Improved Spatial Information Based Semisupervised Classification of Remote Sensing Images

Neeta S. Kothari, *Student Member, IEEE*, Saroj K. Meher, *Senior Member, IEEE*, and Ganapati Panda, *Senior Member, IEEE*

***Abstract*—Motivation in the use of semisupervised learning method is because of its ability to strategically explore and use abundantly available unlabeled samples along with the limited number of labeled samples, as seen in the remote sensing (RS) imagery. In this direction, the present article proposes a semisupervised classification model with spatial information based self-learning methodology to classify land covers in RS images. The model uses granular neural network (GNN) as the base classifier because of its customizable network architecture that is functionally interpretable and costs less computational complexity. Architecture of GNN is governed by fuzzy *if–then* rules that are generated from fuzzy granulation of input feature space. We have used an improved spatial neighborhood learning method for better understanding of data distribution in a semisupervised framework. The method collects the information with collaborative opinions of two independent information extraction approaches, i.e., based on mutual neighborhood criteria and class map of unlabeled samples. Superiority of the proposed model with existing methods are established with different RS images in terms of various performance measurement indexes.**

***Index Terms*—Fuzzy granulation, granular neural network (GNN), land cover classification, neural networks, pattern recognition, remote sensing (RS), semisupervised learning.**

NOMENCLATURE

G_1, G_2, G_3, G_4, G_5	Similarity matrices.
$\mathbf{F}$	n-dimensional sample/pattern/pixel.
M	Loss function used for training the GNN.
L and U	Number of labeled and unlabeled samples.
M_1 and M_2	Component of M for L and U samples.
l_i	$l_i \in L$.
c_{l_i} and p_{l_i}	Actual and predicted class labels of l_i.
$V(.)$	Hinge loss function.
$T(.)$	Regularization function.
u_i and u_j	Pair of unlabeled samples.
p_{u_i} and p_{u_j}	Predicted class label of u_i and u_j.
$\text{SM}_{u_i u_j}$	Similarity matrix for u_i and u_j.
ρ_u	Control parameter for $T(.)$.
η	Learning rate.
p_u and q_u	Classifier o/p and pseudoclass label for u.
C	Total number of classes.
T_c	Total computation time.
DS, DB, and β	Performance measurement indexes.

Manuscript received October 6, 2019; revised November 29, 2019; accepted December 16, 2019. Date of publication January 8, 2020; date of current version February 12, 2020. *(Corresponding author: Saroj K. Meher.)*

N. S. Kothari and S. K. Meher are with the Systems Science and Informatics Unit, Indian Statistical Institute, Bangalore 560069, India (e-mail: sharmaneeta31@gmail.com; saroj.meher@gmail.com).

G. Panda is with the Department of Computer Science, C. V. Raman College of Engineering, Bhubaneswar 752054, India (e-mail: ganapati.panda@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2019.2961985

I. INTRODUCTION

ADVANCEMENT in the sensor technology has increased the information content of remote sensing (RS) images, which can be useful to many applications, e.g., land resource management, soil erosion, and biodiversity. Various types of sensors are used to collect RS images that result in the generation of diverse data and eventually poses several challenges for classification task. Scarcity of labeled samples is a major concern for efficient land cover classification of RS data, with supervised models. Although, collection of labeled samples is time consuming and expensive, unlabeled samples are abundantly available. For this reason, semisupervised learning (SSL) based classification models are successfully being used in the RS domain that explore and learn from both limited number of labeled and large number of unlabeled samples.

In order to address the issue of limited labeled samples in RS data and the best features for classification task, many research works are being carried out using deep learning architectures [1], [2] and tensor-based methods [3]. Recently, two review articles [4], [5] have described a general framework of deep learning and its applications to RS data. The state-of-the-art deep learning models for RS are regarded as special cases of SSL with various deep networks and tuning tricks.

In another approach of dealing with limited labeled samples, several research attempts have been made in the development of efficient semisupervised classification models for RS images [6]–[12]. Among them, self-training [9] and co-training [7] are the most popular approaches. With self-training, size of the available training set is increased strategically with the selected unlabeled samples, whereas co-training [7] and tri-training [10] methods work with two and three different classifiers that are trained simultaneously with separated feature subsets of the labeled data, respectively. However, obtaining these feature subsets that are conditionally independent and individually sufficient is very difficult. These factors motivated us to use the self-training method for the proposed classification model.

With this reasoning, various self-learning classification models have been proposed, for example, support vector machines (SVMs) [13], transductive SVMs (TSVMs) [13], Laplacian SVMs (LapSVMs) [14], [15], graph-based methods [6], and spatial-spectral information based semisupervised classification (S^2ISC) [16]. Although accuracies are good using these models, they are computationally very expensive. In S^2ISC method, the most informative and confident unlabeled samples are selected on the basis of neighborhood information, which are determined collaboratively with spatial and spectral information [16]. This method produces comparatively higher classification accuracy than the methods, that utilizes only spectral information and randomly selected unlabeled samples. In kernel methods, such as SVM and TSVM, training is done by solving a convex optimization problem (i.e., no local minima). These methods can handle the high dimensionality of the samples and can take care of noisy samples effectively. LapSVM is a semisupervised extension of the SVM that performs well and provides improved classification accuracy compared to the conventional SVM and TSVM. However, kernel computation makes the SVM-based methods computationally complex and, thus, may not handle a large number of samples, e.g., RS images. Also, LapSVM uses a functional form of Laplacian eigenmaps that leads to constrained optimization issue.

Ratle *et al.* in [8] proposed a semisupervised self-learning (SSSL) framework with NN (as the base classifier) to address the above-mentioned issues associated with kernel methods, and used an appropriate loss function instead of Laplacian eigenmaps. The method can efficiently process a large number of unlabeled samples. The properties associated with NN, such as adaptivity, speed, fault tolerance, ruggedness, and optimality, make the model more effective to classify large size RS image. With these factors, SSSL NN [8] outperformed other existing SSSL methods [13]–[15]. To improve the SSSL NN further, Im and Taylor [17] proposed an improved method called neighborhood graph learning (NGL), which was implemented with NN to obtain the neighborhood information. However, complexity in the design of model's architecture becomes a huge burden for this model. In addition, this conventional NN behaves like a black box for the developers. It is indeed difficult to interpret the learned information and the way it has learned. This is primarily due to the fact that the features in the upper layers are typically used in complex ways compared to the lower layers and it is hard to understand the relationship between the representations learned at each layer. To mitigate these issues, the present article proposes to design a computationally efficient SSSL model using the framework described in [8], by using granular neural network (GNN) [18] as the base classifier instead of conventional NN. The model is further modified and improved using efficient neighborhood information extraction method that exploits the unlabeled samples and utilizes them for learning the base classifier. GNN is an interpretable, transparent, and computationally cost-effective network with better generalization ability compared to conventional NN [19], [20].

Many research studies have been successfully carried out based on the spatial neighborhood information [8], [17], [21]–[23] of unlabeled samples to improve the performance of semisupervised classification model. Ratle *et al.* [8] trained the NN-based model in the framework of SSSL for the classification of RS images and uses the spatial neighborhood information of unlabeled samples. Ratle *et al.* [8] considered two samples a and b as neighbors, if a is neighbor to b only. There is every possibility that b may not be neighbor to a and, hence, the method becomes an incomplete assessment of neighborhood information. In this article, we have computed the complete/true spatial neighborhood by checking neighboring criteria in both ways. This criterion is more logical to put two samples in the same class or in different classes. Additionally, we have computed neighborhood information of samples with one more method, i.e., based on the class labels of unlabeled samples, where the predicted class map is obtained by a suitable classifier. This step of computation provides one more layer of confidence in the selection of unlabeled samples by the earlier method. Finally, the collaborative spatial neighborhood information obtained by these two methods are used to train the base classifier. Efficacy of the proposed model have been verified in the classification of two multispectral RS (MSRS) and two hyperspectral RS (HSRS) images.

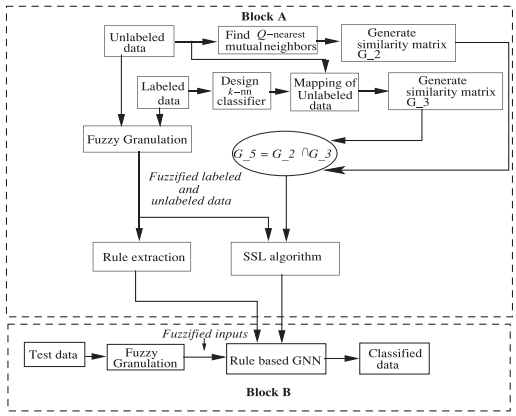


Fig. 1. Schematic flow diagram of the proposed self-learning semisupervised classification model.

The contribution in this article is four-fold. The first one is the characterization of spatial information with the mutual neighborhood criterion of samples. The second one is the characterization of spatial information with the class label of neighbors for the considered sample. The third contribution is the collaborative combination of spatial information obtained through the aforementioned two operational steps. Finally, the use of GNN (as the base classifier) in the learning process of the proposed model that is designed with the adaptive fuzzy rules. Performance of the model was compared with various state-of-the-art methods.

II. Proposed Classification Model

Using the generic framework of semisupervised self-learning (SSSL) approach, we have proposed a classification model with the functional block diagram shown in Fig. 1. The processing steps of the training phase use both labeled and unlabeled samples strategically to train the base classifier, as explained in Block A of Fig. 1, whereas the operations in Block B are kept similar to a generic SSSL model. While developing the proposed

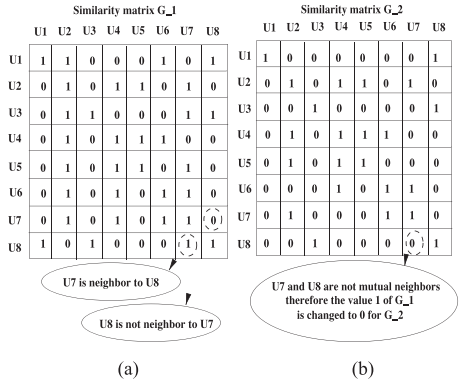


Fig. 2. Generation of similarity matrix. (a) G_1. (b) G_2.

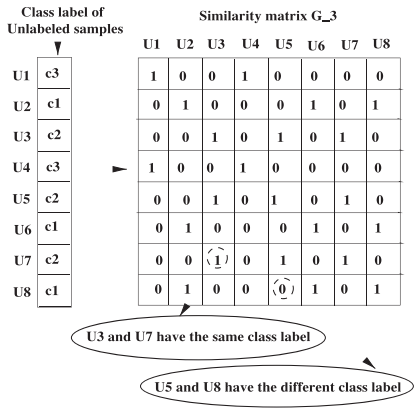


Fig. 3. Generation of similarity matrix G_3.

model, we extract the spatial neighborhood information of each unlabeled sample and represent it in a similarity matrix (G_5). G_5 is a collaborative neighborhood information of similarity matrixes G_2 and G_3 obtained through the mutual agreement of neighborhood and class-map criterions, respectively. GNN is used as the base classifier for the model that takes the fuzzy granulated features as the input and process them. The architecture of the GNN is then designed based on the extracted adaptive fuzzy *if–then* rules obtained from the domain information of the dataset. The detail descriptions of each step of operation for the proposed SSSL GNN classifier are made in the following.

A. Similarity Matrix Generation

Spatial neighborhood information of each of the unlabeled samples is extracted and the respective similarity matrix is generated for the learning process of the semisupervised classification model. Elements of the similarity matrix are assigned a value "1" for the two neighborhood samples and "0" for two nonneighbors. Stepwise operations in the generation of these different similarity matrixes are described in the following.

1) Similarity Matrix G_1: Ratle *et al.* in [8] have used a similarity matrix (now onward we name it as G_1) in the learning process of SSSL NN, which is generated through Q-nearest neighbors of each unlabeled sample. Euclidean distance is used to generate this matrix. Fig. 2(a) illustrates the elements of the matrix G_1 for an example dataset with eight unlabeled samples U1, U2, U3, U4, U5, U6, U7, and U8. In this example, Q is taken as three that means for each sample, three nearest samples (out of seven) are considered as neighbors and others as nonneighbors. Accordingly, the element values of G_1 are assigned. For example, only U2, U4, and U6 are the neighbors to U7, and hence, the corresponding element values in G_1 are 1, as shown in Fig. 2(a).

2) Similarity Matrix G_2: It is logical to define the spatial neighborhood between two samples from both of their perspectives, instead of one to other only. In other words, two samples *a* and *b* are considered neighbors only when each of them is neighbor to other. This criteria of the neighborhood in generating the matrix G_2 can be considered as the generalized form of the criteria used in G_1, which considers one way checking of the neighborhood. The similarity matrix G_1 is revised according to the criteria of this mutual consent to obtain the similarity matrix G_2. Fig. 2(b) depicts the similarity matrix G_2 generated for the same set of samples used in G_1. As shown in Fig. 2(b), the 7th entry of the last row of G_1 (when U7 is neighbor to U8) is changed to "0" to generate its corresponding entry of G_2 because U8 is found as nonneighbor to U7. With the same logic, the element values of G_1 are changed to generate the matrix G_2.

3) Similarity Matrix G_3: To generate the similarity matrix G_3, spatial neighborhood information is obtained through the class map of unlabeled samples. Initially, with a limited number labeled samples, a classifier is trained and used to obtain the labels of unlabeled samples. For this experimental study, *k*-NN classifier is used but any other classifier can also be opted. Using these labeled information, similarity matrix G_3 is generated, where the samples with the same class label are considered as neighbors and for different class labels, they are treated as nonneighbors. For illustration purpose, generation of G_3 for a dataset having eight unlabeled samples is depicted in Fig. 3.

4) Similarity Matrix G_4: Similarity matrix G_4 is generated by combining the spatial neighborhood information obtained with G_1 and G_3. For this purpose, a logical AND operation is performed on the element values of matrixes G_1 and G_3, i.e., $G\_4 = \{G\_1 \cap G\_3\}$.

5) Similarity Matrix G_5: With a logical AND operation on the value of the elements of matrixes G_2 and G_3, similarity matrix G_5 is generated, i.e., $G\_5 = \{G\_2 \cap G\_3\}$. This similarity matrix is used for the SSSL method of the proposed model.

B. Fuzzy Granulation of Input Features

The original input feature space is transformed into fuzzy granulated feature space using three fuzzy sets [24]. The generated fuzzy granules are characterized by linguistic variables that correspond to fuzzy sets and possess the ability to handle the impreciseness and uncertainty of the dataset. The granulated information also provide a better interpretation of the data content compared to the ungranulated or crisp input; thereby enhancing the transparency of the input to a network. In this article, we have used three linguistic properties, i.e., *low*,

TABLE I
FUZZY RULE EXTRACTION FOR A SAMPLE OF CLASS "HABITATION"

Feature	Fuzzy Granulation		
	low	medium	high
F1	**0.9537**	0.1850	0.04627
F2	**0.9589**	0.3932	0.09832
F3	0.6898	**0.9377**	0.3391
F4	0.6671	**0.9903**	0.4330

medium, and *high* for fuzzy granulation of each input feature. A sample F with *n* numeric features ($\mathbf{F} = F_1, F_2, \ldots F_n$) can be granulated using π-type MFs [24].

After fuzzy granulation, *n*-dimensional numeric features ($\mathbf{F} = F_1, F_2, \ldots F_n$) are represented as $(3 \times n)$ dimensional fuzzy granulated features. For example, the fuzzy granulation of four features *F*1, *F*2, *F*3, and *F*4 is illustrated in Table I using three linguistic fuzzy variables *low, medium,* and *high*.

C. Fuzzy $if-then$ Rule Generation

Fuzzy rule extraction aims to obtain the relationship between fuzzy granulated inputs and the associated class labels of the dataset. In this article, fuzzy variables are used to design the fuzzy rules, and semisupervised GNNs are developed.

Sánchez and Trillas [25] explored measures of fuzziness to give new dimension to fuzzy set theory. Fuzziness-based active learning framework was proposed in [26] to enhance the classification performance of hyperspectral image. In the recent past, various methods of fuzzy *if–then* rule generation have been proposed [19], [27], [28]. We have defined the fuzzy rules using the method proposed in [19] because it exploits the domain information in a simple way and requires fewer computations that can be adopted for any dataset. Using this method, a rule for a class is generated by the fuzzy set having maximum membership value for each feature of a sample vector or pixel. As an example, the rule generation for a sample with four features is illustrated in Table I. A sample/pixel is assigned as *habitation* land cover class of an RS dataset, if the fuzzy set *low* of feature F1, *low* of F2, *medium* of F3, and *medium* of F4 are considered, because they have the maximum membership values (boldface values of Table 1) across the three fuzzy granulated features and, thus, these elements together implies *habitation* as the output class. Using this strategy of rule generation, three sets of rules are generated for each class of the dataset to design the GNN in this article.

D. Proposed Semisupervised GNN Classification Model

Modeling process of the proposed semisupervised GNN has adopted a mechanism to deal with both labeled and unlabeled samples in a semisupervised framework. We have used the loss function M to train the model using both labeled and unlabeled samples of the dataset and is defined as

$$M = \underbrace{\frac{1}{L}\sum_{i=1}^{L} V\left(l_i, c_{l_i}, p_{l_i}\right)}_{M1} + \underbrace{\rho_u \frac{1}{U}\sum_{i,j=1}^{U} T\left(p_{u_i}, p_{u_j}, SM_{u_i u_j}\right)}_{M2} \tag{1}$$

where the components $M1$ and $M2$ are meant for L number of labeled and U number of unlabeled samples, respectively. In the subfunction $M1$, c_{l_i} and p_{l_i} are the actual and predicted class labels of the sample $l_i \in L$.$V(.)$ in $M1$ of (1) is the hinge loss function. To strengthen the hypothesis in order to avoid possible misguidance with the limited number of labeled samples, one type of regularization function, i.e., $T(p_{u_i}, p_{u_j}, SM_{u_i u_j})$, is used to process unlabeled samples through the subfunction $M2$. This regularization function $T(.)$ is a modified form of the hinge loss function that is computed for a pair of unlabeled samples u_i and u_j with their corresponding predicted class label p_{u_i} and p_{u_j}, respectively. $SM_{u_i u_j}$ (similarity matrix) characterizes the spatial neighborhood relationship between these two samples. The effectiveness of this regularizer is controlled by an empirically determined parameter ρ_u. Detail description of the processing steps is discussed in the following.

1) Processing the Labeled Samples: The proposed semisupervised model has used GNN as the base classifier and is trained similar to NN using the back propagation learning algorithm (BPA). The BPA requires an appropriate loss function to compute the output error for processing each of the training samples. The most common loss function used in NN is the squared function, which is mostly suitable for Gaussian distributed dataset and this fact is also true in case of GNN. However, if the dataset does not have Gaussian distribution, as in the case of real-time classification problems, such as RS imagery, then the use of hinge loss function is more relevant. Therefore, we have used hinge loss function to compute the error of the network and to process each labeled sample. We have followed the principles of BPA to train the GNN. Hinge loss function is already successfully applied in various kernel methods, such as SVM, LapSVM, and TSVM. Also, Ratle *et al.* [8], Im and Taylor [17], and Bishop [29] suggested that hinge loss function for error computation in classification task of RS datasets is more appropriate. To compute the error for a labeled sample l_i with actual and predicted class label c_{l_i} and p_{l_i}, respectively, the hinge loss function is given as

$$V\left(l_i, c_{l_i}, p_{l_i}\right) = \max\left\{0, \left(1 - c_{l_i} p_{l_i}\right)\right\}. \tag{2}$$

2) Processing the Unlabeled Samples: In the proposed semisupervised framework, unlabeled samples are utilized in the learning process based on their spatial neighborhood relationship. The similarity matrix G_5 represents the neighborhood of unlabeled samples more precisely and accurately and is used in the model to process the unlabeled samples pairwise. The proposed model learns from a randomly picked neighboring unlabeled samples followed by a pair of nonneighbors. The loss function $T(.)$ is defined for a pair of unlabeled samples is given as [30]

$$T\left(p_{u_i}, p_{u_j}, \mathrm{SM}_{u_i u_j}\right) = \begin{cases} \sum_{ij} \left\|p_{u_i} - p_{u_j}\right\|^2, & \text{if } \mathrm{SM}_{u_i u_j} = 1 \\ \sum_{ij} \max\left(0, m - \left\|p_{u_i} - p_{u_j}\right\|\right)^2, & \text{if } \mathrm{SM}_{u_i u_j} = 0 \end{cases} \tag{3}$$

Algorithm 1

1) Given:
   ⇒ L number of labeled sample $\{l_i, c_{l_i}\}$, $i = 1, 2, \ldots L$, where c_{l_i} is the class label for sample l_i.
   ⇒ U number of unlabeled sample $\{u_j\}$, $j = 1, 2, \ldots U$.
2) Generate five similarity matrixes G_1, G_2, G_3, G_4, and G_5 of size $U * U$ (Section II-A).
3) Train the semisupervised GNN
   a) Randomly pick a labeled sample $\{l_i, c_{l_i}\}$ and optimize $V(l_i, c_{l_i}, p_{l_i})$ (2).
   b) Pick a random pair of neighbors (based on G_5), $\{u_r\}$ and $\{u_s\}$ from U and optimize $T(p_{u_r}, p_{u_s}, 1)$ (4).
   c) Pick a random pair of non-neighbors (based on G_5), $\{u_x\}$ and $\{u_y\}$ from U and optimize $T(p_{u_x}, p_{u_y}, 0)$ (4).

Repeat the training steps (a to c) for all samples of L and U.

where p_{u_i} and p_{u_j} are the outputs of the classifier for the unlabeled samples u_i and u_j, respectively. $\mathrm{SM}_{u_i u_j}$ is the neighborhood relationship between u_i and u_j obtained from the pairwise similarity matrix SM computed for all pairs of unlabeled samples. The method is designed in such a manner that a pair of samples with the corresponding element value of similarity matrix "1" must be placed closely and separated for the value "0." With this logic, (3) maps similarly behaved unlabeled samples closely and separates dissimilar samples by a distance *m*. The loss function given in (3) can be reframed for the task of binary classification as [8]

$$T\left(p_{u_i}, p_{u_j}, SM_{u_i u_j}\right) = \begin{cases} \sum_{ij} \eta^{(+)} V\left(u_i, p_{u_i}, c\right) \\ \text{with } c = \text{sign}\left(p_{u_i} + p_{u_j}\right), \text{ if } \mathrm{SM}_{u_i u_j} = 1 \\ \sum_{ij} -\eta^{(-)} V\left(u_i, p_{u_i}, c\right) \\ \text{with } c = \text{sign}\left(p_{u_j}\right), \text{ if } \mathrm{SM}_{u_i u_j} = 0 \end{cases} \tag{4}$$

where $V(.)$ is the hinge loss function and learning is carried out with the learning rates $\eta^{(+)}$ and $\eta^{(-)}$. The classifier learns with the learning rate $\eta^{(+)}$ for a neighboring pair (indicated by $\mathrm{SM}_{u_i u_j} = 1$) to put these samples in the same class. With $\eta^{(-)}$ learning rate, the classifier learns to put nonneighboring (indicated by $\mathrm{SM}_{u_i u_j} = 0$) samples in different classes. For multiclass classification, (4) can be reformulated by computing the hinge loss function to sum over the classes, i.e.

$$V\left(u, p_u, q_u\right) = \sum_{c=1}^{C} \max\left(0, 1 - q_u(c) p_u(c)\right) \tag{5}$$

where p_u and q_u are the classifier output and pseudoclass label for the sample u, respectively, with total number of classes C. $q_u(n) = 1$, if $q_u = c$ and -1 otherwise. In this article, the hinge loss function, as given in (5) for the multiclass case is considered and the training of the model is carried out using stepwise operations illustrated in Algorithm 1. Processing steps of Algorithm 1 is similar to the method described in [8].

3) GNN Architecture: The architecture of GNN depends on the number of granulated input feature, fuzzy *if–then* rules and classes present in the dataset. The granulation operation and fuzzy *if–then* rule generation process are carried out, as discussed in Section II-B and II-C, respectively. The GNN used for this article comprises three layers, i.e., input, hidden, and an output layer. The granulated features are provided as the input to the network and their numbers depend on the linguistic variables used for granulation. In this article, we have considered three linguistic variables *low, medium,* and *high* and accordingly each feature is characterized into three granulated features. Thus, the number of input nodes to the GNN is equal to (3×number of features). The number of hidden-layer neurons is determined by the number of fuzzy *if–then* rules used for the classes of the dataset and each class is represented by one node of the output layer. In the GNN, connections between the nodes of input and hidden layer depend on the number of extracted rule. The connections between input-layer nodes and hidden-layer nodes are established only when the particular granulated input feature is a part of the corresponding rule of hidden node, whose consequence leads to a particular output class. In a similar manner, hidden-layer node and output-layer node connections are established only when particular hidden-layer node yields the corresponding output class. Other than these, node-to-node connections do not exist and, hence, GNN architecture is not fully connected. This strategy of interconnections in GNN is based on the maximum membership, which is derived from fuzzy *if–then* rules and successfully used in [19]. This makes the network more interpretable and transparent compared to a fully connected NN. Similar to the training scheme of a conventional multilayer perceptron NN, the rule-based GNN is trained using the BPA.

III. Performance Measurement Indexes

To evaluate the performance of the proposed semisupervised GNN classification model, various classification measurement indexes, such as overall accuracy (OA), precision, recall, kappa coefficient (KC), dispersion score (DS) [31], [31], computation time (T_c), and clustering indexes, such as β [32], Davies–Bouldin (DB) [33] are used. As per the definition, lower value of DS and DB indexes are better, and for β index, it should be a higher value. Computation time is the total time required for a model to train and test the classifier. In this experimental study, all the simulations are performed in MATLAB (Matrix Laboratory) (version R2017a) environment in Intel core i7 machine with 3.40 GHz processor speed and 64-b operating system.

IV. Dataset Used

Efficacy of the proposed classification model is verified using two MSRS and two HSRS images obtained from different sensors. MSRS images are considered here for the experiment with completely and partially labeled datasets. Partially labeled means ground truth of small number of pixels are only known. To create a completely labeled database, fixed number of samples from each land cover class are collected. To design and test the model, each dataset is randomly partitioned into training and test sets. To evaluate the performance of models using completely

labeled datasets, indexes, such as OA, precision, recall, KC, and DS are used, whereas DB and β indexes are used for partially labeled datasets because class labels of all the samples/pixels of the images are not known.

A. Multispectral RS Dataset

Out of two MSRS images, one dataset was obtained from linear imaging self-scanning sensor (LISS)-II and other from LISS-III sensor. For both the datasets, six major land cover classes are considered; those are *pond* or *fishery water, turbid water, concrete area, habitation, vegetation*, and *open spaces*. Image obtained from LISS-II sensor is composed of four spectral bands, i.e., blue band (band 1), green band (band 2), red band (band 3), and near-infrared band (band 4) with a spatial resolution of 36.25 $\times$ 36.25 m and spectral variation over the wavelength range of $0.45 - 0.86\,\mu$m. Image obtained from LISS-III sensor is having spectral information over the wavelength range of 0.52–1.7 μm and accommodate spatial resolution of 23.5 m. It consists of four spectral bands, two from visible spectrum (green and red) and two from the infrared spectrum (near infrared and shortwave infrared). Two hundred samples are selected randomly from each class of these datasets to form the databases of 1200 samples.

B. Hyperspectral RS Dataset

Brief description of the two HSRS datasets $Indian\,Pines$ and $Pavia\,University$ is made in the following.

1) $Pavia\,University$ and $Indian\,Pines$ Data: This dataset was obtained from reflective optics system imaging spectrometer (ROSIS) sensor, which covers Pavia University region of Italy. The $Pavia\,University$ image is of 610×610 pixels, having spatial resolution of 1.3 m. The dataset has 103 spectral bands for nine land cover classes. We have also considered the $Indian\,Pines$ HSRS dataset, which is commonly used in the analysis of classification algorithms. $Indian\,Pines$ image data comprises 145×145 pixels and is labeled with 16 classes of ground cover. As per the suggestion of the database designer, 20 noisy spectral bands that cover the region of water absorption are removed from the available 220 bands.

V. Experimental Results and Discussions

A. Model Description

Performance of the proposed semisupervised fuzzy rule based GNN classification model is evaluated using both partially and completely labeled RS datasets. In order to design and validate the model in SSSL framework, we have created the data samples accordingly. For completely labeled datasets, the whole samples are divided randomly into two equal parts, such as training set (TR_l) and testing set (TS_l). For each of the sets, an equal percentage of samples is taken from all the classes. In SSSL framework, both labeled and unlabeled samples are required during the training of model. Thus, the training set TR_l obtained from initial partition is again divided into two subsets. One subset is assumed as labeled and the other as unlabeled. To perform the experiment, we have taken five different percentages of labeled samples from TR_l, such as 10%, 20%, 30%, 40%, and 50% and the rest 90%, 80%, 70%, 60%, and 50% are assumed as unlabeled samples. However, in each round of experiment, the trained model is evaluated with the same test set TS_l. For partially labeled datasets, all the aforementioned collected labeled samples (TR_l) are considered for the initial training set to design the classifier. In the SSSL process, varying number of unlabeled samples (i.e., the rest samples of the RS image) are collected based on the candidate sample selection criteria. The trained model is then tested on the whole samples of the RS images for samplewise land cover classification. As the class labels of all samples of RS image is not known, clustering indexes are used to validate the model performance.

1) Model 1: Semisupervised NN with similarity matrix G_1 [8] (see Section II-A1).
2) Model 2: Semisupervised GNN with maximum membership based adaptive rule generation (SSGNN-ARG) and similarity matrix G_1 [8] (see Section II-A1).
3) Model 3: SSGNN-ARG with similarity matrix G_2 (see Section II-A2).
4) Model 4: SSGNN-ARG with similarity matrix G_4 (see Section II-A4).
5) Model 5: Proposed model of SSGNN-ARG with similarity matrix G_5 (see Section II-A5).

All these models (Models 1–5) have used the SSSL Algorithm 1 for training purpose. Because of the random initialization of network parameters, and selection of labeled and unlabeled samples, experimental results vary with each iteration, although rest of the conditions are kept the same. This situation is handled by repeating each set of experiment ten times keeping all the conditions same and the average of them is taken as the final result. With this ten-fold validation approach, the generalization ability of the model is increased for improved performance.

TABLE II
Performance Comparison of Models With LISS-II Data

Model	OA with % of labeled samples from TR_l						For 50% labeled data		
	10	20	30	40	50	All samples of TR_l with NN base classifier	KC	OA_{std}	T_c
1	61.80	75.33	85.80	84.37	91.76	94.74	0.9160	9.6716	15.758
2	62.49	76.53	86.01	85.43	92.28		0.9200	3.0659	10.090
3	63.70	77.96	86.19	88.08	92.71		0.9240	2.9586	10.842
4	65.91	79.99	86.37	88.33	93.90		0.9320	2.7162	10.854
5	66.80	83.18	89.14	89.01	95.56		0.9560	0.8211	10.859

B. Land Cover Classification of Multispectral RS Image

For LISS-II MSRS dataset only, complete critical analysis and discussion of classification performance is presented because similar trend of performance in terms of various performance indexes was obtained with all the considered datasets.

1) IRS LISS-II Dataset:

a) With completely labeled samples: With five different classification models, LISS-II data are classified and corresponding results in terms of OA, KC, OA_{std}, and T_c are shown in Table II. To train these models, five different percentages of labeled and

unlabeled sample are considered. For 50% labeled data (i.e., obtained from TR_l) only, Table II depicts the results of five models, because similar trend of performance was observed for the rest percentages. Model 1 has used similarity matrix G_1 in the learning process of the NN (the base classifier). Models 2–5 (proposed) employed GNN (see Section II-D3) as the base classifier, though the neighborhood information used in these models are different.

It can be observed from the results (see Table II) that for all percentages of labeled samples, Model 2 with GNN performed better than Model 1 with conventional NN. The claim for the improvement of Model 2 over Model 1 is also supported by the KC and OA_{std} values shown for 50% of the labeled samples. This clearly justifies the advantages of GNN. The advantages includes: computational efficiency, simplicity, interpretability, and domain knowledge association (because of the customizable network with adaptive fuzzy rule generation). Computational complexity of Model 2 is actually attributed to the architecture of GNN, which is partially connected. These factors become responsible for an interesting fact, i.e., Model 2 yielded better classification accuracy than Model 1 and as well as with less computational complexity.

In a comparative analysis of models with respect to the effect of similarity matrix for spatial information extraction, Model 3 with G_2 (see Section II-A2) performed better than Model 2 with G_1 (see Section II-A1). For example, Model 3 yielded OA value of 63.70 that is around 1% more than the OA value obtained with Model 2, using 10% of initial training samples. Similarly, for other percentages of training samples also, Model 3 performed relatively better than Model 2 and Model 1. The results are also corroborated by the KC and OA_{std} values, as shown in Table II for 50% of training samples. This analysis shows the worth of considering G_2 that works with the mutual criteria of neighborhood information extraction, unlike the one-way criteria adopted in G_1.

As described in Section II-A3, we have developed the similarity matrix G_3 with the help of class map of unlabeled samples. The information in G_3 is clubbed with both the matrixes G_1 and G_2 using the logical AND operation to develop the matrixes G_4 and G_5, respectively. In order to justify the significance of G_4 and G_5, Models 4 and 5 are developed, respectively, and applied on the LISS_II dataset, and corresponding results are depicted in Table II. It is observed that Model 4 achieved improved performance than Model 2 that uses G_1. Intuitively, it makes sense because Model 4 uses the similarity matrix G_4, which is a strategic combination of information present in G_1 and G_3. This performance of Model 4 is further improved through Model 5 (proposed model) using the matrix G_5, which combines the information of G_2 and G_3. Model 5 outperformed all the four models, and the performance improvement can be seen from Table II. This analysis potentially rationalizes the significance of mutual neighborhood criteria obtained in G_2 and further improvement through the combined information obtained in the proposed similarity matrix G_5. The efficacy of the proposed model is also validated with KC and OA_{std}.

In addition to the OA and KC indexes, performance of the five models are analyzed in terms of computation time T_c, as shown in Table II. As expected, the Model 1 that uses NN as the base classifier takes more time to compute because of its fully connected architecture. On the other hand, GNN-based models (Models 2–5) performed superior to Model 1 in terms of accuracy and at the same time with less computation time, because of the partial connectivity of the GNN. Model 5 and Model 4 take similar amount of time but little more time than Model 2 and Model 3 because of the computation incurred during the design of similarity matrixes.

The experimental results shown in Table II also provides one more critical angle of the superiority of the proposed semisupervised model over the supervised model. In the experiment, part of the whole training samples (TR_l) is used as labeled and rest part as unlabeled to train all models through a strategic manner following the semisupervised principle. In an attempt to analyze semisupervised models with their supervised version, we have trained a conventional NN-based (used as base classifier for Model 1) classification model with whole training samples (TR_l), with no unlabeled samples, and tested on the samples of the test set TS_l. Furthermore, its performance is compared with all semisupervised models (Models 1–5) in terms of OA. The OA obtained is 94.74% with the supervised NN model. The accuracy is higher than the accuracy obtained with Models 1–4, but less than the proposed model (Model 5). However, it is to be noted that the supervised NN model has used 100%, whereas the semisupervised models have used maximum of 50% of labeled samples of TR_l. In spite of this, the proposed Model 5 performed better than the supervised NN model. This way of performance comparison justifies that the proposed semisupervised classification model performed superior to the supervised model, which uses randomly selected training samples.

b) Performance of Model 5 with very few labeled samples: In the aforementioned section, we have discussed the performance of models trained with 10% of the total samples TR_l, as the initial size of labeled samples. Here, we have performed the experiment with the proposed Model 5, using very few number of initial training samples (equal number of samples are taken from each class), such as 12, 30, etc., that are equal to 2%, 5% of TR_l, respectively. In the semisupervised classification, the subsequent training samples (till the size reaches to 50% of TR_l) are being added to the initial training samples, based on model's strategy. We have performed this experiment with only Model 5, as similar trend of performance was observed with rest models. Performances of Model 5 with final training samples (i.e., equal to the size of 50% of TR_l), for different sizes of initial training samples, are tested with the same set (TS_l) of test samples, and corresponding results are depicted in Fig. 4. It can be seen from the results shown in Fig. 4 that Model 5 provided higher classification accuracy, when the initial training sample size is 10% of TR_l. Keeping this result in mind, we have shown the detail experimental results with 10% of initial training sample size.

Superiority of the proposed model on the basis of performance indexes, such as precision, recall, and DS is shown in Table III.

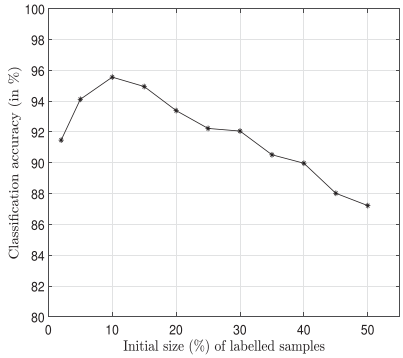


Fig. 4. Accuracies (with 50% labeled samples) of Model 5 with different initial sizes of training set and tested with the same set (TS_l) of test samples.

TABLE III
PRECISION, RECALL, AND DS FOR LISS-II DATA

Model	Class	For 50% labeled data		
		Precision	Recall	DS
1	1	1.0000	0.9900	0.1281
	2	0.9892	0.9200	0.1781
	3	0.9579	0.9100	0.1281
	4	0.9174	1.0000	0.0781
	5	0.9681	0.9100	0.1781
	6	0.9000	0.9900	0.1281
2	1	1.0000	1.0000	0.1157
	2	1.0000	0.8800	0.2157
	3	0.9872	0.7700	0.2157
	4	0.8547	1.0000	0.1157
	5	0.9184	0.9000	0.2157
	6	0.8403	1.0000	0.1157
3	1	1.0000	1.0000	0.0968
	2	1.0000	0.9400	0.1968
	3	1.0000	0.7600	0.1968
	4	0.8333	1.0000	0.0968
	5	0.9300	0.9300	0.1468
	6	0.9000	0.9900	0.1468
4	1	1.0000	1.0000	0.0878
	2	1.0000	0.9300	0.1878
	3	1.0000	0.7700	0.1878
	4	0.8621	1.0000	0.0878
	5	0.9143	0.9600	0.1378
	6	0.9174	1.0000	0.0878
5	1	1.0000	1.0000	0.0610
	2	1.0000	0.9700	0.1610
	3	1.0000	0.8700	0.1610
	4	0.9167	0.9900	0.1110
	5	0.9314	0.9500	0.1110
	6	0.9434	1.0000	0.0610

The performance improvement can be seen from the improved values of precision and recall for all the classes from Model 1 to Model 5. Also, it is observed from Table III that DS values for different classes are decreased from Models 1–5. This improvement in performance, i.e., the decrease values of DS is attributed to the efficiency of the proposed Model 5 because of improved neighborhood information extraction through the similarity matrix G_5, which is a combined information of the similarity matrixes G_2 and G_3. This analysis indicates the improved classwise performance of the proposed semisupervised rule based GNN model.

c) Performance comparison of the proposed models using base classifiers KREGNN and adaptive fuzzy rule based GNN, with different sample selection criteria: So far, we compared the performance of models with respect to the proposed SSSL method and random selection of training samples. Here, we aim to make the comparison of proposed SSSL method with other candidate sample selection criteria, such as random sampling of candidate samples (RanS), mutual information (MI) [34], [35], breaking ties (BTs) [36], and modified BTs (MBTs) [37]. Randomly, samples are selected using RanS method from the candidate sample set. With MI, active label sample selection is made that maximizes the MI between the classifier and class labels. BT method of sample selection is based on the smallest difference of the confidence level/posterior probability for each sample. In a scenario of multiclass problem, the method calculates the difference between the two highest probabilities. Hence, BT method selects the samples that minimizes the distance between the first two most probable classes. The MBT method of sample selection is more generalized than the BT method, where it finds the samples that maximizes the probability of the largest class for each individual class.

TABLE IV
PERFORMANCE COMPARISON OF MODELS WITH DIFFERENT SAMPLE SELECTION CRITERIA

Model	OA for different % of labeled samples from TR_l	
	30	50
KREGNN with proposed SSL	88.38	93.84
KREGNN with RanS	79.97	86.86
KREGNN with BTs	85.89	91.02
KREGNN with MBTs	85.99	91.15
Adp-REGNN with proposed SSL	89.34	95.56
Adp-REGNN with RanS	80.39	88.33
Adp-REGNN with BTs	87.84	92.05
Adp-REGNN with MBTs	88.05	93.14

For the comparison with respect to sample selection criteria, we have considered two base classifiers, such as the proposed GNN with maximum membership value based fuzzy rules that are generated adaptively (Adp-REGNN) and other is the GNN with Kasavov-rules (KREGNN) [28]. It was experimentally shown in Table IV that performance of models with MBTs-based selection criteria yielded little better that BTs-based criteria but certainly better than the RanS-based criteria. However, the proposed similarity matrix based candidate sample selection criteria outperformed the rest, which is justified with both 30% and 50% of labeled samples taken from TR_l. Significance of neighborhood information extraction using the proposed similarity matrix is clearly validating the superiority of the method and in turn, enhancing the accuracy of the semisupervised model. Table IV also validates the improvement of the semisupervised model using Adp-REGNN over KREGNN, as the base classifier.

d) Performance comparison of the proposed model with state of the art methods using completely labeled samples: Classification performance of the proposed semisupervised model (Model 5) is compared with other four semisupervised models (Models 1–4) in the aforementioned section under different scenarios. In this section, we have compared the performance of the proposed model with the state-of-the-art semisupervised methods, such

TABLE V
PERFORMANCE COMPARISON OF THE PROPOSED MODEL WITH THE STATE-OF-THE-ART METHODS USING LISS-II DATASET

Model	OA for different % of labeled samples from TR_l	
	30	50
TSVM [15]	76.88	82.55
LapSVM [16]	83.33	87.16
MLP-NN with existing SSL [10]	85.80	91.76
MLP-NN-based SSL with NGL [19]	86.92	92.69
KREGNN with [22] proposed SSL	88.38	93.83
Adp-REGNN with proposed SSL (Model 5)	89.34	95.56
SS-SSL [11] (Model 6)	85.27	90.25
SS-LapSVM [40] (Model 7)	85.19	90.32
PS^3VM [13] (Model 8)	85.44	91.33
MPM-LBP [41] (Model 9)	86.55	91.54
SS-ELM [42] (Model 10)	88.14	93.26
KNN ($k = 3$) with improved SSL [43] (Model 11)	86.02	91.79
Ensemble of MLP-NN with improved SSL [43] (Model 12)	88.47	94.01

as TSVMs [13], LapSVM [14], [15], neighborhood sample-based SSSL with multilayered perceptron NN (MLP-NN) classifier [8], and SSSL MLP-NN classifier with NGL [17]. The GNN used as the base classifier for Models 2–5 of Table II is designed on the basis of fuzzy $if - then$ rules that are generated from the domain knowledge. We have used the rule generation method using maximum membership value of linguistic variables to infer rules for each class, as discussed in Section II-C and named as Adp-REGNN in Table IV. Using another GNN with Kasavov-rules (KREGNN) [28] and the proposed SSSL is also compared in this section.

The other compared models are Models 6–12. Model 6 is developed [9] with a different approach for SSSL using probabilistic SVM, where the method adapts the available active learning methods to a self-learning framework. Here, a spatial neighborhood criterion is used to derive new candidate samples, which are spatially adjacent to the available (labeled) samples. Model 7 considers the method described by Yang *et al.* [38] that develops a spatio-spectral LapSVM (SS-LapSVM) for semisupervised classification of RS images. Model 8 is a progressive semisupervised SVM [11] with the integration of concepts that are used in conventional active learning methods. Model 9 [39] is the spectral and spatial classification method using loopy belief propagation and active learning. Model 10 [40] is an SSSL model with extreme learning machine (ELM) as the base classifier. Model 11 is an improved SSSL approach based on the DS [31], and Model 12 is the ensemble of four MLP-NNs with the improved SSSL approach based on the DS [31]. The classification results of LISS-II dataset in terms of OA with 30% and 50% labeled samples for the aforementioned semisupervised strategies and proposed model are illustrated in Table V.

It can be inferred from the comparative analysis that the proposed model surpassed the performance of kernel methods, i.e., TSVM and LapSVM. Also, it provided better classification accuracy than SSSL MLP-NN [8] and SSSL MLP-NN with NGL [17], and the advanced methods, such as Models 7 and 8. In kernel methods, LapSVM is computationally efficient and its performance is improved over TSVM, as indicated by OA values (see Table V). Ratle *et al.* [8] provided a better solution than kernel methods to handle large-scale RS dataset and demonstrated improved performance in SSSL approach. Furthermore, Im and Taylor [17] improved the semisupervised classification performance by the incorporation of NGL (requires separate MLP-NN architecture) with MLP-NN, but with complex network architecture. This is primarily due to the fact that the features in the upper layers are typically used in complex ways compared to the lower layers and it is hard to understand the relationship between the representations learned at each layer. The GNN-based semisupervised model using KRE method of rule extraction gives a comparable performance with TSVM for 30% labeled training samples and improved performance than both the kernel methods, i.e., TSVM and LapSVM for 50% labeled training samples. The OA of the proposed model is better than the OA obtained with KREGNN model. Furthermore, the proposed model (Model 5) is compared with three other models (Models 10–12). Among these tree models, Model 12 is performing better with the improved SSSL approach that also uses the ensemble of four MLP-NNs, as the base classifiers. From this observation, it can be concluded that the maximum membership-based rule extraction method can extract better rules than the KRE method [28]. With all aforementioned comparisons, it can be said that the proposed model (Model 5) with comparable computational cost and less complex architecture provided improved and encouraging performance compared to various state-of-the-art methods for the considered RS datasets.

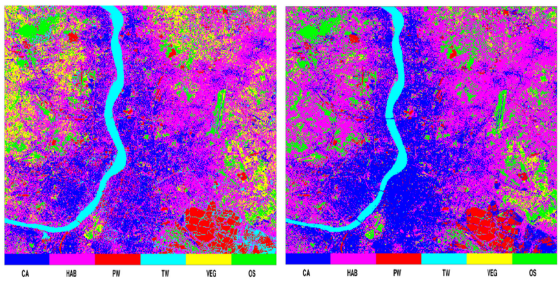
e) Performance analysis of the proposed model with respect to the impacts of its different parameters: So far, we have compared and analyzed the performance of models with respect to different aspects in the processing steps, such as different ratios of training and testing samples, different initial labeled samples require to start the experiment, sample selection methods, different spatial information extraction approaches, and different base classifiers. In this section, we would like to analyze the performance of the proposed model with the impacts of its different parameters, such as the value of Q for evaluating the neighborhood criteria in similarity matrix G_2, value of N in the k-NN classifier for estimating the class labels in G_3 and learning rate η for the NN.

We performed the experiment with different values of Q (i.e., 2,3, ...,10) to evaluate the similarity matrix G_2, but the best one that we considered in the result section is $Q = 6$. The best parameter value for k in k-NN was found to be $k = 5$. The learning rate that performed well for the present experimental study was found to be $\eta = .003$.

f) Performance comparison of the proposed model with others using partially labeled samples: Along with the limited number of labeled samples, SSSL methods take the advantage of unlabeled samples to derive a better hypothesis in designing a classification model. In this section, semisupervised models (Models 1–5) are analyzed on partially labeled datasets, i.e., with limited number of labeled samples only, and having huge amount of unlabeled samples, unlike in completely labeled case, where we have assumed the labeled samples as unlabeled.

TABLE VI
LISS-II IMAGE CLASSIFICATION

Labeled samples	Unlabeled samples	β			DB		
		model 1	model 2	model 5	model 1	model 2	model 5
1200	0	4.59	4.84		3.70	3.36	
1200	5000	4.73	4.96	5.40	3.57	2.04	1.86
1200	10,000	5	5.32	5.98	3.29	1.88	1.30

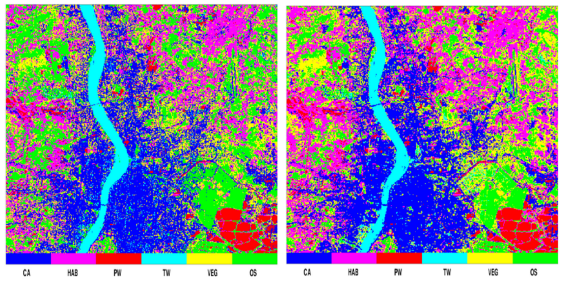


(a) (b)

Fig. 5. LISS-II image classified using (a) Model 2 and (b) Model 5.

TABLE VII
PERFORMANCE COMPARISON OF MODELS WITH LISS-III DATA

Model	OA with different % of labeled data					For 50% labeled data	
	10	20	30	40	50	KC	OA_{std}
1	59.79	73.21	85.87	88.71	91.01	0.8967	4.8734
2	61.99	79.49	85.94	91.01	92.18	0.8834	1.7723
3	63.36	80.83	86.61	91.94	92.36	0.8962	1.0361
4	64.33	82.35	87.09	93.04	92.74	0.9002	1.0931
5	65.93	84.84	88.07	94.01	93.39	0.9088	1.0492



(a) (b)

Fig. 6. LISS-III image classified using (a) Model 2 and (b) Model 5.

The base classifier of each model is trained with all of the labeled samples 1200 (see Section IV-A) and strategically selected fixed number of unlabeled samples (i.e., 5000 and 10 000) from LISS-II RS image. The respective similarity matrixes are used by the models, as described in Section II-A. The trained models are then applied for samplewise land cover classification of the whole image present in the considered area of RS. In this scenario, the performances of the models are evaluated using DB index and β index, and depicted in Table VI. The comparison is shown between Models 1, 2, and 5 (proposed) because similar trend of performance was obtained with Models 3 and 4. It is observed with the values of DB index and β index that performances of all three models are improved with the increase in unlabeled samples for training. Also, β value obtained with the proposed model is the highest among the other two models with both (5000 and 10 000) number of unlabeled samples. This observation justifies that the proposed model selects and utilizes the unlabeled samples in an efficient manner.

The classified LISS-II images by Models 2 and 5 with 1200 labeled and 5000 unlabeled samples are shown in Fig. 5(a) and (b), respectively, for qualitative performance comparison. From Fig. 5(b), it is observed that different land cover classes with Model 5 are more discretely classified than the image shown in Fig. 5(a) with Model 2. These results thus reveal the significance of the proposed model (Model 5) that fulfills the purpose of semisupervised model in the RS image classification.

2) IRS LISS-III Dataset

The classification performance results using all five models (Models 1–5) for LISS-III dataset in terms of OA, KC, and OA_{std} are depicted in Table VII. The proposed model (Model 5) trained with different percentages of labeled and unlabeled samples, outperformed all four models, i.e., Models 1–4. Model 5, as shown in Table VII, yielded an increase of around 2.5% in OA over Model 1, which has used NN as the base classifier. This improvement in classification performance clearly indicates the superiority of GNN over NN and the effect of improved extracted neighborhood information. The results also revealed that Model 5 could produce better classification performance over other GNN-based models (Models 2–4). The reason for this improvement is the incorporation of improved neighborhood information captured through the similarity matrix G_5. Similar trend of performance in terms of various measurement indexes, as discussed with LISS-II dataset, is also found true for this dataset.

We have also classified the whole LISS-III image with partially labeled samples. To train the semisupervised model, all the labeled samples (i.e., 1200) of LISS-III dataset along with a fixed number (5000 and 10 000) of strategically selected unlabeled samples are taken for the training set. With 1200 labeled and 5000 unlabeled samples, the classification maps of LISS-III image, obtained with Models 2 and 5, are shown in Fig. 6(a) and (b), respectively. It can be observed visually that the classified map of Model 5 is better and can segregate discretely distinguishable regions compared to the map obtained with Model 2. This observation corroborates the results obtained in Table VII.

C. Land Cover Classification of Hyperspectral RS Image

1) $Indian\ Pines$ Dataset From AVIRIS Sensor: Land cover classification results of hyperspectral dataset ($Indian\ Pines$) in terms of OA, KC, and OA_{std} are presented in Table VIII. The proposed Model 5 yielded an increase in OA values of 6% to 15% in the classification of land covers over Model 1. It can be observed from Table VIII that the improvement in OA vary with the number of labeled and unlabeled samples taken for training. In Model 1, input to the conventional NN is 200 bands of hyperspectral data ($Indian\ Pines$), this high-dimensional input make the network architecture complex and computationally inefficient. On the other hand, for Model 5 (proposed model), GNN is used as the base classifier, which

TABLE VIII
PERFORMANCE COMPARISON OF MODELS WITH $Indian\ Pines$ DATA OBTAINED FROM AVIRIS SENSOR

Model	OA for different % of labeled samples from TR_l					For 50% labeled samples	
	10	20	30	40	50	KC	OA_{std}
1	60.12	63.22	66.74	68.90	72.04	0.6212	2.4412
2	70.16	73.11	75.24	77.89	79.46	0.6700	2.0359
3	72.20	74.55	77.43	79.44	80.21	0.6899	1.1045
4	73.41	75.99	78.98	80.23	81.88	0.7002	1.5221
5	75.82	78.03	80.21	82.20	84.03	0.7456	0.7922

TABLE IX
PERFORMANCE COMPARISON OF MODELS WITH $Pavia\,University$ DATA OBTAINED FROM ROSIS SENSOR

Model	OA for different % of labeled samples from TR_l					For 50% labeled samples	
	10	20	30	40	50	KC	OA_{std}
1	78.19	78.78	80.16	81.68	83.13	0.611	3.59
2	78.98	79.56	81.77	83.67	85.71	0.655	3.01
3	79.57	81.19	83.54	85.30	86.01	0.700	2.67
4	80.86	82.55	85.03	85.99	87.54	0.709	1.95
5	82.34	84.99	86.07	87.46	89.15	0.781	1.12

is computationally efficient, simple, and interpretable in terms of architecture because of the partially connected network. Also, improved neighborhood information obtained through the similarity matrix G_5 is used in Model 5, which makes the learning process better. In the same line of reasoning, the KC value of Model 5 with 50% labeled is improved compared to the rest models. This improvement is because of the similarity matrix G_5. In addition, KC values of the proposed model are also improved with the proposed model. Various performance analysis of the proposed model (Model 5) discussed in the case of LISS-II dataset is also held true for this dataset in various scenarios (partially and completely labeled dataset) and with different measuring parameters. The advantage of using improved neighborhood information in the form of similarity matrix G_5 and rule-based GNN as the base classifier are actually attributed for the improved classification performance of the proposed Model 5 with $Indian\ Pines$ dataset.

2) $Pavia\,University$ Data From ROSIS Sensor: Similar to AVIRIS sensor based $Indian\ Pines$ data, land cover classification results of $Pavia\,University$ data (i.e., obtained from ROSIS sensor) in terms of OA, KC, and $\mathrm{OA_{std}}$ are illustrated in Table IX. It is observed from the table that Model 5 with the similarity matrix (G_5) yielded the highest classification accuracy in terms OA compared to other models, those are based on different similarity matrixes. The significance of G_5, which is the collaborative contribution of information from the matrixes G_2 and G_3, is clearly visible with this dataset. It is also true for all illustrated percentages of labeled samples. In addition, the performance in terms of KC value revealed the similar trend. Various critical performance analysis of Model 5 discussed in the case of LISS-II dataset is also similar with this dataset in terms of trend for various scenarios (partially and completely labeled dataset) and with different measuring indexes. Advantage of using improved neighborhood information in the form of similarity matrix G_5 and rule-based GNN as the base classifier are actually attributed for the improved classification performance of the proposed Model 5 with $Pavia\,University$ dataset.

VI. CONCLUSION

This article proposed an SSSL model to classify multispectral and hyperspectral RS datasets. The proposed model used the rule-based GNN as the base classifier, which had taken the support through the extracted neighborhood information of the unlabeled samples for SSSL. The quality of neighborhood information was improved by using the collaborative opinion of two independent neighborhood information extraction methods. With the first method mutual neighborhood information of unlabeled samples was captured and the second method discovered the neighborhood on the basis of class-map of unlabeled samples. To take the advantages of both the methods, mutual agreement was considered as the final neighborhood information, which was used in the learning process of the base classifier. The interconnections of GNN are partial in nature that was established according to the adaptive fuzzy *if–then* rules. We also realized the advantages of the base classifier, which is computationally efficient, less complex, interpretable, and manageable in comparison to conventional NN. The proposed model justified its superiority on the basis of various performance parameters in the classification of both multispectral and hyperspectral RS datasets. Comparison of the proposed model with other state-of-the-art methods and different candidate sample selection criteria was made. Along with these comparisons, we also compared the model with a GNN-based model that used the KRE method of rule extraction in designing the GNN architecture. Experimentally, it was analyzed that the proposed method required less computational time comparatively, and at the same time yielded superior classification performance compared to similar other methods.

ACKNOWLEDGMENT

The authors would like to thank Prof. D. Landgrebe and Prof. P. Gamba for providing $Indian\ Pines$ and $Pavia\,University$ datasets. The authors would also like to thank the Editors and the anonymous reviewers for their constructive comments and suggestions to improve the overall quality and presentation of this article.

REFERENCES

[1] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[2] K. Makantasis, K. Karantzalos, A. Doulamis, and K. Loupos, "Deep learning-based man-made object detection from hyperspectral data," in *Proc. Int. Symp. Vis. Comput.: Adv. Vis. Comput.*, 2015, pp. 717–727.

[3] K. Makantasis, A. D. Doulamis, N. D. Doulamis, and A. Nikitakis, "Tensor-based classification models for hyperspectral data analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 6884–6898, Dec. 2018.

[4] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[5] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.
[6] G. Camps-Valls, T. V. B. Marsheva, and D. Zhou, "Semi-supervised graph based hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3044–3054, Oct. 2007.
[7] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.
[8] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2271–2282, May 2010.
[9] I. Dopido, J. Li, P. R. Marpu, A. Plaza, J. M. B. Dias, and J. A. Benediktsson, "Semisupervised self-learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 4032–4044, Jul. 2013.
[10] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, Nov. 2005.
[11] C. Persello and L. Bruzzone, "Active and semisupervised learning for the classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6937–6956, Nov. 2014.
[12] L. Wan, K. Tang, M. Li, Y. Zhong, and A. K. Qin, "Collaborative active and semisupervised learning for hyperspectral remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2384–2396, May 2015.
[13] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for semisupervised classification of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3363–3373, Nov. 2006.
[14] L. Gomez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe, "Semisupervised image classification with Laplacian SVMs," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 3, pp. 336–340, Jul. 2008.
[15] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res*, vol. 7, pp. 2399–2434, Dec. 2006.
[16] L. Wang, S. Hao, Y. Wang, Y. Lin, and Q. Wang, "Spatial spectral information-based semisupervised classification algorithm for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 8, pp. 3577–3585, Aug. 2014.
[17] D. J. Im and G. W. Taylor, "Semi-supervised hyperspectral image classification via neighborhood graph learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1913–1917, Sep. 2015.
[18] W. Pedrycz and G. Vukovich, "Granular neural networks," *Neurocomputing*, vol. 36, pp. 205–224, 2001.
[19] S. K. Meher and D. A. Kumar, "Ensemble of adaptive rule-based granular neural network classifiers for multispectral remote sensing images," *IEEE Trans. Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2222–2231, May 2015.
[20] D. A. Kumar, S. K. Meher, D. Kanhar, and K. P. Kumari, "Unified granular neural networks for pattern classification," *Neurocomputing*, vol. 216, pp. 109–125, 2016.
[21] M. Cui, S. Prasad, W. Li, and L. M. Bruce, "Locality preserving genetic algorithms for spatial-spectral hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1688–1697, Jun. 2013.
[22] X. Jia and J. A. Richards, "Managing the spectral-spatial mix in context classification using Markov random fields," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 311–314, Apr. 2008.
[23] M. Ahmad *et al.*, "Spatial prior fuzziness pool-based interactive classification of hyperspectral images," *Remote Sens.*, vol. 11, 2019, Art. no. 1136.
[24] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, pp. 338–353, 1965.
[25] D. Sánchez and E. Trillas, "Measures of fuzziness under different uses of fuzzy sets," in *Advances in Computational Intelligence*. Berlin, Germany: Springer, 2012, pp. 25–34.
[26] M. Ahmad, S. Protasov, A. Khan, R. Hussain, A. M. Khattak, and W. Khan, "Fuzziness-based active learning framework to enhance hyperspectral image classification performance for discriminative and generative classifiers," *Plos One*, vol. 13, Jan. 2018, Art. no. e0188996.
[27] A. Vasilakos and D. Stathakis, "Granular neural networks for land use classification," *Soft Comput.*, vol. 9, pp. 332–340, 2005.
[28] N. Kasabov, "Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems," *Fuzzy Set Syst.*, vol. 82, pp. 135–149, 1996.
[29] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
[30] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2006, pp. 1735–1742.
[31] S. K. Pal, S. K. Meher, and S. Dutta, "Class-dependent rough-fuzzy granular space, dispersion index and classification," *Pattern Recognit.*, vol. 45, pp. 2690–2707, 2012.
[32] S. K. Pal, A. Ghosh, and B. U. Shankar, "Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation," *Int. J. Remote Sens.*, vol. 21, no. 11, pp. 2269–2300, 2000.
[33] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
[34] D. Mackay, "Information-based objective functions for active data selection," *Neural Comput.*, vol. 4, pp. 590–604, 1992.
[35] B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo, "On semi-supervised classification," in *Proc. 18th Annu. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2004, pp. 721–728.
[36] T. Luo *et al.*, "Active learning to recognize multiple types of plankton," *J. Mach. Learn. Res.*, vol. 6, pp. 589–613, 2005.
[37] J. Li, J. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new Bayesian approach with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3947–3960, Oct. 2011.
[38] L. Yang, S. Yang, P. Jin, and R. Zhang, "Semi-supervised hyperspectral image classification using spatio-spectral Laplacian support vector machine," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 3, pp. 651–655, Mar. 2014.
[39] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 844–856, Feb. 2013.
[40] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.
[41] S. K. Meher, "Semisupervised self-learning granular neural networks for remote sensing image classification," *Appl. Soft Comput.*, vol. 83, 2019, Art. no. 105655.

Neeta S. Kothari (Student Member, IEEE) received the B.E. degree in electronics engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India, in 2009, and the M.E. degree from Devi Ahilya Vishwavidyalaya, Indore, India, in 2013, both in electronics engineering.

She is currently a Visiting Scholar with Systems Science and Informatics Unit, Indian Statistical Institute, Bangalore, India. Her research interests include pattern recognition, and remote sensing.

Saroj K. Meher (Senior Member, IEEE) received the Ph.D. degree in electronics engineering from the National Institute of Technology, Rourkela, India, in 2003.

He is currently an Associate Professor with Systems Science and Informatics Unit, Indian Statistical Institute, Bangalore, India. His research interest includes pattern recognition, granular computing, and computational intelligence.

Ganapati Panda (Senior Member, IEEE) received the Ph.D. degree in electronics and communication engineering from the Indian Institute of Technology (IIT) Kharagpur, Kharagpur, India, in 1982.

He was a Postdoctoral Researcher with The University of Edinburgh, Edinburgh, U.K., from 1984 to 1986. He is currently working as a Research Advisor with the C. V. Raman College of Engineering, Bhubaneswar, India. He is also a Professorial Fellow with the Department of Electrical Sciences, IIT Bhubaneswar. He is currently a Distinguished Professor with AICTE-INAE of India. His research interests include digital signal processing, digital communication, and machine learning.

Prof. Panda is a fellow of the National Academy of Engineering India and the National Academy of Science India.