1. Introduction

In the last few decades, machine learning [1] is a key research area among the researchers due to the dynamic generation and availability of large volume data. Converting this large volume of data into knowledge is one of the biggest challenges. There are various machine learning techniques such as classification [2–5], clustering [6], prediction [7, 8], and system control [9, 10] used for the aforesaid problems. Classification is one of the important machine learning techniques for constructing a model that classifies data into different class labels. It gives the detailed knowledge [11] of the domain that are being classified. To address the real-world problems, ANN [12] is used as a tool for the classification of different tasks such as classification, clustering, and regression similar to the human brain. ANN is well-known for massive parallel in structure that process a large amount of data simultaneously. It has also high precision and high learning ability even in the presence of very little amount of information. ANN is already applied successfully in different problem domains such as time series prediction [13], clustering, and system control [14]. The major drawback of ANN is to handle the imprecise or uncertain data [15]. Due to the presence of imprecise and ambiguous input information, some uncertainties [16, 17] may arise at any stage of the data classification process. Fuzzy Set (FS) is a most suitable technique to handle these uncertainty issues. The degree of membership values of each feature with respect to several class labels are determined using the FS. These membership values can easily deal with the uncertain...
and imprecise information. The distinct feature of FS is that it can work efficiently even with incomplete or imprecise data as compared to other mathematical models. But this fuzzy expansion is a widely used concept in most of the complex problems in order to handle uncertainty. However, this leads to computationally heavy tasks while processing large datasets. But, on the contrary, the use of fuzzy expansion is also essential too. So, we cannot completely eliminate the fuzzification process. Rather, we may avoid the processing of less significant fuzzified values. Hence, it is essential to integrate different individual techniques to form various hybrid techniques. These hybridization techniques provide an intelligent system that performs better than individual techniques to deal with the real-world problems. The motivation behind the hybrid model is to eliminate the drawbacks of individual techniques and build hybrid models that are more efficient and transparent than individual models. The hybridization of various techniques is successfully applied in several applications such as biomedical signal processing [18], cloud optimization [19, 20], forecasting [21], and healthcare [22–24]. The mostly used hybridized model called Adaptive Neuro-Fuzzy Inference System (ANFIS) [25] is successfully applied to different problem domains such as classification, prediction, and pattern recognition. But the major drawback of this model is that it is governed by fuzzy rule sets. These rule sets take more time to train the model and makes the network more complex. It provides better performance only when the fuzzy rule sets are properly designed. Another variant of the NF model that hybridizes both the NN and FS models is described in [26, 27]. In this model, the capability of the fuzzy system is to adapt the problems in a way the human perceives along with the learning ability of NN. It is hybridized to enhance different characteristics such as adaptability, speed, and flexibility of the model. Ghosh et al. [26] used a II-type membership function [28] for the fuzzification process that expands the input features based on the number of classes available in the class label attribute. Due to this fuzzy expansion of these input features, the complexity of the network increases and takes a longer time for training and testing the model. It increases the dimensionality [29] of the problem due to the fuzzification process and leads to a major obstruction that improvises the performance of the model [30–32]. This NF model gives a considerable improvement in accuracy than ANN, but its computational time increases. Apart from this, all the expanded features may not always contribute significantly to the network. Hence, it is essential to remove these features that are not significantly contributing much to the network. In this regard, dimensionality reduction algorithms play an important role to eliminate the features that are not significantly contributing to the network. Therefore, a feature reduction algorithm called Principal Component Analysis (PCA) [33] is used to eliminate the irrelevant features to improve the performance and reduce the computational cost of the model. The NF models have been found successful in many real-life applications [34] in science and engineering. It is suitable to process the imprecise and uncertainty nature of problems [35] by interpreting the input features before processing. It makes the model more robust and transparent in the decision-making process. This attracted many researchers to adopt the NF model for data analysis. Motivated by the above facts, this work projects a scheme for the improvement of the NF model.

The objective of the work is to develop a hybrid model called NF-FR for data classification. In the first step, II-type membership function is employed for fuzzification of input patterns. Then, the feature reduction algorithm is applied on fuzzified variables after the fuzzification process (postfeature reduction) of the NF model to develop neuro-fuzzy with the feature reduction (NF-FR) model. In this model, postfeature reduction has been employed on fuzzified patterns to filter out the irrelevant, redundant, and noisy features. Unlike prefeature reduction, this allows all the features to participate in the fuzzification process and then identify irrelevant features from the fuzzified patterns. This approach allows exploring potential fuzzified features from the weak feature set. The NF-FR model extracts the fuzzified information that is truly contributing to the network for speeding up the classification process by eliminating the irrelevant fuzzified features. The major observation is that the overall time required for running the algorithm decreases considerably by using the FR algorithm called PCA. Thus, the NF-FR model not only provides more accurate results but also reduces the execution time. We have compared four models such as ANN, ANN with FR using PCA (ANN-FR), NF, and NF-FR using ten benchmark datasets from the UCI machine learning repository. Each dataset is then evaluated on various performance measuring parameters such as root mean square error (RMSE), f-measure, precision, and recall from the confusion matrix.

The remainder of this paper is organized as follows: Section 2 provides the review of related works. Section 3 describes the basic preliminaries of NF model, PCA, and the detailed proposed NF-FR model of this research work. Section 4 presents the detailed experimental setup along with simulating environment and result analysis of the proposed model. Section 5 describes the statistical analysis of all the models, and finally, Section 6 concludes the work with future scope of the article.

2. Literature Survey

Several soft computing techniques such as FS, NN, NF and dimensionality reduction play a critical role in the development of hybrid models in the last few last decades. The hybridization of these techniques is also considered to be one of the benchmark works in the field of data mining, machine learning, and pattern recognition. This literature review indicates the most recent development of aforesaid models and their applications in various fields. FS proposed by Zadeh represents the way of representation of human perception specially in diverse fields such as language communication, pattern recognition, and information abstraction that solves uncertainty issues. These uncertainty problems can be resolved through different fuzzification techniques that are used to convert the input features into...
its corresponding fuzzified feature sets. This fuzzification process can be represented in two ways such as class belongingness fuzzification and class nonbelongingness fuzzification. Ghosh et al. [27] proposed a NF classification model in which features are fuzzified based on the bell-shaped membership function. The fuzzified matrix formed from the input features was associated with a degree of belongingness to the different classes. The class labels determined the value of the degree of belongingness towards that class. Pal and Mitra [28] used a membership function that converts the crisp values into its linguistic values, and these linguistic values are used as input patterns to the network instead of numeric values. They have used II-type membership functions for fuzzification and ANN-based MLP with backpropagation model. But they do not consider any prone or addition to the network structure. Meher [34] proposed NF classification using the rough set approach that utilizes the best possible extracted features. This is obtained through feature-wise belongingness of patterns using fuzzy set to deal with imprecision and rough set for uncertainty. Kar et al. [35] provided a recent survey on the NF classification model development during the period of 2002 to 2012 in different application fields such as traffic control, economic system, medical system, and image processing. Viharos and Kis [36] conducted a detailed survey on different NF models such as ANFIS, FALCON, GARIC, NEFCON, and SONFIN along with their architecture. It also gives the detailed survey of the use of these models in the technical diagnostics and measurement field. The detailed survey of NF models from 2000 to 2017 for classification is described in [37]. Das and Pratihar [38] used neuro-fuzzy with multiobjective optimization techniques to inherit fuzziness in the manufacturing process. Škrjanc et al. [39] addressed a review on evolving neuro-fuzzy and fuzzy rule-based models used in real-world environments for classification, clustering, regression, and system identification. In the data analysis process, dimensionality reduction techniques such as feature selection and feature reduction are used in the preprocessing [40] stage in which the original features are transformed into either original feature or transformed features. Chattopadhyay [41] addressed a NF model for the diagnosis of human depression based on certain symptoms. This model used PCA for feature reduction from fourteen features to seven features of those that are relevant and significantly contributing to the decision-making process of disease identification. Ibrahim et al. [42] used a data-adaptive NF inference model for early detection and classification of diabetes disease based on symptoms. Alvanitopoulos et al. [43] proposed a NF classification technique for the identification of damages produced by an earthquake on construction. After the manifestation of the earthquake, the evaluation of the safety of existing structure and measure to be taken for automatic damage classification of buildings is considered. Chen [44] proposed an online NF model for the deadline constraint message scheduling system. It adapts the network structure and parameters to explore the dynamic behaviour of the message scheduling system. Azhari and Kumar [45] addressed a NF approach for text summarization. It filters the high-quality summary sentences on the document understanding conference data corpus. Singh et al. [46] proposed an enhanced NF model used for clustering that reduces the number of linguistic variables as compared to the NF model. Nilashi et al. [47] used ensembles ANFIS model, clustering along with dimensionality reduction for prediction of hepatitis disease diagnosis. Shihabudeen et al. [48] addressed a PSO-based ELM-ANFIS model for regression and classification to reduce the computational cost, randomness, and better generalization.

PCA also plays an important role to eliminate redundant features from the input pattern which improves system performance along with accuracy. It extracts the important information from the datasets and represents it as a new set of orthogonal variables called principal components. It is a statistical method used to reduce the number of variables by collecting highly correlated variables. Polat and Güneş [49] used PCA and ANFIS techniques. They have used the feature reduction algorithm to reduce the number of input features of the diabetes dataset from eight features to four features and conducted the predictive diagnosis by passing the inputs through the ANFIS model. Wang and Paliwal [50] proposed dimensionality-based feature extraction algorithms such as linear discriminant analysis and PCA for vowel recognition. It transforms the input parameter into the feature vector and reduces its dimension to make the classification process more efficient. Azar [51] addressed a feature selection method based on linguistic hedges in the adaptive NF model for medical diagnosis. It reduces the dimensions of the problem and also enhances the performance of the classification by eliminating redundant and noisy features. It also speeds up the computational time of the learning algorithm and simplifies the classification task. Keles et al. [52] proposed a NF tool for prostate cancer classification. This model diagnosis to find a set of rules that can be interpreted linguistically. Gabrys [53] addressed a general fuzzy max-min network model for uncertain information processing in the industry. It analyses and identifies whether to combine or not combine different techniques to form hybridization. Übayli [54] applied the ANFIS model for the classification of ECG signals. They used Lyapunov exponents for feature extraction, and ANFIS is used for classification. Kolodyazhnyi et al. [55] used PCA for dimension reduction and NF Kolmogorov’s network for classification for waste water treatment plant data. Schclar et al. [56] ensemble various models based on the dimensionality reduction. Due to an increase in the dimension of input features in the NF model, the computational cost is increased. To address this issue, various feature reduction techniques are employed in the preprocessing stage. However, our present investigation of the proposed NF-FR model is justified with class belongingness fuzzification of input features. These fuzzified features are filtered out by the PCA to produce the reduced features. These reduced features are passed to the ANN-BPN based model for training and testing. This experimentation is done with ten numbers of both balanced and unbalanced datasets.
3. System Model

Last few decades, researchers are trying to design the hybrid systems by using the fuzzy system and neural network for pattern classification. The basic concepts of NF models, PCA, and the proposed hybrid model of NF-FR models are presented in the following sections.

3.1. Neuro-Fuzzy Model. In real-world problems, uncertainty is one of the major challenges which leads to incomplete and imprecise information about the input data in pattern classification problems. Therefore, it is necessary to provide ample provision to handle uncertainty. In the NF model, instead of a normal crisp input value, fuzzy values are being inputted to the neural network. The fuzzified matrix is a result of the fuzzification process that generates a membership matrix in which the total number of element present in this matrix is equal to the product of the number of features and number of classes present in the dataset, which is input to the neural network. The fuzzified input matrix is associated with the degree of belongingness with respect to the classes, which extracts the feature-wise information of the input pattern. Each feature value of a pattern represents membership values for each class, where membership values are measured by using Π-type membership functions as represented in Figure 1. This fuzzification matrix is passed to the ANN model to train the network.

3.2. Principal Component Analysis. The fuzzification process may result in high dimensional data, where all the features may not carry significant information for discrimination of the pattern. Furthermore, this increase in dimension affects the complexity of machine learning algorithms. This section describes the working principle of a feature reduction algorithm called PCA to extract the relevant features from the original feature set to reduce the dimensions of the data. This can be achieved by transforming the high dimensional features into new small transformed features without losing the essential information of the original datasets. These new sets of features are called principal components in which the data varies as the linear combination of original features. PCA considers only those components that have a larger variance of the data. The major objective of PCA is to identify the hidden patterns of the data, determine the correlation among the features, and decrease the dimensionality of the features by eliminating the redundant and noisy features.

3.3. Proposed Hybrid Neuro-Fuzzy and FR Model. In this model, feature-wise information of input patterns is extracted from the original data with respect to different classes. Since all features are not equally important in discriminating the instances, the feature-wise belonging is expected to help in the classification process. In this section, a detailed schematic diagram of the novel NF-FR model has been proposed for the classification of nonlinear data. This proposed model is moving parts into three major steps such as input feature fuzzification process, feature reduction using PCA, and classification using ANN with backpropagation learning.

Initially, this NF-FR model is used to extract the feature-wise information from the input pattern to its corresponding fuzzified matrix by using the class belongingness fuzzification technique. In this present study, we have used a popular Π-type membership function for fuzzification of the input pattern. Since all the features may not be significantly contributing to the classification process, it is essential to find the class belongingness of each attribute. In order to achieve this, here Π-type membership functions have been used for the fuzzification process which provides the degree of the belongingness of individual features with respect to the class labels. As a result, each feature value of the input patterns has been expanded to C number of values, where C is the number of class labels. Furthermore, such expansion of input patterns may include some insignificant features, and thus, PCA has been used for the pruning of irrelevant features. Finally, the ANN model has been used for the classification process with backpropagation learning, and the output of the ANN model is defuzzified to get the final result. The block diagram of the proposed NF-FR classification model is shown in Figure 2, and the detailed working model is shown in Figure 3. This proposed model has been discussed in detail below.

Step 1 (Fuzzification process). In this step, the n-dimensional input pattern $P_i = (F_{i,1}, F_{i,2}, \ldots, F_{i,n})$ is considered as an input pattern, where n is the number of features available in the dataset. Here, the membership value of each feature of the dataset is computed by using Π-type membership function, which is represented in Figure 1. The membership value of the $j^{th}$ instance of the $i^{th}$ feature with respect to class labels $C$ is denoted as $\mu_C(F_{j,i})$, where $F_{j,i}$ is the $j^{th}$ instance of $i^{th}$ feature of the dataset, and $C = 1, 2, \ldots, k$ represents the number of classes available in the dataset. This fuzzification process $\mu_C(F_{j,i})$ provides the degree of membership of individual features with respect to the different class labels.
Here II type membership has been used for fuzzification and controlling the steepness of the model which may be realized in the following equation:

\[ \mu(F_i) = \begin{cases} \frac{1}{1 + \left(\frac{x}{a} - 1\right)^2}, & x \leq a \\ \frac{1}{1 + \left(\frac{x}{b} - 1\right)^2}, & x > a \end{cases} \]

where \( \mu(F_i) \) represents the membership value of the \( i \)th instance of the dataset, and \( x \) is the crisp value of the \( i \)th feature of the dataset.

The computation of II type membership function can be represented as the following:

\[ \mu(F_i) = \begin{cases} \frac{1}{1 + \left(\frac{x}{a} - 1\right)^2}, & x \leq a \\ \frac{1}{1 + \left(\frac{x}{b} - 1\right)^2}, & x > a \end{cases} \]

and \( g \) is the output of the model.

The equation for II type membership function can be represented as follows:

\[ g = c + \frac{\sum_{i=1}^{n} (\mu(F_i) - \min_{i=1}^{n}(\mu(F_i))) \cdot (\mu(F_i) - \max_{i=1}^{n}(\mu(F_i)))}{\sum_{i=1}^{n} (\mu(F_i) - \min_{i=1}^{n}(\mu(F_i)))} \]

where \( c \) is the center of the complete dataset, \( \mu(F_i) \) represents the membership value of the \( i \)th instance of the dataset, and \( g \) is the output of the model.

The equation for II type membership function can be represented as the following:

\[ g = c + \frac{\sum_{i=1}^{n} (\mu(F_i) - \min_{i=1}^{n}(\mu(F_i))) \cdot (\mu(F_i) - \max_{i=1}^{n}(\mu(F_i)))}{\sum_{i=1}^{n} (\mu(F_i) - \min_{i=1}^{n}(\mu(F_i)))} \]

where \( c \) is the center of the complete dataset, \( \mu(F_i) \) represents the membership value of the \( i \)th instance of the dataset, and \( g \) is the output of the model.

The equation for II type membership function can be represented as the following:

\[ g = c + \frac{\sum_{i=1}^{n} (\mu(F_i) - \min_{i=1}^{n}(\mu(F_i))) \cdot (\mu(F_i) - \max_{i=1}^{n}(\mu(F_i)))}{\sum_{i=1}^{n} (\mu(F_i) - \min_{i=1}^{n}(\mu(F_i)))} \]

where \( c \) is the center of the complete dataset, \( \mu(F_i) \) represents the membership value of the \( i \)th instance of the dataset, and \( g \) is the output of the model.

The equation for II type membership function can be represented as the following:

\[ g = c + \frac{\sum_{i=1}^{n} (\mu(F_i) - \min_{i=1}^{n}(\mu(F_i))) \cdot (\mu(F_i) - \max_{i=1}^{n}(\mu(F_i)))}{\sum_{i=1}^{n} (\mu(F_i) - \min_{i=1}^{n}(\mu(F_i)))} \]

where \( c \) is the center of the complete dataset, \( \mu(F_i) \) represents the membership value of the \( i \)th instance of the dataset, and \( g \) is the output of the model.
the $C$ class level. Here, $C = C_1, C_2, \ldots, C_k$ is the number of classes available in the dataset. The output of this process is a fuzzified matrix that contains the expanded fuzzified features of the input pattern. The example of fuzzification results of one feature (petal width) of the IRIS dataset is represented in Figure 4. Each membership value of all the features of the dataset will be transformed within the range of \([0-1]\) as shown in Figure 4.

**Step 2** (Feature reduction process). Due to expansion of input feature, the complexity of the model increases. To make the classification process more effective and efficient, PCA is used to reduce the features of the fuzzified membership matrix. In this step, this fuzzified membership matrix $\mu(P)$ is used as an input to the PCA algorithm to reduce the dimensions of the features.

Let, the aforesaid fuzzified membership matrix $\mu(P)$ having $q$ number of fuzzified features that can be expressed as $P_i = (F_{i,1}, F_{i,2}, \ldots, F_{i,q})$, $q = n \times C$, $n$ is the number of features, and $C$ is the number of class labels. The covariance matrix of the fuzzified membership matrix is computed by using the following equation:

$$CM = Cov(\mu(P)) = \frac{1}{m-1} \sum_{i=1}^{m} (F_{ij} - \mu(F_{ij})) \\ \times \left( F_{ij} - \mu(F_{ij}) \right)^T,$$

(6)

where $\mu(F_{ij})$ is the sample mean of the feature $F_{ij}$ and $m$ represents the number of samples to be considered. The components of the covariance matrix ($CM_{ij}$) represents the variances of the features $F_i$ and $F_j$.

![Figure 3: The detailed working model of proposed models.](image)
Let \( r \) be the number of principal axes \( A_1, A_2, \ldots, A_r \) which represents the eigenvectors of the covariance matrix, where \( 1 \leq r \leq q \), in which the variance is maximum in the projected space. The mean value of each feature of the fuzzified membership matrix is computed as \( \mu(F_{ij}) \) where \( i = 1, 2, \ldots, q \) and \( j = 1, 2, \ldots, m \). The mean value of each feature is subtracted from each of the data dimensions to produce a dataset whose mean is zero.

The eigen values \( (\alpha_i) \) and eigenvectors \( (A_i) \) of the covariance matrix are computed easily as it is a symmetric matrix. The eigenvectors \( A_i \) and its corresponding eigen values \( \alpha_i \) are computed by using the following equation:

\[
CM \times A_i = \alpha_i \times A_i,
\]

where \( i = 1, 2, \ldots, r \) and \( r \) is the number of principal components that can be derived by using equation (8) and can be represented by the descending order of the eigenvalues of the corresponding eigenvectors.

\[
X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,r}] = A^T \times P_i.
\]

The output of this step is the reduced matrix \( X \) that contains the relevant information of the input features that are necessary in decision-making process of the classification. In the third step, this reduced matrix is passed to the ANN as input.

**Step 3 (Building ANN-BPN model).** In this step, the artificial neural network with backpropagation (ANN-BPN) model is used for the classification process. The ANN-BPN model uses backpropagation as a supervised learning algorithm to train the artificial neural network. It updates the weights of the model to minimize the loss by efficiently computing the gradients. This network uses the reduced fuzzified matrix as input which is generated from the Step 2 to this model. In this model, all the weights of the input layer are fully connected to the hidden layers. All the hidden layers are also fully connected to themselves. The last hidden layer is also connected to the output layer. Initially, all weights are assigned with random weights in between the range of 0 and 1. The number of nodes available in the input layer is equal to the number of features available in the reduced fuzzified matrix. The number of nodes in the output layer is equal to the number of class labels available in the dataset. The number of nodes available in the hidden layer is computed by using equation (9), in which input_nodes, hidden_nodes, and output_nodes represent the number of input nodes, hidden nodes, and output nodes, respectively. The number of hidden layers and the number of neurons in each hidden layer depends on the complexity of the problem. There is no any standard method available to compute it in the literature, but some authors use the following equation to compute the number of neurons available in the hidden layer:

\[
\text{hidden\_nodes} = \sqrt{\text{input\_nodes} \times \text{output\_nodes}}. \tag{9}
\]

In the feedforward step, the model is trained based on the reduced fuzzified matrix input information. The net input is computed by making the sum of product of input patterns and the assigned weights and add the bias. Mathematically, the performance of the net input of the \( N^{th} \) neuron can be expressed in the following equation:

\[
U_N = B_N + \sum_{j=1}^{r} W_{i,j} \times X_{i,j},
\]

where \( B_N \) is the bias of the \( N^{th} \) neuron, \( X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,r}] \) is the input pattern of the reduced fuzzified matrix, \( W_i = [w_{i,1}, w_{i,2}, \ldots, w_{i,r}] \) are connection weights of the \( N^{th} \) neuron, and \( U_N \) is the net input of the model. Similarly, the net input of each layer is computed, and apply the activation function for output, which is determined between different connecting layers. The output of the output layer is computed by using the sigmoid activation function described in the following equation, where \( \phi \) is the activation functions and \( O_N \) is the output of the neuron:

\[
O_N = \phi(U_N). \tag{11}
\]

In the backpropagation step, the error is computed by subtracting the actual output with the target output, and the error is expressed in as follows:

\[
\text{error} = \sum_{j=1}^{m} \text{target\_output} - \text{actual\_output}, \tag{12}
\]

where \( j = 1, 2, \ldots, m \) indicates the number of output neurons. Here, the root mean square error (RMSE) \([5–10]\) can be computed by using the following equation:

\[
\text{RMSE} = \sqrt{\text{error}}. \tag{13}
\]

Similarly, the errors are computed, and the weights and bias are updated in the learning process. The weights of the connecting path between the different layers are adjusted by
computing the change in weights in the model to reduce the overall error of the model, which is realized in equation (14), where $\alpha$ is the learning rate between the range of $[0, 1]$:

$$\Delta \text{weight} = -\alpha \times \frac{\partial \text{error}}{\partial \text{output}} \times \frac{\partial \text{output}}{\partial \text{net}} \times \Delta \text{weight}$$

(14)

The new weights and bias of the model can be computed by using the following equations, respectively:

$$\text{new}_{\text{weight}} = \text{old}_{\text{weight}} + \Delta \text{weight},$$

$$\text{new}_{\text{bias}} = \text{old}_{\text{bias}} + \Delta \text{bias}.$$ 

(15) (16)

This process is repeated multiple times to minimize the root-mean-square error of the model or till the stopping criteria is reached. This proposed approach is different from ANN (feedforward with backpropagation), ANN-FR, and NF as follows: (i) in ANN (feedforward with backpropagation), all the input features are processed in parallel without extracting insignificant features which take more time to train the model and also leads to uncertainty problem. (ii) In the ANN-FR model, PCA is used in the preprocessing stage to eliminate insignificant features, but this model is unable to address the uncertainty issues. (iii) The NF model solves uncertainty issues by using the class belongingness fuzzification process, but it is unable to eliminate the redundant or noisy features. (iv) By considering aforesaid issues, the proposed model solves the uncertainty issue by using the class belongingness fuzzification process and also eliminates the insignificant fuzzified features by using PCA instead of eliminating the complete features which seem insignificant.

4. Result Analysis

In this section, the simulation environment and the dataset used for the training and testing phases for the analysis of the proposed model are presented. Here, four models (ANN, NF, ANN-FR, and NF-FR) are implemented using Matlab (version R2015a) with the Windows 7 operating system. The benchmark datasets from the UCI machine learning repository [57] are collected and tested in different classification models. The detail descriptions about all these datasets [57, 58] can be found at “http://archive.ics.uci.edu/ml/” and “http://keel.es/.”

Several performance comparison techniques such as classification accuracy, root mean square error (RMSE), precision, recall, and f-measure are obtained from the confusion matrix for all the benchmark datasets of each model, and the comparison result is presented. The details of these performance measures are outlined below. The comparison of RMSE of ANN, NF, ANN-FR, and NF-FR models is shown in Table 1. Error plots of four datasets (Titanic, Mammographic, Breast Cancer, and Wine) with four models (ANN, NF, ANN-FR, and NF-FR) are shown in Figure 5. There are some configurationally variations in these four models as follows: ANN is a simple model in which no fuzzification and no PCA are used for analysis. The ANN-FR is a no fuzzification with PCA-based model, NF is a fuzzification with the non-PCA based model, and similarly, NF-FR is a fuzzification with the PCA-based model.

The results presented here are exclusively based on the experiments that are observed. Table 2 describes the comparison of classification accuracy of ANN, NF, ANN-FR, and NF-FR models for worst, average, and best cases. Every model is executed ten times with random weights, and observations are recorded. Based on the ten times execution history of all the models, the worst, average, and best case classification accuracies are obtained and shown in Table 2. Apart from classification accuracy, some additional measures such as precision, recall, and f-measure are also considered here to measure the performance of ANN, NF, ANN-FR, and NF-FR models. Table 3 describes the comparison of precision, recall, and f-measure of ANN, NF, ANN-FR, and NF-FR models.

In Tables 2 and 3, classification accuracy, precision, recall, and f-measure (worst case, average case, and best case) of four models such as ANN, NF, ANN-FR, and NF-FR are presented. There are very few results in Tables 2 and 3 indicating the result of the proposed hybrid method (NF-FR) is lower than other techniques. Generally, this happens in any machine learning model. A single machine learning technique may not be suitable for all the benchmark datasets or problems (as per the principle of “no-free-lunch theorem”). Hence, in order to draw a generalized performance measure, we have conducted various statistical tests such as ANOVA test, Tukey and Dunnett test, Friedman test, and Holm procedure. To validate the performance of the proposed model with other models, several statistical analyses such as ANOVA test, Tukey and Dunnett test, Friedman test, and Holm procedure are made in the next section. For example, in Friedman test, the average rank of four models such as ANN, ANN-FR, NF, and NF-FR is computed based on assigned rank (in Table 4), which is represented in equation (17). This average rank of four models: ANN, ANN-FR, NF, and NF-FR, can be computed and assigned as $\{R4=4, R3=3, R2=1.9, R1=1.1\}$, respectively. Based on these ranks, null-hypothesis has been tested. The results of the aforesaid statistical analysis also show that the overall performance of the proposed NF-FR model is statically significantly different and better than other models. It means that our proposed model may not be suitable for few cases but works well for many datasets.

<table>
<thead>
<tr>
<th>Datasets/models</th>
<th>ANN</th>
<th>NF</th>
<th>ANN-FR</th>
<th>NF-FR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>0.18721</td>
<td>0.10258</td>
<td>0.15471</td>
<td>0.11721</td>
</tr>
<tr>
<td>Mammographic</td>
<td>0.30163</td>
<td>0.19111</td>
<td>0.26163</td>
<td>0.18031</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>0.39734</td>
<td>0.17052</td>
<td>0.21637</td>
<td>0.02546</td>
</tr>
<tr>
<td>Pima Indian</td>
<td>0.41819</td>
<td>0.29887</td>
<td>0.38248</td>
<td>0.30353</td>
</tr>
<tr>
<td>Hayes-Roth</td>
<td>0.32242</td>
<td>0.14770</td>
<td>0.35525</td>
<td>0.16222</td>
</tr>
<tr>
<td>Thyroid</td>
<td>0.15335</td>
<td>0.05315</td>
<td>0.14085</td>
<td>0.08838</td>
</tr>
<tr>
<td>Titanic</td>
<td>0.19373</td>
<td>0.28599</td>
<td>0.07685</td>
<td>0.34237</td>
</tr>
<tr>
<td>Wine</td>
<td>0.05249</td>
<td>0.03854</td>
<td>0.08942</td>
<td>0.10774</td>
</tr>
<tr>
<td>Haberman</td>
<td>0.25329</td>
<td>0.33007</td>
<td>0.35470</td>
<td>0.35749</td>
</tr>
<tr>
<td>Blood Transfusion Service</td>
<td>0.31176</td>
<td>0.31284</td>
<td>0.13597</td>
<td>0.29339</td>
</tr>
</tbody>
</table>
On the similar way, we have tested under ANOVA, Tukey, and Dunnett tests, and the performance of our proposed model was found to be significantly better as compared to other models.

In this experiment, few parameters were considered in the design process of the models. The fuzzy expansion, number of input neurons, and number of output neurons for ten datasets are presented in Table 5. The number of hidden layer used is

<table>
<thead>
<tr>
<th>Datasets/models</th>
<th>ANN</th>
<th>NF</th>
<th>ANN-FR (studied)</th>
<th>NF-FR (proposed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>WC</td>
<td>AC</td>
<td>BC</td>
<td>WC</td>
</tr>
<tr>
<td></td>
<td>85.71</td>
<td>91.56</td>
<td>94.44</td>
<td>90.00</td>
</tr>
<tr>
<td>Mammographic Mass</td>
<td>73.91</td>
<td>79.21</td>
<td>84.62</td>
<td>79.99</td>
</tr>
<tr>
<td>Breast Cancer Wisconsin</td>
<td>85.42</td>
<td>91.07</td>
<td>95.00</td>
<td>91.18</td>
</tr>
<tr>
<td>Pima Indian diabetes</td>
<td>65.52</td>
<td>72.96</td>
<td>78.41</td>
<td>71.76</td>
</tr>
<tr>
<td>Hayes-Roth</td>
<td>56.25</td>
<td>66.34</td>
<td>76.00</td>
<td>60.00</td>
</tr>
<tr>
<td>Thyroid</td>
<td>83.72</td>
<td>86.76</td>
<td>92.85</td>
<td>86.04</td>
</tr>
<tr>
<td>Titanic</td>
<td>75.13</td>
<td>77.64</td>
<td>81.87</td>
<td>76.00</td>
</tr>
<tr>
<td>Wine</td>
<td>85.29</td>
<td>91.24</td>
<td>94.12</td>
<td>88.88</td>
</tr>
<tr>
<td>Haberman</td>
<td>70.21</td>
<td>75.42</td>
<td>78.79</td>
<td>75.00</td>
</tr>
<tr>
<td>Blood Transfusion Service Center</td>
<td>73.37</td>
<td>75.86</td>
<td>80.00</td>
<td>73.61</td>
</tr>
</tbody>
</table>

WC: worst case; AC: average case; BC: best case.

Figure 5: Error plots of four datasets: (a) Titanic; (b) Mammographic; (c) Breast Cancer; (d) Wine.

Table 2: Comparison of classification accuracy of ANN, NF, ANN-FR, and NF-FR.
Table 3: Comparison of different performance parameters (precision, recall, and f-measure) of ANN, NF, ANN-FR, and NF-FR models.

<table>
<thead>
<tr>
<th>Datasets/Models</th>
<th>ANN</th>
<th>NF</th>
<th>ANN-FR</th>
<th>NF-FR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>0.918</td>
<td>0.910</td>
<td>0.911</td>
<td>0.926</td>
</tr>
<tr>
<td>Mammographic Mass</td>
<td>0.733</td>
<td>0.523</td>
<td>0.610</td>
<td>0.846</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>0.972</td>
<td>0.359</td>
<td>0.501</td>
<td>0.914</td>
</tr>
<tr>
<td>Pima Indian</td>
<td>0.393</td>
<td>0.181</td>
<td>0.246</td>
<td>0.903</td>
</tr>
<tr>
<td>Hayes-Roth</td>
<td>0.683</td>
<td>0.706</td>
<td>0.673</td>
<td>0.781</td>
</tr>
<tr>
<td>Thyroid</td>
<td>0.953</td>
<td>0.751</td>
<td>0.803</td>
<td>0.915</td>
</tr>
<tr>
<td>Titanic</td>
<td>0.762</td>
<td>0.389</td>
<td>0.512</td>
<td>0.781</td>
</tr>
<tr>
<td>Wine</td>
<td>0.9155</td>
<td>0.9289</td>
<td>0.9186</td>
<td>0.9361</td>
</tr>
<tr>
<td>Haberman</td>
<td>0.6233</td>
<td>0.0807</td>
<td>0.1411</td>
<td>0.635</td>
</tr>
<tr>
<td>Blood Transfusion</td>
<td>0.5015</td>
<td>0.0839</td>
<td>0.0574</td>
<td>0.7283</td>
</tr>
<tr>
<td>P</td>
<td>0.975</td>
<td>0.928</td>
<td>0.932</td>
<td>0.944</td>
</tr>
<tr>
<td>R</td>
<td>0.944</td>
<td>0.910</td>
<td>0.932</td>
<td>0.944</td>
</tr>
<tr>
<td>FM</td>
<td>0.863</td>
<td>0.840</td>
<td>0.863</td>
<td>0.863</td>
</tr>
<tr>
<td>P</td>
<td>0.953</td>
<td>0.928</td>
<td>0.953</td>
<td>0.944</td>
</tr>
<tr>
<td>R</td>
<td>0.944</td>
<td>0.910</td>
<td>0.953</td>
<td>0.953</td>
</tr>
<tr>
<td>FM</td>
<td>0.863</td>
<td>0.840</td>
<td>0.863</td>
<td>0.863</td>
</tr>
</tbody>
</table>

Table 4: Friedman’s rank to all the models based on avg. classification accuracy.

<table>
<thead>
<tr>
<th>Datasets/models</th>
<th>ANN</th>
<th>NF</th>
<th>ANN-FR</th>
<th>NF-FR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>91.56 (4)</td>
<td>92.63 (3)</td>
<td>94.02 (2)</td>
<td>95.15 (1)</td>
</tr>
<tr>
<td>Mammographic Mass</td>
<td>79.21 (4)</td>
<td>84.19 (3)</td>
<td>84.50 (2)</td>
<td>85.31 (1)</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>91.07 (4)</td>
<td>93.98 (3)</td>
<td>95.39 (1)</td>
<td>95.04 (2)</td>
</tr>
<tr>
<td>Pima Indian Diabetes</td>
<td>72.96 (4)</td>
<td>78.78 (3)</td>
<td>81.66 (2)</td>
<td>82.55 (1)</td>
</tr>
<tr>
<td>Hayes-Roth</td>
<td>66.34 (4)</td>
<td>73.85 (3)</td>
<td>75.20 (2)</td>
<td>77.15 (1)</td>
</tr>
<tr>
<td>Thyroid</td>
<td>86.76 (4)</td>
<td>92.01 (3)</td>
<td>92.53 (2)</td>
<td>95.60 (1)</td>
</tr>
<tr>
<td>Titanic</td>
<td>77.64 (4)</td>
<td>79.25 (3)</td>
<td>80.54 (2)</td>
<td>81.47 (1)</td>
</tr>
<tr>
<td>Wine</td>
<td>91.24 (4)</td>
<td>93.11 (3)</td>
<td>93.94 (2)</td>
<td>95.59 (1)</td>
</tr>
<tr>
<td>Haberman</td>
<td>75.42 (4)</td>
<td>78.05 (3)</td>
<td>80.29 (2)</td>
<td>81.02 (1)</td>
</tr>
<tr>
<td>Blood Transfusion Service</td>
<td>75.86 (4)</td>
<td>76.79 (3)</td>
<td>80.14 (2)</td>
<td>80.22 (1)</td>
</tr>
<tr>
<td>Friedman’s Rank</td>
<td>4</td>
<td>3</td>
<td>1.9</td>
<td>1.1</td>
</tr>
</tbody>
</table>

5. Statistical Analysis of Results

Statistical analysis is a well-known method to analyze the performance of various models with several datasets. Generally, different statistical tools are used to analyze the nature of data and algorithms. In this section, statistical analysis [59] along with the comparison of all the models over multiple datasets is presented. Several statistical tests such as analysis of variance (ANOVA) test [60], Tukey test [61], Dunnett test [62], Friedman test [63, 64], and post hoc test [65, 66] have been used to prove the proposed classification algorithm is more efficient than other existing classification algorithms. It will find out the best classification algorithm among a set of classification algorithms based on certain measuring parameters.

5.1. ANOVA Test. ANOVA [60] is a parametric statistical technique used to be compared among the different models. It generally compares the mean and relative variance in the performance of different models. This method is suitable when more than two models are compared with different datasets. ANOVA uses a null hypothesis and an alternative hypothesis. The null hypothesis is valid only when the performances of all the models are equal or there is no significant difference among these models. Alternatively, the alternative hypothesis is valid only when any one of the models is different from the rest of the models. The one-way ANOVA test has been carried out in SPSS (Version 16.0) with 95% confident interval, and the result has been presented in Tables 6 and 7.

5.2. Tukey and Dunnett Tests. To reject the null hypothesis, Tukey test [61] and Dunnett test [62] have been conducted. In the Tukey test, the comparison of the performance of every model is compared with every other model, but the Dunnett test compares the performance of every model with the proposed model. The control group for this test id NF-FR is compared with different models such as ANN, ANN-FR, and NF. The result of Tukey and
Dunnett tests is presented in Table 8. A homogeneous group of models based on their level of significance is presented in Table 9.

5.3. Friedman Test. Friedman test [63, 64] is a nonparametric statistical technique developed by M. Friedman. It is used to find out the differences among different models by

<table>
<thead>
<tr>
<th>Models</th>
<th>N</th>
<th>Mean</th>
<th>Std. deviation</th>
<th>Std. error</th>
<th>95% confidence interval for mean</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>Std. deviation</td>
<td>Std. error</td>
<td>95% confidence interval for mean</td>
<td>Minimum</td>
<td>Maximum</td>
<td></td>
</tr>
<tr>
<td>(Combined)</td>
<td></td>
<td>212.449</td>
<td>3</td>
<td>70.816</td>
<td>1.125</td>
<td>66.34</td>
<td>91.56</td>
<td>0.352</td>
</tr>
<tr>
<td>Linear term</td>
<td></td>
<td>197.408</td>
<td>1</td>
<td>197.408</td>
<td>3.136</td>
<td>73.85</td>
<td>93.98</td>
<td>0.085</td>
</tr>
<tr>
<td>Contrast</td>
<td></td>
<td>15.041</td>
<td>2</td>
<td>7.521</td>
<td>0.119</td>
<td>75.2</td>
<td>95.39</td>
<td>0.888</td>
</tr>
<tr>
<td>Deviation</td>
<td></td>
<td>14.019</td>
<td>1</td>
<td>14.019</td>
<td>0.223</td>
<td>77.15</td>
<td>95.6</td>
<td>0.64</td>
</tr>
<tr>
<td>Quadratic term</td>
<td></td>
<td>1.022</td>
<td>1</td>
<td>1.022</td>
<td>0.016</td>
<td>6.899</td>
<td>95.6</td>
<td>0.899</td>
</tr>
<tr>
<td>Within groups</td>
<td></td>
<td>2265.856</td>
<td>36</td>
<td>62.94</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>2478.305</td>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Means for groups in homogeneous subsets are displayed. *Dunnett t-tests treat one group as a control and compare all other groups against it.

Dunnett tests is presented in Table 8. A homogeneous group of models based on their level of significance is presented in Table 9.
assigning certain mean ranks to the resultant values represented in Table 4. The average rank of algorithms can be computed by using the following equation, where $r^j_i$ is the rank of the $j^{th}$ model on the $i^{th}$ dataset and $N$ is the number of dataset:

$$R_j = \frac{1}{N} \sum_i r^j_i.$$  

(17)

The average rank of four models: ANN, ANN-FR, NF, and NF-FR is computed based on assigned rank, which is represented in equation (17). This average rank of four models: ANN, ANN-FR, NF, and NF-FR can be represented as [$R4 = 4$, $R3 = 3$, $R2 = 1.9$, $R1 = 1.1$], respectively. The value of $X^2$ is computed from average rank $R_i$ and can be realized by the following equation (18), where $N$ is the number of datasets and $m$ is the number of models. In this case, the value of $X^2$ is 48.2:

$$X^2_F = \frac{12N}{m(m-1)} \left[ \sum_j R^2_j - \frac{m(m+1)^2}{4} \right].$$  

(18)

The Friedman statistics $F_F$ is measured by using $X^2_F$ with $(m - 1)$ degree of freedom and can be realized in equation (19). The critical value [64] can be obtained from Friedman statistics $F_F$ with $(m - 1)$ and $(m - 1) \times (N - 1)$ degree of freedom. In this approach, four numbers of models $m$ and ten numbers of datasets $N$ are used. In this case, the value of Friedman statistics $F_F$ is 241 with four numbers of models and ten numbers of datasets:

$$F_F = \frac{(N - 1)X^2_F}{(m - 1) - X^2_F}. $$  

(19)

The performances of models are different, if the corresponding average rank differs by at least the critical difference. The critical value is computed as 4.6 with $(4 - 1 = 3)$ and $(4 - 1 = 3) \times (10 - 1 = 9)$ degree of freedom and significance level $\alpha = 0.01$. The density plot is obtained and shown in Figure 6 with a degree of freedom (3, 27). It is noted that the null hypothesis is rejected as the critical value (4.6) is less than the Friedman statistics ($F_F = 241$). The post hoc test experiment has been conducted by using the Holm procedure, after the rejection of the null hypothesis.

5.4. Holm Procedure. This Holm procedure [65–67] computes the performance of every individual model with the rest of the models by using the $z$-value and $p$ value. The $z$-value is computed by using equation (20), and the $p$ value is computed from the $z$-value and the normal distribution table accordingly:

$$z = \frac{(R_i - R_j)}{\sqrt{m(m+1)/6N}}.$$  

(20)

where $m$ is the number of models, $z$ is the $z$-score value, and $N$ is the number of datasets. The average rank of the $i^{th}$ and $j^{th}$ models is denoted by $R_i$ and $R_j$, respectively. All the three models are compared with the proposed model based on $z$-value, $p$ value, and $\alpha/(m - i)$, and the result is represented in Table 10. Here, we noticed that, in almost all the cases, the $p$ values are less than $\alpha/(m - i)$ by using the Holm test. Hence, it is concluded that the null hypothesis is rejected. Thus, it indicates that the proposed model NF-FR is statistically significantly different and better than other classification models.

6. Conclusion and Future Scope

In this paper, the proposed NF-FR model is demonstrated successfully for solving data classification problems in data mining. Initially, this model uses the fuzzification process for expansion of the input features class-wise belongingness of the features to various classes which provide to handle imprecise and uncertainty problems. Due to the expansion
of features, the model structure becomes massively parallel and also found that all the features may not contribute significantly to the model. In the next step, PCA is applied to reduce the dimension of the expanded features by selecting the best suitable relevant and nonredundant features. As a result, the learning time of the proposed model was also reduced with the selected relevant features. However, a particular II-type membership function considered for the fuzzification process may not always be suitable for the entire datasets. In such cases, the selection of suitable membership functions may be taken into consideration for data analysis. As per the experimental analysis, the proposed method is able to classify the datasets with superior classification performance as compared to ANN, NF, and ANN-FR models. After statistical analysis, it is found that the proposed NF-FR model is valid and efficient as compared to FR models. After statistical analysis, it is found that the proposed NF-FR model is valid and efficient as compared to ANN, NF, and ANN-FR models. In the future, this proposed model can be used in various real-life problems such as gene expression classification, document classification, and satellite image classification.

Data Availability

The data used to support the findings of this study are included in Section 4 within the article. We have used ten benchmark datasets from the UCI machine learning repository. The detail descriptions about all these datasets can be found at “http://archive.ics.uci.edu/ml/” and “http://keel.es/.”

Disclosure

This study was not funded by any research organization.

Conflicts of Interest

All authors declare that there are no conflicts of interest.

Acknowledgments

This research work was supported by the Science and Engineering Research Board (SERB), Department of Science and Technology (DST), New Delhi, Govt. of India, under the research project grant Sanction Order No. EEQ/2017/000355.

References


