Neural Networks 48 (2013) 91–108

Contents lists available at ScienceDirect

Neural Networks

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

Fuzzy rough sets, and a granular neural network for unsupervised feature selection

Avatharam Ganivada [a,*], Shubhra Sankar Ray [a,b], Sankar K. Pal [a]

[a] Center for Soft Computing Research, Indian Statistical Institute, Kolkata, India
[b] Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India

ARTICLE INFO

Article history:
Received 29 October 2012
Received in revised form 4 May 2013
Accepted 30 July 2013

Keywords:
Granular computing
Rough fuzzy computing
Rule based layered network
Feature evaluation

ABSTRACT

A granular neural network for identifying salient features of data, based on the concepts of fuzzy set and a newly defined fuzzy rough set, is proposed. The formation of the network mainly involves an input vector, initial connection weights and a target value. Each feature of the data is normalized between 0 and 1 and used to develop granulation structures by a user defined α-value. The input vector and the target value of the network are defined using granulation structures, based on the concept of fuzzy sets. The same granulation structures are also presented to a decision system. The decision system helps in extracting the domain knowledge about data in the form of dependency factors, using the notion of new fuzzy rough set. These dependency factors are assigned as the initial connection weights of the proposed network. It is then trained using minimization of a novel feature evaluation index in an unsupervised manner. The effectiveness of the proposed network, in evaluating selected features, is demonstrated on several real-life datasets. The results of FRGNN are found to be statistically more significant than related methods in 28 instances of 40 instances, i.e., 70% of instances, using the paired t-test.

1. Introduction

Granulation, a computational paradigm, is one of the important steps in the human cognition process. In the granulation process, granules are extracted from a dataset, where, data points having similar characteristics are collected within each granule. Granular Computing (GrC) is a very convenient approach for handling incomplete and uncertain information involved in the data. Here, computation and operations are performed on the information granules. The concept of information granulation is described by Zadeh (1997), where, perceptions are considered to have fuzzy boundaries and the values of attributes they can take are granulated (a clump of indistinguishable data points/patterns). Rough set theory is also used to deal with uncertainty arising between the data points or a set of data points. The uncertainty arising between the set of data points is handled by defining the crisp equivalence granules based on an indiscernibility relation. Computation of granules using fuzzy sets, rough sets and neural networks is given by Dick and Kandel (2001), Herbert and Yao (2009) and Pawlak and Skowron (2007).

Dick and Kandel (2001), Szczuka (2001) and Zhang, Jin, and Tang (2008) have shown that different granular neural networks for classification can be developed by incorporating the information granules, generated by the concept of either fuzzy sets or rough sets, into artificial neural networks. The concepts of fuzzy rough set proposed by Cornelis, Jensen, Hurtado, and Slezak (2010) and Dubois and Prade (1990), based on the fuzzy tolerance relation. Here, the uncertainty arising between the overlapping boundaries of attribute values of the data points is dealt with by assigning a membership value to every data point using a membership function, belonging to the lower and upper approximations, based on the fuzzy similarity relation or fuzzy tolerance relation.

Feature selection is essential in analyzing datasets, when the data has a large number of features and high dimensions. Some of the features may be redundant/irrelevant and they can degrade the performance & increase the computational complexity of the networks. Selection of salient features, by discarding irrelevant features, from large datasets is an important task in pattern recognition. These selected features can increase the performance of the networks and can also decrease the computational complexity.

Different algorithms for feature selection have been given by several researchers. Some of them like, branch and bound, sequential forward & backward search and sequential floating point search methods can be used for selecting an optimal set of features. An efficient algorithm for feature selection on the basis of the feature similarity measure, called maximum information compression index, is described by Mitra, Murthy, and Pal (2002).

Investigations for feature selection are also performed in the framework of neural networks. Verikas and Bacauskiene (2002)

* Corresponding author. Tel.: +91 8013902064.
E-mail address: avatharg@yahoo.co.in (A. Ganivada).

0893-6080/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.neunet.2013.07.008

proposed a neural network for identifying salient features, based on multi-layer feed forward networks, where, the network is trained with an augmented cross entropy error function. A neuro-fuzzy network for feature selection is developed by Pal, De, and Basak (2000). Here, a membership function, defined by incorporating the weighted distance between two patterns and minimizing the feature evaluation index using unsupervised training, is used in the formation of the neuro-fuzzy network. Recently, Ghosh, Shankar, and Meher (2009) described a neuro-fuzzy model for classification, based on multi-layer perceptron. Here, the input vector of the proposed model is defined in terms of membership values, defined by a membership function. Ganivada, Dutta, and Pal (2011); Ganivada, Ray, and Pal (2012) have proposed fuzzy rough granular neural networks for classification and clustering, where the fuzzy rules representing the information granules generated by fuzzy rough sets are incorporated into artificial neural networks.

In this article, we propose a fuzzy rough set, where, new notions of lower and upper approximations of a set are defined. A three-layered fuzzy rough granular network (FRGNN) is then introduced for feature selection, based on the concept of fuzzy sets and the proposed fuzzy rough set. Initially, a pair wise similarity matrix is generated by finding the similarity between all possible pairs of patterns, based on fuzzy logical connectives. An α-cut is applied on the similarity matrix, in order to develop granulation structures. The mean for every granulation structure is calculated and a mean matrix, using the means of all the granulation structures, is constructed. A row in the matrix represents a mean and a column represents a feature. Each column vector is then used as an input vector, while the average of each column, corresponding to a feature, is defined as a target value of FRGNN. The granulation structures are then labeled with integers, representing the classes, and are presented to a decision table. The decision table helps in extracting the domain knowledge about data, in terms of dependency factors, using the concept of the proposed fuzzy rough set. The dependency factors are determined as the initial connection weights of FRGNN. In a part of the investigation, we also develop a new feature evaluation index and the FRGNN is trained, using the minimization of the proposed index. During training, the connection weights between the nodes in the hidden and output layers are updated in an unsupervised manner; thereby optimal weights between the nodes of the hidden and output layers of the proposed network are determined. The optimal weights provide the importance of the individual features of the data for selection. Note that the proposed feature evaluation index is based on the network error value, which is based on the mean value of the feature and not dependent on the class information.

This article is organized as follows: First, in Section 2, we examine the preliminaries on granulation and approximation of set using the concept of rough sets, fuzzy sets and fuzzy rough sets. New notions of lower and upper approximations of a set, representing a fuzzy rough set, are defined in Section 3. The diagrammatic representation of FRGNN and the formation of the proposed fuzzy rough granular network (FRGNN), involving an input vector, initial connection weights and a target vector are described using the concept of fuzzy sets and fuzzy rough sets in Section 4. Different well-known existing methods for feature selection and feature evaluation, which are used for testing the performance of FRGNN, are discussed in Section 6. Experimental results of the proposed FRGNN, by comparing with different existing algorithms, and discussion are provided in Section 6. Finally, conclusions are provided in Section 7.

2. Preliminaries on rough sets, fuzzy sets and fuzzy rough sets: Granulations and approximations

In this section, we outline the preliminaries on rough sets, fuzzy sets and fuzzy rough sets. In rough sets, the granulation structure is typically a partition of a universe. Pawlak and Skowron (2007) described the basics on granulation and approximation of a set using the concept of rough sets.

A data point, x, in a fuzzy set $A \subseteq U$, where U is a universe, may be assigned a membership value, denoted by $\mu_A(x)$, by a membership function. The fuzzy set is defined as

$$A = \{(\mu_A(x), x)\}, \quad x \in U,\ \mu_A(x) \in [0, 1]. \tag{1}$$

In Eq. (1), the membership value $\mu_A(x)$ and $x \in \mathbf{R}^n$, within a range $[0, 1]$, can be defined by the π-membership function,

$$\pi(x, C, \lambda) = \begin{cases} 2\left(1 - \dfrac{\| x - C \|_2}{\lambda}\right)^2, & \text{for } \dfrac{\lambda}{2} \leq \| x - C \|_2 \leq \lambda, \\ 1 - 2\left(\dfrac{\| x - C \|_2}{\lambda}\right)^2, & \text{for } 0 \leq \| x - C \|_2 \leq \dfrac{\lambda}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where, $\lambda > 0$ is a scaling factor (radius) of the π function with C as a central point, and $\| \cdot \|_2$ denotes the Euclidean norm. Ganivada et al. (2011) shown the choice of the parameters, scaling factor $\lambda > 0$ and center C, of the π-membership function.

A fuzzy relation $R(x, y)$ between two patterns x and $y \in A \subseteq U$ is a $A \times A \rightarrow [0, 1]$ mapping such that $R(x, y)$ is a fuzzy set in A. For each $y \in U$, the R-foreset of y is a fuzzy set R_y, defined by Cornelis et al. (2010) as $R_y(x) = R(x, y)$, for all $x \in U$.

In fuzzy rough sets, the similarity between any two patterns in a set A is modeled by a fuzzy relation R, which is defined in Ganivada et al. (2011) as

$R(x, x) = 1$ (reflexive),
$R(x, y) = R(y, x)$ (symmetry), and
$T(R(x, y)R(y, z)) \leq R(x, z)$ (T-transitivity),

for all x, y and z in A. Given a t-norm (or a T-norm), the symmetry and the T-transitivity properties need not be satisfied by the fuzzy relation R. In general, the fuzzy reflexive relation R is called a fuzzy T-equivalence relation or fuzzy tolerance relation and can be treated as a fuzzy equivalence granule. The set $A \subset U$ is approximated in the fuzzy tolerance space by giving a membership value to a pattern in A for belonging to lower and upper approximations of the set. The concept of the fuzzy rough set, denoted by $(\mu_{R(A)}, \mu_{\bar{R}(A)})$, is defined by Dubois and Prade (1992) as

$$\mu_{\underline{R}(A)}(x) = \inf_{y \in U} \max\{1 - R(x, y), \mu_A(x)\}, \tag{3}$$

$$\mu_{\bar{R}(A)}(x) = \sup_{y \in U} \min\{R(x, y), \mu_A(x)\}, \tag{4}$$

where, $\mu_{\underline{R}(A)}(x)$ and $\mu_{\bar{R}(A)}(x)$ represent the membership degrees of a pattern x for belonging to the lower and upper approximations of the set A.

Sen and Pal (2009) proposed a new notion to the fuzzy rough set, $(\underline{R}(A), \overline{R}(A))$, when R is a fuzzy tolerance relation and the set A is a crisp or a fuzzy set. Here, $\underline{R}(A)$ and $\overline{R}(A)$ represent the lower and upper approximations of a set A, and are defined as

$$\underline{R}A = \{(x, \underline{M}(x)) | x \in U\}, \tag{5}$$

$$\overline{R}A = \{(x, \overline{M}(x)) | x \in U\}, \tag{6}$$

where,

$$\underline{M}(x) = \sum_{Y \in U/R} m_Y(x) \times \inf_{\varphi \in U} \max(1 - m_Y(\varphi), \mu_A(\varphi)), \tag{7}$$

$$\overline{M}(x) = \sum_{Y \in U/R} m_Y(x) \times \sup_{\varphi \in U} \min(m_Y(\varphi), \mu_A(\varphi)). \tag{8}$$

Here, $Y \in U/R$ is a granule, $m_Y(x)$ signifies a membership value of a pattern belonging to $Y \in U/R$, and $\mu_A(\varphi)$ is the membership function associated with A. The symbols, $\sum$ and $\times$, in Eqs. (7) and (8) denote fuzzy union and intersection operations, respectively.

Some of the fuzzy logical counterparts of connectives involved in the generalization of lower and upper approximations are needed for the approximation of a set. An operator T, mapping from $[0, 1]^2$ to $[0, 1]$, satisfies $T(1, x) = x$, for all $x \in [0, 1]$. Let T_M, T_P and T_L denote t-norms, and these are defined as

$$T_M(x, y) = \min(x, y) \quad \text{(minimum operator)}, \tag{9}$$

$$T_P(x, y) = x * y, \quad \text{and} \tag{10}$$

$$T_L(x, y) = \max(0, x + y - 1) \quad \text{(Lukasiewicz } t\text{-norm)}, \tag{11}$$

for all x and $y \in [0, 1]$. On the other hand, a mapping $I : [0, 1] \times [0, 1] \to [0, 1]$ satisfies $I(0, 0) = 1, I(1, x) = x$, for all $x \in [0, 1]$, where, I is an implication operator. For all x and $y \in [0, 1]$, the implication operators I_{KD}, I_L and I_{KDL} are defined as

$$I_{KD}(x, y) = \max(1 - x, y) \quad \text{(Kleene–Dienes implicator)}, \tag{12}$$

$$I_L(x, y) = \min(1, 1 - x + y) \quad \text{(Lukasiewicz implicator)}, \tag{13}$$

$$I_{KDL}(x, y) = 1 - x + x * y$$

$$\text{(Kleene–Dienes–Lukasiewicz implicator)}. \tag{14}$$

A new notion to fuzzy rough set, denoted by $(R \downarrow A, R \uparrow A)$, is proposed by Radzikowska and Kerre (2002) in fuzzy tolerance approximation space. Here, $R \downarrow A$ and $R \uparrow A$ denote the lower and upper approximations, respectively, of a set A, which are defined with a similarity relation R under the fuzzy logic connectives, Eqs. (11) and (13), as

$$(R \downarrow A)(x) = \inf_{y \in U} I(R(x, y), A(x)), \quad \text{and} \tag{15}$$

$$(R \uparrow A)(x) = \sup_{y \in U} T(R(x, y), A(x)), \tag{16}$$

for all x in U.

3. Proposed fuzzy rough set: granulations and approximations

In this section, a new fuzzy rough set is proposed by defining new notions to lower and upper approximations of a set in the tolerance approximation space. Granulation and approximation of a set is proposed using the concept of fuzzy rough set.

For a non empty universe U, a fuzzy reflexive relation or fuzzy tolerance relation R on U, is considered to express the approximation equality between two objects in a set $A \subseteq U$. Let $S = (U, \mathcal{A} \cup \{d\})$ denote a decision system. Here, $\mathcal{A}$ represents the attributes, say $\{a_1, a_2, \ldots, a_n\}$. The decision attribute d is defined as $X_k, k = 1, 2, \ldots, c$, where, c denotes the number of decision classes. Classification of a pattern x in U can be performed depending on the decision classes. A fuzzy reflexive relation R_a, between any two patterns x and y in U, with respect to a quantitative attribute $a \in \mathcal{A}$, is defined by Ganivada et al. (2012).

3.1. Defining decision classes using fuzzy sets

When a qualitative attribute $a \in \{d\}$, then the relation R_d between any two patterns x and $y \in U$, with respect to the attribute 'a' (fuzzy decision classes), is defined by Ganivada et al. (2012) as follows: Assume that a decision system S contains c-classes corresponding to a decision attribute. Let O_{kj} and $V_{kj}, j = 1, 2, \ldots, n$, denote mean and standard deviation, respectively, of the patterns belonging to the kth class. The weighted distance Z_{ik} of a pattern $\overrightarrow{x}_i, i = 1, 2, \ldots, s$, where, s is the total number of patterns from the kth decision class, is defined by Pal and Dutta Majumder (1977) as

$$Z_{ik} = \sqrt{\sum_{j=1}^{n} \left[\frac{x_{ij} - O_{kj}}{V_{kj}}\right]^2}, \quad \text{for } k = 1, 2, \ldots, c, \tag{17}$$

where, x_{ij} represents jth component of the ith pattern. Note that, when the labeled values of all the patterns in a class are the same, then the standard deviation will be zero. In that case, we consider $V_{kj} = 0.000001$ (for the sake of computation) so that, the weighting distance Z_{ik} becomes high and the membership value of the ith pattern, belonging to the kth class along that feature, becomes low. The membership value of the ith pattern in the kth class is defined by Pal and Dutta Majumder (1977) as

$$\mu_k(\overrightarrow{x}_i) = \frac{1}{1 + \left(\frac{Z_{ik}}{f_d}\right)^{f_e}}, \tag{18}$$

where, f_e and f_d are fuzzifiers. It may be noted that, the decision attribute is quantitative for the patterns with different membership values corresponding to a decision attribute. It can be defined in two different ways, namely,

(1) the membership values of all the patterns in the kth class to its own class is defined as

$$D_{kk} = \mu_k(\overrightarrow{x}_i), \quad \text{if } k = l, \tag{19}$$

where, $\mu_k(\overrightarrow{x}_i)$ represents the membership value of the ith pattern to the kth class, and

(2) the membership values of all patterns in the kth class to other classes is defined as

$$D_{kl} = 1, \quad \text{if } k \neq l, \tag{20}$$

where, k and $l = 1, 2, \ldots, c$. For any two patterns x and $y \in U$, with respect to an attribute $a \in \{d\}$, the fuzzy decision classes are defined as

$$R_d(x, y) = \begin{cases} D_{kk}, & \text{if } a(x) = a(y), \\ D_{kl}, & \text{otherwise.} \end{cases} \tag{21}$$

Now, we define the lower approximation, denoted by $(R_a \downarrow R_d)$, and the upper approximation, denoted by $(R_a \uparrow R_d)$, of a set A, based on the fuzzy reflexive relation R_a and fuzzy decision value R_d between two objects in U.

3.2. Lower and upper approximations

The membership value of pattern x in $A \subseteq U$ for belonging to the lower approximation of a set A, based on a fuzzy reflexive relation R_a, and fuzzy logic connectives, (Eqs. (10) and (12)), is defined as

$$(R_a \downarrow R_d)(x) = \min\{\underline{\gamma}(x), \underline{\gamma}^c(x)\} \tag{22}$$

where,

$$\underline{\gamma}(x) = \inf_{y \in A}\{R_a(x, y) * R_d(x, y)\}, \tag{23}$$

$$\underline{\gamma}^c(x) = \inf_{y \in U - A}\{\max(1 - R_a(x, y), R_d(x, y))\}. \tag{24}$$

Here, $\underline{\gamma}(x)$ represents a weighted membership value of a pattern x computed by taking the product of $R_a(x, y)$ and $R_d(x, y)$. While $R_a(x, y)$ represents a membership value of the pattern appearing in the fuzzy reflexive relational matrix, $R_d(x, y)$ represents a membership value of the pattern appearing in the fuzzy decision class, when the patterns x and y belong to the same set A. The value of $\underline{\gamma}^c(x)$, lying between $[0, 1]$, is computed (using Eq. (12)), when

Table 1
Dataset.

U	a	b	c	d
1	−0.4	−0.3	−0.5	1
2	−0.4	0.2	−0.1	2
3	−0.3	−0.4	0.3	1
4	0.3	−0.3	0	2
5	0.2	−0.3	0	2
6	0.2	0	0	1

the patterns x and y belong to two different sets (not in the same set A).

The membership value of a pattern x in $A \subseteq U$ for belonging to the upper approximation of a set A, based on the fuzzy reflexive relation R_a, and fuzzy logic connectives, (see Eqs. (9) and (14)), is defined as

$$(R_a \uparrow R_d)(x) = \max\{\overline{\gamma}(x), \overline{\gamma}^c(x)\}, \tag{25}$$

where,

$$\overline{\gamma}(x) = \sup_{y \in A}\{1 - R_a(x, y) + (R_a(x, y) * R_d(x, y))\}, \quad \text{and} \tag{26}$$

$$\overline{\gamma}^c(x) = \sup_{y \in U - A}\{\min(R_a(x, y), R_d(x, y))\}, \tag{27}$$

for all x in U. Here, the term $\overline{\gamma}(x)$ represents a membership value, (computed by using Eq. (14)), when the two patterns x and y belong to a set A. The value of $\overline{\gamma}^c(x)$ lies between [0, 1] and is computed (using Eq. (9)), when two patterns belong to two different sets (not in the same set A).

For any $B \subseteq \mathcal{A}$ and for $x \in U$, the fuzzy positive region can be defined, depending on the B-indiscernibility relation R_B, as

$$POS_B(y) = \left(\bigcup_{x \in U} R_B \downarrow R_d x\right)(y), \tag{28}$$

for all y in U. The degree of dependency of γ, depending on the set of attributes $B \subseteq \mathcal{A}$, is defined as

$$\gamma_B = \frac{\sum_{x \in U} POS_B(x)}{|U|}, \tag{29}$$

where, $|\cdot|$ denotes the cardinality of a set U, and the value of γ is $0 \leq \gamma \leq 1$.

3.3. Example

Let us consider a dataset containing two classes given by Cornelis et al. (2010), shown in Table 1, as a typical example. Here, the data is normalized within 0 to 1. The membership values of the patterns, corresponding to every feature, for belonging to lower and upper approximations are based on the Eqs. (22) and (25), respectively. Fig. 1 shows a 3-dimensional scattered plots of features, defined in terms of membership values for belonging to lower and upper approximations, in F_1–F_2–F_3 space.

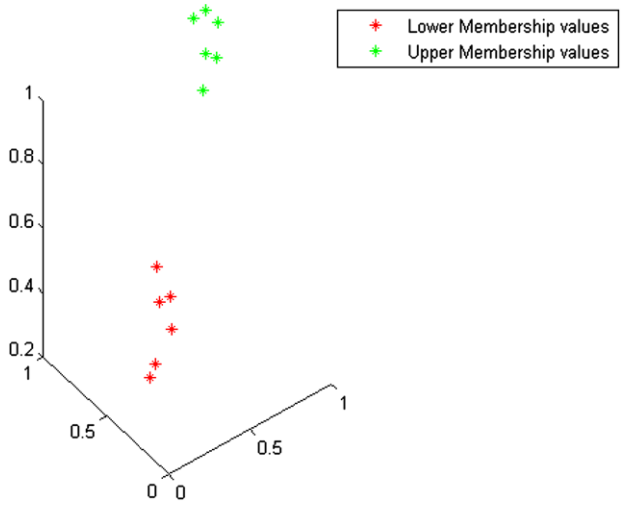

Fig. 1. 3-dimensional scattered features of the data, defined in terms membership values for belonging to lower and upper approximations, in F_1–F_2–F_3 space.

4. Proposed approach for the formation of a fuzzy rough granular neural network

In this section, we first provide a diagrammatic representation, in Fig. 2, of the proposed fuzzy rough granular neural network (FRGNN) for unsupervised feature selection. The main steps are divided into three blocks, shown with dotted lines. These are as follows:

(1) *Generate c-granulation structures:* The first block within dotted lines (Block 1) represents the procedure for determining the c-granulation structures, which includes data normalization, formation of a similarity matrix using fuzzy implication and t-norm, generation of p-granulation structures (groups) and selection of top c-granulation structures from them. These are explained in Sections 4.1 and 4.2.

(2) *Determine the input vector and target value of FRGNN:* The second block (Block 2) provides an approach for determining the input vector and target value of FRGNN, where, the input vector is represented by the mean values of the c-granulation structures, and the target value is defined by the average value of the mean values, corresponding to a feature. This approach is explained in Section 4.3.

(3) *Determine initial connection weights of FRGNN:* The procedure for determining the connection weights of FRGNN, including membership values of the patterns for belonging to the lower approximation and positive region of a set corresponding to the features, dependency values for the features with respect to the decision classes, and average values of the dependency values for the features (using the concepts of the proposed fuzzy rough set as explained in Section 3) is provided in the third block (Block 3) and is described in Section 4.4.1.

Now, the output of Blocks 1 and 3 are used for setting the number of nodes in the input layer (c) & hidden layer (n) and initializing the connection weights between the nodes in the input & hidden and hidden & output layers of FRGNN, respectively. Block 2 provides the input vector and target value for training FRGNN. The architecture of FRGNN is then formulated and is explained in the first paragraph of Section 4.4.

The mechanism involved in the forward propagation and minimization of the proposed feature evaluation index with respect to the connection weights between nodes of the hidden and output layers, which are used in training of FRGNN, is explained in Section 4.4.2.

4.1. Normalization of data

Let $\{x_{ij}\}, i = 1, 2, \ldots, s; j = 1, 2, \ldots, n$; be a set of n-dimensional data, where, s represents the total number of patterns in the data. The data is then normalized between 0 and 1, for every feature. For $j = 1$, the data is x_{i1}. The procedure for normalization of x_{i1} is defined as

$$x'_{i1} = \frac{x_{i1} - x_{\min_{i1}}}{x_{\max_{i1}} - x_{\min_{i1}}}, \tag{30}$$

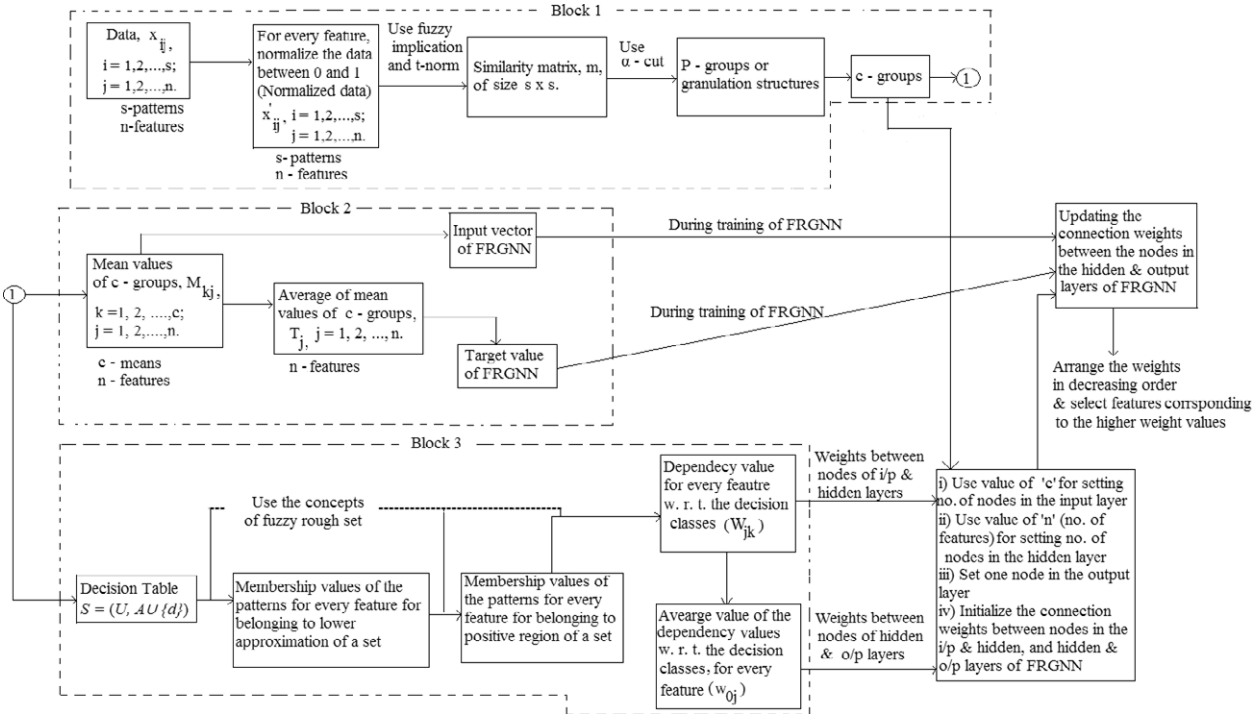

Fig. 2. Diagrammatic representation of the proposed FRGNN.

where, $x_{\min_{i1}}$ and $x_{\max_{i1}}$ are the minimum and maximum values of x_{i1}, respectively, along a feature $j = 1$. In a similar way, normalization is performed for $j = 2, \ldots, n$. The resultant normalized data is then used to determine granulation structures by an α-cut.

4.2. Determination of granulation structures based on α-cut

The data is partitioned into granulation structures by computing a pairwise similarity matrix using x'_{ij} (normalized data), and applying a user defined α-cut on it. As an example, the steps for computing the similarity m_{12}, between x'_{1j} and x'_{2j}, by applying the implication operator (Eq. (13)) and t-norm (Eq. (11)), are as follows:

1. $I_{1j} \leftarrow \min(1 - x'_{1j} + x'_{2j}, 1)$, $I_{2j} \leftarrow \min(1 - x'_{2j} + x'_{1j}, 1)$,
2. $T_j \leftarrow \max(I_{1j} + I_{2j} - 1, 0)$,
3. $m_{12} \leftarrow \min\{T_j\}$.

In a similar manner, we find the similarity between all possible pairs and construct a similarity matrix, m, of size $s \times s$, where, s is the total number of patterns. By using the similarity matrix m, we generate p number of granulation structures/groups, based on an α-cut, chosen between 0 and 1. The steps are as follows:

S1: Let i and i' denote the number of rows and columns of the similarity matrix m, where, i and $i' = 1, 2, \ldots, s$.
S2: Choose an α value between 0 and 1.
S3: Apply the α-value or a threshold on the first row ($i = 1$) of the similarity matrix m, and find how many similarity values in that row are greater than the α-value.
S4: Find the positions of the columns, corresponding to the similarity values greater than the α-value, in the first row of the similarity matrix.
S5: Select the patterns, corresponding to the positions of the columns in the first row of the similarity matrix, to form one group.
S6: Skip the rows and also columns corresponding to the selected positions in step S4 in the similarity matrix.
S7: Apply steps S4–S6 on the remaining rows in the similarity matrix, until all the patterns are assigned to groups.

An algorithm for generating the granulation structures, described in the aforesaid steps, is provided by Ganivada et al. (2012). The granulation structures (p groups) generated using the above mentioned procedure, are arranged in decreasing order according to their sizes, where, the size is defined by the number of patterns within a group. We choose the top c groups, out of the p groups, for formation of the proposed FRGNN, where, c is a user defined value. The data points in $c + 1$ to p groups are then added into the top c-groups, as described in the following Steps:

(1) Find average values (means) for all p-groups (including the top c-groups and ($c + 1$ to p)).
(2) Find the Euclidean distances from the means of top c-groups to the mean of the $c + 1$ group.
(3) Find the minimum Euclidean distance and add all the patterns in $c + 1$ group into the group corresponding to the minimum distance.
(4) Repeat steps 2 and 3 for $c + 2$ to p-groups, until all the patterns in these groups are added into the top c-groups.

The resultant c-groups (granulation structures) are then used in the formation of the proposed FRGNN.

4.3. Determination of input vector and target value of FRGNN

Let $\{x'^{k}_{ij}\}$ denote a set of n-dimensional patterns belonging to the c-granulation structures, and the values are randomized between 0 and 1, where $k = 1, 2, \ldots, c$; $i = 1, 2, \ldots, s_k$; $j = 1, 2, \ldots, n$. Here, s_k is the number of patterns in the kth granulation structure. The average values of the patterns in c-granulation structures are calculated and a mean matrix, denoted by M_{kj}, of the size $c \times n$, is constructed, where, rows represent the means of the groups and columns represent the features. Let $\overrightarrow{I}^{j}$ and T_j denote an input

Table 2
Input vector and target value of the proposed FRGNN.

	Means of groups				
	Feature f_1	Feature f_2	Feature f_3	$\cdots$	Feature f_n
Group_1	M_{11}	M_{12}	M_{13}	$\cdots$	M_{1n}
Group_2	M_{21}	M_{22}	M_{23}	$\cdots$	M_{2n}
Group_3	M_{31}	M_{32}	M_{33}	$\cdots$	M_{3n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
Group_c	M_{c1}	M_{c2}	M_{c3}	$\cdots$	M_{cn}
	Average of data (average of the means)				
Data in groups $1, \ldots, c$	T_1	T_2	T_3	$\ldots$	T_n

vector and a target value corresponding to a feature j. $\overrightarrow{I}^j$ is defined as

$$\overrightarrow{I}^j \equiv \{M_{kj}\}, \quad k = 1, 2, \ldots, c. \tag{31}$$

$\overrightarrow{I}^j$ represents an input vector corresponding to the jth column in the mean matrix M_{kj}. T_j is defined as

$$T_j = \frac{\sum_{k=1}^{c} M_{kj}}{c}. \tag{32}$$

In other words, T_j represents a target value computed by taking an average of the components of the input vector corresponding to the jth feature. Here, the component is the mean of the granulation structure along the feature. We concisely explain the concept of input vector and target value by the mean matrix shown in Table 2.

From Table 2, we can see that a mean matrix of the size $c \times n$, where, c is the number of groups and n is the number of features. For example, considering feature f_1, the mean values of all groups, $\{M_{11}, M_{21}, M_{31}, \ldots, M_{c1}\}$, and the average of the means, T_1, are taken as input vector and target value of FRGNN, respectively.

4.4. Formation of the proposed fuzzy rough granular neural network (FRGNN) and an algorithm for training FRGNN

We consider a three-layer neural network consisting of interconnected neurons (nodes) for formation of FRGNN. Number of nodes in the input layer is set equal to c as there are c-rows in the mean matrix M_{kj}. In the hidden layer, the number of nodes is set equal to the number of features (n). One node is set in the output layer. The links connecting the nodes in the input layer and hidden layer, and hidden layer and output layer are initialized by the connection weights, say W_{jk} and w_{0j}, respectively. The initial architecture of FRGNN is shown in Fig. 3.

4.4.1. Determination of initial connection weights

The concepts of fuzzy rough set, based on a decision table, are used for extracting the domain knowledge about data. It is incorporated into FRGNN as its initial connection weights, W_{jk} and w_{0j}. The procedure for defining the connection weights is described as follows:

Procedure: The c-granulation structures, obtained by an α-cut, representing as decision classes, are presented to a decision system $S = (U, \mathcal{A} \cup \{d\})$. Here, U represents a universe and $\mathcal{A}$ represents the attributes, say $\{a_1, a_2, \ldots, a_n\}$. The decision attribute d is defined as X_k, $k = 1, 2, \ldots, c$, where, c represents the number of decision classes. The following steps are applied to a decision table $S = (U, \mathcal{A} \cup \{d\})$ for extracting domain knowledge about data.

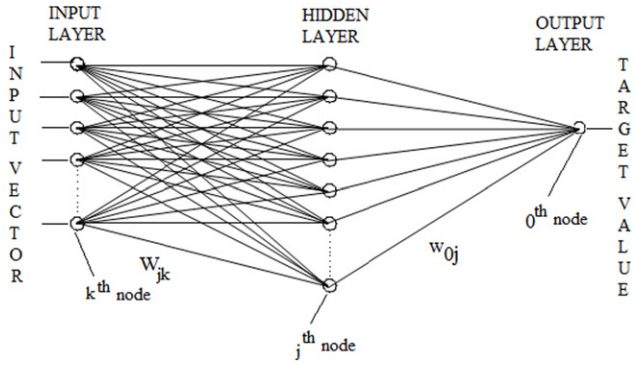

Fig. 3. FRGNN architecture.

S1: Generate a fuzzy reflexive relational matrix by using the fuzzy reflexive relation on all possible pairs of patterns and obtain additional granulation structures based on the relational matrix.
S2: Use the fuzzy reflexive relational matrix to compute the membership value, belonging to the lower approximation (using Eq. (22) or Eq. (15)) of every pattern of a concept, for each conditional attribute with respect to the decision classes (using Eq. (21)).
S3: Calculate the fuzzy positive region (using Eq. (28)) of every pattern for each conditional attribute.
S4: Calculate the degree of dependency (using Eq. (29)), of each conditional attribute within the concept, with respect to the decision classes. Assign these dependency factors as initial connection weights, denoted by W_{jk}, between nodes of the input layer and the hidden layer of FRGNN.
S5: Calculate the average dependency degree of all the dependency degrees of the conditional attributes within the concepts. Here, the average dependency degrees can be represented as the dependency degrees of the conditional attributes. Assign the resultant dependency degrees of the conditional attributes as initial connection weights, denoted by w_{0j}, between nodes of the hidden and output layers of FRGNN.

FRGNN is then trained, in the presence of the input vector and the target vector, using forward propagation and minimization of feature evaluation index with respect to the connection weights between the nodes in the hidden layer and output layer (w_{0j}), in an unsupervised manner. The connection weights, w_{0j}, between nodes of the hidden and output layers are updated, as these weights (w_{0j}) represent dependency degrees of the conditional attributes (see Step S5). The updated weights provide the ordering or importance of the individual features. A higher dependency degree of the attribute signifies that the attribute/feature is the best for selection. It may be noted that the connection weights between the nodes in the input layer and hidden layer (W_{jk}) are not updated during training.

4.4.2. Determination of an algorithm for training FRGNN

We first explain forward propagation of FRGNN. A feature evaluation index is then defined using an error value, defined at the node of output layer, after completing the forward propagation. Minimization of the feature evaluation index with respect to the connection weights between the nodes in the hidden and output layers is described, using gradient decent method, as it is involved in updating the connection weights (w_{0j}), during training. Finally, an algorithm for training FRGNN is provided in this Section.

Forward propagation: During forward propagation, an input vector $\overrightarrow{I}^j$ (Eq. (31)) corresponding to a feature j is presented at the input layer.

Assume that W_{jk} (see Step 4) is an initial connection weight from the kth node in the input layer to jth node in the hidden layer. Here, W_{jk} represents the dependency factor of the jth feature, corresponding to the kth decision class. Therefore, the input of the jth hidden node is the sum of outputs of k input nodes received via connection weights W_{jk} and is represented by

$$O_j^{(1)} = \sum_{k=1}^{c} W_{jk} I_k^j, \quad 1 \le j \le n. \tag{33}$$

The total output of the jth hidden node is the input for single node 0 in the output layer. Let w_{0j} be the initial connection weight (see Step 5) from the jth node in the hidden layer to single node 0 in the output layer. Therefore, the sum of activations received via connection weights w_{0j} by the single node in the output layer is

$$O^{(2)} = \sum_{j=1}^{n} w_{0j} O_j^{(1)}. \tag{34}$$

After the forward propagation of FRGNN is completed, we calculate the error at the output layer node, denoted by E, using the following equation:

$$E = \eta(T - O^{(2)})^2, \tag{35}$$

where, $O^{(2)}$ is the obtained output or actual output at the node in output layer, T is the target value and η is a parameter chosen in the interval (0, 1] and is used to put the value of E within 0 to 1, while an input vector is presented at the nodes in the input layer of FRGNN. Here, the error is defined by calculating the Euclidean distance between the target value and the obtained output. A fuzzy feature evaluation index is then given in terms of E as follows:

Feature evaluation index: A fuzzy feature evaluation index, denoted by H, is defined in terms of the error E (Eq. (35)) as

$$H = \frac{1}{c \ln 2}[-E \ln(E) - (1 - E) \ln(1 - E)], \tag{36}$$

where, c is a constant and is set equal to the number of input nodes in FRGNN (which is equal to the number of granulation structures). Fig. 4 shows the plot of the proposed fuzzy feature evaluation index H with $c = 2$ and base e for different values of $E \in [0, 1]$ in the 2-dimensional plane. The properties of the feature evaluation index H are discussed as follows:

1. Sharpness: $H = 0$ iff $E = 0$ or 1.
2. Maximality: H attain maximum value iff $E = 0.5$.
3. Symmetric: $H = \overline{H}$ iff $E = \overline{E}$, where, $\overline{H}$, complement of H, attains a value corresponding to a value of $\overline{E}$, complement of E.
4. Resolution: $H \ge H^*$, when $E \ge E^*$. Here, H^* varies with E^*, a sharpened version of E.
5. Continuity: H is a continuous function of E, where $E \in [0, 1]$.

The feature evaluation index H is now minimized with respect to the weights, w_{0j}, between nodes of the hidden and output layers. The minimization of the feature evaluation index H, with respect to the weights, signifies that the convergence towards the improved values for the weights is achieved. The improved values for the weights provide an ordering of the individual features. A higher value of the weight signifies that the importance of the corresponding feature is high.

Minimization of feature evaluation index: The minimization of the feature evaluation index H with respect to the weights, w_{0j}, is performed by using gradient descent method. This implies that the changes in the weights w_{0j}, say $\triangle w_{0j}$, are proportional to $-H/\partial w_{0j}$.

$$\triangle w_{0j} = -\delta \frac{\partial H}{\partial w_{0j}} \tag{37}$$

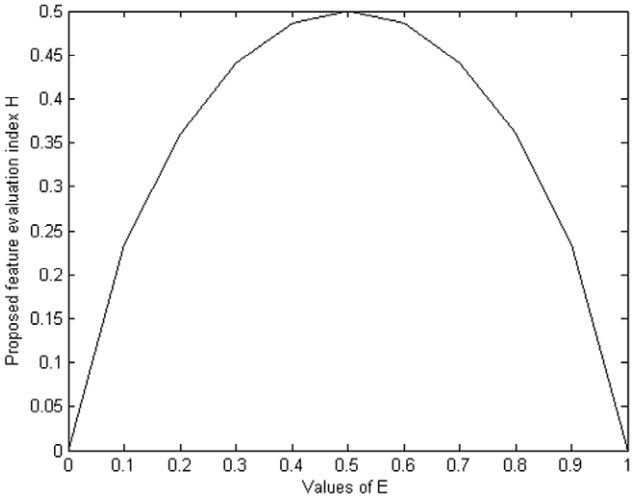

Fig. 4. Feature evaluation index for different values of E.

where, δ is the learning rate chosen between 0 and 1. The partial derivative $\frac{\partial H}{\partial w_{0j}}$ can be evaluated using the chain rule

$$\frac{\partial H}{\partial w_{0j}} = \frac{\partial H}{\partial O^{(2)}} \frac{\partial O^{(2)}}{\partial w_{0j}}. \tag{38}$$

Using Eq. (34), the expression $\frac{\partial O^{(2)}}{\partial w_{0j}}$ can be obtained as

$$\frac{\partial O^{(2)}}{\partial w_{0j}} = \frac{\partial}{\partial w_{0j}} \sum w_{0j} O_j^{(1)} = O_j^{(1)}. \tag{39}$$

The expression $\frac{\partial H}{\partial O^{(2)}}$ signifies the rate of change of error with respect to the output $O^{(2)}$ and using Eq. (36) it can be written as

$$\frac{\partial H}{\partial O^{(2)}} = \frac{\partial}{\partial O^{(2)}} \left(\frac{1}{c \ln 2}[-E \ln(E) - (1 - E) \ln(1 - E)] \right). \tag{40}$$

Using Eq. (35), Eq. (40) can be written as

$$\frac{\partial H}{\partial O^{(2)}} = \frac{\partial}{\partial O^{(2)}} \left(\frac{1}{c \ln 2} \{ -(\eta(T - O^{(2)})^2) \ln(\eta(T - O^{(2)})^2) - (1 - \eta(T - O^{(2)})^2) \ln(1 - \eta(T - O^{(2)})^2) \} \right). \tag{41}$$

After completion of the differentiation, Eq. (41) can be obtained as

$$\frac{\partial H}{\partial O^{(2)}} = \frac{1}{c \ln 2} (2\eta(T - O^{(2)}) \ln(\eta(T - O^{(2)})) - (2\eta(T - O^{(2)}) \ln(1 - \eta(T - O^{(2)})^2))). \tag{42}$$

Now, we substitute Eqs. (39) and (42) in Eq. (38). Therefore, Eq. (38) is obtained as

$$\frac{\partial H}{\partial w_{0j}} = \frac{1}{c \ln 2} (2\eta(T - O^{(2)}) \ln(\eta(T - O^{(2)})) - (2\eta(T - O^{(2)}) \ln(1 - \eta(T - O^{(2)})^2))) O_j^{(1)}. \tag{43}$$

Using Eqs. (37) and (43), we obtain,

$$\triangle w_{0j} = -\delta \left(\frac{1}{c \ln 2} (2\eta(T - O^{(2)}) \ln(\eta(T - O^{(2)})^2) - (2\eta(T - O^{(2)}) \ln(1 - \eta(T - O^{(2)})^2))) O_j^{(1)} \right). \tag{44}$$

Algorithm for training the network: During training of FRGNN, the weights w_{0j} of the links, connecting the hidden and the output layers, are updated, and the updated equation is defined as

$$w_{0j}(l+1)+ = \triangle w_{0j}(l); \tag{45}$$

where, l represents an epoch or iteration. For every iteration, the training of the network is performed using the following steps:

S1: Present an input vector $\overrightarrow{I}_j$ (Eq. (31)) at the nodes of the input layer and a target vector T_j (Eq. (32)) at the output layer node.
S2: Train the network using Eqs. (33) and (34).
S3: Find $\triangle w_{0j}$ using Eq. (44).
S4: Update the weights w_{0j} using Eq. (45).

After the training of FRGNN is completed, a higher value (closer to 1) of w_{0j} signifies that the corresponding feature j is the best for selection. We arrange the weights in decreasing order and select different sets of features for performing further experiments. The number of features in a set is represented by k, which is less than or equal to the total number of features. The results for the best k number of features, out of all the values of k, are considered.

5. Methods for comparison and evaluation of the performance of FRGNN

Here, we describe the existing algorithms for feature selection, and these are used for comparing the performance of FRGNN. Some methods used for evaluation of the selected features are also discussed.

5.1. Methods for comparison

We compare the performance of FRGNN with that of "unsupervised feature selection using feature similarity (UFSFS)" described by Mitra et al. (2002), sequential forward search (SFS) given by Devijver and Kittler (1982) and Relief-F proposed by Kira and Rendell (1992). The code of UFSFS, implemented in Matlab, is downloaded from the home page of Mitra (2002). Matlab codes of SFS and Relief-F are available with Matlab 7.11. Now, we describe the methods, UFSFS, SFS and Relief-F, as follows:

Unsupervised feature selection using feature similarity (UFSFS): A feature similarity matrix is computed by finding the similarity between the features using a similarity measure. The features are partitioned into distinct groups, based on the K-NN principle, by using the similarity matrix. For a group, the most representative feature is then selected and the remaining features are discarded. Here, the representative feature means a feature having the farthest distance from the remaining features in the group. The same process is repeated for all the remaining groups until all the features are selected are discarded. A reduced feature subset is then constituted by collecting all the representative features from the groups. Further information of this algorithm is available in the article Mitra et al. (2002).

Sequential forward search (SFS): It was defined by Devijver and Kittler (1982) and is a bottom up search process. The process starts with an empty subset and sequentially adds a feature to the subset that gives the best criterion value. In general, misclassification rate is considered as the criterion value. For every iteration, a feature, which will be combined with the features already included in the subset in order to extend the size of the subset, is selected from the remaining available features (which are not added to the subset so far). Hence, the extended subset of features should provide a minimum misclassification value. This process is repeated until the size of the subset of features reaches an expected number, say ('k'), of features. The value of k is chosen as less than the total number of features in the data.

Relief-F: Relief-F is a supervised feature selection algorithm proposed by Kira and Rendell (1992). For each feature, a pattern is chosen randomly from the dataset, and the distances between the randomly selected pattern & its nearest neighbor from the same class (say 'hit') and the randomly selected pattern & its nearest neighbor from the different class (say 'miss') is calculated. A weight for every feature is computed by taking the difference between the hit and the miss. This process is repeated for all features and the features are ranked, based on their weights. Finally, the features, whose average weight is greater than a threshold, are selected.

5.2. Methods for evaluation

The features selected by the aforesaid methods are evaluated using Naive Bayes and K-NN and the entropy measure for feature evaluation introduced by Pal and Mitra (1999). Their classification results are shown in terms of percentages of accuracy. The significance of percentages of accuracy of the proposed method, as compared to the other algorithms, is evaluated by Macro precision, recall and F-score.

K-NN: The K-NN classifier is used for evaluating the effectiveness of the selected attributes in classification of the data. The K-NN needs class information for performing classification. During training of the K-NN, we used 10-fold cross validation, designed with stratified sampling. Training is done on 1-fold of data, selected randomly from each of the classes. The remaining 9-folds of data are treated as test data. This process is repeated 10 times for the same number of selected features, and the overall performance of K-NN is determined by taking the average result. The value of 'K' in K-NN is chosen as the square root of the number of patterns in the training data.

Naive Bayes: For the assessment of the efficiency of selected features in classification of the datasets, the Naive Bayes classifier is used in a similar way to the K-NN classifier assuming either normal distribution or multivariate multinomial distribution of the classes for the different datasets.

Entropy: Let p and q be the two patterns. The distance between the two patterns can be defined by Pal and Mitra (1999) as

$$D_{pq} = \left[\sum_{j=1}^{M}\left(\frac{x_{pj}-x_{qj}}{\max_j - \min_j}\right)^2\right]^{\frac{1}{2}}, \tag{46}$$

where, x_{pj} represents a feature value of the pattern p along the jth axis, $\max_j \& \min_j$ represent minimum and maximum values, respectively, among all the patterns along the jth axis, and M is the total number of selected features. The similarity between p and q is given by $sim(p, q) = \exp -\alpha D_{pq}$, where, α is a positive constant. The value of α can be computed by $\frac{-\ln 0.5}{\overline{D}}$, where, $\overline{D}$ is defined by the average of distances for all possible pairs of patterns of the data. Entropy is, therefore, defined as

$$E = -\sum_{p=1}^{s}\sum_{q=1}^{s}(sim(p,q)\log(sim(p,q)) + (1-sim(p,q))\log(1-sim(p,q))). \tag{47}$$

A lower value of entropy for a group of patterns signifies that they are clustered in a better fashion.

The following Macro precision, recall & F-score are computed using the classification accuracies.

Table 3
Characteristics of data.

Dataset name	No. of patterns	No. of features	No. of classes
Iris	150	4	3
Wisconsin cancer	684	9	2
Waveform	5000	40	3
Spam base	4601	57	2
Ionosphere	351	33	2
Multiple features	2000	649	10
Arrhythmia	452	180	13
Secom data	1567	591	2

Macro precision, recall and F-score: Macro precision and recall and F-score are described by Salton and McGill (1983) as follows. The precision (p_k) and recall (r_k) of a class k are defined as

$$p_k = \frac{\text{Number of patterns correctly classified into class } k}{\text{Number of patterns classified into class } k}, \quad (48)$$

$$r_k = \frac{\text{Number of patterns correctly classified into class } k}{\text{Number of patterns that are truly present in class } k}. \quad (49)$$

Then, the harmonic mean between the precision and recall of class k, denoted by F_k, is defined as

$$F_k = \frac{2 \times p_k \times r_k}{p_k + r_k}, \quad (50)$$

where, F_k gives equal importance to both precision and recall to the kth class. The Macro average values of the precision, recall and F-score, lying between 0 and 1, of all the classes over 10-folds are then defined. The closer the value of Macro average F-score is to 1, the better the classification of the dataset is.

In addition, a self-organizing map is used for clustering the microarray gene expression data, based on the selected features. The resultant clusters of microarray gene expression datasets are evaluated using the entropy, β-index defined by Pal, Ghosh, and Shankar (2000), Davies–Bouldin index (DB-index) described by Davies and Bouldin (1979) and fuzzy rough entropy developed by Ganivada et al. (2012). A higher value of β-index indicates that the clustering solutions are compact whereas, it is the opposite for entropy, DB-index and fuzzy rough entropy.

6. Experimental results

In this section, the effectiveness of the proposed FRGNN is demonstrated on different real life datasets including microarray gene expression data. The characteristics of datasets, which are collected from the UCI machine learning repository provided by Blake and Merz (1998), are shown in Table 3.

Datasets are chosen in three categories based on their dimensions such as *low* (dimension <10), *medium* ($10 <$ dimension <100) and *high* (dimension >100). These are summarized as follows:

(1) *Iris:* The iris dataset contains 150 patterns with 4 features/attributes and 3 classes. Each class contains 50 patterns, referring to a type of iris plant, and is linearly separable from the other classes. Attribute information is given in terms of sepal length, sepal width, petal length and petal width representing the four features.
(2) *Wisconsin cancer:* This dataset contains 684 patterns and 9 features. All patterns belong to 2 classes, represented by benign tumor and malignant tumor.
(3) *Ionosphere:* This data refers to radar data. The radar signals are represented as autocorrelation functions of radar measurements. It contains 352 patterns, 33 attributes and 2 classes.

Table 4
Summary for different microarray datasets.

Dataset name	No. of genes	No. of time points	No. of functional categories
Cell cycle	634	93	16
Yeast complex	979	79	16
All Yeast	6072	80	18

(4) *Waveform:* This dataset contains 5000 patterns, 40 attributes and 3 wave classes. While, all the attribute values are continuous between 0 and 6, 19 attributes among them are noise values with mean 0 and variance 1.
(5) *Spam base:* This data refers to spam and non spam emails. The task is to determine whether a given email is spam or not. It contains 4601 instances, 57 continuous valued attributes representing word frequencies, and 2 classes.
(6) *Multiple features:* This dataset consists features of handwritten numerals, ('0'–'9'), extracted from a collection of Dutch utility maps. There are 2000 patterns, 625 attributes, and 10 classes. The values of attributes are integers and real type.
(7) *Arrhythmia:* The cardiac arrhythmia data contains 452 samples and 279 attributes, 206 of which are linear and the rest are nominal. Out of 206 linear attributes, 179 are used in our experiments and the rest, containing missing values, are removed. All the attributes represent ECG measurements. The aim is to classify it into one of the 13 classes of cardiac arrhythmia.
(8) *Secom data:* The secom data consists of 1567 samples, 591 features, and 2 classes. Each attribute represents a signal, which is operated by a semi-conductor manufacturing process, collected by sensors and labeled with process measurements.
(9) *Microarray gene expression data:* Microarray gene expression datasets, like Cell Cycle provided by Ray, Bandyopadhyay, and Pal (2007), Yeast Complex prepared by Bar-Joseph, Gifford, and Jaakkola (2001) and Eisen, Spellman, Brown, and Botstein (1998), All Yeast generated by Eisen et al. (1998), are used in experiments. Based on the functional annotations, shown by Ray, Bandyopadhyay, and Pal (2012), of the Munich Information for Protein Sequences(MIPS) database, the genes in Cell Cycle, Yeast Complex, and All Yeast are classified into 16, 16 and 18 groups, respectively. The characteristics of these datasets, namely, name, number of genes, the number of time points (attributes), and number of top level functional categories (classes), are given in Table 4. The Cell Cycle data contains total 653 genes. Out of 653 genes, 19 genes, containing missing gene expression values, are removed. The remaining 634 genes are then used in the experimental results. Similarly, for All Yeast data, out of 6221 genes, 6072 genes are used in the experimental results after 149 genes, containing missing gene expression values, are removed. The Yeast Complex data contains 979 genes without any missing gene expression values.

6.1. Results

We first explain the connectionist mechanism of FRGNN and then describe the process of selecting the best features by using FRGNN for iris data. Let f_1, f_2, f_3 and f_4 denote the four consecutive features of iris data. Figs. 5–10 show the scatter plots of the pairwise combinations of the four features of three classes in the 2-dimensional (2D) plane.

Initially, each feature is normalized between 0 and 1 using Eq. (30) (see Section 4.1). A similarity matrix is computed using the procedure explained in Section 4.2. Different numbers of granulation structures are then generated for different values of α, chosen between 0 and 1. As a typical example, let us consider the iris data. Here, 11 groups are generated for $\alpha = 0.69$. These groups are then

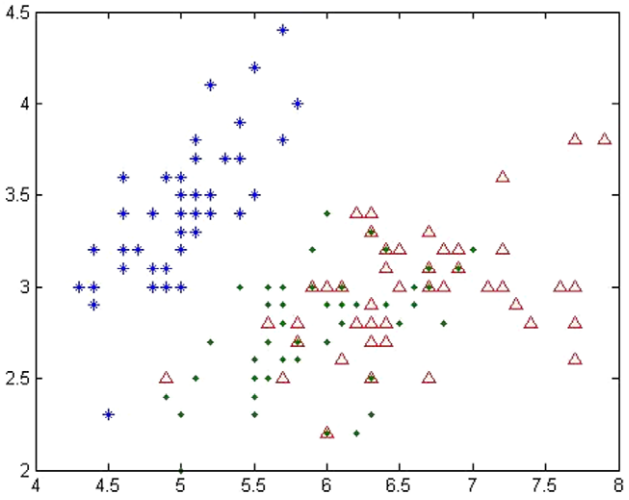

Fig. 5. Features f_1–f_2 plane.

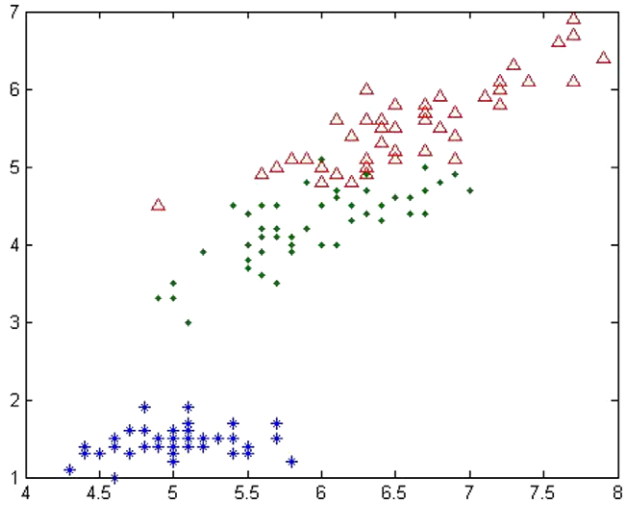

Fig. 6. Features f_1–f_3 plane.

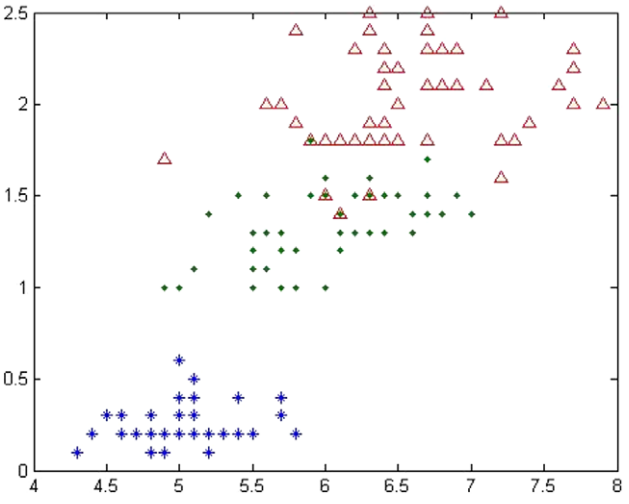

Fig. 7. Features f_1–f_4 plane.

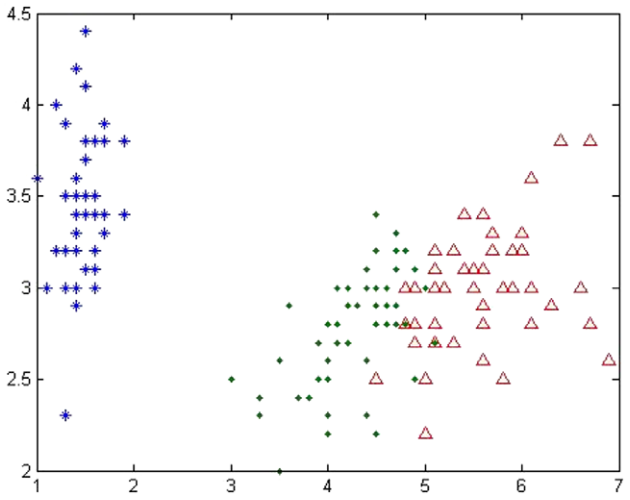

Fig. 8. Features f_2–f_3 plane.

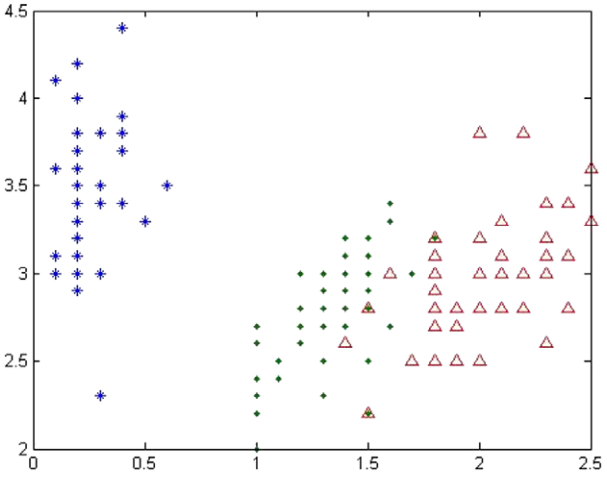

Fig. 9. Features f_2–f_4 plane.

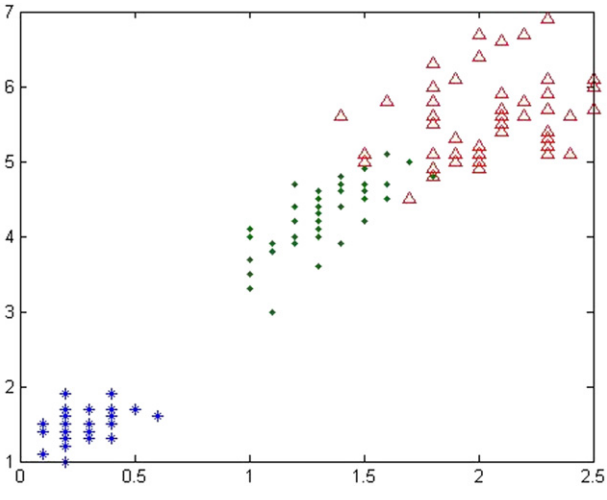

Fig. 10. Features f_3–f_4 plane.

arranged in a descending order according to their sizes. We choose the top 5 groups ($c = 5$) out of 11 groups. The remaining groups (6 to 11) are added into the top 5-groups using steps 1–5, given in Section 4.2. The input vectors and the target values are then determined using Eqs. (31) and (32), respectively, and are shown in Table 5.

By observing the results from Table 5, an input vector and target value, corresponding to feature f_1, is represented by {0.3908, 0.8049, 0.6355, 0.3680 and 0.5833} and 0.5712, respectively. In

Table 5
A mean matrix, representing input vector, defined by using the 5 groups for iris data.

	f_1	f_2	f_3	f_4
Input vector	0.3908	0.6048	0.3033	0.3376
	0.8049	0.4036	0.9212	0.9409
	0.6355	0.7533	0.4583	0.485
	0.3680	0.4930	0.2274	0.0520
	0.5833	0.6726	0.5544	0.5416
Target	0.5712	0.5594	0.5325	0.5419

Table 6
Initial connection weights of FRGNN for iris data.

Initial connection weights from each of the input nodes to 4 nodes in the hidden layer					Initial connection weights from 4 nodes in the hidden layer to the node in the output layer
0.1010	0.1584	0.1711	0.1512	0.0767	0.1271
0.1057	0.1495	0.1541	0.1617	0.0643	0.1513
0.1197	0.2003	0.1783	0.1829	0.0754	0.1492
0.1247	0.1806	0.1531	0.1956	0.0921	0.1317

Table 7
Ordering of the features for iris data using FRGNN.

Features	Updated weights	Order
f_1	0.3457	4
f_2	0.3563	3
f_3	0.4028	1
f_4	0.3732	2

FRGNN, the number of nodes in the input layer is set to 5 for iris data as the mean matrix has 5 means corresponding to the 5 groups. The number of nodes in the hidden layer is set to 4 as the iris data has four features. The 5 groups are presented to a decision system S. The decision table S is used to extract domain knowledge about data in terms of dependency factors using the concept of fuzzy rough sets. The dependency factors of the features, corresponding to the 5 groups, and the average of them are initialized as connection weights between nodes of the input layer and hidden layer, and the hidden layer and output layer, respectively. These are shown in Table 6.

During training of FRGNN, the input vector is presented at nodes of the input layer, while the target value is stored at the single node of the output layer. The connection weights between nodes of hidden and output layers are updated by using Eq. (45). The values of η (a parameter used to put the error within 0 to 1) in Eq. (35) and δ (learning rate) in Eq. (45) are set to be 0.9 and 0.00091, respectively, for iris data. After the training of FRGNN is completed, the updated weights (w_{0j}) between nodes of the hidden and output layers provide the ordering of the features as shown in Table 7. The ordering is found to be $f_3 > f_4 > f_1 > f_2$. For evaluating the effectiveness of FRGNN, we selected the top k number of features, say $k = 2$, i.e., features f_3 and f_4, and classified the dataset separately using Naive Bayes and K-NN classifiers, and the results are shown in terms of percentage of accuracies. The importance of the features are also evaluated with a entropy measure, where, for a particular method, say FRGNN, the entropy value for the top k number of selected features is calculated. The results are provided in Table 8. The same procedure, explained so far for iris data, is applied for selecting salient features from the remaining datasets.

6.2. Effect of top-k features on classification accuracies for K-NN and Naive Bayes

Here, we study the variation of classification accuracies with the numbers of top selected features for Wisconsin breast cancer, waveform and multiple features data with dimensions 9, 40 and 649, respectively (belonging to *low* (<10), *medium* (10–100) and *high* (>100)), using K-NN and Naive Bayes classifiers, as an example. Different sets of features, which are obtained by FRGNN, UFSFS, SFS and Relief-F, are used for the classification purpose. The number of selected features in a set is denoted by k. The variation of average classification accuracies with the top k number of selected features are shown in 2D-plots.

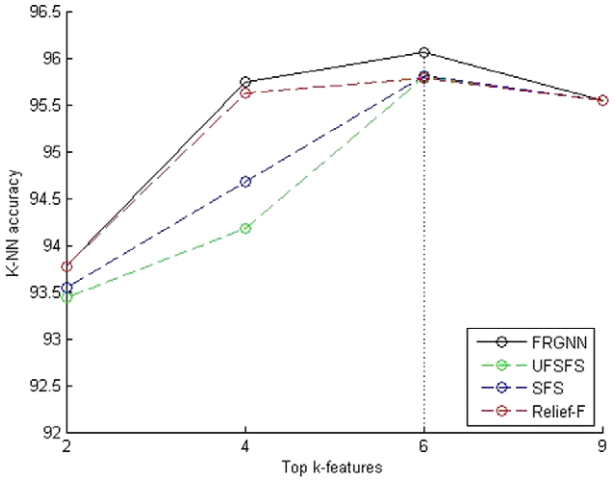

Fig. 11. Variation of the average classification accuracies of K-NN with the number of top selected features for Wisconsin breast cancer data.

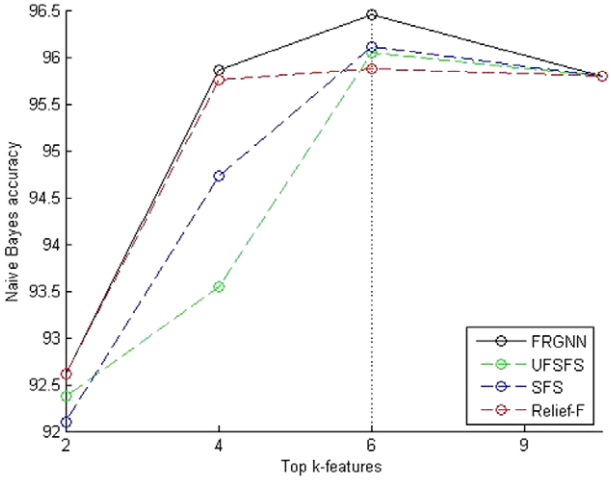

Fig. 12. Variation of the average classification accuracies of Naive Bayes with the number of top selected features for Wisconsin breast cancer data.

Wisconsin breast cancer data: For Wisconsin breast data, the classification accuracies of K-NN and Naive Bayes, corresponding to different numbers of features ($k = 2$, 4, 6 and 9) selected by FRGNN, UFSFS, SFS and Relief-F, are shown in Figs. 11 and 12, respectively.

It is observed from Figs. 11 and 12 that the plots of average classification accuracies of K-NN and Naive Bayes for FRGNN are gradually increased for $k = 2$, 4 and 6 and are decreased for $k = 9$. Similar observations can also be seen for the remaining algorithms, but the performance of FRGNN is better than related methods for $k = 2$, 4 and 6 whereas, Relief-F is the second best.

It is also observed for all feature selection algorithms that K-NN and Naive Bayes provide the highest classification accuracies for $k = 6$, out of all other values of k. Hence $k = 6$ is marked by a vertical dotted line in Figs. 11 and 12.

Waveform data: For waveform data, the variation of the average classification accuracies with numbers of selected features, obtained by all the algorithms, are shown in Figs. 13 and 14. It is

Table 8
Comparison of the classification values of K-NN and Naive Bayes for Relief-F, SFS, UFSFS and FRGNN for different datasets.

Dataset	Method	K-NN			Naive Bayes			Entropy	CPU time in s.
		Min. acc. (%)	Max. acc. (%)	Avg. acc. (%)	Min. acc. (%)	Max. acc. (%)	Avg. acc. (%)		
Iris ($k = 2$)	Relief-F	87.41	96.29	93.04	86.67	97.03	94.00	0.2078	0.4595
	SFS	62.96	91.11	82.37	60.71	93.93	81.87	0.2191	0.4056
	UFSFS	87.41	96.29	93.04	86.67	97.03	94.00	0.2078	0.0468
	FRGNN(5)	87.41	96.29	93.04	86.67	97.03	94.00	0.2078	0.5000
Wisconsin breast cancer ($k = 6$)	Relief-F	94.28	96.57	95.79	95.09	96.24	95.44	0.2158	1.1112
	SFS	94.77	96.41	95.75	95.42	97.05	96.11	0.2169	0.6252
	UFSFS	94.42	96.73	95.62	94.77	97.05	96.04	0.2158	0.0506
	FRGNN(2)	95.42	96.57	96.06	95.75	97.05	96.45	0.2130	2.2560
Waveform ($k = 15$)	Relief-F	83.68	85.37	84.58	78.71	80.75	79.42	0.2424	26.4391
	SFS	82.97	84.86	83.80	78.66	80.22	79.19	0.2426	20.3433
	UFSFS	81.55	82.35	82.06	77.13	79.88	78.23	0.2433	0.6978
	FRGNN(3)	83.60	85.51	84.62	79.26	80.77	79.62	0.2418	146.0420
Spam base ($k = 29$)	Relief-F	67.65	69.29	68.24	51.75	67.72	57.79	0.2297	25.6744
	SFS	74.71	80.24	77.80	54.88	67.47	60.57	0.2281	36.8786
	UFSFS	80.02	82.51	81.45	53.08	67.02	60.64	0.2286	1.1544
	FRGNN(4)	80.96	85.09	82.76	59.10	75.06	61.08	0.2174	216.5280
Ionosphere ($k = 16$)	Relief-F	65.71	84.44	73.05	73.33	89.21	80.76	0.2370	1.3340
	SFS	64.76	77.14	70.50	66.98	86.98	79.61	0.2333	4.1596
	UFSFS	74.60	83.49	76.00	71.74	86.03	82.34	0.2319	0.1062
	FRGNN(2)	64.44	84.12	78.47	79.36	91.42	84.06	0.2317	1.510
Multiple features ($k = 325$)	Relief-F	68.88	74.94	70.70	91.27	94.66	92.78	0.2368	327.9212
	SFS	67.56	72.22	70.22	91.66	94.05	93.27	0.2475	19477.9917
	UFSFS	78.72	84.44	81.35	91.44	94.00	92.55	0.2465	63.5080
	FRGNN(8)	81.05	87.11	85.15	91.94	94.66	93.42	0.2454	850.9470
Arrhythmia ($k = 100$)	Relief-F	53.08	55.80	54.37	51.61	57.77	54.72	0.2410	25.2878
	UFSFS	49.38	55.30	53.58	43.20	55.80	52.17	0.2370	2.1528
	FRGNN(6)	53.33	56.79	55.20	50.74	60.29	55.23	0.2271	13.6860
Secom data ($k = 100$)	Relief-F	92.94	93.58	93.33	82.78	92.12	87.71	0.2460	73.3049
	UFSFS	93.23	93.80	93.33	84.64	89.53	86.60	0.2450	40.9971
	FRGNN(2)	93.09	93.73	93.40	80.56	91.68	86.68	0.2406	788.2932
Average results of first six datasets	Relief-F	78.12	84.51	80.90	79.47	87.60	83.36	0.2282	63.8232
	SFS	78.69	84.60	81.86	79.04	87.13	83.79	0.2293	3256.7429
	UFSFS	82.43	87.36	84.94	79.13	86.83	83.96	0.2290	10.9273
	FRGNN	82.14	89.11	86.68	82.01	89.33	84.77	0.2262	202.9638
Average results of all datasets	Relief-F	76.84	82.05	79.14	76.40	84.43	80.32	0.2321	60.1915
	UFSFS	79.65	84.15	82.08	75.33	83.29	80.32	0.2320	13.5892
	FRGNN	79.91	85.65	83.57	77.92	85.99	81.31	0.2281	252.4703

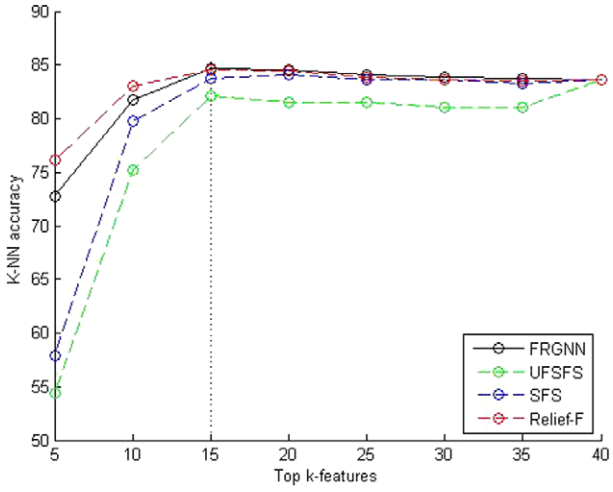

Fig. 13. Variation of the average classification accuracies of K-NN with the number of top selected features for waveform data.

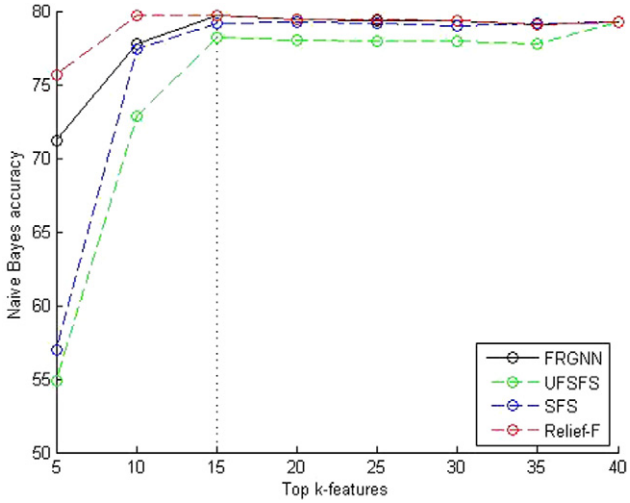

Fig. 14. Variation of the average classification accuracies of Naive Bayes with the number of top selected features for waveform data.

observed that the classification accuracies of both K-NN and Naive Bayes for all the algorithms are gradually increased for $k = 5$, 10 and 15, and are gradually decreased for $k = 20$, 25, 30 and 35.

For $k = 5$ and 10, the performance of FRGNN is inferior to Relief-F and is superior to UFSFS and SFS. For the remaining values of k (other than $k = 5$ and 10), it is evident from Figs. 13 and 14 that the performance of FRGNN is better than the related methods. At value 15, k is marked with a vertical dotted line, where, the maximum performance is achieved for all algorithms.

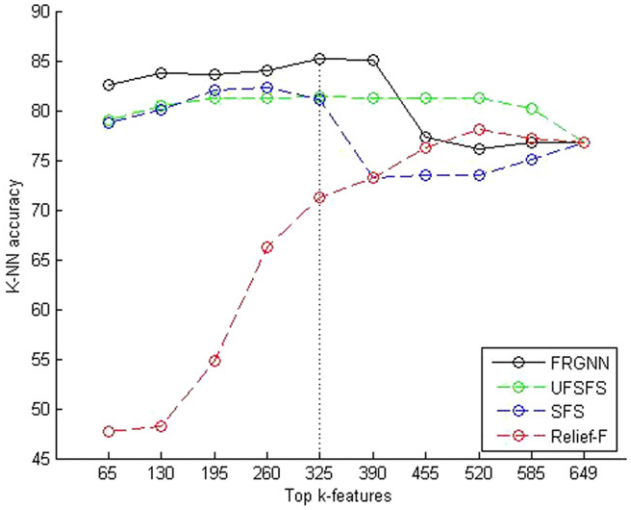

Fig. 15. Variation of the average classification accuracies of K-NN with the number of top selected features for multiple features data.

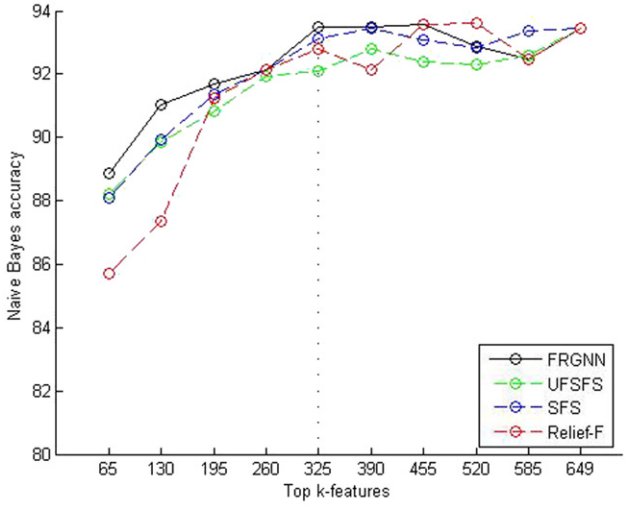

Fig. 16. Variation of the average classification accuracies of Naive Bayes with the number of top selected features for multiple features data.

Multiple features data: For multiple features data, the values of k are chosen as 65, 130, 195, 260, 325, 390, 455, 520, 585 and 649, while, the smallest one ($k = 65$) is 10% of the actual feature size, and the other values are obtained by increasing the percentage by 10 at a time. Thereafter, the data with the selected number of features, obtained by FRGNN, UFSFS, SFS, Relief-F algorithms, is classified using K-NN and Naive Bayes. The plots of average classification values of K-NN and Naive Bayes, corresponding to different values of k, are shown in Figs. 15 and 16, respectively.

From Fig. 15, it can be seen that FRGNN provides higher classification accuracies, as compared with UFSFS, SFS, Relief-F, for $k = 65$, 130, 195, 260, 325 and 390 using K-NN. Using Naive Bayes, a similar observation can also be made for all the algorithms from Fig. 16 for $k = 65$, 130, 195, 325 and 390. At value 260, the performance of FRGNN seemed to be equal to SFS and Relief-F. However, the best classification accuracy, using K-NN and Naive Bayes, for all the algorithms is achieved at value 325, out of all other values of k. Hence, this value is marked with a vertical dotted line in Figs. 15 and 16.

The results for a particular value of k, for which, K-NN and Naive Bayes provide the best average performance (average of 10 fold cross validation results) for all algorithms, is shown in Table 8. The value of k is mentioned within parenthesis at column 1. The CPU time for feature selection, and classification accuracies and entropy value of the selected features are shown in the table for all the methods and datasets. The number of nodes set in the input layer of FRGNN for the data is shown within parenthesis at column 2.

For iris data, by using FRGNN, features 3 and 4 are selected, which are also selected by UFSFS and Relief-F. Hence, the results, shown in Table 8 for K-NN, Naive Bayes and entropy, are found to be equal for these algorithms. In contrast, by using SFS, features 1 and 3 are selected and the results for SFS, shown in Table 8, are inferior to FRGNN. These results reveal that, the overlapping between the class boundaries of features 3 and 4, shown in Fig. 10, is less than the other features, as shown in Figs. 5–9.

From Table 8, it is observed that, for Wisconsin breast cancer data, the average percentage of accuracies of K-NN and Naive Bayes are 96.06% and 96.45%, respectively, for FRGNN, whereas, using UFSFS, SFS and Relief-F, these are 95.62% & 96.04%, 95.75% & 96.11%, and 95.79% & 95.44%, respectively. Entropy value for FRGNN is seen to be lower (0.2130) than that of UFSFS (0.2158), SFS (0.2169) and Relief-F (0.2158). Based on these results, it can be stated that FRGNN provides a better performance than the other algorithms.

For waveform data, the average percentage of accuracies of K-NN and Naive Bayes for FRGNN are 84.62% and 79.62% respectively, and are seen to be higher than those of Relief-F (84.58%) and (79.42%), SFS (83.80%) and (79.19%), and UFSFS (82.06%) and (78.23%). The entropy value for FRGNN is found to be lower than the remaining algorithms.

For multiple features of high dimension, it can be observed that the performance of FRGNN, in terms of entropy, the average percentage of accuracy using K-NN and Naive Bayes, is seen to be superior to the other algorithms. Similar comparisons can also be made for the remaining datasets based on the results shown in Table 8.

It may be noted that SFS does not work for arrhythmia and secom data sets as the covariance of the training data is seen to be non positive definite, when the misclassification rate is defined for these two data sets using Naive Bayes classifier. The performance of FRGNN is compared with Relief-F, SFS and UFSFS by considering the average results of the first six datasets. By using FRGNN, the average percentage of accuracies for K-NN and Naive Bayes are obtained as 86.68% and 84.77%, respectively, whereas, these are obtained for K-NN and Naive Bayes as 80.90% & 83.36% using Relief-F, 81.86% & 83.79% using SFS, and 84.94% & 83.96% using UFSFS, respectively. The average of all entropy values for FRGNN is 0.2262, which is smaller than that of Relief-F (0.2282), SFS (0.2293) and UFSFS (0.2290).

The CPU times in seconds, for all the methods and datasets are also provided in the last column of Table 8. For iris, Wisconsin breast cancer, waveform, spam base, arrhythmia and secom data, the CPU times for FRGNN are 0.5000, 2.2560, 146.0420, 216.5280, 13.6860 and 788.2932 respectively. It is observed that these values are the highest for FRGNN, as compared to the related methods except for ionosphere and multiple features data which are the second highest. For most of the datasets, the CPU time is second highest for SFS. The lowest CPU time is observed for UFSFS for all the datasets. Although FRGNN takes higher CPU time, the top features selected by FRGNN provide higher classification accuracies in the most of the cases.

The significance of FRGNN and related methods are provided in Table 9 in terms of the average of precision, recall & F-score of all the classes for six datasets. For FRGNN, the average values of precision are 0.9502, 0.9612, 0.8480, 0.8246, 0.8277 & 0.8624 and recall are 0.9487, 0.9517, 0.8466, 0.8141, 0.7179 & 0.8554 for iris, Wisconsin breast cancer, waveform, spam base, ionosphere and multiple features data, respectively, using K-NN. For the same data sets, the best average values of F-score for FRGNN are 0.9481,

Table 9
Comparison of average precision, recall and F-score of K-NN and Naive Bayes for Relief-F, SFS, UFSFS and FRGNN for the datasets with different dimensions.

Dataset	Method	K-NN			Naive Bayes		
		Average precision	Average recall	Average F-score	Average precision	Average recall	Average F-score
Iris ($k = 2$)	Relief-F	0.9502	0.9487	0.9481	0.9484	0.9452	0.9468
	SFS	0.8480	0.8148	0.8300	0.8935	0.8661	0.8794
	UFSFS	0.9502	0.9487	0.9481	0.9484	0.9452	0.9468
	FRGNN(5)	0.9502	0.9487	0.9481	0.9484	0.9452	0.9468
Wisconsin breast cancer ($k = 6$)	Relief-F	0.9591	0.9488	0.9539	0.9546	0.9569	0.9558
	SFS	0.9595	0.9478	0.9536	0.9580	0.9568	0.9574
	UFSFS	0.9594	0.9475	0.9534	0.9582	0.9547	0.9564
	FRGNN(2)	0.9612	0.9517	0.9564	0.9594	0.9634	0.9613
Waveform ($k = 15$)	Relief-F	0.8478	0.8461	0.8470	0.8274	0.7963	0.8115
	SFS	0.8402	0.8384	0.8393	0.8206	0.7939	0.8104
	UFSFS	0.8226	0.8211	0.8218	0.8110	0.7844	0.7975
	FRGNN(3)	0.8480	0.8466	0.8473	0.8290	0.7984	0.8133
Spam base ($k = 29$)	Relief-F	0.6774	0.6767	0.6769	0.6459	0.6446	0.6452
	SFS	0.7702	0.7801	0.7814	0.6713	0.6630	0.6571
	UFSFS	0.8089	0.7870	0.8038	0.6634	0.6560	0.6596
	FRGNN(4)	0.8246	0.8141	0.8192	0.6696	0.6650	0.6674
Ionosphere ($k = 16$)	Relief-F	0.8194	0.6437	0.7155	0.8381	0.7774	0.8065
	SFS	0.7984	0.5680	0.6884	0.8049	0.7476	0.7961
	UFSFS	0.8396	0.6733	0.7450	0.8264	0.7925	0.8090
	FRGNN(2)	0.8277	0.7179	0.7685	0.8510	0.8311	0.8408
Multiple features ($k = 325$)	Relief-F	0.7304	0.7138	0.7220	0.9338	0.9281	0.9309
	SFS	0.8279	0.8127	0.8290	0.9355	0.9304	0.9329
	UFSFS	0.8342	0.8150	0.8260	0.9264	0.9214	0.9239
	FRGNN(8)	0.8624	0.8554	0.8589	0.9369	0.9314	0.9341
Average results of the datasets	Relief-F	0.8307	0.7963	0.8116	0.8580	0.8414	0.8495
	SFS	0.8407	0.7986	0.8171	0.8473	0.8296	0.8383
	UFSFS	0.8691	0.8321	0.8456	0.8567	0.8439	0.8502
	FRGNN	0.8790	0.8557	0.8609	0.8647	0.8542	0.8595

Table 10
Comparison of UFSFS and FRGNN for microarray gene expression datasets.

Dataset	Method	Entropy	β-index	DB-index	Fuzzy rough entropy	CPU time in s
Cell cycle ($k = 45$)	UFSFS	0.244304	0.0883736	5.575267	0.155386	0.6905
	FRGNN(8)	0.243300	0.082622	5.297885	0.145353	60.7200
Yeast complex ($k = 40$)	UFSFS	0.244290	0.095343	3.473932	0.138780	0.6136
	FRGNN(8)	0.243914	0.109309	3.349231	0.126026	30.0960
All Yeast ($k = 40$)	UFSFS	0.244092	0.061760	17.747542	0.085130	2.3817
	FRGNN(8)	0.243497	0.068388	10.903942	0.109613	586.8830

0.9564, 0.8473, 0.8192, 0.7685 & 0.8589, using K-NN. In contrast, the second best average values of F-score for Wisconsin breast cancer and waveform data are obtained using Relief-F (0.9539 & 0.8470, respectively), and for spam base and ionosphere data are obtained using UFSFS (0.8038 and 0.7450, respectively). The value of the F-score for multiple features data is the second best (0.8290) for SFS. Similar observations can also be found for FRGNN, using the results of Naive Bayes. Hence, we can say that the performance of FRGNN is more significant compared to the remaining algorithms.

Microarray gene expression data: To study the importance of the selected features of microarray gene expression data, we used the entropy measure and also clustered the genes of microarray data, based on the selected features, by using a self-organizing map. The clustering solutions are then evaluated based on the β-index, DB-index and fuzzy rough entropy.

Comparisons between the FRGNN and UFSFS for Cell Cycle, Yeast Complex and All Yeast microarray gene expression data are shown in Table 10. It may be noted that, Relief-F and SFS can be applied on datasets, where, each pattern belongs to a single class only and these methods cannot be applied on microarray gene expression datasets as each gene belongs to multiple functional categories according to Munich Information for Protein Sequences (MIPS).

For Cell Cycle data, the values of entropy, DB-index and fuzzy rough entropy are seen to be lower for FRGNN than those of UFSFS. The value of β-index for UFSFS is higher than FRGNN. This means, for Cell Cycle data, FRGNN is superior to UFSFS in terms of entropy, DB-index and fuzzy rough entropy, whereas the converse is true only in terms of β-index.

For All Yeast data, on the other hand, FRGNN is seen to perform better in terms of entropy, β-index and DB-index except fuzzy rough entropy.

Interestingly, for Yeast Complex data, FRGNN is superior to UFSFS in terms entropy, β-index, DB-index and fuzzy rough entropy. While the overall performance of the FRGNN has an edge over UFSFS, it requires more CPU time as is evident from Table 10.

Mean square error: The mean square error of FRGNN for iris data is shown in Fig. 17. It can be observed that the error value is 0.09 at zeroth iteration and is gradually decreased with the increase in the number of iterations. Finally, FRGNN converges to a minimum value at 2000th iteration for iris data. It can also be observed that the results of FRGNN for iris data in Table 10 are supported by the error value of FRGNN as shown in Fig. 17.

The mean square error of FRGNN for waveform data is shown in Fig. 18. The mean square error is initially observed to be 0.00863 at zeroth iteration and is seen to be 0.0064 after few iterations. Thereafter, this value remains the same for FRGNN for all the

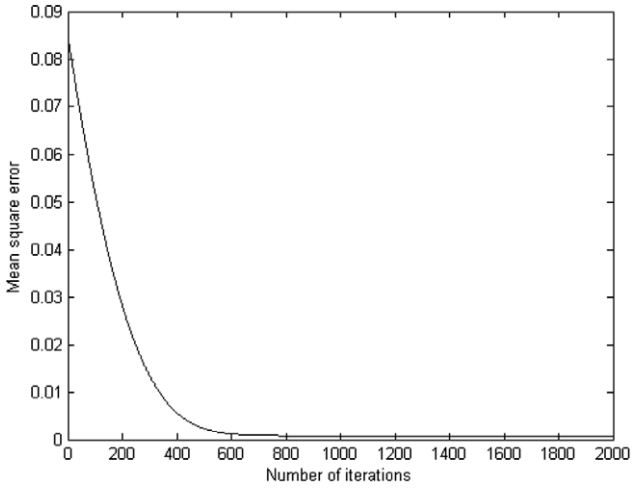

Fig. 17. Variation of mean square error with number of iterations of FRGNN for iris data.

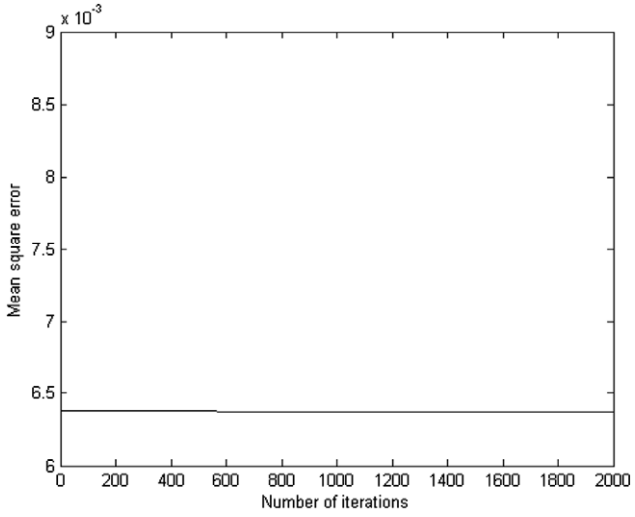

Fig. 18. Variation of mean square error with number of iterations of FRGNN for waveform data.

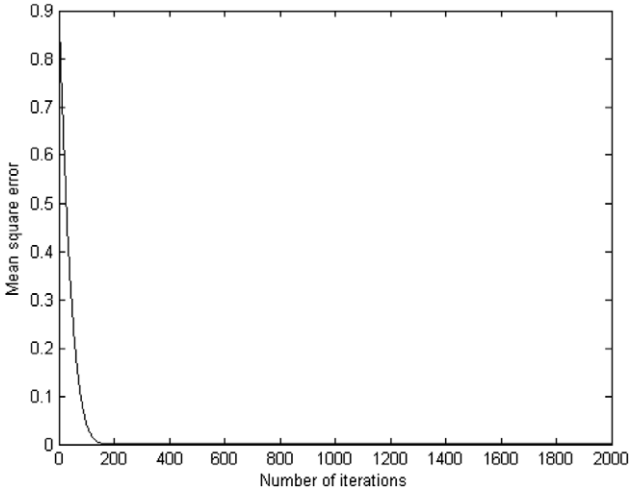

Fig. 19. Variation of mean square error with number of iterations of FRGNN for spam base data.

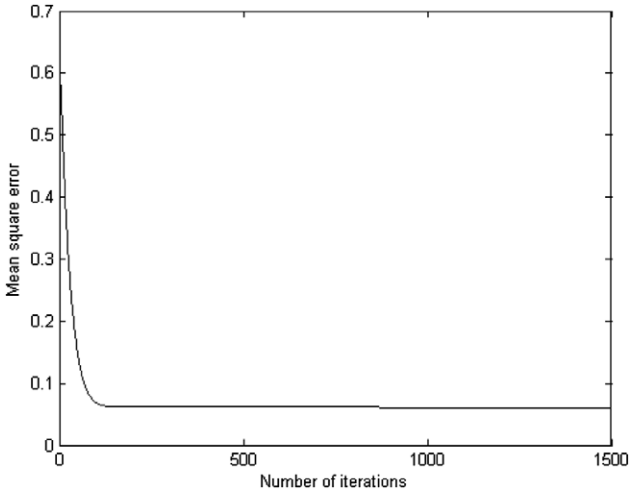

Fig. 20. Variation of mean square error with number of iterations of FRGNN for multiple features data.

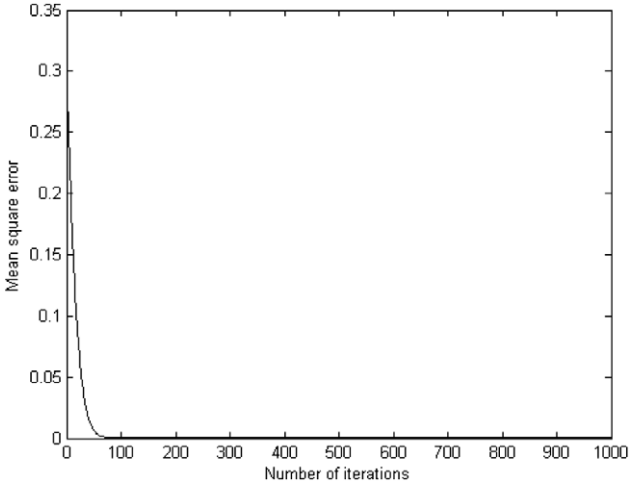

Fig. 21. Variation of mean square error with number of iterations of FRGNN for arrhythmia data.

iterations. Figs. 19–21 show variation of mean square errors with number of iterations of FRGNN for spam base, multiple features and arrhythmia data, respectively. The error values support the results related to superior performance of FRGNN, as compared to algorithms like UFSFS, Relief-F and SFS.

For Cell Cycle, Yeast Complex, and All Yeast microarray gene expression data, FRGNN converses to the minimum values at 3000 iterations as shown in Figs. 22–24, respectively. Similar observations can also be made for the remaining datasets.

6.3. Statistical analysis of feature selection algorithms using t-test

The performance of FRGNN and the other related methods is analyzed statically using a paired t-test. It is performed with classification accuracies for two feature selection methods and a single classifier, at a time, for all values of k and the results of 10-fold cross validation in a particular dataset using

$$t = \frac{\bar{M_1} - \bar{M_2}}{\sqrt{\frac{V_1}{n_1} + \frac{V_2}{n_2}}}, \tag{51}$$

where, $\bar{M_1}$ & V_1 represent the mean and variance of all classification accuracies of FRGNN for a particular dataset and a single classifier, and $\bar{M_2}$ & V_2 denote the same variables for any other feature selection method (UFSFS, SFS or Relief-F). The alternative hypothesis (H_1) that "the percentages of accuracies of FRGNN are better than

Table 11
The results of 10-fold cross validation corresponding to $k = 2$, 4 and 6 for all the feature selection methods for Wisconsin breast cancer data using K-NN.

		Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
FRGNN	$k = 2$	93.95	92.81	94.77	91.67	93.95	95.59	94.61	91.01	93.14	95.10
	$k = 4$	96.08	95.92	96.57	96.73	94.93	95.42	95.59	95.42	95.1 0	95.75
	$k = 6$	95.42	96.08	96.57	95.75	96.24	96.08	96.08	96.41	96.24	95.75
UFSFS	$k = 2$	91.50	94.77	94.28	94.44	93.14	94.77	92.32	94.77	94.28	92.97
	$k = 4$	92.65	92.48	93.79	95.59	94.44	94.44	94.28	93.79	94.44	95.42
	$k = 6$	94.42	95.59	96.08	95.10	96.73	96.08	95.75	95.92	95.59	94.92
SFS	$k = 2$	92.65	93.14	94.77	94.12	92.81	94.93	93.30	93.95	92.65	93.14
	$k = 4$	94.03	95.10	94.44	95.59	93.79	94.61	94.93	94.77	95.10	92.03
	$k = 6$	95.92	95.92	95.42	94.77	96.41	96.90	95.92	94.93	96.08	95.25
Relief-F	$k = 2$	93.12	93.79	95.59	94.61	93.30	92.65	93.14	94.12	94.77	91.03
	$k = 4$	96.24	95.42	95.59	95.10	95.26	95.42	95.59	95.92	95.42	96.41
	$k = 6$	95.92	96.57	94.28	95.10	95.59	96.08	96.41	96.24	95.75	95.92

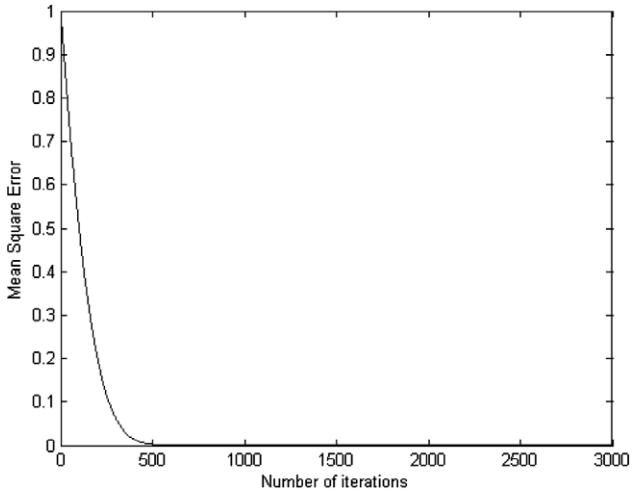

Fig. 22. Variation of mean square error with number of iterations of FRGNN for Cell Cycle data.

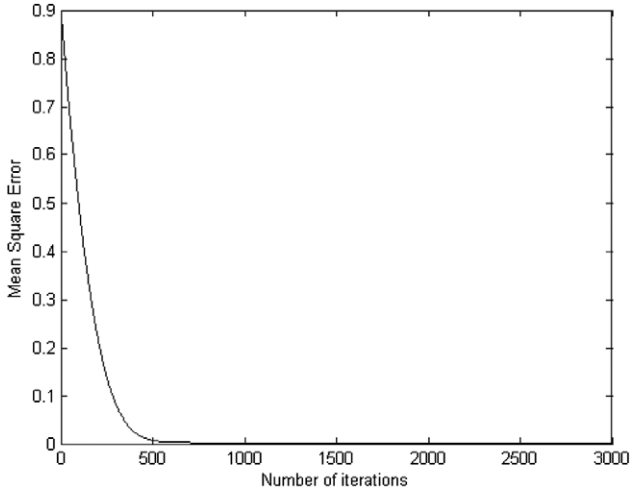

Fig. 23. Variation of mean square error with number of iterations of FRGNN for Yeast Complex data.

UFSFS, SFS or Relief-F" is used in the calculation of t-statistics. The p-values corresponding to t-values are obtained using the t-table and we either reject or accept the null hypothesis that "there is no difference between the percentages of accuracy for two related methods", with a particular significance level.

The percentages of accuracy, for each of the feature selection algorithms, for all values k and all 10-fold cross validations are shown in Table 11 for Wisconsin breast cancer data using K-NN, as

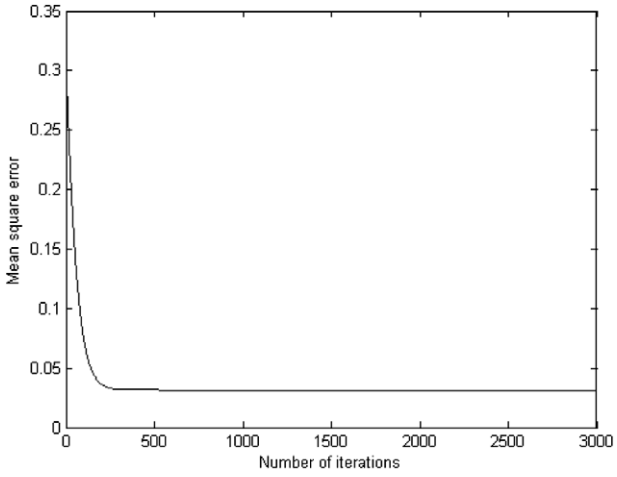

Fig. 24. Variation of mean square error with number of iterations of FRGNN for All Yeast data.

an example. Here each row provides the percentages of accuracy of 10-fold for a particular value of k. The t- tests (using Eq. (51)) for the pairs FRGNN & UFSFS, FRGNN & SFS and FRGNN & Relief-F are then performed using the results in the table. The t-value for the pair FRGNN & UFSFS is found to be 1.9301 for K-NN. The p-value, corresponding to this t-value, is seen to be 0.0292. Similarly, the t-values for the pairs FRGNN & SFS and FRGNN & Relief-F using K-NN are 1.6871 and 0.4138, respectively. The related p-values for the same pairs are 0.0485 and 0.3403. Hence, by observing the p-values, the null hypothesis that "there is no difference between the percentages of accuracy for the two methods" is rejected for all the pairs with significance levels 0.0292, 0.0485 and 0.3403, which suggests that there is strong evidence against the null hypothesis, in favor of the alternative for the first two cases. Note that, although both SFS and Relief-F are supervised feature selection methods, the results of FRGNN, an unsupervised method, are still more significant than SFS. However, the significance level is much lower for the pair FRGNN & Relief-F. Here, the degrees of freedom (df) are $30 \times 2 - 2 = 58$ as there are 30 results for every algorithm (either FRGNN or UFSFS) using K-NN.

Similar t-tests for the same pairs of feature selection algorithms are performed, with percentages of accuracy for all values of k and all results of 10-fold cross validation, for remaining datasets using K-NN and Naive Bayes. The t-values and related p-values for all the datasets are shown in Table 12. The degrees of freedom (df) for every data are provided within parenthesis in the table.

From Table 12, we can observe that the p-values for the pair FRGNN & UFSFS for breast cancer, waveform, spam base, ionosphere, multiple features, arrhythmia and secom datasets are 0.0292, 0.000002, 0.0067, 0.0395, 0.0019, 0.0013 and 0.0477,

Table 12
Results of t-tests for different pairs of feature selection algorithms for all the datasets for K-NN and Naive Bayes.

Dataset	Pair of algorithms	K-NN	Naive Bayes
Iris ($df = 18$)	FRGNN & SFS	$t = 3.0966, p = 0.0031$	$t = 5.5525, p = 0.00001$
Breast cancer data ($df = 58$)	FRGNN & UFSFS	$t = 1.9301, p = 0.0292,$	$t = 2.4319, p = 0.0091$
	FRGNN & SFS	$t = 1.6871, p = 0.0485,$	$t = 1.6895, p = 0.0483$
	FRGNN & Relief-F	$t = 0.4138, p = 0.3403,$	$t = 0.5145, p = 0.3044$
Waveform data ($df = 118$)	FRGNN & UFSFS	$t = 4.7465, p = 0.000002,$	$t = 3.5966, p = 0.0002$
	FRGNN & SFS	$t = 1.9198, p = 0.0285$	$t = 1.6595, p = 0.0498$
	FRGNN & Relief-F	$t = 0.4098, p = 0.3413$	$t = 1.0128, p = 0.1565$
Spam base ($df = 98$)	FRGNN & UFSFS	$t = 2.5195, p = 0.0067,$	$t = 1.6833, p = 0.0478$
	FRGNN & SFS	$t = 1.7145, p = 0.0448,$	$t = 1.6661, p = 0.0494$
	FRGNN & Relief-F	$t = 7.6920, p = 0,$	$t = 1.7574, p = 0.0410$
Ionosphere ($df = 58$)	FRGNN & UFSFS	$t = 1.7883, p = 0.0395,$	$t = 1.6944, p = 0.0478,$
	FRGNN & SFS	$t = 1.3104, p = 0.0976,$	$t = 0.9003, p = 0.1858$
	FRGNN & Relief-F	$t = 3.1066, p = 0.0015,$	$t = 1.8127, p = 0.0375$
Multiple features ($df = 118$)	FRGNN & UFSFS	$t = 2.9504, p = 0.0019,$	$t = 1.3488, p = 0.0900$
	FRGNN & SFS	$t = 0.4314, p = 0.3335,$	$t = 0.6509, p = 0.2582$
	FRGNN & Relief-F	$t = 1.7856, p = 0.0384,$	$t = 2.5719, p = 0.0057$
Arrhythmia ($df = 58$)	FRGNN & UFSFS	$t = 3.4889, p = 0.0013$	$t = 3.6145, p = 0.0003$
	FRGNN & Relief-F	$t = 0.7011, p = 0.2430$	$t = 1.9728, p = 0.0266$
Secom ($df = 58$)	FRGNN & UFSFS	$t = 1.6951, p = 0.0477,$	$t = 1.6033, p = 0.0572$
	FRGNN & Relief-F	$t = 1.5025, p = 0.0692,$	$t = 1.8383, p = 0.0369$

respectively, using K-NN. These p-values indicate that there is strong evidence, against the null hypothesis, in favor of the alternative hypothesis (H_1) (favoring FRGNN over UFSFS) for all the cases. Using Naive Bayes, the p-values are less than 0.05 for the same pair and most of the data sets except for multiple features and secom data, where, the p-values are 0.0900 and 0.0572, respectively. Hence, out of 14 instances (7 for K-NN and 7 for Naive Bayes), the results of FRGNN are statistically more significant than UFSFS in 12 instances.

After observing the p-values corresponding to t-values, for the pair FRGNN & SFS form Table 12, for iris, breast cancer, waveform and spam base, we reject the null hypothesis in favor of the alternative hypothesis (H_1) that the percentages of accuracies of FRGNN are better than SFS with significance levels 0.00031, 0.0485, 0.0285 and 0.0448, respectively for K-NN. Whereas, using Naive Bayes, the null hypothesis is also rejected in favor of alternative hypothesis (H_1) for the same datasets with the significance levels 0.00001, 0.0483, 0.0498 and 0.0494. By observing the p-values for the same pair, using ionosphere and multiple features data, we accept the null hypothesis, instead of the alternative, for K-NN and Naive Bayes. Hence, out of 12 instances (6 for K-NN and 6 for Naive Bayes), the results of FRGNN are statistically more significant than SFS in 8 instances. The p-values for the pairs FRGNN & UFSFS and FRGNN & Relief-F, for iris data, are not provided in Table 12 as FRGNN, UFSFS and Relief-F select the same features for iris data. Also note that, since SFS does not work for selecting features for arrhythmia and secom data, the t-values and related p-values for these datasets are not provided in the table.

Similarly, for the pair FRGNN & Relief-F, we reject the null hypothesis in favor of the alternative hypothesis (H_1) that the percentages of accuracy of FRGNN are better than Relief-F for spam, ionosphere and multiple features data, using k-NN, with the significance levels 0, 0.0015 and 0.0384, respectively. The null hypothesis is also rejected in favor of the alternative hypothesis (H_1) for spam, ionosphere and multiple features, arrhythmia and secom data, using Naive Bayes, with significance levels 0.0410, 0.0375, 0.0057, 0.0266 and 0.0369, respectively. But, we accept the null hypothesis, instead of the alternative hypothesis (H_1), for the breast cancer, waveform, arrhythmia and secom datasets using K-NN and for breast cancer and waveform datasets using Naive Bayes. Hence, out 14 instances (7 for K-NN and 7 for Naive Bayes), the results of FRGNN are statistically more significant than Relief-F in 8 instances. It may be noted that, since the SFS and Relief-F are supervised feature selection algorithms, the t-tests for the pairs FRGNN & SFS and FRGNN & Relief-F for some of the datasets favor the null hypothesis rather than the alternative. In brief, the results of FRGNN are found to be statistically significant in 28 instances out of 40 instances, i.e., 70% of the instances.

7. Conclusion

In this article, new notions of lower and upper approximations of a fuzzy rough set are proposed by using fuzzy logical operators, where, the conditional attributes and decision attributes are defined in terms of fuzzy membership values. Here, granules are developed by the concepts of fuzzy set and fuzzy rough set. The network (FRGNN) integrates granular computing and neural networks, two different components of natural computing, in a soft computing framework. The granules shown in terms of granulation structures are based on the concept of fuzzy sets. These structures are also used to determine a decision system and to extract domain knowledge about data. The domain knowledge, defined in terms of dependency factors and representing the fuzzy rules, is incorporated into a three layered network as initial connection weights. The fuzzy rules can be represented as granules. The network is trained through minimization of the proposed feature evaluation index in an unsupervised manner. After training is completed, the importance of individual features are obtained from the updated weights between the nodes of the hidden layer and output layer of FRGNN. A higher weight corresponding to a feature indicates that, the feature is more important for selection.

The performance of FRGNN, in terms of average percentage of classification accuracies of K-NN & Naive Bayes and the value of entropy, is found to be superior to UFSFS, SFS and Relief-F. These are shown on real life datasets with different dimensions, ranging from 3 to 649. The variation in the mean square error of FRGNN with the number of iterations also supports the superior performance of the FRGNN in feature selection task to the other algorithms, like UFSFS, SFS and Relief-F. Superior results of FRGNN as compared to UFSFS, in terms of β-index, DB-index and fuzzy rough entropy, are shown for the data sets, like microarrays with different dimensions, ranging from 79 to 93.

The weakness of the proposed FRGNN is that it takes higher computational time than the related methods for feature selection tasks. We also like to mention that the selection of α resulting in the number of nodes in the input layer with the granulation structures is crucial.

The proposed FRGNN can be recommended for feature selection when the data has no class information (class labels), has uncertainty arising in the feature space and has a large number of features. Although, both FRGNN and UFSFS are unsupervised feature selection algorithms, the performance of FRGNN is superior to UFSFS for all the datasets because the concepts of fuzzy set and fuzzy rough set are used in the proposed algorithm. Methods, like Relief-F and SFS, can be used, when the data has class labels as these are supervised feature selection algorithms. But still the t-test results show that the accuracies of FRGNN are more significant than SFS in 8 instances out of 12 and Relief-F in 8 instances out of 14, whereas, FRGNN outperformed UFSFS (also an unsupervised method) in 12 instances out of 14. In brief, the p-values related to t-tests yield that the results of FRGNN are statistically significant in 70% of the instances.

Acknowledgment

S. K. Pal acknowledges the J. C. Bose Fellowship of the Government of India.

References

Bar-Joseph, Z., Gifford, D. K., & Jaakkola, T. S. (2001). Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, *17*(1), 22–29.

Blake, C.L., & Merz, C.J. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/mlearn/MLrepository.html.

Cornelis, C., Jensen, R., Hurtado, G., & Slezak, D. (2010). Attribute selection with fuzzy decision reducts. *Information Sciences*, *180*(2), 209–224.

Davies, D. L., & Bouldin, D. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis Machine Intelligence*, *1*(2), 224–227.

Devijver, P. A., & Kittler, J. (1982). *Pattern recognition: a statistical approach*. New Jersey: Prentice Hall.

Dick, S., & Kandel, A. (2001). Granular computing in neural networks. In W. Pedrycz (Ed.), *Granular computing: an emerging paradigm*. Heidelberg: Physica-Verlag.

Dubois, D., & Prade, H. (1990). Rough fuzzy sets and fuzzy rough sets. *International Journal of General Systems*, *17*(2–3), 191–209.

Dubois, D., & Prade, H. (1992). Putting fuzzy sets and rough sets together. In R. Slowiniski (Ed.), *Intelligent decision support*. Dordrecht: Kluwer Academic.

Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of National Academy of Sciences of the United States of America*, *95*(25), 14863–14868.

Ganivada, A., Dutta, S., & Pal, S. K. (2011). Fuzzy rough granular neural networks, fuzzy granules, and classification. *Theoretical Computer Science*, *412*(42), 5834–5853.

Ganivada, A., Ray, S. S., & Pal, S. K. (2012). Fuzzy rough granular self-organizing map and fuzzy rough entropy. *Theoretical Computer Science*, *466*, 37–63.

Ghosh, A., Shankar, B. U., & Meher, S. K. (2009). A novel approach to neuro-fuzzy classification. *Neural Networks*, *22*(1), 100–109.

Herbert, J. P., & Yao, J. T. (2009). A granular computing frame work for self-organizing maps. *Neurocomputing*, *72*(13), 2865–2872.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In *9th international conference on machine learning* (pp. 249–256). Morgan Kaufmann.

Mitra, P. (2002). http://www.facweb.iitkgp.ernet.in/pabitra/paper.html.

Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(3), 301–312.

Pal, S. K., De, R. K., & Basak, J. (2000). Unsupervised feature evaluation: a neuro-fuzzy approach. *IEEE Transactions on Neural Networks*, *11*(2), 366–376.

Pal, S. K., & Dutta Majumder, D. (1977). Fuzzy sets and decision making approaches in vowel and speaker recognition. *IEEE Transactions on Systems, Man and Cybernetics*, *7*(8), 625–629.

Pal, S. K., Ghosh, A., & Shankar, B. U. (2000). Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation. *International Journal of Remote Sensing*, *21*(11), 2269–2300.

Pal, S. K., & Mitra, S. (1999). *Neuro-fuzzy pattern recognition: methods in soft computing*. New York: John Wiley.

Pawlak, Z., & Skowron, A. (2007). Rudiments of rough sets. *Information Sciences*, *177*(1), 3–27.

Radzikowska, A. M., & Kerre, E. E. (2002). A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems*, *126*(2), 137–155.

Ray, S. S., Bandyopadhyay, S., & Pal, S. K. (2007). Dynamic range based distance measure for microarray expressions and a fast gene ordering algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *37*(3), 742–749.

Ray, S. S., Bandyopadhyay, S., & Pal, S. K. (2012). A weighted power framework for integrating multi-source information: gene function prediction in yeast. *IEEE Transactions on Biomedical Engineering*, *59*(4), 1162–1168.

Salton, G., & McGill, M. J. (1983). *An introduction to modern information retrieval*. New York: McGrawHill.

Sen, D., & Pal, S. K. (2009). Generalized rough sets, entropy, and image ambiguity measures. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *39*(1), 117–128.

Szczuka, M. (2001). Refining classifiers with neural networks. *International Journal of Computer and Information Sciences*, *16*(1), 39–55.

Verikas, A., & Bacauskiene, M. (2002). Feature selection with neural networks. *Pattern Recognition Letters*, *23*(11), 1323–1335.

Zadeh, L. A. (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, *90*(2), 111–127.

Zhang, Y. Q., Jin, B., & Tang, Y. (2008). Granular neural networks with evolutionary interval learning. *IEEE Transactions on Fuzzy Systems*, *16*(2), 309–319.