Nonlinear Dimensionality Reduction for Data Visualization: An Unsupervised Fuzzy Rule-Based Approach

Suchismita Das and Nikhil R. Pal, *Fellow, IEEE*

***Abstract*—In this article, we propose a general framework for the unsupervised fuzzy rule-based dimensionality reduction primarily for data visualization. This framework has the following important characteristics relevant to the dimensionality reduction for visualization: preserves neighborhood relationships; effectively handles data on nonlinear manifolds; capable of projecting out-of-sample test points; can reject test points, when it is appropriate; and interpretable to a reasonable extent. We use the first-order Takagi–Sugeno model. Typically, fuzzy rules are either provided by experts or extracted using an input–output training set. Here, neither the output data nor experts are available. This makes the problem challenging. We estimate the rule parameters minimizing a suitable objective function that preserves the interpoint geodesic distances (distances over the manifold) as Euclidean distances on the projected space. In this context, we propose a new variant of the geodesic *c*-means clustering algorithm. The proposed method is tested on several synthetic and real-world datasets and compared with the results of six state-of-the-art data visualization methods. The proposed method is the only method that performs equally well on all the datasets tried. Our method is found to be robust to the initial conditions. The predictability of the method is validated by suitable experiments. We also assess the ability of our method to reject test points when it should. The scalability issue of the scheme is also discussed. Due to the general nature of the framework, we can use different objective functions to obtain projections satisfying different goals. To the best of our knowledge, this is the first attempt to manifold learning using unsupervised fuzzy rule-based modeling.**

***Index Terms*—Fuzzy rules, geodesic distance, predictability, Takagi–Sugeno system (T–S system), visualization.**

I. Introduction

VISUALIZATION is one of the prominent exploratory data analysis schemes as it provides insights into the data. We come across high-dimensional data in various real-world problems related to, as examples, finance, meteorology, computer vision, medical imaging, multimedia information processing, and text mining [1]–[5]. However, plotting more than three dimensions directly is not feasible. Data visualization schemes provide ways to visualize high-dimensional data. They can be broadly divided into two categories. Methods in the first category provide some mechanism to display more than two dimensions graphically. The Chernoff faces [6] is an example of this category. The second category reduces the dimensionality of the data to two or three. They aim to represent the data in a lower dimension keeping the "relevant" information of the original data as intact as possible. Note that, the schemes under the first category do not explicitly extract/summarize any information of the data. On the other hand, dimensionality reduction-based schemes try to carry the information present in the original data to its lower dimensional representation. Dimensionality reduction for data visualization can be achieved in many ways such as principal component analysis (PCA) [7], multidimensional scaling (MDS) [8], and manifold learning [9]. Some of these methods are linear; for example, PCA, canonical correlation analysis [10], linear discriminant analysis [11], factor analysis [12], and locality preserving projections [13], which are not suitable if the dataset has nonlinear structures. Note that, a special case of linear projection is feature selection [14]–[17]. When data are projected by feature selection, the features in the reduced space maintain their original identity, while in other cases of projections, the new features are difficult to interpret. Nonlinear projections such as Sammon's method [18] and manifold learning algorithms preserve some geometric properties of the data. Although the physical meaning of such features is difficult to comprehend, they produce more useful visualization. A class of nonlinear data projection methods for visualization can be categorized as manifold learning. In a p-dimensional manifold, each point has a local neighborhood that is homeomorphic to the Euclidean space of the same dimension. In manifold learning for data visualization, the objective is to produce a low-dimensional (usually two or three) representation of the high-dimensional data by preserving the local neighborhood (local geometry) [19]–[22].

An important aspect is whether the dimensionality reduction method is equipped with predictability or not. Parametric methods such as PCA and multilayer autoencoder-based methods provide a direct mapping from the high-dimensional space to the low-dimensional space. Thus, the trained model can produce lower dimensional representations for any test points. With nonparametric methods, such predictions for new points are usually not possible in a *straightforward* manner. Fuzzy rule-based systems (FRBSs) are parametric models that are extensively used in different machine learning tasks such as control, classification,

Manuscript received March 26, 2020; revised January 16, 2021 and March 26, 2021; accepted April 19, 2021. Date of publication April 29, 2021; date of current version July 1, 2022. *(Corresponding author: Nikhil R. Pal.)*

The authors are with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700108, India (e-mail: suchismitasimply@gmail.com; nikhil@isical.ac.in).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TFUZZ.2021.3076583.

Digital Object Identifier 10.1109/TFUZZ.2021.3076583

and forecasting [23]–[25]. An FRBS learns a function from a given set of training points and directly predicts outputs for test points. FRBSs can model a nonlinear relationship between input and output. Moreover, they are easy to understand and develop. They provide rules that are "explainable"/"comprehensible" at least to some extent. For these characteristics, FRBSs seem to be suitable for realizing dimensionality reduction-based data visualization models. However, the literature is not rich in this area. Recently, Lughofer and Nikzad-Langerodi [26] proposed an interesting dimensionality reduction technique preserving the local neighborhood structure to design fuzzy rule-based systems. This method constructs a latent variable space having reduced dimensionality using a local structure preserving variant of the partial least-square method. Then, the FRBS is designed in that latent variable space to predict the target variables of the time-series data. However, in [26], the fuzzy system is not used to perform the dimensionality reduction task. Apart from the work in [27], which is a *supervised* method, we could not find any investigation employing FRBSs for data visualization applications through the dimensionality reduction. In this context, it is worth noting that there are many fuzzy-rough-set-based methods for feature selection [28], [29]. One can use any feature selection method, including the fuzzy-rough-set-based methods to select two or three features and use them for visualization. But such feature selection methods are neither intended for nor can do manifold learning where the local neighborhood structure needs to be preserved. Moreover, these methods are also not fuzzy rule-based methods.

A. Motivation, Challenge, and Novelty

Most of the unsupervised manifold learning methods for data visualization such as local linear embedding (LLE) [19], ISOMAP [20], and t-SNE [30], cannot project test data points in a *straightforward manner*, and the same is true for other nonlinear projection methods like Sammon's method. Later, some of these methods have been extended/adapted so that they can handle out-of-sample points. Some of these extended versions need the training data to deal with out-of-sample points. More importantly, these adapted versions *always* produce some output given *any* input, even when the test input is far from the sampling window of the training data. Thus, there is a need for dimensionality reduction methods that satisfy at least the following desirable characteristics: *D1:* are unsupervised in nature; *D2:* can preserve the *local geometry* of the data in the lower dimension; *D3:* can make a faithful projection of test data points in a *straightforward* manner without requiring the training data; *D4:* can refuse to project test points which are coming from an area far from the "sampling window" of the training data; and *D5:* are interpretable/understandable at least to some extent. Motivated by these goals, we decided to use a Takagi–Sugeno (T–S) [23] type fuzzy rule-based system because the antecedent of a fuzzy rule represents a small hyper-ellipsoidal region in the input space (i.e., captures the local geometry) and its consequent represents a local linear model for the projected data. This is exactly what manifold learning demands—projections preserving the local structure. Identification of fuzzy rule-based systems for classification/regression problems demands either experts to provide the rules or the use of labeled data where each instance in the training data should have a target output. However, here, neither we have the output data nor experts can provide such rules. This makes the problem challenging. Here, our objective is: given an unlabeled high-dimensional dataset, X, we want to find a lower dimensional representation, Y, of X so that the local neighborhood structure of X is preserved in Y. Due to the absence of target data, traditional approaches to extract fuzzy rule-based systems cannot be used. We propose to use a suitable objective function to design the fuzzy system. The novelty of the scheme lies in the fact that it extracts the desired fuzzy rules for nonlinear projection of data without using any target data. Moreover, it provides a *general* framework, which enjoys the following advantages.

1) Most of the unsupervised methods for the dimensionality reduction cannot project out-of-sample test data points in a *straightforward* manner, but ours can.
2) This is one of the few methods of nonlinear data projection that has the *reject* option.
3) Our method enjoys some level of interpretability and because of the structure of the fuzzy reasoning, it is not likely to make a poor generalization.
4) The proposed model is very robust to the initial condition.
5) It provides a general framework for the unsupervised dimensionality reduction. We can use different objective functions to preserve different characteristics of the original data in the projected space. For example, in [31], we have used Sammons' stress [18] as the objective function.
6) The modified form of the geodesic c-means clustering ensures that the cluster centers are always on the manifold, and thereby, providing better initial rules.
7) The proposed method performs better or comparable to several nonfuzzy methods.
8) Finally, to the best of our knowledge, this is the first unsupervised method based on fuzzy rules for nonlinear manifold learning.

II. Literature Review

Let $\mathbf{X} = \{\mathbf{x_i} = (x_{i1}, x_{i2}, \ldots, x_{id_h})^T \in \mathcal{R}^{d_h} : i \in \{1, 2, \ldots, n\}\}$ be the input dataset, where, n is the number of instances and d_h is the dimension of the data. To visualize the data, we want to map it to a d_l-dimensional space, where $d_l < d_h$. Let the lower dimensional data be represented by $\mathbf{Y} = \{\mathbf{y_i} = (y_{i1}, y_{i2}, \ldots, y_{id_l})^T \in \mathcal{R}^{d_l} : i \in \{1, 2, \ldots, n\}\}$. Dimensionality reduction methods project data in R^{d_h} to R^{d_l} keeping some relevant information of $\mathbf{X}$ as intact as possible in $\mathbf{Y}$. Generally, for visualization, d_l is chosen as two or three. Other than visualization, dimensionality reduction methods are also applied in data denoising, compressing, extracting suitable features for classification, clustering, and so on [2], [4], [32]. We discuss here only the methods which attempt to preserve the structure of the data and aid in visualization.

Dimensionality reduction methods could be divided into two groups: linear and nonlinear. The most extensively used linear method is PCA [7], [32], where the principal components correspond to the directions that explain maximum variance of the data. Maximum autocorrelation factors (MAFs) [33], [34] preserves temporally interesting structures present in the input to the projected output data. The MAF assumes that there is an

underlying d_l-dimensional temporal signal, which is smooth and the remaining $d_h - d_l$ dimensions are noise with little temporal correlation. However, there are datasets where linear projections cannot do the required job. For example, consider a dataset consisting of points from a sphere and a spherical shell surrounding the sphere in three or higher dimension, where the sphere and shell represent two classes. These two classes are well separated but no linear projection to two dimensions can make them separable, but a nonlinear projection like Sammon's method [18] can. A detailed review of various linear dimensionality reduction methods is available in [32]. For datasets where points lie near or on a nonlinear manifold, preserving the local distance structure is important, but linear mapping like PCA fails to represent the desirable characteristics of such data. Nonlinear dimensionality reduction methods like LLE [19] and ISOMAP [20] could be useful for such datasets.

A group of nonlinear methods tries to preserve the interpoint distances in the high-dimensional space as the interpoint distances in the estimated low-dimensional space. Examples of such methods are Sammon's projection [18], curvilinear component analysis [35], ISOMAP [20], and curvilinear distance analysis [36]. Sammon's projection minimizes the following cost function with respect to $\mathbf{y_i}$s:

$$E = \left(1/\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}\right) \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \left(d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|\right)^2 / d_{ij} \quad (1)$$

where d_{ij} represents the pairwise Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. Note that, the first term in (1) is a constant for any given dataset, and hence, can be dropped as it does not affect the optimization.

Sammon's projection, however, fails on datasets such as a Swiss Roll [19]. In Swiss Roll, the Euclidean distance between two points can be small but their distance over the manifold on which the data points reside can be large. To model such datasets, we should consider the pairwise distance computed using the distances over the manifold, i.e., the geodesic distance. In differential geometry, a geodesic is a curve defining the shortest path between a pair of points on a surface or in a Riemannian manifold. The length of the geodesic defines the geodesic distance between the two points. Thus, it is the shortest distance between two points while moving along the surface. Mathematically, the distance between two points p and q over a Riemannian manifold is defined to be the infimum of the lengths $L(\gamma)$ over all the piece-wise smooth curve segments γ from p to q [37].

ISOMAP is one of the methods that considers the geodesic distance as the distance measure for the input space. It applies the classical MDS over the input geodesic distance matrix to compute the lower dimensional embedding $\mathbf{Y}$. The vectors $\mathbf{y}_i;\ i = 1, 2, \ldots, n$, are estimated minimizing the cost function $E = \|\tau(\mathbf{D}_G) - \tau(\mathbf{D}_Y)\|_{L^2}$. Here, $\mathbf{D}_G$ denotes the matrix of geodesic distances over the input space, $\mathbf{D}_Y$ denotes the matrix of Euclidean distances over the output space, and $\|\mathbf{A}\|_{L^2} = \sqrt{\sum_{ij} A_{ij}^2}$. The τ operator converts distances to inner products [20].

Almost at the same time as that of ISOMAP, another manifold learning algorithm was introduced named LLE. For data points residing on or approximately on a manifold, the LLE assumes that points in a small neighborhood lie on a linear patch. The entire manifold is made of numerous such small linear patches. This local linear model for each data point is estimated in the LLE. The low-dimensional data representation is computed in two steps. First, w_{ij}s are estimated by minimizing $\Phi(\mathbf{w}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} w_{ij}\mathbf{x}_j\|^2$, where $\mathcal{N}(\mathbf{x}_i)$ represents the set of neighbors of the point $\mathbf{x}_i$. Second, after w_{ij}s are estimated, the lower dimensional representation $\mathbf{y}_i$s are estimated minimizing the objective function $\psi(\mathbf{Y}) = \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} w_{ij}\mathbf{y}_j\|^2$ keeping w_{ij}s constant.

The Laplacian Eigenmaps (LE) [22] is another approach similar to the LLE. Here, the local property is characterized by the distances of a point to its neighbors. The weight w_{ij} associated with a data point $\mathbf{x}_i$ and its neighbor $\mathbf{x}_j \in \mathcal{N}(\mathbf{x_i})$ is computed using (2).

$$\text{If } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i), w_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2} \text{ else } w_{ij} = 0 \quad (2)$$

where $\sigma > 0$. Then, the lower dimensional representations $\mathbf{y}_i$s are computed minimizing the cost function $\phi(\mathbf{Y}) = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2$. An alternative weight function is also used: $w_{ij} = 1$, if $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$; 0, otherwise.

Another method named t-SNE [30] is able to create a single map that reveals structure at many different scales. The t-SNE method converts the Euclidean interpoint distances in the high-dimensional input space to symmetrized conditional probabilities $p_{ij} = \dfrac{p_{j|i} + p_{i|j}}{2n}$, where $p_{j|i}$ signifies the conditional probability that $\mathbf{x}_i$ would pick $\mathbf{x}_j$ as its neighbor. The conditional probability $p_{j|i}$ follows a Gaussian distribution. Similarly, the joint probability of the lower dimensional outputs, $\mathbf{y}_i$ and $\mathbf{y}_j$, is represented by q_{ij} and it is considered to follow the student's t distribution with one degree of freedom. The lower dimensional representations $\mathbf{y}_i$s are obtained by minimizing the Kullback–Leibler divergence between the joint probability distributions of the input and its lower dimensional representation.

A very recent algorithm that is competitive with t-SNE is uniform manifold approximation and projection (UMAP) [38], [39]. This method constructs a topological representation of the high-dimensional data by approximating the manifold locally through local fuzzy simplicial set representations. Lower dimensional representations are estimated by minimizing the cross entropy between the said topological representations of higher and lower dimensional data. There are several other nonlinear methods. To name a few: maximum variance unfolding [40], diffusion map [41], Hessian LLE [21], and local tangent space analysis [42]. In this context, it is worth discussing a nonlinear variant of PCA [43], called the kernel PCA (KPCA). Here, the data are implicitly mapped into a high-dimensional space, and then, PCA is done in that high-dimensional space. The process does not require an explicit mapping but is realized using a kernel function. Despite being a nonparametric method, the KPCA can project out-of-sample points provided, all the training data points are available during the test phase.

Generally nonparametric methods, like LLE [19], ISOMAP [20], and t-SNE [30] cannot project out-of-sample points in a *straightforward* manner. There have been a number of attempts to extend such methods for the projection of out-of-sample points [44]–[47]. For example, in [44], the authors proposed some out-of-sample extensions of methods

like LLE and ISOMAP, where the problem of generalizing the embeddings to out-of-sample data points, is modeled as learning the eigenfunction of a kernel. Two extensions of the LLE are proposed in [45], while t-SNE is extended in [47] to project out-of-sample data points.

Next, we discuss some parametric models. In [48], the multilayer perceptron (MLP) is used for the structure preserving dimensionality reduction. In this work, Sammon's projections of the sampled inputs or of the input prototypes generated by a self-organizing map were used as target outputs of an MLP, which is trained by the usual gradient descent algorithm. The investigations in [49] and [50] used the MLP for the dimensionality reduction using Sammon's stress as the objective, while in [50], some other networks were proposed for data projection including one for nonlinear discriminant analysis.

In [51] and [52], multilayer autoencoders with an odd number of hidden layers were employed for the dimensionality reduction. In an autoencoder, to project the original data to d_l dimension, the middle hidden layer is chosen to have d_l nodes. After the autoencoder is trained, given an input $\mathbf{x}_i$, we take the output of the d_l-dimensional middle hidden layer as the projected vector $\mathbf{y}_i$. However, these models do not consider the interrelationship between data points and the underlying manifold structure. The work in [53] proposes a scheme called "generalized autoencoder" (GAE), which incorporates concepts of manifold learning. The GAE extends the traditional autoencoder in the following two aspects:

1) each instance $\mathbf{x}_i$ reconstructs a set of instances rather than reconstructing only itself;
2) the reconstruction error of an instance $\mathbf{x}_i$ is weighted by a relational function of $\mathbf{x}_i$ and $\mathbf{x}_j$ defined on the learned manifold.

GAE learns the underlying manifold by minimizing the weighted distance between the original and reconstructed data. In a very recent work [54], an autoencoder-based scheme incorporating manifold learning, termed diffusion net, is proposed. Here, a multilayer encoder is trained to encode the diffusion net embedding of the input. Then, a multilayer decoder is trained to reconstruct the original input from the encoder output. The encoder and decoder networks are stacked to form the final autoencoder network.

There are other methods to find explicit maps between high- and low-dimensional data. For example, in [55], the authors proposed an algorithm named neighborhood preserving polynomial embedding (NPPE). The method considers the objective function of the LLE as its learning objective where the lower dimensional output vectors are represented as a polynomial function of higher dimensional input vectors. The unknown coefficients of the polynomial function are estimated from the modified LLE objective function by solving a generalized eigenvalue problem.

Works exploring FRBSs to reduce the data dimensionality for visualization are scarce. In [27], a *supervised* fuzzy rule-based model is proposed for the structure preserving dimensionality reduction. Here, given the high-dimensional inputs and its lower dimensional projections (generated by any method such as Sammon's projection), a fuzzy rule-based system is used to learn the projecting map. In this context, both Mamdani–Assilian (MA) [56] and Takagi–Sugeno (T–S) [23] models have been used.

III. Proposed Method

We want to develop a system to project high-dimensional data for visualization. The system should have at least the desirable attributes D1 to D5 listed in Section I. To realize this, we consider an FRBS as the most appropriate modeling framework. Although an MA model could be used, considering greater flexibility we use the T–S model. Since this is an *unsupervised* method, the main challenge is to define a suitable objective function. The second challenge is how to generate the initial rule base. Typically, we use a clustering algorithm to generate the initial rules. But when the data points are on a manifold whose intrinsic dimension is lower than the original dimension of the data, the c-means or the fuzzy c-means algorithm may not be useful as these algorithms may (usually will) find cluster centers that are significantly away from the manifold.

As described in Section II, we denote the input dataset by $\mathbf{X} = \{\mathbf{x}_i; i = 1, 2, \ldots, n\}$ and the corresponding projected dataset by $\mathbf{Y} = \{\mathbf{y}_i; i = 1, 2, \ldots, n\}$. *Note that, only X is given and Y has to be estimated along with other system parameters.* Let the qth input variable be x_q and the mth output variable be y_m. Let, there be n_c rules for each output variable. For the T–S model, the kth rule for the mth output variable is of the form

$$R_{km}: \text{ If } x_1 \text{ is } F_{k1} \text{ AND } x_2 \text{ is } F_{k2} \text{ AND } \cdots \text{ AND } x_{d_h} \text{ is } F_{kd_h}, \text{ then } y_m^k = a_{km0} + \sum_{q=1}^{d_h} a_{kmq} x_q \tag{3}$$

where $k = 1, 2, \ldots, n_c$; $m = 1, 2, \ldots, d_l$; F_{kq} is the kth fuzzy set (linguistic value) defined on the qth input feature; and a_{kmq}'s are consequent parameters. Let us define the matrix $A = (A_1, A_2, \cdots A_{n_c})^T$, where $A_k = (a_{k10}, a_{k11}, \ldots, a_{k1d_h}, a_{k20}, a_{k21}, \ldots, a_{k2d_h}, \ldots, a_{kd_l0}, a_{kd_l1}, \ldots, a_{kd_ld_h})$. Let the firing strength of the rule R_{km} for the point $\mathbf{x}_i$ be $\alpha_{k,i}; k = 1, 2, \ldots, n_c$. Note that, for an antecedent, "x_1 is F_{k1} AND x_2 is F_{k2} AND $\cdots$ AND x_{d_h} is F_{kd_h}," there are d_l consequents, "$y_m^k = a_{km0} + \sum_{q=1}^{d_h} a_{kmq} x_q$"; $m = 1, 2, \ldots, d_l$, resulting in d_l rules. But the firing strength for each of these d_l rules remains the same ($\alpha_{k,i}$). The system output is computed as

$$y_{im} = \sum_{k=1}^{n_c} \left(\alpha_{k,i} / \sum_{p=1}^{n_c} \alpha_{p,i} \right) y_{im}^k \tag{4a}$$

$$= \sum_{k=1}^{n_c} \left(\alpha_{k,i} / \sum_{p=1}^{n_c} \alpha_{p,i} \right) \left(a_{km0} + \sum_{q=1}^{d_h} a_{kmq} x_{iq} \right) \tag{4b}$$

where $i = 1, 2, \ldots, n$; and $m = 1, 2, \ldots, d_l$. Typically, for designing an FRBS, the target outputs for the training data are known and the rule base parameters are estimated by minimizing the square error defined by the target outputs and the estimated outputs. But for us, the target output is *not* available. So, we need to define a suitable objective function that can help to learn the manifold of the input data $\mathbf{X}$ as well as the parameters of the rule base. One promising approach would be to preserve the geodesic neighborhood relationship, i.e., the geodesic distance structure of the manifold, on to the projected lower dimension. Let $gd_{ij}^{\mathbf{X}}$

be the geodesic distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^{d_h}$ and $ed_{ij}^{\mathbf{Y}}$ be the Euclidean distance between $\mathbf{y}_i$ and $\mathbf{y}_j$, $\mathbf{y}_i, \mathbf{y}_j \in \mathbb{R}^{d_l}$. A good objective function to estimate $\mathbf{y}_i$s is

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(gd_{ij}^{\mathbf{X}} - ed_{ij}^{\mathbf{Y}}\right)^2 / gd_{ij}^{\mathbf{X}}. \tag{5}$$

The objective (5) is introduced by Yang in [57]. Note that (5) is similar to Sammon's stress function that uses the Euclidean distance in both high- and low-dimensional spaces. In place of (5), we can use other functions also. Next, we address the issue of identification of the rule base.

A. Initial Rule Extraction

When both input and output data are provided there are many ways of generating an initial rule base and its refinement [27], [58]–[62]. Some of the popular methods use evolutionary algorithms [60] or clustering [27], [61], [62] of input–output data. Here, we do not have the output data. So, we cluster the input data into a predefined number of clusters and translate each cluster into a rule (to be precise, to a rule antecedent). The proposed method being a fuzzy rule-based scheme, the obvious choice seems to be the fuzzy c-means (FCM) clustering algorithm. But we did not do so for the following two reasons. First, for the FCM, the cluster centers may not fall on the data manifold. Since our idea is to use a cluster centroid to model a set of points in the neighborhood of the centroid, it is better to have the centroid on the data manifold. Second, the geodesic distance is not defined in terms of an inner product induced norm, and hence, the FCM convergence theory does not hold for the geodesic distance-based FCM. So, we use a slightly modified c-means called geodesic c-means (GCM) algorithm. Here, we cluster the input, aiming to obtain some representative points on the input manifold. To extract information regarding the input manifold structure, we use geodesic distance as the dissimilarity measure for clustering. Like the conventional c-means, data points are assigned to a cluster using the minimum distance (here, minimum geodesic distance) criterion and the cluster centroids are computed as the mean vector of the points in a cluster. Since such centroids may not lie on the manifold, we use an extra step. For each computed centroid, we find the input data point closest to the centroid and use that data point as a cluster centroid. We note here that the use of the geodesic distance in the c-means clustering algorithm has been investigated in other studies [63]–[65] but their approaches are different from ours. In [63], the authors introduced a class of geodesic distance that took into account local density information and employed that geodesic distance in the c-means clustering algorithm. The study in [64] incorporated the geodesic distance in a soft kernel c-means algorithm. The authors in [65] integrated the geodesic distance measure into the fuzzy c-means clustering algorithm. The algorithmic description of our clustering algorithm (GCM) is given in Algorithm S-1 (see Supplementary Materials). To estimate the geodesic distance, we have followed an approach similar to that in ISOMAP [20]. First, we construct a neighborhood graph of the input data points based on the Euclidean distance. For approximating the geodesic distance, every edge is assigned a weight/cost, which is basically the Euclidean distance between the pair of points on which the edge is incident. For any point, the geodesic distances to its neighboring points are approximated by the Euclidean distance. The geodesic distance between two points that are not neighbors, i.e., not connected by an edge is approximated by evaluating the shortest path on the neighborhood graph. We compute the shortest path distance using the Floyd–Warshall algorithm [66].

The kth cluster centroid, $\mathbf{v}_k = (v_{k1}, v_{k2}, \ldots, v_{kd_h})$, obtained by the GCM is translated into the antecedent of the kth rule, R_{km}, as follows:

$$\text{If } x_1 \text{ is "close to } v_{k1}\text{" AND } x_2 \text{ is "close to } v_{k2}\text{"}$$
$$\text{AND } \cdots x_{d_h} \text{ is "close to } v_{kd_h}\text{."} \tag{6}$$

The fuzzy set F_{kq} in (3) is defined linguistically as " close to v_{kq}"; $q = 1, 2, \ldots, d_h$. Thus, the kth rule given in (3) becomes

$$R_{km}: \text{If } x_1 \text{ is "close to } v_{k1}\text{" AND } x_2 \text{ is "close to } v_{k2}\text{" AND}$$
$$\ldots \text{AND } x_{d_h} \text{ is "close to } v_{kd_h}\text{" then } y_m^k = \sum_{q=0}^{d_h} a_{kmq} x_q \tag{7}$$

where $x_0 = 1$; $k = 1, 2, \ldots, n_c$; and $m = 1, 2, \ldots, d_l$. We model "x_q is 'close to v_{kq}'" using a Gaussian membership function with the center at v_{kq}. To exploit the differentiability property, we use Gaussian membership functions. This also makes the antecedent of the kth fuzzy rule to model a hyperellipsoidal volume centering at $\mathbf{v}_k$ in the input space. For the ith point, the membership to the set "x_q is close to v_{kq}" is computed as

$$\mu_{kq,i} = \exp\left\{-(x_{iq} - v_{kq})^2 / 2\sigma_{kq}^2\right\}. \tag{8}$$

Here, σ_{kq} is the spread of the Gaussian membership function of the fuzzy set F_{kq} centered at v_{kq}. We use the product T-norm to aggregate $\mu_{kq,i}$'s to obtain the firing strength $\alpha_{k,i}$ as follows:

$$\alpha_{k,i} = \prod_{q=1}^{d_h} \mu_{kq,i}. \tag{9}$$

We denote the matrix of cluster centroids as $V = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n_c})^T = [v_{kq}]_{n_c \times d_h}$. Similarly, the matrix of spreads $\Sigma = (\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \sigma_{\mathbf{n_c}})^T = [\sigma_{kq}]_{n_c \times d_h}$. An initial estimate of σ_{kq} can be taken as the standard deviation of the qth feature in the kth cluster or can be initialized using some other method. The set of consequent parameters A is initialized randomly. Note that these are initial choices that will be refined during the training.

B. Objective Function and Its Optimization

To learn the lower dimensional representations $\mathbf{Y}$ of the given input $\mathbf{X}$, the error function defined in (5) is minimized with respect to the rule base parameters V, Σ, and A. Using (8) and (9) in (4b), we obtain the following relation:

$$y_{im} = \sum_{k=1}^{n_c} \frac{\left(\prod_{q=1}^{d_h} \exp\left\{-\frac{(x_{iq} - v_{kq})^2}{2\sigma_{kq}^2}\right\}\right)\left(\sum_{q=0}^{d_h} a_{kmq} x_{iq}\right)}{\sum_{p=1}^{n_c} \prod_{q=1}^{d_h} \exp\left\{-\frac{(x_{iL} - v_{pq})^2}{2\sigma_{pq}^2}\right\}}. \tag{10}$$

Here, $x_{i0} = 1$ and $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{id_l})^T$. Hence, $\mathbf{y}_i$ is dependent on v_{kq}, σ_{kq}, and $a_{kmq}; \forall k = 1$ to $n_c, q = 1$ to d_h, and $m = 1$ to d_l, which form V, Σ, and A, respectively. It is evident that we can write the lower dimensional vectors $\mathbf{y}_i$s

TABLE I
COMPUTATIONAL COMPLEXITY OF VARIOUS METHODS

Method	Complexity Computational	Space
Proposed Method	O(n^2t)	O(n^2)
Sammon's Projection [4]	O(n^2t)	O(n^2)
ISOMAP [4]	O(n^3)	O(n^2)
LLE [4]	O(pn^2)	O(pn^2)
t-SNE [30]	O(n^2t)	O(n^2)
UMAP [39]	O($n^{1.14}t$)	O(n^2)
Diffusion Net [54]	$O(n^3 + nwt)$	O(n^2)

Here, n = Size of training set, t = Number of iterations, p = Ratio of nonzero to total number of elements in the sparse matrix related to LLE, and w = Number of weights in the network.

as a function of the rule base parameters V, Σ, and A as follows:

$$\mathbf{y}_i = f(V, \Sigma, A; \mathbf{x}_i). \tag{11}$$

Replacing $ed_{ij}^{\mathbf{Y}}$ in (5) as $\|\mathbf{y}_i - \mathbf{y}_j\|$, and then, using (11), we get

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(gd_{ij}^{\mathbf{X}} - \|\mathbf{y}_i - \mathbf{y}_j\|\right)^2 / gd_{ij}^{\mathbf{X}}$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(gd_{ij}^{\mathbf{X}} - \|f(V,\Sigma,A;\mathbf{x}_i) - f(V,\Sigma,A;\mathbf{x}_j)\|\right)^2 / gd_{ij}^{\mathbf{X}}. \tag{12}$$

The optimum values of V, Σ, and A are obtained as

$$(V_{\text{opt}}, \Sigma_{\text{opt}}, A_{\text{opt}}) = \arg\min_{V,\Sigma,A}(E). \tag{13}$$

To minimize the error function E in (13), a momentum-based gradient descent approach is used here. The overall process is summarized in Algorithm S-2 (see Supplementary Materials).

C. Scalability Analysis

Scalability is a desirable property of any algorithm. The error function (5) of our method considers all the interpoint distances, i.e., a total of $n(n-1)/2$ distances. During the training phase, to update the rule parameters, in each iteration, $n(n-1)/2$ values need to be computed. So, for t iterations, the computational complexity of the proposed method is O(n^2t) for a fixed input dimension. In Table I, we have compared the computational complexity of the proposed method during the training phase, with six popular dimensionality reduction methods for data visualization.Note that, LLE and ISOMAP perform convex optimization, whereas the other methods including ours perform nonconvex optimization with iterative steps. Similar to our proposed method, Sammon's projection and t-SNE also involve interpoint distances in their objective functions. Hence, they have the same complexity as that of the proposed method. The computational complexity of the UMAP is bounded by the approximate nearest neighbor search, which is empirically equal to O($n^{1.14}$) [39]. Considering total t iterations, computational complexity would be O($n^{1.14}t$). Since the UMAP involves estimation of the nearest neighbors, it needs to store the interpoint distances. Hence, its space complexity is O(n^2). We emphasize that being an explicit mapping method, during the test phase, the proposed method exhibits a computational complexity of O(1). In [54], the authors did not discuss the complexity of training diffusion net. Diffusion net involves computation of diffusion map [complexity $O(n^3)$] and training an autoencoder [complexity $O(nwt)$, where w is the number of weights in the autoencoder]. Except diffusion net, the other comparing methods in their original form cannot predict out-of-sample points. They exhibit the same computational complexity in the test phase as in the training phase. However, their extensions have lower complexities for projecting test data. In terms of space requirement, like most of the other methods, our method also exhibits a complexity of O(n^2). It is also important to understand the dependence of an algorithm with the dimension of the input feature vectors (d_h). In our method, the number of rule parameters varies linearly with the input dimension. So, both the computational and space complexity scale up linearly with the number of input features.

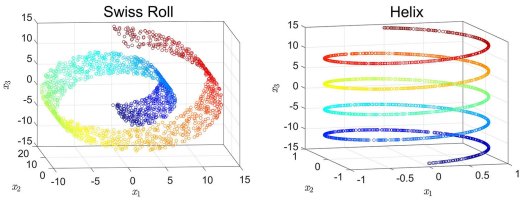

Fig. 1. Datasets. (a) Swiss Roll. (b) Helix.

IV. EXPERIMENTATION AND RESULTS

A. Dataset Description

For our experiments, we consider three synthetic datasets and three real-world datasets. We choose these datasets as they are commonly used in manifold learning studies [18], [19], [30], [55]. The synthetic datasets are: Swiss Roll, S Curve, and Helix as shown in Figs. 1(a), S-3(a) (in the Supplementary Material), and 1(b), respectively. All of these are three-dimensional data but contained completely within a two-dimensional space. Each of these synthetic datasets consists of 2000 points. The first two datasets are generated using the "scikit-learn" package [67] (version 0.19.2) of python. Helix dataset is generated by the equations: $x_3 = t/\sqrt{2}; x_1 = \cos x_3$; and $x_2 = \sin x_3$. The variable t is varied from -20 to 20 with steps of 0.02. The real-world datasets considered are Frey face [68], COIL [69], and USPS handwritten digits [68]. Frey face consists of 1965 gray scale images each of size 20×28 pixels, i.e., the input dimension is 560. These are of the face of a single person with different face orientations and facial emotions. The COIL dataset consists of images of 20 different objects viewed from 72 equally spaced angels. We have considered the first object of the COIL dataset. Each image is of resolution 32×32. Thus, the input dimension is 1024 and the number of instances $= 72$. USPS handwritten digits dataset contains 1100 images for each of the handwritten digits: 0–9. Each image is of size 16×16, resulting in inputs of dimension 256. We have considered the images of 0 only.

B. Experiment Settings

We compare the results obtained by the proposed method on the aforementioned datasets with six other data visualization methods: Sammon's projection, ISOMAP, LLE, t-SNE, UMAP,

and diffusion net. The synthetic datasets are used directly without any preprocessing. For image datasets, pixel values are divided by 255 to have their values in [0,1]. We compare the predictability of the proposed method with that of the out-of-sample extension of ISOMAP and LLE [44], autoencoder [51], and simplified NPPE (SNPPE) [55]. For this, we use the Swiss Roll and the Frey face datasets. Following [55], we choose the number of nearest neighbors, $\epsilon = 1\%$ of the training samples (rounded to the nearest integer). For the first object of the COIL dataset (we shall refer to it as COIL), the number of instances is 72. So, we use $\epsilon = 5$ instead of $\epsilon = 1$ to construct a reasonable graph. To choose the number of fuzzy rules, n_c, we perform an experiment. Details of this experiment and its results are included in the Section S-I of the Supplementary Materials. From that experiment, we decide to set $n_c = 1\%$ of the training samples (rounded to the nearest integer). Here also for the COIL dataset to avoid underfit, we choose $n_c = 5$ instead of $n_c = 1$. For synthetic datasets, we initialize, spreads of the Gaussian memberships, σs in two different ways: 0.2 times and 0.3 times the feature-specific range. For high-dimensional real datasets, we need to choose higher values of σs to avoid generating several zero rule-firing cases. For 0 s of USPS dataset (we shall refer to it as USPS) and Frey face, we initialize σs as 0.4 times the feature-specific range. For COIL dataset, σs are initialized as 0.5 times the feature-specific range. In all cases, consequent parameters are initialized with random values in $(-0.5, 0.5)$.

For both synthetic and real-world datasets, we repeat the experiments five times for each choice of the spread. For the three synthetic datasets, there are two initial choices for spread. So, ten runs are conducted giving ten rule-based systems for each synthetic dataset. On the other hand, for each real-world dataset, we use only one choice of the initial spread, and thereby, generating five rule-based systems. Then, based on the minimum value of the objective function in (5), we choose the best rule-based system for a given dataset. We do *not* look at any test data and the best result can always be chosen in an automatic manner without any human intervention based on (5) as this gives the training error.

To optimize the objective function (5), we use the "train.MomentumOptimizer" class of the "TensorFlow" framework [70] with learning rate $= 0.1$ and momentum $= 0.5$. For a fair comparison, for the comparing methods, we apply the synthetic datasets directly and the real-world datasets are used after feature-wise zero-one normalization. For computing Sammon's projection, we use the MATLAB implementation of multidimensional scaling [8], *mdscale* with the parameter "Criterion" set to "sammon." The dimension of the output is set to two. For other parameters, the default values are used. The ISOMAP, LLE, and t-SNE are implemented, respectively, using the classes "manifold.Isomap," "manifold.LocallyLinearEmbedding," and "manifold.TSNE" of the "scikit-learn"(sklearn) package [67] (version 0.19.2) of python. UMAP is implemented using the python API "umap"[38]. The diffusion net is implemented with the available code [71]. For LLE, ISOMAP, UMAP, and diffusion net, we use the same neighborhood size (number of nearest neighbors) as used by the proposed method. For t-SNE, there is no provision of setting the neighborhood size directly. The "perplexity"[30] parameter of t-SNE provides an indirect measure of the effective number of neighbors. So, we set the "perplexity" parameter to the number of nearest neighbors used in the proposed method. For these methods, all other parameters are kept at their default values as supported by the respective routines. For out-of-sample testing, except for SNPPE, the other three comparing methods are realized using MATLAB Toolbox for dimensionality reduction (drtoolbox) by Laurens van der Maaten [72]. The results of SNPPE are generated by the code provided by the authors of SNPPE. In these cases also, the neighborhood size, when required, is set to the same value as used by the proposed method. All the other parameters are kept at their default values.

C. Results and Comparisons

From Figs. 2(a), S-3(b) (see Supplementary Material), and 3(a), we can observe that the proposed method successfully unfolds the nonlinear structures present in the Swiss Roll, S Curve, and Helix data, respectively, to nearly a linear structure. For all three datasets, Sammon's projection [see Figs. 2(b), S-3(c) (in Supplementary Material), and 3(b)] cannot unfold the original data to its intrinsic linear structure but it tries to preserve the global geometric structure. ISOMAP tries to preserve the pairwise distances over the manifold by using the geodesic distance in the classical multidimensional scaling error function [20]. Preserving geodesic distance, ISOMAP is able to unfold the nonlinear structure as shown in Figs. 2(c) and S-3(d) (in Supplementary Material). But a drawback of the system is that large distances play a stronger role compared to small distances in its objective. As a result, in the case of the Helix dataset, as seen in Fig. 3(c), it cannot unfold the data as desired. LLE, t-SNE, UMAP, and diffusion net try to capture the local structure present in the original data, in the projected space. Figs. 2(d), S-3(e) (see the Supplementary Material), and 3(d) reveal that the LLE can unfold the underlying structure to a reasonable extent but the quality of unfolding is far from that by the proposed method and ISOMAP. Similarly, t-SNE and UMAP although have the same aim, are not able to unfold the datasets properly as revealed by Figs. 2(e), 2(f), S-3(f) (in the Supplementary Material), S-3(g) (in the Supplementary Material), 3(e), and 3(f). Figs. S-2, S-3(h), and S-4 in the Supplementary Material indicate that, apart from the Helix dataset (see Fig. S-4 in the Supplementary Material), the diffusion net is not successful in capturing the variations along the intrinsic dimensions. For both Swiss Roll and S Curve, the diffusion net embedding captures variations along one intrinsic feature out of the two. In summary, of the seven methods, only the fuzzy rule-based method is successful for all three datasets and the next best-performing method is ISOMAP.

The proposed method can also faithfully project the real-world datasets in two dimensions. In Fig. 4, the two-dimensional embedding of the Frey face dataset is shown. Fig. 4 reveals that the gradual change in the face orientation from right to left is roughly related to the increase in the projected feature y_1 on the horizontal axis. Similarly, the change in the expressed emotion varies roughly with the extracted feature (projection) y_2. Smiling, neutral, and annoyed faces are mapped in the top, middle, and bottom regions, respectively. The results of the five

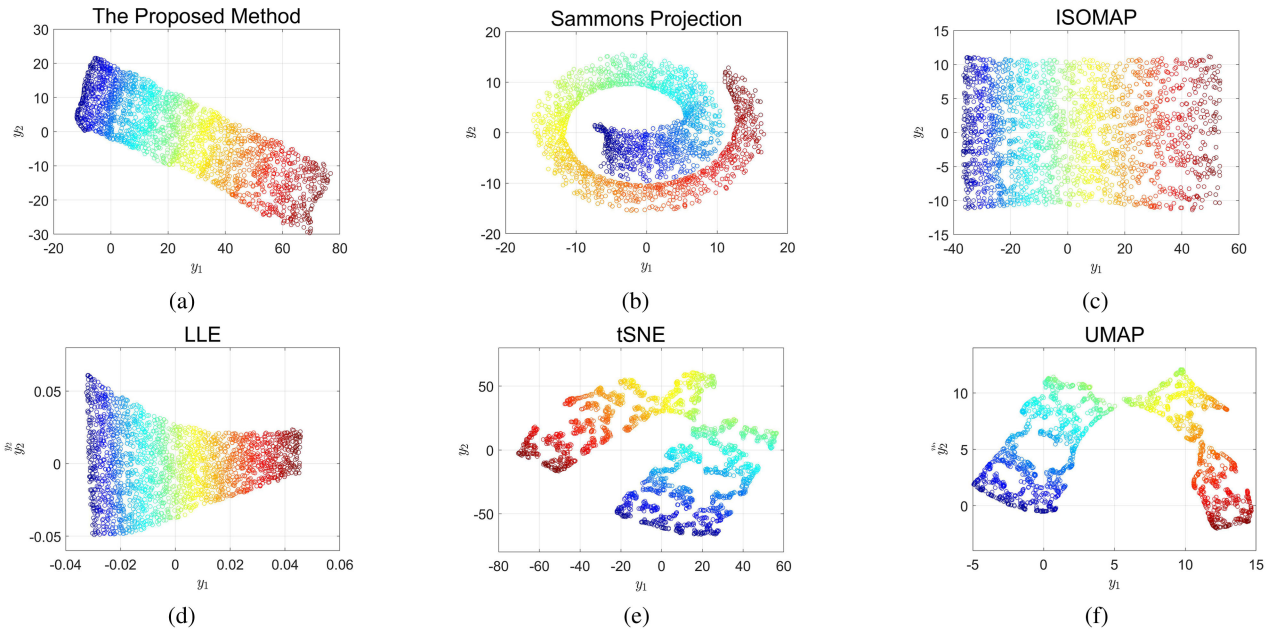

Fig. 2. Visualization of the Swiss Roll data with (a) proposed method, (b) Sammon's projection, (c) ISOMAP, (d) LLE, (e) t-SNE, and (f) UMAP.

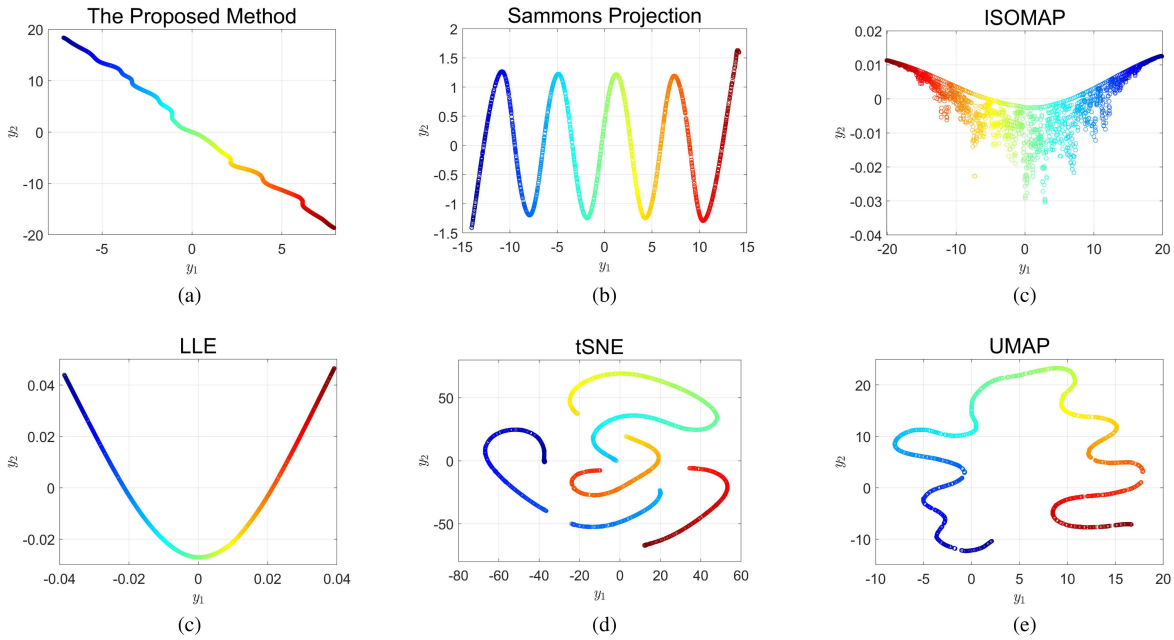

Fig. 3. Visualization of the Helix data with (a) proposed method, (b) Sammon's projection, (c) ISOMAP, (d) LLE, (e) t-SNE, and (f) UMAP.

comparing methods are placed in the Supplementary Materials. Sammon's projection being a Euclidean distance preserving method has placed the faces expressing different emotions in comparatively overlapping regions as seen in Fig. S-5 in the Supplementary Material. Other neighborhood preserving methods have unfolded the high-dimensional manifold. Comparing the results in Figs. S-6–S-10 in the Supplementary Material with the result in Fig. 4, we can infer that the proposed method has mapped the variations of Frey face dataset comparatively in a more consistent and desired manner. Fig. S-11 in the Supplementary Material depicts the two-dimensional embedding of the COIL dataset. For this dataset, the expected outcome is obtained by the proposed method (see Fig. S-11(a) in the Supplementary Material) as well as by ISOMAP (see Fig. S-11(c) in the Supplementary Material), t-SNE (see Fig. S-11(e) in the Supplementary Material), and diffusion net (see Fig. S-11(g) in the Supplementary Material). However, results obtained from Sammon's projection, LLE, and UMAP shown

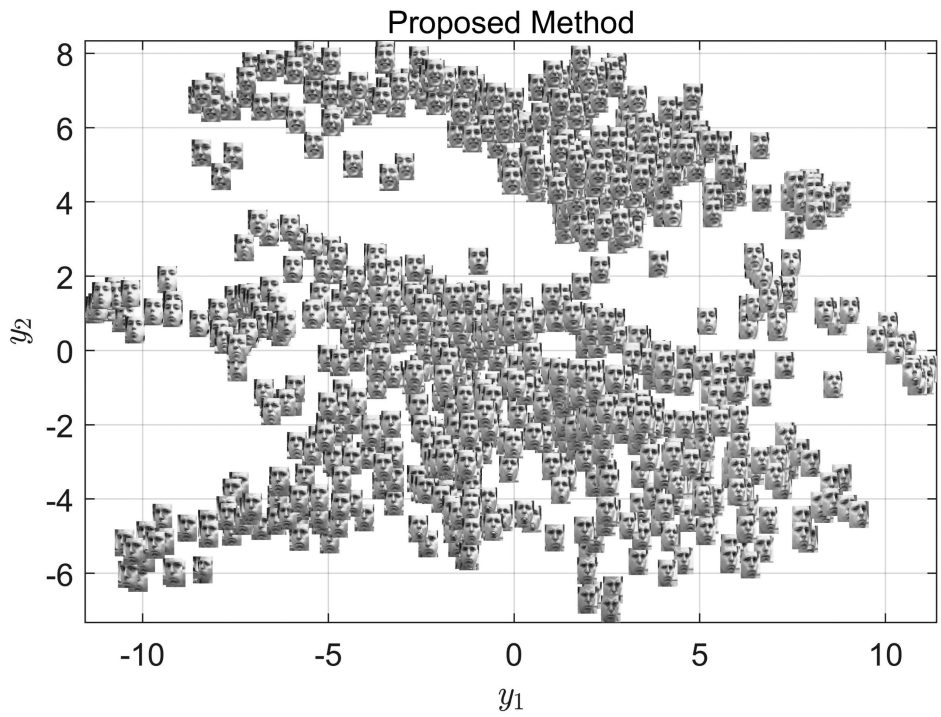

Fig. 4. Visualization of the Frey face dataset with the proposed method.

in Figs. S-11(b), S-11(d), and Fig. S-11(f) in the Supplementary Material, respectively, cannot recover the manifold faithfully on the two-dimensional space. Fig. S-12 in the Supplementary Material shows a visualization of the handwritten character 0 s of the USPS dataset by the proposed method. In this case also, the proposed method successfully projects zeros with different orientations and linewidths in different regions in a continuous manner. The projections by the other six methods shown in Figs. S-13–S-18 in the Supplementary Material reveal that LLE (see Fig. S-15 in the Supplementary Material), t-SNE (see Fig. S-16 in the Supplementary Material), UMAP (see Fig. S-17 in the Supplementary Material), and diffusion net (see Fig. S-18 in the Supplementary Material) could not represent the data in a way so that orientations and linewidths of the character vary smoothly in the projected space. In all three cases of the real-world datasets, the proposed method successfully learns the approximate two-dimensional manifold on which the original data lie.

D. Impact of Initial Rules

Our method initializes the rule antecedents using the centroids of the clusters obtained by the Algorithm S-I in the Supplementary Material. To show the impact of the initial rules, we have initialized the rule antecedents using uniformly distributed random values generated from the smallest hyperbox containing the training data. The obtained projection for Swiss Roll data is shown in Fig. S-19(b) in the Supplementary Material. Fig. S-19(a) in the Supplementary Material shows the result obtained by the proposed method with rule antecedents defined by the cluster centroids. From Fig. S-19(a) and S-19(b) in the Supplementary Material, it is evident that with or without judicious initialization of rule antecedents, the proposed method can unfold the nonlinear structure satisfactorily. This observation is also true in the case of the Frey face dataset. Fig. S-20 in the Supplementary Material shows the visualization of Frey face dataset using the proposed method with random antecedents initialized as mentioned previously. To investigate further, we have computed the average of the final error [see (5)] over the ten/five runs (performed as explained in Section IV-B) for the two considered datasets with both types of initialization of the rule antecedents. Table II shows that the averages of the final error values are less in the case of initialization of the rule antecedents by cluster centroids compared to random initialization for both datasets. This suggests that although in both cases, the algorithm lands up in useful local minima, for the cluster-based initial rules, generally the minima are superior.

TABLE II
AVERAGE OF THE FINAL ERROR VALUES FOR DIFFERENT INITIALIZATIONS

Data Set	Initial rule antecedents defined by	
	Cluster centroids	Random
Swiss Roll	0.04740	0.06490
Frey face	0.50459	0.50500

We have also done ten experiments on the Swiss Roll data with rules that are initialized with random values in [0,1] hypercube. The results are shown in Fig. S-21 in the Supplementary Material. Even in this case, the results are reasonably good, but not as good as the cases with judiciously initialized rules. In a few cases, the system could not unfold. Thus, the system is quite robust with respect to the initial choice of rules but certainly, cluster-based rules help.

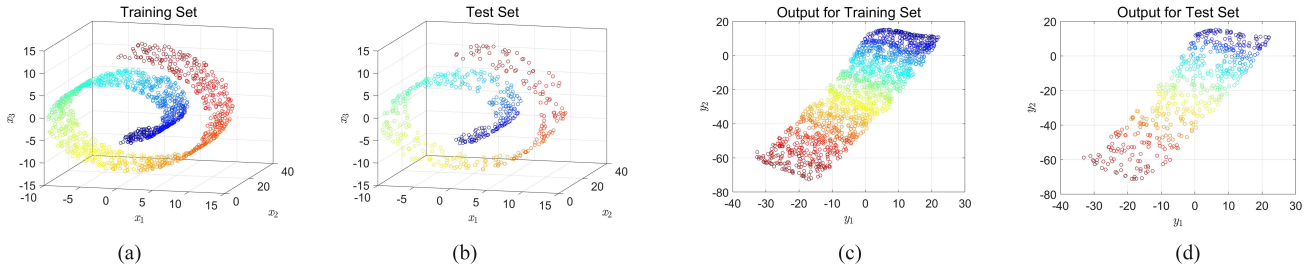

Fig. 5. For experiment 1 on the validation of predictability with the Swiss Roll data. (a) Training set. (b) Test set. (c) Proposed method output for the training set. (d) Proposed method output for the test set.

E. Validation of Predictability for Out-of-Sample Points

One of the key advantages of utilizing an FRBS is that it is parametric, and thereby, providing an explicit mapping function from the high-dimensional inputs to a lower dimensional one. So, after the system is trained, it is possible to predict the lower dimensional embedding for new test points. As mentioned earlier, to compare the predictability of the proposed method, we consider the following four methods: out-of-sample extension of ISOMAP and LLE [44] and two parametric methods: autoencoder [51] and SNPPE [55]. Note that SNPPE involves a polynomial mapping. Following [55], we use the second-order and third-order polynomial-based SNPPE.

To assess the predictability of the system, we have designed the following three experiments:

1) randomly dividing the dataset into training and test sets;
2) using a contiguous portion of the manifold as the test set and rest as the training set;
3) leave one out validation.

We have performed the first test on the Swiss Roll and Frey face datasets. We use 75% of the Swiss Roll dataset, i.e., 1500 randomly selected points to form the training set and the rest 25%, i.e., 500 points to form the test set. The training and test datasets are shown in Fig. 5(a) and 5(b), respectively. While partitioning the dataset, the associated label (color) vectors are also partitioned accordingly. Fig. 5(c) and 5(d) are the results corresponding to the training and test datasets by the proposed method. From Fig. 5(d), it is evident that the predicted locations determined by the proposed method for the test points are at their expected positions in the lower dimension. Figs. S-22–S-26 in the Supplementary Material show the projections of the training and test sets by ISOMAP, LLE, autoencoder, second-order SNPPE, and third-order SNPPE, respectively. Except autoencoder, the other comparing methods successfully unfold the Swiss Roll structure in the training phase and appropriately predict the test points. From Fig. S-24, we observe that the autoencoder-based method is unable to preserve the geometry of the training data. In Fig. 5(d) and Figs. S-22(b), S-23(b), S-25(b), and S-26(b) (see the Supplementary Material), color assignment to the projected test points by the label vector shows that the test dataset is unfolded in the desired manner in all five cases. For the Frey face dataset, we have selected 15 random instances as the test set and the rest of the points as the training set. Fig. S-27 in the Supplementary Material illustrates the projected output by the proposed method for the training and test data together where the images with black rectangular borders correspond to the test instances. For all comparing methods for this data, we follow the same representation scheme for the training and test data. In Figs. S-27–S-29 in the Supplementary Material, we can see that the test instances are projected to the regions where the training instances have the same/similar facial expression and orientation. Thus, for high-dimensional image datasets also, the proposed method, as well as LLE and ISOMAP, are successful in prediction. The autoencoder-based lower dimensional projection, shown in Fig. S-30 in the Supplementary Material clustered different facial expressions and different face orientations in different regions. However, the autoencoder-based method is unable to unfold the intrinsic structure properly because in some cases, it placed faces with opposite orientations (left and right) closer to each other compared to the faces with the same orientations. A similar placement has occurred for opposite expressions (e.g., happy and annoyed) and neutral expressions. Figs. S-31 and S-32 in the Supplementary Material reveal that SNPPE-based methods neither unfolded the higher dimensional data nor predicted the positions of the test points in a desirable manner.

The second test is also done with the Swiss Roll dataset. A contiguous portion of the dataset consisting of 50 points has been removed and the remaining 1950 points serve as the training dataset as shown in Fig. 6(a). The removed portion serves as the test set. Fig. 6(c) displays the output of the proposed method corresponding to the training set. Fig. 6(d) shows the test set outputs of the proposed method along with the training set. The test set points are demarcated with the plus (+) symbol. For the comparing methods, we represent the results in the same manner. Figs. S-33(a), S-34(a), S-35(a), S-36(a), and S-37(a) in the Supplementary Material show the projections of the training data for ISOMAP, LLE, autoencoder, second-order SNPPE, and third-order SNPPE, respectively. Similarly, Figs. S-33(b), S-34(b), S-35(b), S-36(b), and S-37(b) in the Supplementary Material show the projections of the test set (demarcated with the plus (+) symbol) along with the training set for ISOMAP, LLE, autoencoder, second-order SNPPE, and third-order SNPPE, respectively. These results reveal that the proposed method, as well as the comparing methods except for the autoencoder, has appropriately interpolated the test points.

The third test uses the leave-one-out validation on the COIL dataset. Each of the 72 instances is considered as the test instance and the remaining 71 instances are used as the training set. For each of the 72 training–test sets, outputs are generated. Results of all 72 tests are shown in the Supplementary Materials (see

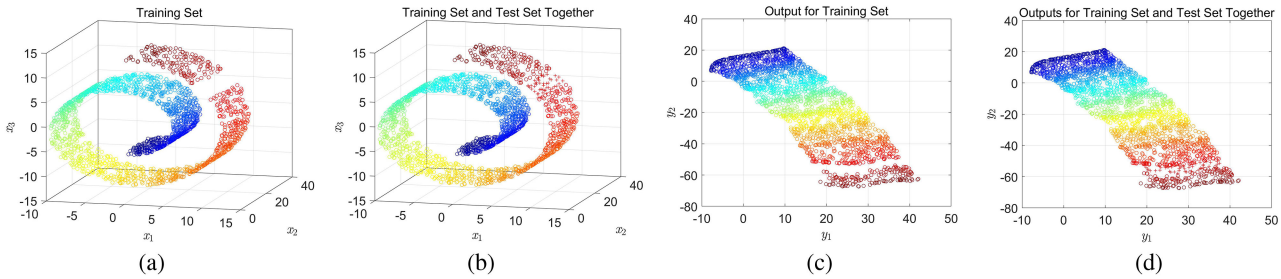

Fig. 6. For experiment 2 on validation of predictability with the Swiss Roll data. (a) Training set. (b) Training set and test set. (c) Proposed method output for the training set. (d) Proposed method outputs for training set and test set.

Figs. S-38–S-49 in the Supplementary Material). The training points are demarcated in green and the test point is demarcated in red. Instead of showing the images of the object, the consecutive viewing angles of the images are represented by consecutive numbers. In most of the cases, it is found that the test object is placed in the desired position, i.e., in between points corresponding to input images captured in plus-minus five degrees of the viewing angle of the given test point.

F. Rejection of Outputs

The proposed method is capable of identifying the test cases where the system may not produce reliable outputs. In a T–S type fuzzy rule-based system, the output is computed by (4). The rule having the maximum firing strength mainly characterizes the output. For test points far away from the training data, no fuzzy rule is expected to fire strongly. Consequently, for those points, the outputs obtained may not be reliable. Fig. S-50(a) in the Supplementary Material shows a set of training points and test points for the Swiss Roll dataset. The training points are depicted in blue, while the test points are in red. Test points are located away from the training points. The histogram of the maximum firing strengths of rules is shown in Fig. S-50(b) in the Supplementary Material. Here also, the training and test firing strengths are indicated with blue and red colors, respectively. Fig. S-50(b) in the Supplementary Material shows that the maximum rule firing strengths of the test points are noticeably lower than those of the training points. This information can be used to decide when the system should reject its output (refuse to make a decision). For example, if the maximum firing strength is less than 0.15, the system output can be rejected. Note that this threshold can be decided based on the training data only. In Fig. S-50(c) in the Supplementary Material, we have depicted the projected outputs for the training and test data corresponding to Fig. S-50(a) in the Supplementary Material without exercising the rejection option. It is clear that the prediction by the proposed method for the test points (red circles) is inappropriate and should be discarded. Note that except diffusion net, other compared methods do not have any provision for rejecting outputs.

G. Generalized Nature of the Proposed Model

The proposed model gives a general framework for learning an unsupervised FRBS for manifold learning or data projection. Instead of using the objective function in (5), we can use different objectives to preserve different characteristics of the data. For example, the objective function of Sammon's projection in (1) can be used to preserve the global geometric structure as in [31]. Fig. S-51 in the Supplementary Material shows the Swiss Roll data and its corresponding lower dimensional output using the proposed method with (1) as the objective function for learning. Similarly, Figs. S-52 and S-53 in the Supplementary Material depict the output produced by the FRBS using (1) as the objective function for the S Curve and Helix data, respectively. In both cases, the projection reveals what Sammon's method is expected to do. For these experiments, the other settings are kept the same as described in Section IV-B. This system can predict for out-of-sample points and that is an advantage over the usual Sammon's method. We also note that unlike the fuzzy system in [27], the proposed system does not require any target output for learning.

V. Conclusion and Discussion

Fuzzy rule-based systems have been used for various machine learning tasks but unfortunately, FRBSs are almost unexplored for manifold learning or data visualization. Here, we have proposed a framework for unsupervised learning of a Takagi–Sugeno FRBS for the projection of high-dimensional data. Without using any target data, the parameters of the FRBS are estimated so that the FRBS can project the high-dimensional data preserving the local neighborhood structure.

Some advantages of the proposed scheme are listed as follows.

1) It is an unsupervised system, yet it learns a parametric system for explicit mapping of high-dimensional data into a lower dimensional space. Thus, out-of-sample instances can be projected in a straightforward manner.
2) It can refuse to project an out-of-sample instance, when it is far away from the sampling window of the training data, i.e., the system is equipped with an appropriate reject option. Most machine learning systems cannot do so.
3) The system is reasonably transparent as it is based on fuzzy reasoning, where each rule-antecedent represents a small hyperellipsoidal region in the input space and its consequent is a local linear function of the inputs.
4) It provides a general framework that can use different objective functions.
5) The learning algorithm is found to be quite robust with respect to initialization.

We have compared the performance of the proposed system with Sammon's method, ISOMAP, LLE, t-SNE, UMAP, and diffusion net on several datasets. The proposed system is the only

one that performs consistently well on all datasets tested. The predictability of the proposed system is validated using three types of experiments involving both synthetic and real-world datasets.

A. Limitations and Scope of Improvement

Like several other data projection methods, both the time and space complexities of the proposed method are quadratic functions of the number of data points, n. So, for a very large n, the computational overhead will be high. The use of graphics processing units can reduce the effective complexities significantly. Moreover, a proper sampling scheme [48] that produces a subset of the original data, which adequately represents the original data distribution may be used. The firing strength of a rule is computed as the product of the memberships of the atomic antecedent clauses in the rule. So, for a very high-dimensional data, it is possible that for some instances, no rule fires with a significant strength leading to an undesirable situation. However, this issue is true for *any* FRBS, it is not special to the system proposed here. Like several other manifold learning schemes, the proposed scheme is sensitive to the number of neighbors that is used to construct the neighborhood graph for computing geodesic distances. In our study, the number of fuzzy rules has been chosen empirically. The choice of the number of fuzzy rules is related to the number of clusters we choose for clustering the training data. We did not explore the possibilities of using cluster validity indices or other schemes to decide on the optimal number of clusters as this was not the primary objective of this study. Moreover, a cluster validity index helps to find the "optimal" number of clusters in the pattern recognition sense. But here even if the data do not have any cluster in the pattern recognition sense, we can use clustering to find the initial rule base. However, for a given dataset, there may be a desirable number of rules, which may simplify the system and improve its performance further.

The clustering algorithm we use, GCM, depends on the construction of a neighborhood graph using the input data points based on the Euclidean distance. For very high-dimensional data, the GCM algorithm may suffer from the curse of dimensionality as Euclidean distance does not behave properly in a very high-dimensional space [73]. However, it is worth noting that methods like LLE, LE, and ISOMAP, which use neighborhood defined using the Euclidean distance also suffer from the same problem for high-dimensional data. For nonimage data, a possible approach to address this problem is to project the data onto a space having moderate dimension using an unsupervised feature selection scheme or using a parametric feature extraction scheme that does not influence much the intrinsic manifold structure but eliminates redundant features or information. For image data, if the size of the image is big, it will also result in high-dimensional inputs. But here, we cannot use feature (pixel) selection as that will not be effective. However, we may use feature embedding techniques like an autoencoder to reduce the size of the image. Then, we can develop the proposed fuzzy rule-based system in the latent space / reduced space. We have restricted our work to datasets lying on a single manifold. Through neighborhood graph construction, we can only approximate the geodesic distance over a single manifold or overlapping manifolds. But if the data lie in two (or more) widely separated manifolds, using a reasonable number of nearest neighbors, we may not get a connected neighborhood graph. We have not proposed any technique to approximate the geodesic distance over a multicomponent neighborhood graph. These issues are kept for future investigation.

REFERENCES

[1] A. M. Álvarez-Meza, J. A. Lee, M. Verleysen, and G. Castellanos-Dominguez, "Kernel-based dimensionality reduction using Renyi's α-entropy measures of similarity," *Neurocomputing*, vol. 222, pp. 36–46, 2017.

[2] D. Meng, Y. Leung, T. Fung, and Z. Xu, "Nonlinear dimensionality reduction of data lying on the multicluster manifold," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 1111–1122, Aug. 2008.

[3] A. Talwalkar, S. Kumar, and H. Rowley, "Large-scale manifold learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[4] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: A comparative," *J. Mach. Learn. Res.*, vol. 10, pp. 66–71, 2009.

[5] Y. Wang *et al.*, "A perception-driven approach to supervised dimensionality reduction for visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 5, pp. 1828–1840, May 2018.

[6] H. Chernoff, "The use of faces to represent points in k-dimensional space graphically," *J. Amer. Statist. Assoc.*, vol. 68, no. 342, pp. 361–368, 1973.

[7] R. O. Duda, D. G. Stork, and P. E. Hart, *Pattern Classification*. New York, NY, USA: John Wiley, 2001.

[8] I. Borg and P. Groenen, "Modern multidimensional scaling: Theory and applications," *J. Educ. Meas.*, vol. 40, no. 3, pp. 277–280, 2003.

[9] L. Cayton, "Algorithms for manifold learning," Univ. California at San Diego, San Diego, CA, USA, Tech. Rep. 12, 2005, pp, 1–17.

[10] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[11] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[12] H. Akaike, "Factor analysis and AIC," in *Selected Papers of Hirotugu Akaike*. Berlin, Germany: Springer, 1987, pp. 371–386.

[13] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 153–160.

[14] E. Tuv, A. Borisov, G. Runger, and K. Torkkola, "Feature selection with ensembles, artificial variables, and redundancy elimination," *J. Mach. Learn. Res.*, vol. 10, pp. 1341–1366, 2009.

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

[16] K. Nag and N. R. Pal, "A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 499–510, Feb. 2016.

[17] I.-F. Chung, Y.-C. Chen, and N. R. Pal, "Feature selection with controlled redundancy in a fuzzy rule based framework," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 734–748, Apr. 2018.

[18] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. COM-18, no. 5, pp. 401–409, May 1969.

[19] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[20] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[21] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," in *Proc. Nat. Acad. Sci.*, vol. 100, no. 10, pp. 5591–5596, 2003.

[22] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[23] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.

[24] S. K. Ng and H. J. Chizeck, "Fuzzy model identification for classification of gait events in paraplegics," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 4, pp. 536–544, Nov. 1997.

[25] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2-3, pp. 191–203, 1984.

[26] E. Lughofer and R. Nikzad-Langerodi, "Robust generalized fuzzy systems training from high-dimensional time-series data using local structure preserving PLS," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 11, pp. 2930–2943, Nov. 2020.
[27] N. R. Pal, V. K. Eluri, and G. K. Mandal, "Fuzzy logic approaches to structure preserving dimensionality reduction," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 3, pp. 277–286, Jun. 2002.
[28] Y. Lin, Y. Li, C. Wang, and J. Chen, "Attribute reduction for multi-label learning with fuzzy rough set," *Knowl.-Based Syst.*, vol. 152, pp. 51–61, 2018.
[29] N. Mac Parthaláin, R. Jensen, and R. Diao, "Fuzzy-rough set bireducts for data reduction," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 8, pp. 1840–1850, Aug. 2020.
[30] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[31] S. Das and N. R. Pal, "An unsupervised fuzzy rule-based method for structure preserving dimensionality reduction with prediction ability," in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov.*, 2019, pp. 413–424.
[32] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 2859–2900, 2015.
[33] P. Switzer and A. Green, "Min/max autocorrelation factors for multivariate spatial imagery," Stanford Univ., Stanford, CA, USA, Tech. Rep. SWINSF6, Apr.1984.
[34] R. Larsen, "Decomposition using maximum autocorrelation factors," *J. Chemometrics*, vol. 16, no. 8–10, pp. 427–435, 2002.
[35] P. Demartines and J. Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 148–154, Jan. 1997.
[36] J. A. Lee *et al.*, "A robust non-linear projection method," in *Proc. Eur. Symp. Artif. Neural Netw.*, 2000, pp. 13–20.
[37] J. M. Lee, *Introduction to Smooth Manifolds*. Berlin, Germany: Springer, 2001.
[38] L. McInnes, J. Healy, N. Saul, and L. Groberger, "UMAP: Uniform manifold approximation and projection," *J. Open Source Softw.*, vol. 3, no. 29, 2018, Art. no. 861. [Online]. Available: https://doi.org/10.21105/joss.00861
[39] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.
[40] K. Q. Weinberger and L. K. Saul, "An introduction to nonlinear dimensionality reduction by maximum variance unfolding," in *Proc. 21st Nat. Conf. Artif. Intell.*, vol. 6, 2006, pp. 1683–1686.
[41] R. R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 5–30, 2006.
[42] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, 2004.
[43] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. Int. Conf. Artif. Neural Netw.*, 1997, pp. 583–588.
[44] Y. Bengio, J.-f. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 177–184.
[45] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, 2003.
[46] T.-J. Chin and D. Suter, "Out-of-sample extrapolation of learned manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1547–1556, Sep. 2008.
[47] L. Van Der Maaten, "Learning a parametric embedding by preserving local structure," in *Proc. Artif. Intell. Statist.*, 2009, pp. 384–391.
[48] N. R. Pal and V. K. Eluri, "Two efficient connectionist schemes for structure preserving dimensionality reduction," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1142–1154, Nov. 1998.
[49] A. K. Jain and J. Mao, "Artificial neural network for nonlinear projection of multivariate data," in *Proc. Int. Joint Conf. Neural Netw.*, 1992, vol. 3, pp. 335–340.
[50] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 296–317, Mar. 1995.
[51] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[52] D. DeMers and G. W. Cottrell, "Non-linear dimensionality reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, 1993, pp. 580–587.
[53] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 490–497.
[54] G. Mishne, U. Shaham, A. Cloninger, and I. Cohen, "Diffusion nets," *Appl. Comput. Harmon. Anal.*, vol. 47, no. 2, pp. 259–285, 2019.
[55] H. Qiao, P. Zhang, D. Wang, and B. Zhang, "An explicit nonlinear mapping for manifold learning," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 51–63, Feb. 2013.
[56] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Mach. Stud.*, vol. 7, no. 1, pp. 1–13, 1975.
[57] L. Yang, "Sammon's nonlinear mapping using geodesic distances," in *Proc. IEEE 17th Int. Conf. Pattern Recognit.*, 2004, pp. 303–306.
[58] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 1–7, Feb. 1993.
[59] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, Nov./Dec. 1992.
[60] O. Cordón and F. Herrera, "A two-stage evolutionary process for designing TSK fuzzy rule-based systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 6, pp. 703–715, Dec. 1999.
[61] N. R. Pal and S. Saha, "Simultaneous structure identification and fuzzy rule generation for Takagi-Sugeno models," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 6, pp. 1626–1638, Dec. 2008.
[62] Y.-C. Chen, N. R. Pal, and I.-F. Chung, "An integrated mechanism for feature selection and fuzzy rule extraction for classification," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 4, pp. 683–698, Aug. 2012.
[63] N. Asgharbeygi and A. Maleki, "Geodesic k-means clustering," in *Proc. IEEE 19th Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.
[64] J. Kim, K.-H. Shim, and S. Choi, "Soft geodesic kernel k-means," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2007, vol. 2, pp. II-429–II-432.
[65] B. Feil and J. Abonyi, "Geodesic distance based fuzzy clustering," in *Soft Computing in Industrial Applications.*Berlin, Germany: Springer, 2007, pp. 50–59.
[66] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, 1962, Art. no. 345.
[67] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[68] S. Roweis, "Data for MATLAB hackers, faces, Frey face," 2020. [Online]. Available: https://cs.nyu.edu/∼roweis/data/frey_rawface.mat
[69] S. A. Nene *et al.*, "Columbia object image library (coil-20)," Tech. Rep. CUCS-005-96, Feb. 1996. [Online]. Available: https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[70] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Des. Implementation*, 2016, pp. 265–283.
[71] G. Mishne, "Diffusionnet—A geometric autoencoder," 2017. [Online]. Available: https://github.com/gmishne/DiffusionNet
[72] L. van der Maaten, "Matlab toolbox for dimensionality reduction," 2020. [Online]. Available: https://lvdmaaten.github.io/drtoolbox/code/drtoolbox.tar.gz
[73] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *Proc. Int. Conf. Database Theory*, 2001, pp. 420–434.

Suchismita Das received the B.Tech. degree from the West Bengal University of Technology, Kolkata, India, in 2012, and the M.Tech. degree from the Indian Institute of Engineering Science and Technology, Shibpur, India, in 2014. She is currently working toward the Ph.D. degree with the Indian Statistical Institute, Kolkata.

Her research interests include manifold learning, feature selection, and machine learning.

Nikhil R. Pal, photograph and biography not available at the time of publication.