An Efficient Classification Model using Fuzzy Rough Set Theory and Random Weight Neural Network

Ramsha Javed[1], Rana Aamir Raza[2], Maruf Pasha[3], Asif Yaseen[4], and Amjad Ali[5]
[1,2]Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan
[3,5]Department of Information Technology, Bahauddin Zakariya University, Multan, Pakistan
[4]Department of Commerce, Bahauddin Zakariya University, Multan, Pakistan
Email: aamir@bzu.edu.pk

(Received: 19 Aug 2021; Accepted:11 Sep 2021; Issue Published: 12 Sep 2021)

ABSTRACT

In the area of fuzzy rough set theory (FRST), researchers have gained much interest in handling the high-dimensional data. Rough set theory (RST) is one of the important tools used to pre-process the data and helps to obtain a better predictive model, but in RST, the process of discretization may loss useful information. Therefore, fuzzy rough set theory contributes well with the real-valued data. In this paper, an efficient technique is presented based on Fuzzy rough set theory (FRST) to pre-process the large-scale data sets to increase the efficacy of the predictive model. Therefore, a fuzzy rough set-based feature selection (FRSFS) technique is associated with a Random weight neural network (RWNN) classifier to obtain the better generalization ability. Results on different dataset show that the proposed technique performs well and provides better speed and accuracy when compared by associating FRSFS with other machine learning classifiers (i.e., KNN, Naive Bayes, SVM, decision tree and backpropagation neural network).

Keywords: Fuzzy Set, Random Neural Network, Rough Set Theory, Feature Selection**.**

1. INTRODUCTION

In machine learning and pattern recognition, many real world problems require supervised learning (SL)to build a better classification model. In SL, class probabilities and class-conditional probabilities are unknown, and the instances that participate in the classification process are associated with label (i.e., class). In the classification process, significant features or attributes are often unknown *a priori* [1], and many existing features, either partially or completely irrelevant and redundant to the target concept (i.e., class). According to [2], relevant features are neither irrelevant nor redundant to the target concept and contribute well during the classifier's training, but the irrelevant features never affect the target concept, such features do not contribute anything new to the target concept. To solve these issues, researchers have been conducted many experiments to extract important features from the data [3], [4]. For a better classification model, feature selection becomes prominent, especially in the data sets with majority of features, this process eliminates unimportant features and improves the generalization ability of a classifier. Therefore, feature dimensionality reduction (FDR) is considered one the important problems in the domain of pattern recognition, this technique consists of two approaches, i.e., feature selection and feature extraction [5]. The feature selection process helps determine and filter the redundant or irrelevant features from data [6], which can be discarded to optimize the generalization ability of a classifier. The feature selection process (i.e., features or attributes reduction), can be viewed as one of the most important techniques in RST, but it is hard to handle the hybrid attributes with this traditional theory of rough set [7]. One way to solve this problem is a discretization of the such attributes but this process may lead to the information loss. Another approach is to use the concept of fuzzy rough set (FRST), which encapsulates the distinct concepts of indiscernibility and fuzziness[8], [9].In the process of feature selection, many methods have been proposed to deal high dimensional data, therefore, an effective and efficient reduction method is required that maintain the original meaning of the data after removing the irrelevant features. In literature [10]–[13], many theories have been proposed to minimize unnecessary and unwanted features. Recently, data reduction has become a topic of interest for researchers and many techniques and methodologies have been proposed to deal with imprecision and uncertainty in the dataset. RST applied to many domains e.g., classification, medical science, system monitoring, clustering, text classification, expert system etc., and it is a considered as successful approach due to the three main reasons; firstly, it only analyzes the hidden facts in dataset, secondly no additional information required to analyze the data such as threshold or expert knowledge about specific domain and thirdly, it gets a minimum representation of the knowledge for datasets.

Using RST, it is only possible to find a features subset of the original dataset with discretized features, hence the features with minimal information are removed from the dataset and the most predictive features are selected as a feature subset. Traditional RST encounters a problem because sometimes we have to work with continuous or real-valued features. Therefore, fuzzy set theory combines with rough set theory to get the useful features from the large data. Hence, it is necessary to develop techniques for real-valued feature. Therefore, it is necessary to develop techniques for real valued feature dataset to provide the data reduction on the basis of the concept of values similarity. This can be achieved by using the fuzzy rough set theory (FRST). As discussed above, FRST allow to perform data analysis on dataset with real valued datasets directly. The foundation of RST is sited in the late 1980s [14], this theory presented a new mathematical approach for data reduction and data analysis. This theory is extended to fuzzy rough set theory to directly deal with real valued datasets [15], [16]. Both theories can tolerate the uncertainty in the datasets but the difference between these theories is indiscernibility. That's why it is useful to hybridize these concepts to deal with the uncertainty and imperfection of the data [17]. In this paper, we proposed a feature reduction (FR) algorithm based on fuzzy rough set feature selection (FRSFS) method, which applies FRST on a real-valued noisy dataset and effectively removed the conflicting and irrelevant features from the dataset and selected the better subset of features that will maintain the original meaning of features. The reduced feature subset will feed to the different machine learning classifiers and the accuracy and computational time is obtained to measure the generalization ability of a classifier.

This paper is organized as follows: an overview of the FRST based feature selection methods, and the foundation of rough set and fuzzy rough set theories are presented in section 2. Section 3 describes the random weight neural network (RWNN). In section 4, the proposed technique based on FRSFS and RWNN is presented. Experimental analysis and discussions are presented in section 5. Finally, section 6 provides the concluding remarks and future work.

2. FUZZY ROUGH SET BASED FEATURE SELECTION METHODS

Ludmila [18] was the pioneer who first proposed feature selection using fuzzy rough set (FRS). In [18], author considered the problem of evaluating the *hypoxic resistance*of a patient on the basis of his blood pressure values during a *barocamera examination*. Authors in [19]introduced a formal concept of fuzzy rough attribute reduction and applied to the problem of web categorization. Their results showed considerable The foundation of RST is established in the late 1980s [14], this theory presents a new mathematical approach for data reduction and data analysis, and also deals with

reduction of dimensionality withminimal information loss. Feature selection methods have been applied to any sized dataset to find out the most relevant and informative features for later use [20], [21].In the literatures, many feature reduction techniques in FRS environment have been discussed. Many FRS techniques used the concept of degree of dependency-based feature selection; where the features are selected on the basis of dependency of decision feature over the set of conditional features [22]. A feature is an attribute or characteristics of an object its quality is very important becausethe accuracy of a system can be improved by increasing the quality of the features [23]. The basic idea behind the feature selection is to select a set of features from the specified datasets that can best define the whole data after minimizing the effects of redundant and irrelevant features resulting better predictive model [24], [25]. Many researches have been conducted for the real time application using FRST. Recently, authors in [26], used fuzzy rough sets theory (FRST) for prototype selection to enhance SVM in intrusion detection system (IDS). Their experimental results show that the proposed technique provides better generalization in terms of precision, recall, and accuracy rate.

2.1. Criteria for Feature Selection

The context of the feature selection is to find out the useful subset of features that makes the feature process of selection meaningful and useful. There are many feature selection criteria, some of which are discussed below. Dependency defines the connection between features, its means how strongly features are linked or associated with each other. It specifies that to what extent the values of features are uniquely determines the values of other features. In Supervised Machine Learning, we can represent the dependency of the label of the C on the features X and Y. If $A(X)$ and $A(Y)$ is dependent on class C on feature X and Y and if the dependency degree of feature X is greater than the feature Y such as $A(X) > A(Y)$ then feature X will be preferred over feature Y. And the other one is classification accuracy. Classification accuracy is one of the feature selection criteria and it is fully dependent on the machine learning classification algorithm. The context of the classification accuracy is to find out the useful features that give the better training time and classification accuracy. The drawback of this approach is avoiding overfitting and may lead to incorrect accuracy due to noise and redundancy in data.

2.2. Rough Set Theory (RST)

uncertainty, ambiguity and vagueness in dataset [14], [25]. This theory doesn't require any additional parameter other than the dataset and this is the main advantage of this theory. It doesn’t mean that this theory makes no model assumption; in fact, it assumes assumptions on the given dataset because it assumes that the data is accurate. This is the major difference between fuzzy set theory and Dempster–Shafer theory [27].

2.3. ROUGH SET FEATURE SELECTION

Let’s consider $I =< U, C, D >$ be a decision system where universal set is represented by, the set of conditional attributes is represented by C and decision set of an attribute is represented by D. Each feature or attribute $\alpha \grave{o} A$ is related with a set V_a and its value is called a domain of a. We can divide the feature set into the subsets called conditional feature C and decisional feature D. Let $R \subset A$ be a features subset, the equivalence relation of indiscernibility relation is represented by $IND(R)$ and can defined as Equation (1)

$$IND(R) = \{(s,t) \in U \times U : \forall a \in R, a(s) = a(t)\} \quad (1)$$

In Equation 1, $a(s)$ represents the value of features a of object s. If $(s,t) \in IND(R)$, where s and t are said to be indiscernible with respect to R. The family of all equivalence classes of $IND(R)$ is denoted by $\frac{U}{IND(R)}$.

In [14], author presented two approximations of subset that are lower approximation (LA) and upper approximation (UA). Lower approximation contains those objects that are definitely in set S defined as (2).

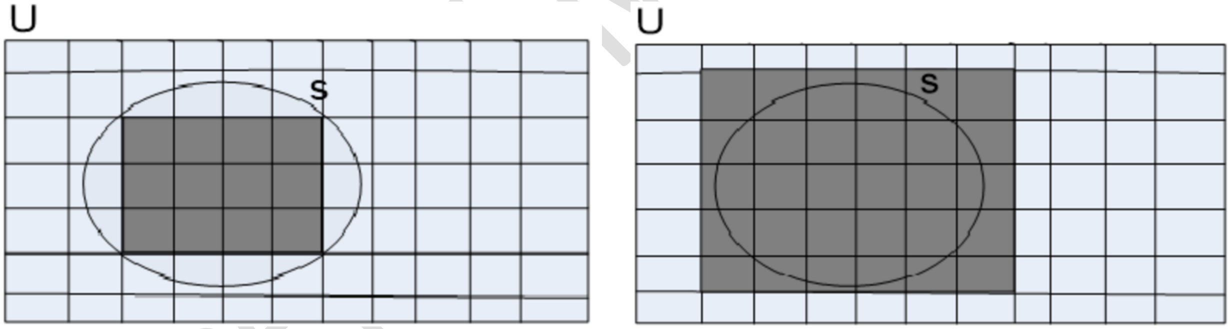
$$\underline{R}(s) = \cup\left\{E \in \frac{U}{IND(R)} : E \subseteq A\right\} \quad (2)$$

The lower approximation of RST is shown in Figure 1 (b). The UA of the rough set theory contains those elements that possibly belong to x and it can be represented as (3).

$$\bar{R}(s) = \cup\left\{E \in \frac{U}{IND(R)} : E \cap A \neq \varphi\right\} \quad (3)$$

The Upper approximation of RST is shown in Figure 1 (a). The boundary region of *S* can be constructed by using lower approximation and upper approximation. It contains the elements of U that are not surely inside or outside *S*. It can be defined by the difference of the upper approximation and lower approximation as presented in (4).

$$BND_R(s) = \bar{R}(s) - \underline{R}(s) \quad (4)$$



(a) **(b)**

Figure 1: Lower and upper approximation

R and D are set of features of equivalence relation over universal set U. The positive region of decision class U/ IND(D) with respect to conditional feature R is defined as (5).

$$POS_R(D) = \cup\underline{R}(s) \quad (5)$$

Finding the dependency between two features is an important issue in analysis of data. Dependency define how distinctively the value of a features defines the value of other features. A feature D depends on other feature R by degree K (0 ≤k ≤1) and it can be shown as (6).

$$k = \gamma(R,D) = \frac{|POS_R(D)|}{|U|} \quad (6)$$

Reduction is a one way of taking only those features that reserve the indiscernibility relation. By using the selected number of features that can be achieved by using the complete feature set, it gives the same set of equivalence classes. The rest of the features are irrelevant and noisy. These features can be removed to enhance the predictive performance of a classifier. The reduced feature subset is called Reducts. It can be defined by using the dependency degree as mentioned in Equation (7).

$$\gamma(R,D) = \gamma(R',D) \ \mathit{for} \ \mathrm{R}' \subseteq C \quad (7)$$

An attribute set $R' \subseteq \mathrm{R}$ will be called Reduct with respect to *D*, if the dependency of D on R' will be same as that of its dependency on C. This theory has been used in many applications [7], [28], [29] like prediction of business failure, financial investment, Fault diagnosis, medicine, feature selection and many other applications [30]–[32].

2.4. Fuzzy rough set theory (FRST)

When we work with real valued datasets, it may be possible that features values are numeric and symbolic. In such situations, it will be challenging to find out the pattern by using the RST. If the feature values are symbolic and not numeric then fuzzy set theory is necessary to apply on the given dataset. In some cases, clustering of objects to the given features values might not be useful. The two objects may be very close to each other but in reality a minor difference between the feature's values may classify in different classes [33]. Another technique by performing discretization, replace the exact feature values by interval codes. Another solution is the use of tolerance rough set [31], if feature values are sufficiently close then it is considered indiscernible. These techniques are not suitable because of information loss. So, an alternative approach is considered that is fuzzy Rough sets [27], [30], [34], [35]. values objects of the universal sets are classified in the interval of [0,1]. Most dataset contained real-valued features so, there is need of a discretization step to perform as implemented by standard fuzzification techniques [29], [37], [38]. L. A. Zadeh [36] introduced a Fuzzy Set Theory to model concept of vagueness. Based on the feature

Fuzzy Equivalence Class is an essential part of FRST. In the literatures [5], [32], [38]–[40], Fuzzy Rough Set Theory (FRST) have fuzzy equivalence classes that is a main part of the FRST as the crisp equivalence classes are the main part of the rough set Theory. FRST can be defined by LA and UA based on fuzzy relationships. Each sample in the decision table may be correlated with many fuzzy equivalence class using a degree of membership function in the range of [0,1]. FRST must hold the following axioms as depicted in (8).

- $\exists x, \mu_F(x) = 1$
- $\mu_F(x) \wedge \mu_E(x,y) \le \mu_F(y)$ (8)
- $\mu_F(x) \wedge \mu_F(x) \le \mu_E(x,y)$

Whereas, the first state of the axioms is that fuzzy equivalence class is a nonempty, second state of the axiom is that the features in *y* neighborhood are in fuzzy equivalence class of y and last state of the axioms is that each two features in F are related on feature x [41]. This theory can be defined by using the lower approximation and upper approximation of RST on the basis of fuzzy relationship. The fuzzy lower approximation (FLA) and fuzzy upper approximation (FUA) are presented in (9) and (10) respectively.

$$\mu_{\underline{p}\mathrm{x}}(x) = \sup_{F \in U/P} \min\left(\mu_F(x), \mathop{inf}_{y \in U} max\{1 - \mu_F(y), \mu_x(y)\} \right) \qquad (9)$$

$$\mu_{\bar{P}\mathrm{x}}(x) = \sup_{F \in U/P} \min\left(\mu_F(x), \sup_{y \in U} \min\{\mu_F(y), \mu_x(y)\} \right) \qquad (10)$$

In Equation (9) and (10), F is the fuzzy equivalence class and x is the approximation of the fuzzy concept, $\mu_x(y)$ is the degree of membership function of object y to fuzzy subset x. Expand the FRST by using the min, max operators, where each object is represent by a implicator I and t-norm T [42]. Fuzzy lower approximation and fuzzy upper approximation is called the FRST. A minor difference between FRST and rough fuzzy set is that the approximation of the fuzzy set in crisp approximation space is called rough fuzzy set and the approximate the crisp set in fuzzy approximation space is called FRST [8]. Some researchers consider the fuzzy rough set as standard [41].

FRST based feature reduction is based on the FLA. The process is similar to the RST approach. Positive region in RST is defined as a union of lower approximations. A fuzzy positive region is presented in (11).

$$\mu_{POS\,p(Q)}(x) = \mathop{sup}_{X \in \cup/Q} \mu_{\underline{P}X}(x) \qquad (11)$$

The fuzzy rough dependency function can be defined by using fuzzy positive region as depicted in (12).

$$\mathrm{Y}'_p(Q) = \frac{|\mu_{POS\,p(Q)}(x)|}{|\cup|} = \sum x \in \cup \frac{\mu_{POS\,p(Q)}(x)}{|\cup|} \qquad (12)$$

In FRST, the dependency function is similar to RST, the dependency of Q to P is the part of the objects that are distinct out of the complete dataset.

There are some issues with FRST [43] which will explain later in the study. Sometimes fuzzy lower approximation becomes bigger than fuzzy upper approximation and fuzzy lower approximation might not be the subset of fuzzy upper approximation. This is not desirable as it suggest that upper approximation is more certain than lower approximation that is meaningless for FLA [37] and the complexity of calculating Cartesian product of fuzzy equivalence classes gets larger for large feature subset. A compact computational domain is proposed in [28] to reduce the computational effort required to calculate the fuzzy lower approximation for larger datasets.

Due to these issues, alternative approaches of Fuzzy lower approximation and upper approximation have been proposed in [5] and presented in (13) and (14) respectively.

$$\mu_{\overline{R_P}X}\left(x\right)=sup_{y\in U}T\left(\mu_{R_P}\left(x,y\right),\mu_X\left(y\right)\right) \qquad (13)$$

$$\mu_{\underline{R_p}X}\left(x\right)=inf_{y\in U}I\left(\mu_{R_P}\left(x,y\right),\mu X\left(y\right)\right) \qquad (14)$$

In Equations (13) and (14), *T* is the t-norm and *I* is the fuzzy implicator. R_p is the fuzzy similarity relation induced by the features P as presented in (15).

$$\mu_{R_P}\left(x,y\right)=\underset{a\in P}{\cup}\{\mu_{R_a}(x,y)\} \qquad (15)$$

where $\mu_{R_a}\left(x,y\right)$ is the degree to which object *x* and *y* are similar for feature *a*. Many similarity relations can be constructed but we use the similarity relation as used by [44] in (16).

$$\mu_{R_a}\left(x,y\right)=max\left(\min\left(\frac{\left(a\left(y\right)-\left(a\left(x\right)-\sigma_b\right)\right)}{\left(a\left(x\right)-\left(a\left(x\right)-\sigma_a\right)\right)},\frac{\left(\left(a\left(x\right)+\sigma_a\right)-a\left(y\right)\right)}{\left(\left(a\left(x\right)+\sigma_a\right)-a\left(x\right)\right)}\right),0\right) \qquad (16)$$

In Equation (16), σ_a is the variance of feature *a*. Fuzzy positive region can be defined as (17).

$$\mu_{POS_{R_p}(Q)}\left(s\right) = \sup_{S\in\bigcup/Q}\mu_{\underline{R_P}S}\left(s\right) \qquad (17)$$

Positive region also leads to defining the dependency function as shown in equation (18).

$$\Upsilon'_p(Q)=\frac{|\,\mu_{POS_{R\ p}(Q)}\left(s\right)|}{|\bigcup|} \qquad (18)$$

3. RANDOM WEIGHT NEURAL NETWORK

In traditional neural networks, computational complexity is high because backpropagation is iterative and time consuming to tune its parameters. In this way, a non-iterative methodology is needed to train neural network with low computational complexity. In 1992, Schmidt et al. [45] examined the effect of random initialization on the generalization performance of single layer feed forward neural network (SLFN). The results of the experiments show that SLFN can acquire better generalization ability after selecting the random weights of hidden layer nodes and input layer nodes and calculating the weights of output layer nodes rationally. This is the first study of selecting the weight of neural network (NN) randomly. The analysts also inferred that the output layer's weight is more significant than the weights of the hidden layer. Authors in [45] did not propose the name of the SLFN that’s why we may call it random weights neural network (RWNN) to recognize their work. The structure of RWNN is shown in Figure 2.

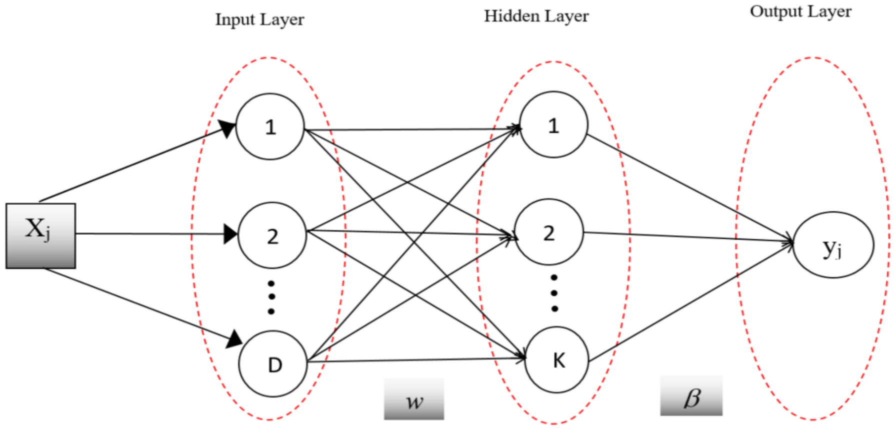


Figure 2: Structure of random weight neural network

The idea of randomization of the hidden layer weights has been proposed many times. Pao et al. in [46] explored the learning and generalization of the random vector functional link network (RVFLN), but its generalization performance was investigated in [47]. The approach used in this paper consist of formulating a limited interval representation of the function and afterward evaluating that integral with the Monte-Carlo method. In [48], authors used this technique for adjusting the weights of SLFN before training it with Back Propagation (BP). In each layer, the optimal initial weights are evaluated by using a least square method.

In the literatures [49]–[51], one can see that numerous concepts are presented for randomization of the weights and biases of the hidden layer. In RWNN, the biases of the hidden layer and weights of the input layer can be randomly selected and the weights of output layer and hidden layer can be determined analytically with Moore-Penrose generalized inverse. One can see in the literatures [35], [52], [53], [49] that many ideas have been presented for determining the random initialization of weight of the hidden layer in the NN where training was performed by using Pseudo-Inverse.

RWNN provides better training speed than backpropagation because it does not require iterative tuning parameters at hidden layer nodes. Conventional NN have great approximation ability but the behavior is heavily depending on training set during the training process. The boundaries generated by the neural network for classification are sometimes unpredictable in the presences of the small datasets.

The main idea of the RWNN is the randomization of the hidden layer weights and the subsequent training consist of the calculating the least square solution to the linear system defined by the targets and the outputs of the hidden layer.

$$X=\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{pmatrix} and, T=\begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$

SLFN with $\tilde{K}$ hidden nodes approximating N samples with zero-error means that there exist β_i, w_i and b_i where $i=1,\cdots,\tilde{K}$ such that

$$\sum_{i=1}^{\tilde{K}} \beta_i g\left(w_i, b_i, x_i\right) = t_j,\ j=1,\cdots,N \qquad (19)$$

Whereas

$$w=\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{\tilde{K}} \end{pmatrix}^T = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1D} \\ w_{21} & x_{22} & \cdots & w_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ w_{\tilde{K}1} & w_{\tilde{K}2} & \cdots & w_{\tilde{K}D} \end{pmatrix}^T$$ are input layer weights and $$b=\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{\tilde{K}} \end{pmatrix}$$ are the are the input layer biases.

Hereafter, we can write the equation (19) as:

$$H\beta = T \qquad (20)$$

In equation (20), H is the hidden layer output matrix.

$$H=\begin{pmatrix} g(w_1,b_1,x_1) & g(w_2,b_2,x_1) & \cdots & g(w_{\tilde{K}},b_{\tilde{K}},x_1) \\ g(w_1,b_1,x_2) & g(w_2,b_2,x_2) & \cdots & g(w_{\tilde{K}},b_{\tilde{K}},x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g(w_1,b_1,x_N) & g(w_2,b_2,x_N) & \cdots & g(w_{\tilde{K}},b_{\tilde{K}},x_N) \end{pmatrix}$$

, and $\beta=\begin{pmatrix}\beta_1 \\ \beta_2 \\ \vdots \\ \beta_{\tilde{K}}\end{pmatrix}^T$ is the weights of output (21)

layer, and $g(z)=\frac{1}{1+e^{-z}}$ is a sigmoid function.

A sigmoid function has a curve in two directions and resembles to the English letter *S*. This function transforms an input value to an output in the [0,1] interval. This function only returns the positive value if anyone need the neural network return the negative values than this function is unsuitable. In this way, the solution (21) turns into a system of linear equations and in most situations, it tends to be moved to a regular system of linear equation.

$$H^T H\beta = H^T T \qquad (22)$$

Let $H^T H$ is a non-singular, then according to the Equation (22), solution of the system can be presented as (23).

$$\beta=\left(H^T H\right)^{-1} H^T T \qquad (23)$$

RWNN compute the weights of the output layer β according to the Equation (23). In Random Weight Neural Network (RWNN), the weights of the input layer w_i and the hidden layer biases b_i are considered to be arbitrary variable, independent and can be fixed during experimental simulation

4. PROPOSED ALGORITHM FOR FEATURE SELECTION

The basic feature selection process is shown in the Figure 3. The Feature set is selected by using the fuzzy rough feature selection technique. The proposed classification model using feature selection technique is shown in the Figure 4. In this model, the first step is to pass the data to the FRSFS algorithm, FRSFS calculates the dependency using the lower approximation. Select the useful features that have high dependency degree and called a reduced dataset. This reduced dataset passes to the RWNN classifier and calculates the accuracy using different layer weights and biases. The reduced dataset also passes to the different classification algorithms, KNN, Naïve Bayes, Decision tree, SVM and neural network, and calculates the accuracy using the same parameters. For a given data set, define the equivalence classes with membership degree in the range of [0,1] to calculate the similarity relation of each feature. After creating the fuzzy equivalence class of each feature, compute the lower approximation of each object for each feature and each decision concept, then calculate the positive region for each object to calculate the dependency degree of each feature. Select feature having maximum dependency degree. The algorithm stops when we find the maximum dependency or equal to 1. Select the features subset that have maximum dependency make it a reduct subset. Feed this subset to the random weight neural network machine learning classifier. Initialize the nodes hidden layers. Compute the hidden layer output matrix and output weights . Our proposed algorithm is depicted in Table 1.

8.

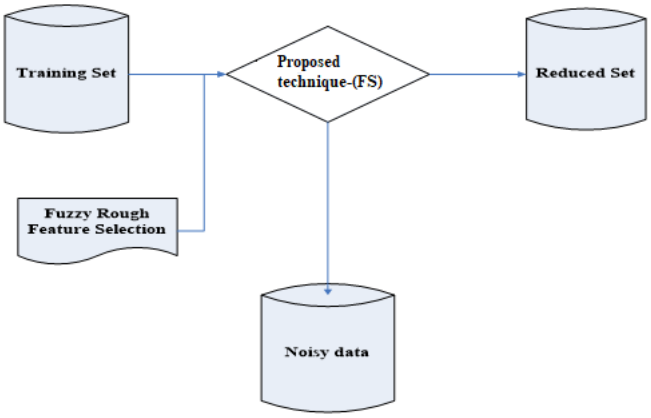


Figure 3: Feature selection process

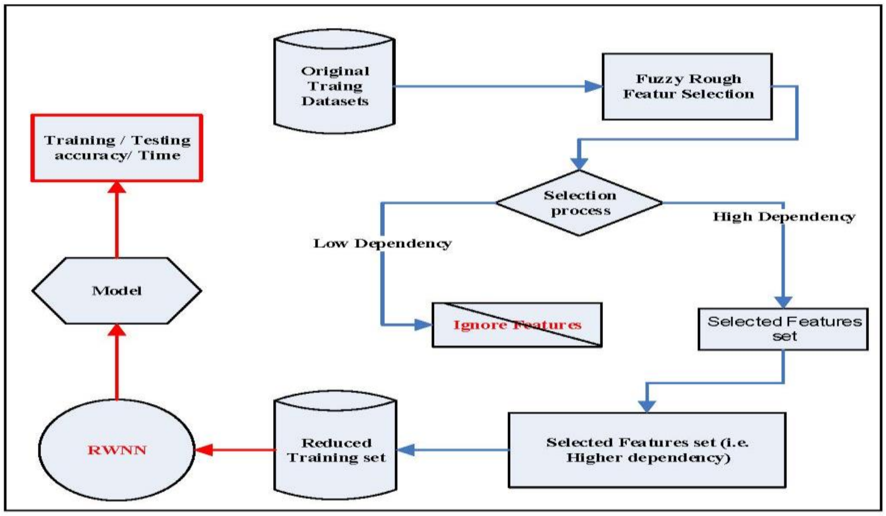


Figure 4: Proposed fuzzy rough set based classification model

Table 1. Proposed algorithm

Input: Data set D = (x_i , y_i |$1 \leq i \leq N$)

Output: Classification accuracy and training time

```
1.    Fuzzy equivalence class according to Equation (8)
2.    Calculate the similarity relation $\mu_{R_p}(s,t)$ w.r.t. Equation (16)
3.    Compute the FLA '$\mu_{\underline{R_p}s}(s)$' w.r.t. Equation (14)
4.    Compute the positive region '$\mu_{POS_{R_p}(Q)}(s)$' with respect to Equation (17)
5.    Compute the dependency degree '$\Upsilon'$ with respect to Equation (18)
6.    Repeat the process and select feature subset with high dependency degree.
       Now we will execute the process of RWNN with optimized feature set
7.    Choose hidden layer activation function $g(z) = \frac{1}{1+e^{-z}}$
8.    Adjust the nodes $\tilde{K}$ in hidden layer.
9.    Assign input parameters $w_i$ and $b_i$ where $i = 1, \cdots, \tilde{K}$ w.r.t. [50]
```

```
10.    Calculate $H$ according to Equation (21)
11.    Calculate $\tilde{\beta}$ according to Equation (23).
```

5. EXPERIMENTAL ANALYSIS AND DISCUSSION

5.1. DATASETS DESCRIPTION

To evaluate the proposed classification model, we choose seven different datasets with different classes and features. We collect the datasets from UCI machine learning repository [55]. Table 2 depicts the detail of these datasets.

Table 2. Datasets description

Datasets	**Number of features**	**Number of instances**	**Number of classes**
Breast-Cancer	10	286	2
Credit-G	21	999	2
Pima_Native_American_Diabetes	8	768	2
Horse-Colic	27	368	3
Sonar	60	208	2
SpectFheart	44	267	2
Wine	13	962	3

In our experiment, our techniques using RWNN, and NN cannot process symbolic data or discrete data; therefore, different techniques can be used to convert symbolic data into continuous data without affecting the performance [56], [57]. It is also necessary to scale the data for RWNN. Therefore, we performed necessary scaling to normalize the training data and also testing data sets. Each dataset is converted into numerical data by using the nominal to numerical techniques. Each dataset is normalized in the range [0,1]. We denote the universal set, feature set, and decision set by U, F, and D for each dataset. For each real-valued feature $f \in F$, it is normalized by using Equation (24).

$$f'_j(x_i) = \frac{f_j(x_i) - \min(f_j(x_i))}{\max(f_j(x_i)) - \min(f_j(x_i))}, \mathrm{x}_i \in U \qquad (24)$$

In Equation (24), $f'_j(x_i) \in [0,1]$ for each $\mathrm{x}_i \in U$ and $f'_j(x_i)$ stands for the jth feature of sample *i*. Here we still use a f_i to denote the corresponding normalized conditional attribute for simplicity.

5.2. EVALUATION CRITERIA

The performance measurement of the proposed classification model is one of the major task and that's why our ultimate purpose to propose this classification model is the improvement of accuracy of machine learning classifiers on large datasets. There are many techniques used to evaluate the results of the proposed model but we use the *Accuracy* metric for evaluation, which is used to compare the different algorithms. Classification accuracy is the ratio of number of correct estimations to the total number of input data.

$$Accuracy = \frac{TruePositive + TrueNegative}{TotalSample} \qquad (25)$$

In equation (25), True Positive is which we predicted Yes and the actual output is also Yes, True Negative in which we predicted No and the actual value is also No and total sample is the total number of datasets. After selecting the important features from FRSFS algorithm, the KNN, decision tree, Naïve Bayes, SVM and RWNN are employed. We use 10-fold validation to perform our experiment. Before feeding the datasets to the classifiers, we set the classifier values as KNN set to 3 nearest neighbor, SVM kernel set to the polynomial kernel, and Neural Network hidden nodes are set to 5. In RWNN, we use the sigmoid activation function with hidden layers.

5.3. EXPERIMENTAL PERFORMANCE USING PROPOSED TECHNIQUE

Fuzzy rough feature selection provides us with important features by removing the redundant and noisy features as shown in Table 3. We normalize the datasets by applying the normalization formula using Equation (24). Then apply the FRSFS algorithm on the normalized datasets and find the important features of each dataset. The reduct of these datasets are given in Table 3.

Table 3. Remaining features after applying FRSFS technique

Datasets	Number of Features	Reduct
Breast-Cancer	10	7
Credit-G	21	12
Pima_Native_American_Diabetes	8	7
Horse-Colic	27	7
Sonar	60	12
SpectFheart	44	8
Wine	13	6

After necessary scaling and feature reduction, we perform our experiment using proposed technique. Figures 5,6,7,8,9,10,11 depicts the performance of our proposed methodology. In our experiment we chose the hidden layer's nodes from 10 to 50 to analyze the impact of no. of hidden nodes on classifiers generalization ability.

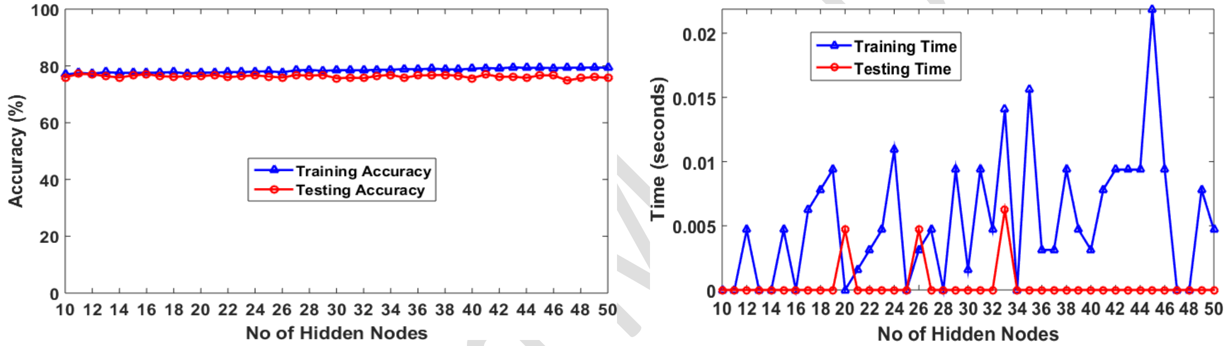


Figure 5: Training/testing accuracy and time (diabetes dataset)

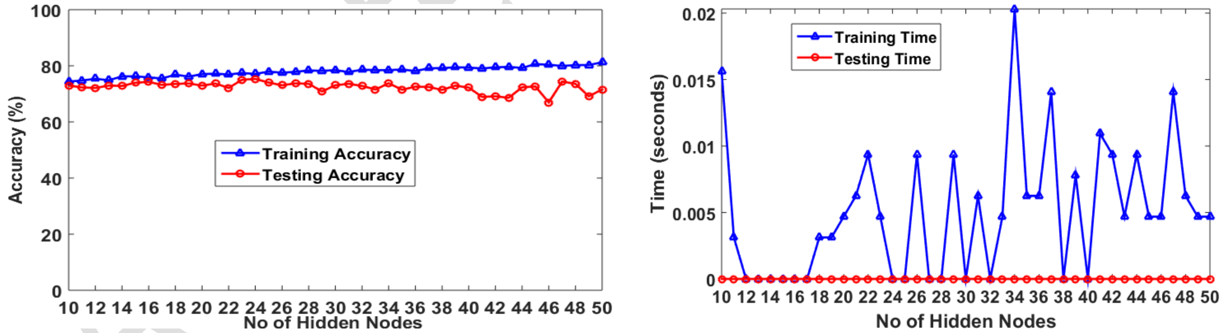


Figure 6: Training/testing accuracy and time (breast cancer)

5.4. Performance Comparison

Now, the reduced datasets are fed to the different machine learning classifiers (i.e., KNN, SVM, decision tree Naïve Bayes, neural network) and the results are compared with RWNN; where different hidden layer nodes are used to measure the generalize ability of RWNN. The experimental results are presented in Tables 6 and 7.

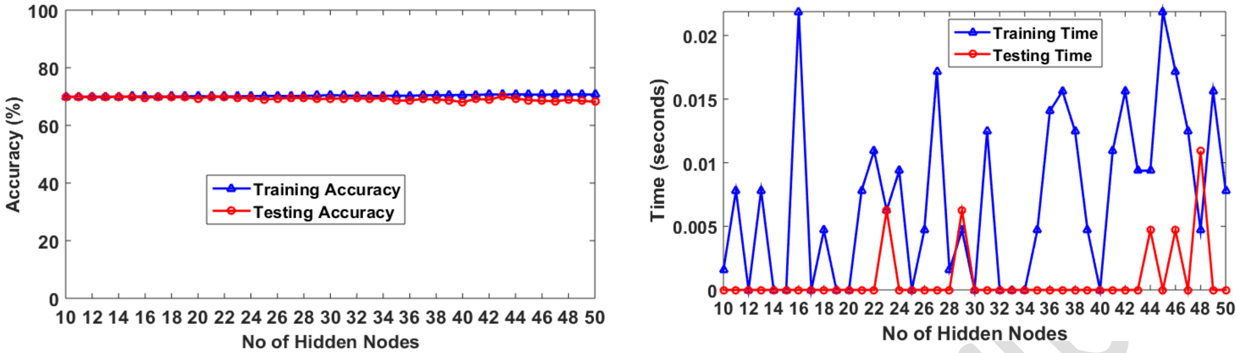


Figure 7:Training/testing accuracy and time (credit-G dataset)

Table 4. Comparison of different ML algorithms with the proposed algorithm

Dataset	Naive Bayes	Decision Tree	Neural Network	KNN	SVM	Proposed algorithm
Breast Cancer	69.23	69.28	66.78	69.58	72.07	**75.32**
Credit-G	64.46	69.97	69.86	63.76	67.86	**70.455**
Pima_Native_American_Diabetes	75.48	75.09	75.35	72.09	76.66	**77.563**
Horse_Colic	58.94	60.98	58.96	58.2	60.49	**63.13**
Sonar	48.57	53.36	53.36	47.11	49.03	**81.64**
Spectfheart	70.41	73.78	75.65	75.28	79.42	**79.965**
Wine	69.66	39.88	34.26	62.35	69.10	**99.012**

Table 5. Training time of proposed technique and other supervised learning algorithms

Dataset	Naive Bayes	Decision Tree	Neural Network	KNN	SVM	Proposed algorithm
Breast Cancer	0.00	0.00	7.32s	0.00	0.52s	0.00
Credit-G	0.01s	0.03	74.47s	0.00	6.96s	0.00
Pima_Native_American_Diabetes	0.02s	0.06s	1.23s	0.00	0.24s	0.00
Horse_Colic	0.00	0.00	3.02s	0.00	0.61s	0.00
Sonar	0.00	0.00	36.66s	0.00	0.64s	0.00
Spectfheart	0.01s	0.01s	0.65s	0.00	0.07s	0.00
Wine	0.00	0.00	9.76s	0.00	0.89s	0.00

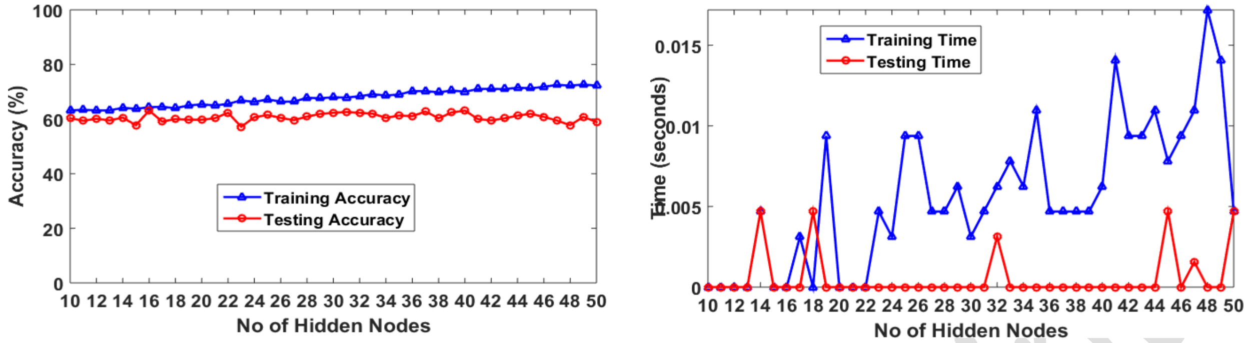


Figure 8: Training/testing accuracy and time (horse colic dataset)

In Table 4, one can see that the proposed methodology is dominating on other classifiers. As discussed earlier, different number of hidden layer nodes are used i.e., from 10 to 50. Table 5 represents the testing time of proposed algorithm. Table 6 depicts the number of hidden layer nodes providing higher accuracy rates and better training and testing time.

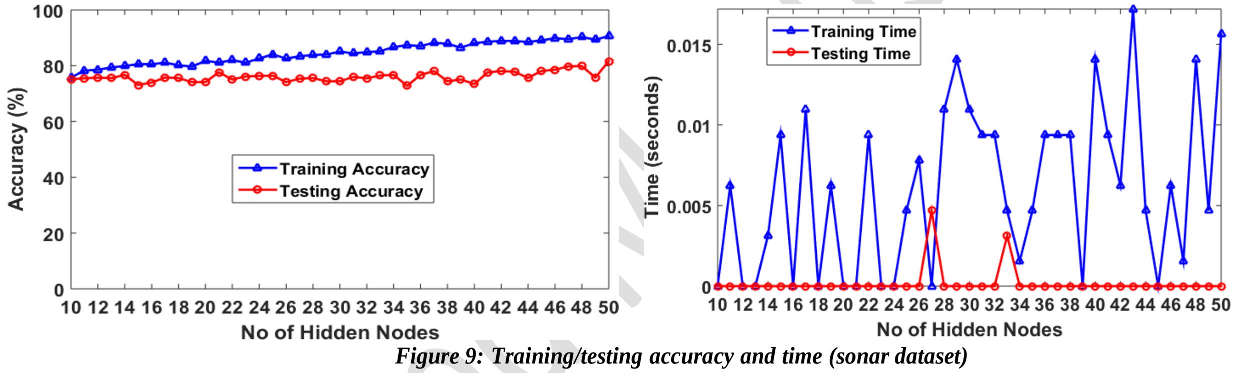


Figure 9: Training/testing accuracy and time (sonar dataset)

Table 6. Hidden nodes depicting the highest accuracy & time

Datasets	**Hidden layers**
Breast-Cancer	10
Credit-G	21
Pima_Native_American_Diabetes	8
Horse-Colic	27
Sonar	60
SpectFheart	44
Wine	13

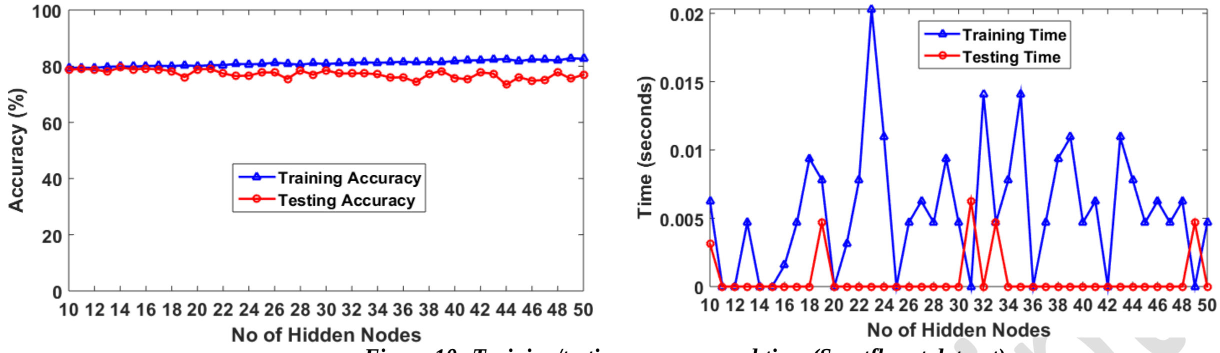


Figure 10: Training/testing accuracy and time (Spectfheart dataset)

The initialization interval in this experiment is $\left[0,\varphi\right]$, $\left[1<\varphi\le 10\right]$. The input weights w_i and biases b_i at the hidden layer nodes were the random variables that followed a uniform distribution over the interval $\left[0,\varphi\right]$. Hence, with the reduced training data set, which is obtained by using proposed methodology, a smaller interval, i.e., [0, 1] leads to the better accuracy.

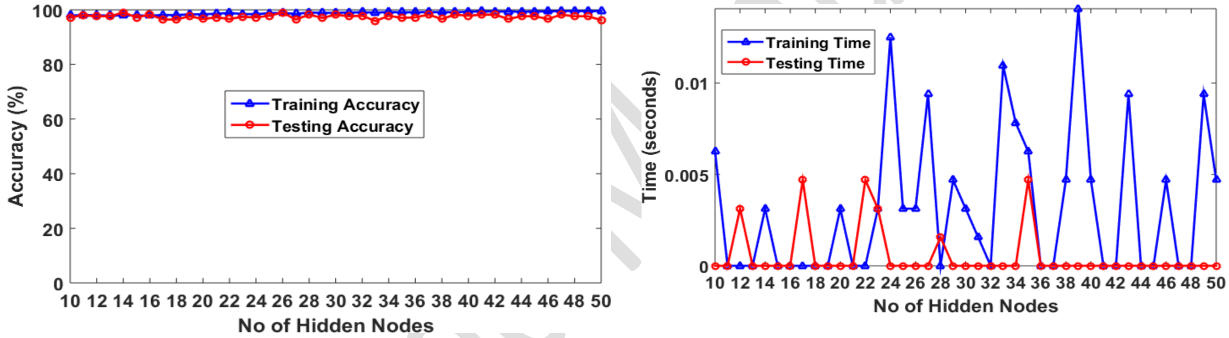


Figure 11: Training/testing accuracy and time (wine dataset)

6. CONCLUSION

In this study, We presented an efficient classification model using FRST and RWNN after necessary pre-processing on datasets to eliminate the redundancy and noise for better generalization ability. In our study, we used FRSFS to determine the most relevent features after removing noisy and conflicting features by calculating the dependency degree of each feature and obtained a feature set that is having high dependency degree. These reduced datasets are further processed by RWNN using different hidden layer nodes and the results are compared with other machine learning classifiers, i.e., SVM, kNN, Naïve Bayes, decision tree and BP neural network. The experimental results show that RWNN provides a better predictive performance and training time than other machine learning algorithms. This model can be effectively utilzed for every type of datasets (specialy for big-data) to achieve better generalization ability. This research work is limited to the feature selection, in future it can be extended by using hybrid technique for both instance and feature selection.

REFERENCES

[1] M. Dash and H. Liu, “Feature selection for classification,” *Intell. Data Anal.*, vol. 1, no. 3, pp. 131–156, 1997.

[2] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant Features and the Subset Selection Problem,” in *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 121–129.

[3] M. Dash and H. Liu, “Feature selection for classification,” *Intell. Data Anal.*, vol. 1, no. 3, pp. 131–156, 1997.

[4] X. D. Wang, R. C. Chen, F. Yan, Z. Q. Zeng, and C. Q. Hong, “Fast Adaptive K-Means Subspace Clustering for High-Dimensional

Data," *IEEE Access*, vol. 7, pp. 42639–42651, 2019.

[5] R. B. Bhatt and M. Gopal, "On fuzzy-rough sets approach to feature selection," *Pattern Recognit. Lett.*, vol. 26, no. 7, pp. 965–975, 2005.

[6] K. S. Fu, "Introduction to Syntactic Pattern Recognition," 1977, pp. 1–30.

[7] Z. Pawlak, "Rough sets," *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341–356, Oct. 1982.

[8] D. Dubois and H. Prade, "Rough fuzzy sets and fuzzy rough sets," *Int. J. Gen. Syst.*, vol. 17, no. 2–3, pp. 191–209, 1990.

[9] W. Wang and Z.-H. Zhou, "On multi-view active learning and the combination with semi-supervised learning," in *Proceedings of the 25th international conference on Machine learning - ICML '08*, 2008, pp. 1152–1159.

[10] M. Liu *et al.*, "The Applicability of LSTM-KNN Model for Real-Time Flood Forecasting in Different Climate Zones in China," *Water*, vol. 12, no. 2, p. 440, Feb. 2020.

[11] M. Q. Tran, M. Elsisi, and M. K. Liu, "Effective feature selection with fuzzy entropy and similarity classifier for chatter vibration diagnosis," *Measurement*, vol. 184, p. 109962, Nov. 2021.

[12] J. Chen, S. Yuan, D. Lv, and Y. Xiang, "A novel self-learning feature selection approach based on feature attributions," *Expert Syst. Appl.*, vol. 183, p. 115219, Nov. 2021.

[13] F. Aghaeipoor and M. M. Javidi, "A hybrid fuzzy feature selection algorithm for high-dimensional regression problems: An mRMR-based framework," *Expert Syst. Appl.*, vol. 162, p. 113859, Dec. 2020.

[14] Z. Pawlak, "Rough sets," *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341–356, 1982.

[15] D. Dubois and H. Prade, "Twofold fuzzy sets and rough sets-Some issues in knowledge representation," *Fuzzy Sets Syst.*, vol. 23, no. 1, pp. 3–18, 1987.

[16] M. Wygralak, "Rough sets and fuzzy sets-some remarks on interrelations," *Fuzzy Sets Syst.*, vol. 29, no. 2, pp. 241–243, Jan. 1989.

[17] P. Lingras and R. Jensen, "Survey of rough and fuzzy hybridization," *IEEE Int. Conf. Fuzzy Syst.*, no. August, 2007.

[18] L. I. Kuncheva, "Fuzzy rough sets: Application to feature selection," *Fuzzy Sets Syst.*, vol. 51, no. 2, pp. 147–153, 1992.

[19] R. Jensen and Q. Shen, "Fuzzy-rough attribute reduction with application to web categorization," *Fuzzy Sets Syst.*, vol. 141, no. 3, pp. 469–485, Feb. 2004.

[20] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, no. 1–2, pp. 245–271, 1997.

[21] R. Jensen and Q. Shen, "Fuzzy-rough sets assisted attribute selection," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 1, pp. 73–89, 2007.

[22] T. Som, S. Shreevastava, A. K. Tiwari, and S. Singh, "Fuzzy Rough Set Theory-Based Feature Selection," in *Mathematical Methods in Interdisciplinary Sciences*, Wiley, 2020, pp. 145–166.

[23] H. Sever, V. V. Raghavan, and T. D. Johnsten, "The status of research on rough sets for knowledge discovery in databases," in *Proceedings of the Second International Conference on Nonlinear Problems in Aviation and Aerospace (ICNPAA98)*, 1998, vol. 2, pp. 673–680.

[24] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.

[25] I. Iguyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[26] S. Kejia, H. Parvin, S. N. Qasem, B. A. Tuan, and K. H. Pho, "A classification model based on svm and fuzzy rough set for network intrusion detection," *J. Intell. Fuzzy Syst.*, vol. 39, no. 5, pp. 6801–6817, Oct. 2020.

[27] J. Kohlas and P. A. Monney, "Theory of evidence - A survey of its mathematical foundations, applications and computational aspects," *ZOR Zeitschrift für Oper. Res. Math. Methods Oper. Res.*, vol. 39, no. 1, pp. 35–68, Feb. 1994.

[28] R. B. Bhatt and M. Gopal, "On the compact computational domain of fuzzy-rough sets," *Pattern Recognit. Lett.*, vol. 26, no. 11, pp. 1632–1640, 2005.

[29] D. Chen, X. Wang, and S. Zhao, "Attribute reduction based on fuzzy rough sets," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4585 LNAI, pp. 381–390, 2007.

[30] Q. Shen and R. Jensen, "Rough sets, their extensions and applications," *Int. J. Autom. Comput.*, vol. 4, no. 3, pp. 217–228, 2007.

[31] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Inf. Sci. (Ny).*, vol. 177, no. 1, pp.

3–27, 2007.

[32] M. Zhang and J. T. Yao, “A rough sets based approach to feature selection,” in *Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS*, 2004, vol. 1, pp. 434–439.

[33] Z. Pawlak, “Some issues on rough sets,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. LNCS 3100, pp. 1–58, 2004.

[34] G. Chen, T. T. Pham, and N. Boustany, “Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems,” *Appl. Mech. Rev.*, vol. 54, no. 6, pp. B102–B103, Nov. 2001.

[35] Z. Pawlak, “Rough set approach to knowledge-based decision support,” *Eur. J. Oper. Res.*, vol. 99, no. 1, pp. 48–57, 1997.

[36] L. A. Zadeh, “Fuzzy sets,” *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[37] E. C. C. Tsang, D. Chen, D. S. Yeung, X. Z. Wang, and J. W. T. Lee, “Attributes reduction using fuzzy rough sets,” *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, pp. 1130–1141, 2008.

[38] X. Z. Wang, J. H. Zhai, and S. X. Lu, “Induction of multiple fuzzy decision trees based on rough set technique,” *Inf. Sci. (Ny).*, vol. 178, no. 16, pp. 3188–3202, 2008.

[39] A. Mieszkowicz-Rolka and L. Rolka, “Fuzzy rough approximations of process data,” *Int. J. Approx. Reason.*, vol. 49, no. 2, pp. 301–315, 2008.

[40] C. Wang *et al.*, “A Fitting Model for Feature Selection with Fuzzy Rough Sets,” *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 741–753, 2017.

[41] D. Dubois and H. Prade, “Putting Rough Sets and Fuzzy Sets Together,” *Intell. Decis. Support*, pp. 203–232, 1992.

[42] A. M. Radzikowska and E. E. Kerre, “A comparative study of fuzzy rough sets,” *Fuzzy Sets Syst.*, vol. 126, no. 2, pp. 137–155, 2002.

[43] R. Jensen and Q. Shen, “Fuzzy-rough data reduction with ant colony optimization,” *Fuzzy Sets Syst.*, vol. 149, no. 1, pp. 5–20, 2005.

[44] R. Jensen and Q. Shen, “New approaches to fuzzy-rough feature selection,” *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 824–838, 2009.

[45] W. F. Schmidt, M. A. Kraaijveld, and R. P. W. Duin, “Feedforward neural networks with random weights,” in *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, pp. 1–4.

[46] Y. H. Pao, G. H. Park, and D. J. Sobajic, “Learning and generalization characteristics of the random vector functional-link net,” *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[47] B. Igelnik and Y. H. Pao, “Stochastic Choice of Basis Functions in Adaptive Function Approximation and the Functional-Link Net,” *IEEE Trans. Neural Networks*, vol. 6, no. 6, pp. 1320–1329, 1995.

[48] Y. F. Yam, T. W. S. Chow, and C. T. Leung, “A new method in determining initial weights of feedforward neural networks for training enhancement,” *Neurocomputing*, vol. 16, no. 1, pp. 23–32, 1997.

[49] M. Alhamdoosh and D. Wang, “Fast decorrelated neural network ensembles with random weights,” *Inf. Sci. (Ny).*, vol. 264, pp. 104–117, 2014.

[50] B. Igelnik and Yoh-Han Pao, “Stochastic choice of basis functions in adaptive function approximation and the functional-link net,” *IEEE Trans. Neural Networks*, vol. 6, no. 6, pp. 1320–1329, 1995.

[51] S. Scardapane, D. Wang, M. Panella, and A. Uncini, “Distributed learning for Random Vector Functional-Link networks,” *Inf. Sci. (Ny).*, vol. 301, pp. 271–284, 2015.

[52] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, “Fuzziness based semi-supervised learning approach for intrusion detection system,” *Inf. Sci. (Ny).*, vol. 378, pp. 484–497, 2017.

[53] R. A. R. Ashfaq, Y.-L. He, and D.-G. Chen, “Toward an efficient fuzziness based instance selection methodology for intrusion detection system,” *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 6, 2017.

[54] F. Cao, H. Ye, and D. Wang, “A probabilistic learning algorithm for robust modeling using neural networks with random weights,” *Inf. Sci. (Ny).*, vol. 313, pp. 62–78, 2015.

[55] A. Frank and A. Asuncion, “{UCI} Machine Learning Repository.” 2010.

[56] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.

[57] J. Neter, *Applied linear statistical models*, 1st ed. WCB/MacGraw-Hill, 1996.