H (2019) Multi-layer radial basis function neural network based on multi-scale kernel learning. Applied Soft Computing, 82: 105541

Hussain SS, Usman M, Siddique THM, Naseem I, Togneri R, and Bennamoun M (2021) q-RBFNN: A Quantum Calculus-based RBF Neural Network. arXiv preprint arXiv:2106.01370.

Zhao H, Zheng J, Xu J, and Deng W (2019) Fault diagnosis method based on principal component analysis and broad learning system. IEEE Access, 7: 99263-99272.

Artoni F, Delorme A, and Makeig S (2018). Applying dimension reduction to EEG data by Principal Component Analysis reduces the quality of its subsequent Independent Component decomposition. NeuroImage 175: 176-187.

Patanè G (2021). Continuous Fuzzy Transform as Integral Operator. IEEE Transactions on Fuzzy Systems 29(10): 3093-3102.

Manchanda M, and Sharma R (2018) An improved multimodal medical image fusion algorithm based on fuzzy transform. Journal of Visual Communication and Image Representation 51: 76-94.

Grassi S, and Pareschi L (2021) From particle swarm optimization to consensus based optimization: stochastic modeling and mean-field limit. Mathematical Models and Methods in Applied Sciences 31(08): 1625-1657.

Zhang Y, Liu X, Bao F, Chi J, Zhang C, and Liu P (2020) Particle swarm optimization with adaptive learning strategy. Knowledge-Based Systems 196(105789).

Shanthi MB, Meenakshi DK, Ramesh PK (2018) Particle swarm optimization. In Advances in Swarm Intelligence for Optimizing Problems in Computer Science (pp. 115-144). Chapman and Hall/CRC.

Zemmal N, Azizi N, Sellami M, Cheriguene S, Ziani A, AlDwairi M, and Dendani N (2020) Particle swarm optimization based swarm intelligence for active learning improvement: Application on medical data

classification. Cognitive Computation 12(5): 991-1010.

El-Kenawy ES, and Eid M (2020). Hybrid gray wolf and particle swarm optimization for feature selection. International Journal of Innovative Computing Information and Control 16(3): 831-844.

Zhang S, Xu J, Lee LH, Chew EP, Wong WP, and Chen CH (2016) Optimal computing budget allocation for particle swarm optimization in stochastic optimization. IEEE Transactions on Evolutionary Computation 21(2): 206-219.

Chicco D, and Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics 21(1): 1-13.

Irigaray D, Dufrechou E, Pedemonte M, Ezzatti P, and López-Vázquez C (2019) Accelerating the calculation of Friedman test tables on many-core processors. In Latin American High Performance Computing Conference. Springer, Cham: 122-135.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Highlights.docx