Abstract—A Bayesian deep restricted Boltzmann–Kohonen architecture for data clustering termed deep restricted Boltzmann machine (DRBM)-ClustNet is proposed. This core-clustering engine consists of a DRBM for processing unlabeled data by creating new features that are uncorrelated and have large variance with each other. Next, the number of clusters is predicted using the Bayesian information criterion (BIC), followed by a Kohonen network (KN)-based clustering layer. The processing of unlabeled data is done in three stages for efficient clustering of the nonlinearly separable datasets. During the first stage, DRBM performs nonlinear feature extraction by capturing the highly complex data representation by projecting the feature vectors of $d$ dimensions into $n$ dimensions. Most clustering algorithms require the number of clusters to be decided $a$ priori; hence, here, to automate the number of clusters in the second stage, we use BIC. In the third stage, the number of clusters derived from BIC forms the input for the KN, which performs clustering of the feature-extracted data obtained from the DRBM. This method overcomes the general disadvantages of clustering algorithms, such as the prior specification of the number of clusters, convergence to local optima, and poor clustering accuracy on nonlinear datasets. In this research, we use two synthetic datasets, 15 benchmark datasets from the UCI Machine Learning repository, and four image datasets to analyze the DRBM-ClustNet. The proposed framework is evaluated based on clustering accuracy and ranked against other state-of-the-art clustering methods. The obtained results demonstrate that the DRBM-ClustNet outperforms state-of-the-art clustering algorithms.

Index Terms—Bayesian information criterion (BIC), data clustering, deep restricted Boltzmann machine (DRBM), Kohonen network (KN).

I. INTRODUCTION

CLUSTERING is the task of the division of data into groups of similar objects [1]. The main objective of a clustering algorithm is to explore the inherent structure present in the samples and group the related samples [2]. Clustering is very useful in organizing data and understanding the hidden structure of the data. Clustering finds its plethora of applications in text analytics [3], [4], anomaly detection [5], hand-written digit recognition [6], social network analysis [7], customer grouping based on preferences in marketing sites [8], dynamic clustering [9], face recognition [10], model order detection [11], and heterogeneous system placement problems [12].

There are various types of clustering algorithms, such as partitional clustering, hierarchical clustering, model-based clustering, and density-based clustering [13]. The most popular partitional clustering algorithms are K-means and the Kohonen network (KN). In the case of model-based clustering, the expectation-maximization (EM) algorithm is most popular [14]–[16]. For K-means, the cluster centers are initialized randomly, and the samples are assigned to any one of the clusters using a similarity measure [14]. In every iteration, these cluster centers are updated until convergence is reached. In the KN, the attributes that are assigned weights are updated iteratively using a competitive learning mechanism [15]. The EM algorithm groups the data based on the maximum likelihood approach using the Gaussian mixture model [16]. The main drawback of the aforementioned clustering algorithms is that their performance mainly depends on the dimensionality of the feature vector and the prior assignment of the number of clusters [17], [18]. By using the Bayesian information criterion (BIC), we can remove the problem of specifying the number of clusters $a$ priori [19].

A common misconception in clustering is that having a larger number of samples with cluster information leads to a better discriminative cluster since the features contain information about the target (class/cluster). However, there are many reasons, such as the presence of irrelevant, redundant, or noisy features, which thwarts the clustering accuracy. Thus, it is essential to project features into a higher or lower order (and vice versa) by transforming nonlinearly separable data into linearly separable data. In the literature, the restricted Boltzmann machine (RBM) is the most popular machine learning model that can extract nonlinear features from linearly separable data [20]. However, non-negative data cannot be represented by negative data; for this reason, the deep RBM (DRBM) has been proposed to model positive data [21].
machine (RBM) has overcome such issues [20]–[22]. The RBM is a nonlinear feature extraction technique that is superior in comparison to linear feature extraction techniques, such as the principal component analysis [23].

The first Boltzmann machine (BM) model was developed in the 1980s and was inspired by statistical mechanics [24]. Smolensky [25] in 1986 implemented the Harmonium, which is a variant of the BM. The RBM is often described as the Monte Carlo version of Hopfield networks [26]. Even though the idea of BM has existed since the 1980s, it was more widely used from 2006 when Hinton et al. [27] implemented a modified version of BM, termed RBM, for a fast and efficient way of learning. RBM overcomes problems with the time required for the network to converge, which grows exponentially with the size of the samples [28]. In BM, it is difficult to obtain an unbiased sample from the posterior distribution given the data vector; the complexity is due to the connectivity within and between the layers. In RBM, the restrictedness arises due to the absence of visible to visible and hidden to hidden interactions in the network.

The learning algorithm used to train the RBM is based on a k-step contrastive divergence (CD) technique, which approximates the Markov chain Monte Carlo (MCMC) technique by sampling all the units in a layer at once [29]. After the emergence of this technique, and with advances in the availability of computational power, many researchers realized that shallow neural networks were not sufficient. Thus, there was a need for networks to learn deeper representations of the data [30], which is essential for computer vision [31], time-series forecasting, and other applications [32]. Later, Salakhutdinov and Hinton [33] developed deep BMs by stacking many layers of RBM and forming an undirected graphical model, which is unlike deep belief networks and uses directed graphical models [27], [34]. The deep RBM (DRBM) has been used as a generative model to learn the nonlinear distribution of a dataset [32]. The development of better computational hardware and high-performance processors has recently led to applications of the approach with larger datasets [35], [36]. Sankaran et al. [37] proposed an RBM-based semisupervised class sparsity signature by combining the unsupervised generative training with a supervised sparsity regularizer to learn better generative features. The RBM has also been applied for cancer detection [38], time-series prediction [40], detecting event-related potential [39], and estimation of 3-D trajectories from 2-D trajectories [41].

In this article, we propose the DRBM-ClustNet framework, which performs feature extraction using DRBM. The output of the first stage is used to predict the number of clusters with BIC. The feature-extracted data obtained with DRBM and the predicted number of clusters are used as an input for the clustering algorithm using the KN. The proposed framework’s performance is compared with six state-of-the-art clustering methods, namely, K-means [42], self-organizing maps (SOMs) [43], EM [44], density-based spatial clustering of applications with noise (DBSCAN) [45], unsupervised extreme learning machine (US-ELM) [46], and Bayesian extreme learning machines KN (BELMKN) [47]. The proposed DRBM-ClustNet and the state-of-the-art clustering algorithms are applied to two synthetic datasets, 15 benchmark datasets from the UCI Machine Learning Repository [48], and four image datasets. The performance evaluation is carried out using various statistical approaches.

The remaining content of this article is organized as follows. The architecture diagram of the DRBM-ClustNet framework with an abstract code is discussed in Section II. The clustering performance for two synthetic datasets with the DRBM-ClustNet framework and other available clustering algorithms is narrated in Section III. The results for various clustering methods when applied on 15 benchmark tabular datasets and four image datasets compared to the proposed approach are discussed in Section IV. Conclusions are drawn in Section V.

II. DRBM-CLUSTNET ARCHITECTURE

The proposed DRBM-ClustNet architecture for data clustering is discussed in this section. The DRBM is used for feature extraction, followed by BIC that is applied to predict the number of cluster centers for the feature extracted data. Finally, the above two levels are used as input for data clustering using the KN. The detailed flow of the DRBM-ClustNet architecture is shown in Fig. 1.

A. Feature Extraction Using DRBM

The RBM is an energy-based undirected bipartite graphical model, as shown in layer 1 ($L_1$) of Fig. 1. For a given dataset, initially, we use the min–max normalization on a real-valued dataset for feature extraction, whereas the conventional RBM uses binary-valued data. The normalization of the dataset performed is the same as in [49] and [50]

$$F(x_i) = \frac{0.9 - 0.1}{\max(x_i) - \min(x_i)}(x_i - \min(x_i)) + 0.1$$

(1)

where $x_i$ is a dataset with $s = \{1, 2, \ldots, N\}$, and $\min(x_i)$ and $\max(x_i)$ correspond to the minimum and maximum values for a particular feature with $i = \{1, 2, \ldots, d\}$.

The normalized dataset is passed to $L_1$ of the DRBM network, but $L_1$ consists of a visible layer and a hidden layer. The visible units $v_i, i = 1, d$, are real-valued, and the hidden units are given by $h_j, j = 1, p$. Let $W^1 \in \mathbb{R}^{d \times p}$ denote the weights between the hidden layer and visible layer units in $L_1$, and $w_{ij}$ indicates the weights between the $i$th visible unit and the $j$th hidden unit. As RBM is an energy-based model, the probability distribution of the visible layer and hidden layer units is stated in the form of an energy equation. The joint probability distribution of the visible layer and the hidden layer units $P(v, h)$ is given by

$$P(v, h) = \frac{1}{Z}e^{-E(v, h)}$$

(2)

where $E(v, h)$ is the energy of the joint configuration between visible layer and hidden layer units, and is given by

$$E(v, h) = -\sum_{i=1}^{d} \sum_{j=1}^{p} w_{ij} v_i h_j - \sum_{i=1}^{d} b_i v_i - \sum_{j=1}^{p} c_j h_j$$

(3)
where \( b_i \) is the visible layer bias and \( c_j \) is the hidden layer bias. \( Z \) is the partition function defined as

\[
Z = \sum_{i=1}^{d} \sum_{j=1}^{p} e^{-E(v_i, h_j)}.
\]

The partition function \( Z \) is intractable. As a result, the joint probability distribution of the visible and the hidden units \( P(v, h) \), which depends on \( Z \), is also intractable. Therefore, we express the joint probability distribution \( P(v, h) \) in (2) in terms of the conditional probability distribution \( P(v|h) \) and \( P(h|v) \). This is much easier to compute and is defined as

\[
P(v|h) = \prod_{i=1}^{d} P(v_i|h)
\]

(5)

\[
P(h|v) = \prod_{j=1}^{p} P(h_j|v).
\]

(6)

From (5) and (6), we can observe that the conditional probabilities are all independent. This indicates that the state of the hidden layer units is independent given the states of the visible layer units (without interconnecting the neurons in the same layer either in visible or hidden layer). Therefore, this type of BM is defined as an RBM. The conditional probabilities from (5) and (6) are derived as a sigmoidal function given by

\[
P(h_j|v) = s\left(\sum_{i=1}^{d} v_i w_{ij} + c_j\right)
\]

(7)

\[
P(v_i|h) = s\left(\sum_{j=1}^{p} h_j w_{ij} + b_i\right)
\]

(8)

where \( s(\cdot) \) denotes the sigmoidal function and \( s(x) = (1/1 + e^{-x}) \).

RBM is a generative model from which we predict \( P(h|v) \) by clamping the visible units and \( P(v|h) \) by using the hidden activation obtained from \( P(h|v) \). Hinton [29] proposed the \( k \)-step CD algorithm for training the RBM, which jointly performs the Gibbs sampling for all variables in one layer instead of sampling the new values of all variables one by one. The one-step CD is divided into two phases, namely, positive CD and negative CD, followed by weight updating.

1) Positive CD: The probability that a hidden state is evaluated by clamping all the visible units

\[
P(h|v) = s(v_{<0}> W + c) = h_{<0>}
\]

(9)

Then, we calculate the interactions between the visible layer and hidden layer units as

\[
a^+ = v_{<0>}, h_{<0>} = (vh)_{<0>}
\]

(10)

where \( a^+ \) denotes the positive associations.

2) Negative CD: The probability of a visible state given the hidden activations from (9) to get \( v_{<1>}. \) Then, we use these visible units to find \( h_{<1>}. \)

\[
P(v|h) = s(W h_{<1>} + b) = v_{<1>}
\]

(11)

\[
P(h|v) = s(W v_{<1>} + c) = h_{<1>}
\]

(12)

Then, we calculate the negative associations between the visible layer and hidden layer units

\[
a^- = v_{<1>}, h_{<1>} = (vh)_{<1>}
\]

(13)

3) Weight Update Step: Update the weights by considering the difference between the positive associations from (10) and negative associations from (13) using adaptively decreasing learning rate \( \epsilon \)

\[
\Delta w = \epsilon (a^+ - a^-)
\]

(14)

The above one-step CD algorithm is applied, and the states of visible and hidden units are sampled alternately.
In many cases, a single-layer \((L_1)\) RBM will not be able to capture the complete nonlinearity in the dataset. As a result, we need to stack multiple RBM layers \((L_1, L_2,\ldots, L_n)\) to produce a DRBM. In DRBM, the activation of one hidden layer forms the training samples for the next hidden layer. This technique is efficient in learning complex data representations. DRBM can efficiently learn a generative model of the high-dimensional input and large-scale dataset [20].

Consider the DRBM network in the proposed architecture shown in Fig. 1, where feature extraction is carried out using the layers \((L_1, L_2,\ldots, L_n)\). Let \(h^l\) \((l \leq 1 \text{ to } n)\) denote the \(l\)th hidden layer and \(W^l\) denote the weights between the layers. The energy equation for the DRBM is given by

\[
E(\psi, h^1, \ldots, h^n; \theta) = -\theta W^1 h^1 - h^1 W^2 h^2 - \cdots - h^{n-1} W^n h^n
\]

where \(\theta = \{W^1, W^2, \ldots, W^n\}\) are the model parameters that have to be trained and \(n\) is the number of hidden layers in the DRBM network [51].

In DRBM, the first layer RBM is trained using the one-step CD to obtain the reconstructions of the visible vectors. Then, the next layer RBM is trained using the first layer sampled hidden layer activations \(h^1\) obtained from \(P(h^1|v; W^1)\) as the training data for the second layer RBM and obtain the reconstructions of the visible units. Continue this process recursively until layers \(n-1\). Using all the trained weights \(\theta = \{W^1, W^2, \ldots, W^n\}\) and the input data features as the visible units, a feedforward pass is performed until the last hidden layer \(n\). The final activations of the hidden layer \(n\) are the feature-extracted data \((\hat{x}^n_i)\).

B. BIC for Cluster Prediction

To automate the process of cluster prediction, DRBM-ClustNet uses the BIC. This is applied to the feature-extracted data obtained from DRBM instead of the actual dataset, as shown in Fig. 1. Most of the clustering algorithms require a parameter called the number of clusters to be inputted manually. As in real-world problems, datasets may be with or without labels; hence, BIC is applied to statistically predict the number of clusters.

BIC uses multivariate Gaussian distribution and parameters that are the mean and covariance matrices. These parameters are estimated using the EM algorithm. BIC for cluster prediction is defined as

\[
\text{BIC} = \ln(N)k - 2\ln(\hat{L})
\]

where \(N\) is the total number of samples, \(\hat{L}\) is the maximized value of the likelihood function, and \(k\) is the total number of free parameters to be estimated. The BIC is computed for \(c = 1, 2, \ldots, N\); the model value with minimum BIC value is selected, and its corresponding number of clusters \(n_c\) is used for clustering the data.

C. Kohonen Network for Data Clustering

The DRBM-ClustNet uses the feature-extracted data from DRBM and the BIC selected number of clusters \((n_c)\). The data clustering is performed using the KN. The KN is made up of input and output layer. The number of output layer neurons \((n_o)\) depends on the total number of clusters \((n_c)\) predicted using BIC. The weights between the input and the output layer are \(W_k \in R^{n_i \times n_o}\), where \(n_i\) is the number of input neurons and the weight matrix is randomly initialized. The weight is updated iteratively using the discriminant function considering the Euclidean distance given by

\[
d(j) = \sqrt{\sum_{j=1}^{n_o} (\hat{x}_i - \hat{w}_{ij})}, \quad j \in 1 \text{ to } n_o.
\]

The winning neuron is the one, which has minimum separation with the input sample. The weight is updated iteratively considering the neighborhood function (i.e., the neurons that are within its vicinity) using

\[
h_{ci}(t) = \alpha(t) \exp\left(-\frac{d_{ij}^2}{2\sigma(t)}\right)
\]

where \(t\) is the iteration, \(\alpha(t)\) is the learning rate for given iteration \(t\), and \(\sigma(t)\) is the spread of the data points in consideration and is given by

\[
\alpha(t) = \alpha_0 \exp\left(-\frac{t}{T_1}\right)
\]

\[
\sigma(t) = \sigma_0 \exp\left(-\frac{t}{T_2}\right)
\]

where \(T_1\) and \(T_2\) are time constants.

The updating of the weights for the winning neuron and its neighboring neurons within its vicinity is computed as follows:

\[
\delta w_{ij} = h_{ci}(t)(y_i - \hat{w}_{ij}).
\]

DRBM-ClustNet uses this as a final layer for data clustering. To know the grouping of the data points to individual class labels is evaluated using a performance measure. The clustering efficiency can be evaluated using the clustering accuracy \((\eta)\)

\[
\eta = \frac{\sum_{i=1}^{n_o} a_i}{N}
\]

where \(a_i\) is the correctly clustered samples and \(N\) is the total number of samples in the data. The proposed DRBM-ClustNet is summarized in Algorithm 1.

III. DRBM-CLUSTNET APPLIED ON SYNTHETIC DATASET

In this section, we discuss the results of the proposed framework and other clustering algorithms for two synthetic datasets, as shown in Figs. 2 and 3. The results obtained are evaluated and benchmarked against the state-of-the-art clustering algorithms, such as K-means [42], SOM [43], EM [44], and DBSCAN [45]. Similarly, clustering is compared with feature learning algorithms, such as US-ELM [46], BELMKN [47], and single-layer RBM [18].

The first dataset is of the flame distribution, which consists of 600 instances. It has two classes with 300 instances each. The second dataset is of the moon distribution, which consists of two classes (two half-moons) each of which has
Fig. 2. Clustering of flame pattern distribution using (a) K-means, (b) SOM, (c) EM, (d) DBSCAN, (e) US-ELM, (f) BELMKN, (g) single-layer RBM, and (h) DRBM-ClustNet.

Fig. 3. Clustering of moon pattern distribution using (a) K-means, (b) SOM, (c) EM, (d) DBSCAN, (e) US-ELM, (f) BELMKN, (g) single-layer RBM, and (h) DRBM-ClustNet.

150 samples. The flame distribution and the moon distribution datasets are non-linearly distributed, and clustering them accurately is a challenge.

For each of the two datasets, DRBM-ClustNet is applied and compared with the aforementioned clustering techniques, and their performance is analyzed. DRBM-ClustNet uses DRBM for feature extraction by setting the parameters such as the number of hidden layers and the number of hidden neurons in each layer. We can observe from Fig. 2 that K-means, SOM, EM, DBSCAN, and US-ELM have failed to cluster the dataset efficiently. However, BELMKN clusters the dataset efficiently as it uses ELM for feature learning and the KN for clustering. The clustering accuracy using BELMKN for the flame distribution dataset is 98.5%. The single-layer RBM has few misclassification samples to capture the nonlinearity in the dataset. It is overcome by stacking the multiple layers of RBM to form a DRBM, which is used in the proposed DRBM-ClustNet. The DRBM-ClustNet architecture for the flame distribution dataset consists of five hidden layers. For the proposed DRBM-ClustNet, we can observe that there is no misclassification for the flame distribution dataset, where the performance is better compared to the other clustering algorithms used in this article.

Similarly, we can observe from Fig. 3(a)–(g) that, for the moon distribution dataset, all the clustering algorithms fail to cluster the corners of the moon-shaped distribution. Fig. 3(h) shows the proposed method DRBM-ClustNet that can cluster even the corners without any samples being misclassified.

By observing the two synthetic datasets used here, we can infer that the proposed DRBM-ClustNet that uses DRBM
we conclude that the DRBM-ClustNet can be used for feature extraction gives a better clustering accuracy in comparison to state-of-the-art clustering algorithms. Hence, we conclude that the DRBM-ClustNet can be used for the efficient clustering of nonlinearly distributed datasets.

IV. RESULTS AND DISCUSSION

This section discusses the results for the DRBM-ClustNet framework on 15 different benchmark datasets from the UCI repository. Initially, we discuss the characteristics of the datasets shown in Table I. Next, the number of clusters is predicted using BIC with the feature-extracted data obtained using three feature learning techniques, namely, ELM, single-layer RBM, and DRBM-ClustNet. The cluster prediction of these feature learning techniques, namely, ELM, single-layer RBM, and DRBM-ClustNet. The cluster prediction of these feature learning techniques, namely, ELM, single-layer RBM, and DRBM-ClustNet.

A. Dataset Description

In this research, we have applied the proposed DRBM-ClustNet on 15 benchmark datasets, as shown in Table I (https://archive.ics.uci.edu/ml/index.php). We have adopted 15 widely used datasets of the UCI repository to compare with other prominent clustering methods. The number of samples/observations, the number of features/attributes, and the actual number of clusters for every dataset used in this work are shown in Table I. The detailed description for the datasets is given as follows.

Dataset 1: The Balance data classify a sample into one of the three classes, namely, balance scale tip to the right, tip to the left, or to be balanced. It consists of four attributes, which includes the left weight, the left distance, the right weight, and the right distance, and 625 samples.

Dataset 2: The Cancer data categorize a tumor either as benign or malignant. It has 30 attributes and 569 observations in total.

Dataset 3: The Cancer-Int data classify a breast tumor either as malignant or benign. It has 699 observations and nine attributes.

Dataset 4: The Credit data have two classes, namely, to grant a credit approval or not. It has 14 attributes with 690 observations.

Dataset 5: The Dermatology data have different diagnosis details for the erythemato-squamous diseases in dermatology. It has six classes, 34 attributes, and 366 observations.

Dataset 6: The Diabetes data are to diagnostically predict whether a patient has diabetes based on certain diagnostic measurements. It has 768 observations and nine attributes.

Dataset 7: The E. coli data have details of the proteins’ cellular localization sites. It contains five classes, 327 observations, and seven attributes.

Dataset 8: The Glass data have details for the oxide content and refractive index for different glass types. It consists of six classes, 214 observations, and nine attributes.

Dataset 9: The Heart data consist of details for the diagnosis of heart diseases. It consists of two classes, 270 observations, and 13 attributes.

Dataset 10: The Horse data have the details for the different types of horses. It consists of three classes, 364 observations, and 26 attributes.

Dataset 11: The Iris data classify a sample into one of the three types of flower classes, namely, setosa, virginica, or versicolor. It has 150 observations and four attributes, namely, the width of petals, the length of petals, the width of sepals, and the length of the sepals.

Dataset 12: The SECOM data have signals from various sensors during the process of manufacturing semiconductors. There are 1567 observations and 590 features. The dataset consists of two classes, nonfail and fail, and is highly imbalanced. The missing values in attributes were imputed with zeroes.

Dataset 13: The Thyroid data consist of three classes, namely, normal functional, under functional, and over functional. There are 215 observations and five attributes.

Dataset 14: The Vehicle data consist of different types of vehicles, namely, bus, Opel, van, and Saab. There is a total of 846 observations and 18 attributes.

Dataset 15: The Wine data list different types of chemical analyses of wine. It consists of three classes, 178 observations, and 13 attributes.
B. Parameters for Algorithm

All the datasets used in this article are preprocessed, which includes imputation, encoding, and scaling. In the imputation process, all the missing values were imputed with a zero. All the categorical features were label encoded or one-hot encoded based on feature description. All distance calculation is based on the Euclidean distance. Distance-based techniques are sensitive to the scale of features. Therefore, to avoid bias toward high-value features, all the features were normalized between 0 and 0.9 using min–max scaling. This also ensures a faster convergence for neural network-based approaches. It is necessary to tune the hyperparameters for effective learning. Since our hyperparameters are in a well-defined range, the standard grid search approach [52] has been used to find an optimal set of hyperparameters for the data clustering. A single RBM had one hidden layer with 50 hidden units. The DRBM has five hidden layers with 50 hidden units in each of the first four layers and ten hidden units in the final layer. The maximum epochs and the learning rate are set to 50 000 and 0.1, respectively. Furthermore, as all these algorithms are dependent on initial random points or weights, ten runs were carried out, and the average and the standard deviation were recorded.

C. Assessment of the Cluster Prediction by DRBM-ClustNet

The BIC is used initially to evaluate the number of clusters for a given dataset. The proposed DRBM-ClustNet uses DRBM for feature extraction on the abovementioned dataset. The feature-extracted dataset is used to predict the number of clusters with BIC. Furthermore, the obtained results are compared with the feature-extracted data based on ELM, the single-layer RBM for the number of clusters predicted by BIC, along with the BIC applied to the actual data, as shown in Table II.

In Table II, we observe a slight change in the actual number of clusters in the database compared with the BIC predicted number of clusters for various feature learning techniques. From Table II, we can observe that, for the Cancer and Cancer-Int datasets, none of the feature-extracted data with BIC were able to predict the actual number of clusters. Here, instead of two clusters, all the feature learning techniques with BIC predict three clusters. This kind of behavior is observed in Cancer and Cancer-Int datasets as they are linearly separable. By applying feature learning to these datasets, we are introducing additional nonlinearity. As a result, feature learning techniques, such as ELM, single-layer RBM, and DRBM, fail for these two datasets. In contrast, for the Dermatology, Iris, Thyroid, and Wine datasets, the methods predicted the exact number of clusters. Furthermore, the Glass and SECOM datasets are highly nonlinear; hence, we get the actual number of clusters by applying the feature learning techniques to these two datasets. For the Balance, Horse, and Vehicle datasets, the USELM, BELMKNN, and single-layer RBM feature learning data on BIC fail to accurately predict the number of clusters. Only the DRBM feature learning method accurately predicts the exact number of clusters as three, three, and four for Balance, Horse, and Vehicle datasets, respectively. This is observed, as these three datasets are highly nonlinear, and the nonlinearity in the datasets can be captured by stacking many layers of RBM. As a result, we can conclude that the DRBM-BIC is a good approach for the prediction of the number of clusters for nonlinearly distributed data.

D. Data Visualization for DRBM

To analyze the behavior of data distribution in some of the scenarios where DRBM can pick the exact clusters, it is possible to visualize the data from high dimension to low dimension, and vice versa. This is analyzed with and without applying DRBM on the dataset using a parallel coordinates plot, a scatter plot, and a correlation matrix [53].

The architecture used for DRBM feature learning has ten hidden neurons in the last layer. The Iris dataset has four features, and the DRBM feature learning, obtained in the last layer, contains ten dimensions. Similarly, the Wine dataset has 13 features reduced to ten features using ten hidden neurons in the last layer of the DRBM architecture. The parallel coordinates and scatter plots for the Iris and Wine datasets without DRBM (actual data) and with DRBM feature-extracted data

| Dataset     | Actual Clusters | BIC on actual data | ELM with BIC | Single Layer RBM BIC | DRBM with BIC |
|-------------|-----------------|--------------------|--------------|----------------------|---------------|
| Balance     | 3               | 5 ± 0.04           | 7 ± 0.31     | 5 ± 0.00             | 3 ± 0.00      |
| Cancer      | 2               | 3 ± 0.47           | 3 ± 0.26     | 3 ± 0.00             | 3 ± 0.00      |
| CancerInt   | 2               | 2 ± 0.63           | 3 ± 0.37     | 3 ± 0.00             | 3 ± 0.00      |
| Credit      | 2               | 2 ± 0.39           | 5 ± 0.31     | 4 ± 0.00             | 6 ± 0.00      |
| Dermatology | 6               | 5 ± 0.99           | 6 ± 0.87     | 6 ± 0.44             | 6 ± 0.39      |
| Diabetes    | 2               | 2 ± 0.93           | 3 ± 0.79     | 5 ± 0.31             | 5 ± 0.22      |
| E.Coli      | 5               | 4 ± 0.67           | 5 ± 0.61     | 5 ± 0.00             | 5 ± 0.00      |
| Glass       | 6               | 3 ± 0.81           | 6 ± 0.73     | 6 ± 0.00             | 6 ± 0.00      |
| Heart       | 2               | 2 ± 1.07           | 3 ± 1.18     | 4 ± 0.94             | 4 ± 0.81      |
| Horse       | 3               | 2 ± 0.75           | 2 ± 0.88     | 2 ± 0.18             | 3 ± 0.18      |
| Iris        | 3               | 3 ± 0.31           | 3 ± 0.02     | 3 ± 0.00             | 3 ± 0.00      |
| SECOM       | 2               | 3 ± 0.15           | 2 ± 0.54     | 2 ± 0.44             | 2 ± 0.22      |
| Thyroid     | 3               | 3 ± 0.39           | 3 ± 0.31     | 3 ± 0.00             | 3 ± 0.00      |
| Vehicle     | 4               | 3 ± 1.25           | 3 ± 1.09     | 3 ± 0.98             | 4 ± 0.65      |
| Wine        | 3               | 3 ± 0.49           | 3 ± 0.23     | 3 ± 0.00             | 3 ± 0.00      |
Fig. 4. Parallel coordinates plot for (a) Actual Iris dataset and (b) Iris dataset after DRBM feature extraction.

Fig. 5. Scatter plot for (a) Actual Wine dataset and (b) Wine dataset after DRBM feature extraction.

are shown in Figs. 4 and 5. From Fig. 4(a), for the Iris dataset, we can observe that there is an overlap between the two classes (class 2 and class 3). However, after applying DRBM feature learning, in Fig. 4(b), we can observe that the overlap is reduced with a clear separation between these classes. For the Wine dataset, we can observe from Fig. 5(a) and (b) that there is a significant difference between the two scatter plots. Because of the nonlinear dimensionality reduction from 13 to ten dimensions in the DRBM feature extraction, linear separability is introduced to the dataset. Hence, by applying DRBM feature extraction to the Iris dataset, we can significantly reduce the correlation between the feature vectors, which, in turn, results in high clustering accuracy.

Overall, from the above analysis, we can conclude that, by decreasing the dimension, DRBM performs well for five datasets (Dermatology, Horse, SECOM, Vehicle, and Wine). Hence, DRBM with BIC performs well for ten out of the 15 datasets, i.e., in 66.6% of the cases, whereas BIC applied on the actual dataset performs well for seven out of 15 datasets, i.e., for 46.6% of the datasets. Both ELM and the single-layer RBM perform well for seven out of 15 datasets, i.e., in 46.6% of the datasets. Another observation from Table II is that DRBM with BIC predicts clusters consistently as the standard deviation is almost zero for most of the datasets, whereas a large variation is present for other techniques. Hence, DRBM works effectively and efficiently for nonlinearly distributed datasets.
E. Assessment of the Cluster Accuracy of DRBM-ClustNet

The proposed DRBM-ClustNet is applied to the above-mentioned 15 benchmark UCI machine learning repository datasets [48]. In the DRBM-ClustNet, the feature-extracted data from the DRBM network and the cluster prediction output from BIC are given for clustering using the KN. This approach shows the best performance compared to the traditional clustering algorithms, such as K-means, EM, SOM, and DBSCAN, and feature learning-based clustering techniques, such as USELM, BELMKN, and single-layer RBM.

The K-means algorithm involves random initialization of centroids, and the cluster centers are iteratively updated until it converges. The main drawback of the K-means is the random choice of initial clusters. In K-means, we assume that each attribute has the same weight, i.e., all attributes are assumed to have the same contribution toward clustering. In addition, it gets stuck in local optima, and we can observe from Table III that the performance of the K-means algorithm is the worst of all the approaches. Unlike K-means, SOM gives weights for each attribute by using the neighborhood concept and iteratively computes the cluster centers. The main drawback of SOM is ineffective in clustering nonlinearly separable datasets as the hidden layers are absent. The EM algorithm iteratively computes the cluster centers using the maximum likelihood approach. The performance of EM is better than SOM and K-means as it is a soft clustering algorithm. The DBSCAN finds core samples of high density and expands clusters from them. However, as the quality of DBSCAN also depends on the distance measure used in the function region, it becomes challenging at higher dimensions to find appropriate separation or clusters.

The USELM and BELMKN perform better in comparison to K-means, DBSCAN, SOM, and EM. USELM and BELMKN use ELM with K-means and KN as the clustering algorithms, respectively. As BELMKN uses the KN, this clustering algorithm overcomes the stated drawbacks of K-means. As both USELM and BELMKN use ELM for feature learning, they suffer from drawbacks of ELM. ELM is a single hidden layer neural network where the weights between the input layer and hidden layer are randomly initialized, and the weights between the hidden layer and the output layer are computed using a closed-form solution. This, sometimes, turns out to be a major drawback as it increases the amount of randomness in the network, and hence, the clustering results obtained through ELM feature learning are not consistent, which may result in either underfitting or overfitting. From Table III, we can observe that the standard deviation for USELM and BELMKN is greater than 1 for all the datasets. For the datasets that are highly nonlinear, such as Balance, Credit, Dermatology, Glass, and Heart, the standard deviation is very high compared to the other datasets, which indicates that the clustering results are not consistent for ELM. Hence, for USELM and BELMKN, we need to run the program in several trials to obtain better accuracy. Another disadvantage of ELM is that it sometimes overfits the data. As a result, more samples are drawn into one of the clusters that dominate the data, and the remaining clusters suffer from the sparsity of samples. Hence, we can observe from Table III that, for datasets such as Balance, Dermatology, and Thyroid, the clustering accuracies of the ELM-based algorithms, USELM and BELMKN, are lower than the ones obtained by the proposed DRBM-ClustNet.

DRBM-ClustNet is modeled as a bipartite graphical model to exploit the generative nature of data and estimate better probability distributions. Each hidden neuron in a DRBM-ClustNet is a stochastic processing unit, which learns a probability distribution over the inputs. The hidden units in a DRBM-ClustNet capture higher order correlations between the inputs. In DRBM-ClustNet, the convergence to the minima is achieved in two steps, namely, based on the positive CD and the negative CD, which is accomplished by Gibbs sampling [20]. Hence, the convergence to a global minimum is better guaranteed compared to ELM-based feature learning. As a result, the clustering accuracy for the datasets using DRBM-ClustNet is higher compared to USELM and BELMKN. In addition, there are higher accuracies obtained by DRBM-ClustNet for the nonlinear datasets, such as Balance, Credit, Dermatology, Glass, and Heart. As DRBM-ClustNet is less immune to the nonlinearity in the dataset, it does not overfit the data.

Fig. 6. Correlation matrix plot for (a) Original Iris dataset and (b) Iris dataset after RBM feature extraction.
TABLE III
CLUSTERING ACCURACY PERCENTAGE WITH STANDARD DEVIATION AND RANKING (IN PARENTHESIS BELOW) OF VARIOUS TECHNIQUES ON EACH DATASET

| Dataset | K-means | SOM | EM | USELM | BELMKN | DBSCAN | Single RBM | DRBM-ClustNet |
|---------|---------|-----|----|-------|--------|--------|-----------|---------------|
| Balance | 51.4 ± 0.9 | 52.5 ± 0.2 | 53.0 ± 0.6 | 45.9 ± 3.3 | 47.1 ± 3.4 | 52.5 ± 0.5 | 68.1 ± 0.16 | 70.8 ± 0.06 |
| Cancer  | 83.2 ± 1.9 | 86.0 ± 1.0 | 91.2 ± 1.6 | 83.8 ± 1.6 | 86.7 ± 2.0 | 88.8 ± 0.8 | 95.0 ± 0.2 | 93.4 ± 0.2  |
| Cancerint| 95.8 ± 1.1 | 94.2 ± 0.4 | 93.6 ± 1.6 | 92.0 ± 1.5 | 92.5 ± 1.8 | 91.9 ± 0.9 | 96.1 ± 0.7 | 97.0 ± 0.41 |
| Credit  | 52.2 ± 2.8 | 54.8 ± 1.2 | 51.4 ± 1.8 | 58.5 ± 4.1 | 61.5 ± 3.8 | 67.1 ± 1.3 | 85.3 ± 0.13 | 86.5 ± 0.26 |
| Dermatology | 24.2 ± 3.9 | 29.8 ± 1.5 | 67.7 ± 0.4 | 71.0 ± 3.9 | 76.6 ± 3.6 | 63.2 ± 0.9 | 92.8 ± 0.4 | 95.5 ± 0.28 |
| Diabetes | 63.6 ± 1.5 | 64.8 ± 0.8 | 52.4 ± 0.5 | 65.0 ± 1.9 | 67.9 ± 2.4 | 61.7 ± 0.8 | 69.2 ± 0.23 | 71.0 ± 0.25 |
| E.Coli | 53.8 ± 5.7 | 61.0 ± 3.8 | 82.0 ± 0.5 | 80.1 ± 2.7 | 80.7 ± 1.8 | 52.8 ± 4.2 | 80.9 ± 1.2 | 83.1 ± 0.19 |
| Glass | 52.1 ± 3.8 | 53.8 ± 2.9 | 47.7 ± 0.4 | 40.2 ± 4.1 | 41.5 ± 3.5 | 39.8 ± 3.7 | 54.4 ± 0.21 | 57.6 ± 0.26 |
| Heart | 58.2 ± 3.4 | 58.8 ± 2.2 | 52.6 ± 1.6 | 66.6 ± 2.2 | 71.4 ± 2.9 | 58.3 ± 2.0 | 79.2 ± 0.27 | 82.4 ± 0.24 |
| Horse | 49.7 ± 4.4 | 48.3 ± 2.3 | 43.4 ± 1.9 | 58.6 ± 4.8 | 60.9 ± 4.1 | 47.6 ± 3.1 | 62.0 ± 0.56 | 65.3 ± 0.26 |
| Iris | 88.6 ± 4.0 | 81.3 ± 0.8 | 90.0 ± 0.3 | 91.5 ± 3.3 | 92.3 ± 4.8 | 77.2 ± 1.8 | 91.5 ± 0.33 | 93.8 ± 0.33 |
| SECOM | 62.2 ± 2.2 | 60.6 ± 0.3 | 66.1 ± 0.7 | 65.9 ± 1.2 | 69.4 ± 2.3 | 61.6 ± 0.9 | 73.1 ± 0.40 | 73.5 ± 0.28 |
| Thyroid | 85.8 ± 2.2 | 86.2 ± 0.3 | 90.1 ± 0.7 | 86.5 ± 1.2 | 87.6 ± 2.3 | 85.6 ± 0.9 | 94.1 ± 0.09 | 92.0 ± 0.09 |
| Vehicle | 44.3 ± 2.2 | 44.0 ± 1.5 | 45.0 ± 1.8 | 40.5 ± 3.8 | 42.0 ± 2.2 | 43.3 ± 1.5 | 46.0 ± 0.62 | 48.4 ± 0.22 |
| Wine | 70.0 ± 3.4 | 75.0 ± 0.0 | 90.4 ± 0.3 | 91.9 ± 1.1 | 95.5 ± 1.7 | 74.8 ± 0.4 | 95.8 ± 0.08 | 97.2 ± 0.08 |

The RBM network with one hidden layer is not sufficient to capture the nonlinearity of a dataset even with an increase in the number of hidden neurons in that layer. This is evident for almost all the datasets from Table III. The proposed DRBM-ClustNet overcomes this problem during the processing of input data. Initially, it has one hidden layer, but the output of this layer is taken to the next hidden layer, and a stack of RBM layers is built up producing a better generative model. The DRBM-ClustNet has the potential to learn internal representations of the data that are increasingly complex. As a result, DRBM-ClustNet performs better in terms of accuracy compared to the RBM with one hidden layer and also in comparison with the other clustering algorithms used in this study.

Furthermore, Table III shows that the clustering results using single-layer RBM and DRBM-ClustNet are consistent for every dataset. The standard deviation is less than about 0.7 and 0.41 in all cases for the single-layer RBM and DRBM-ClustNet, respectively. The proposed approach also works for the clustering of high-dimensional and highly imbalanced datasets. This can be observed by looking at the results in the table for the SECOM data, which has 590 attributes and a minority class of 6%. Here, it can be stated that the DRBM-ClustNet not only outperforms the other algorithms in terms of accuracy but also does well when a minority class is in the data.

From Table III, the Thyroid dataset, average clustering accuracies with RBM with one hidden layer and DRBM-ClustNet is 94.1% and 92.0%, respectively. The same kind of results is also observed for the Cancer dataset. There is an exception here in that the feature extraction with RBM with one hidden layer produces a higher clustering accuracy compared to the DRBM-ClustNet. This occurs because the Thyroid dataset is a more linearly separable dataset. Increasing the number of hidden layers for a simple dataset may result in overfitting. As a result, a drop in the accuracy of the DRBM-ClustNet is observed. Overall, from Table III, it is evident that, for 13 out of 15 datasets (i.e., 86.6%), the DRBM-ClustNet performs best in terms of accuracy compared to all other clustering approaches.

Table IV shows the average clustering accuracy for each of the clustering approaches applied to the 15 benchmark datasets. Here, we can infer that the DRBM-ClustNet has the maximum average clustering accuracy. The single-layer RBM ranks second, then BELMKN, followed by USELM. Among the traditional clustering approaches, EM fares better than DBSCAN, SOM, and K-means. Table V represents the sum of the ranks for all the datasets and clustering approaches shown in Table III. The ranking based on the sum of ranks indicates that the proposed DRBM-ClustNet outperforms the other clustering approaches. The Single RBM, BELMKN, EM, USELM, SOM, K-means, and DBSCAN follow DRBM-ClustNet in ranking order.

F. Assessment of the Clustering on Image Datasets

In this section, the proposed algorithm is compared with state-of-the-art algorithms based on RBM, namely,
Graph RBM, mixed GraphDBN (mGraphDBN), and full GraphRBM-based DBN (fGraphDBN) [54]. The performance of the DRBM-ClustNet algorithm is assessed on three publicly available image datasets, namely, COIL-20, the Extended Yale database B (YaleB), and MNIST. All the observations in COIL-20 and YaleB are used for experimentation. In the case of MNIST data, 60,000 training dataset observations are used in the experimental study as in [54]. The Whitening Principal Component Analysis is applied for reducing the dimension of all the three image datasets to 400 as in [54]. The setting of DRBM-ClustNet remains the same as in Section IV-E. The obtained results are also compared with the normalized mutual information (NMI) as performance metric [54], [55] and tabulated in Table VI. The populated values are from ten runs and indicate the mean with its standard deviation. The performance of other algorithms in [54] is listed for reference in the table.

From Table VI, it can be observed that DRBM-ClustNet performs best for the MNIST and YaleB datasets. In the case of COIL-20, DRBM-ClustNet performs better than GraphRBM and mGraphDBN but shows fractionally lower performance than fGraphDBN. This performance could be attributed to better subspace clustering by DRBM-ClustNet in the final stage, which uses the KN. Furthermore, it is also observed that the average numbers of clusters predicted by the second stage of DRBM-ClustNet for COIL-20, YaleB, and MNIST datasets are 19 ± 0.87, 38 ± 0.61, and 10 ± 0.21, respectively. The actual numbers of clusters for COIL-20, YaleB, and MNIST datasets are 20, 38, and 10, respectively. This indicates the ability of DRBM-ClustNet to accurately pick the number of clusters without having to define them explicitly like other methods. Overall, these results demonstrate the superior performance of the DRBM-ClustNet framework in comparison to other state-of-the-art RBM algorithms for various nonlinear image datasets.

**Assessment of the Flooded Region Using Satellite Imagery:** The DRBM-ClustNet is further evaluated for clustering actual satellite images of the flood-prone region. This application would aid in the quick identification of the extent of flood-prone regions. Here, three MODIS satellite images of various stages, namely, before the flood, during the flood, and after the flood, are used as shown in Fig. 7. The dimension of each of the three images is 482 × 627 pixels. Details of the MODIS images used in this study are discussed in [56]. All three images were processed using DRBM-ClustNet for spectral clustering to assess the extent of water (flood) cover. The extracted flood regions across all three images are highlighted in the figure with blue (actual river) and red (flooded region) colors.

**MODIS satellite images are of low spatial resolution (250 m); hence, the study prominently focuses on city granularity for the current application.** Fig. 7(b) shows results of the proposed approach during a flood timeline, wherein white dots signify cities that were not flooded, while the flooded cities are indicated with white dots with black dots inside. Here, it can be observed that the DRBM-ClustNet is able to correctly identify flooded regions for 12 out of 15 cities with just one false positive. Other clustering approaches, such as K-means and SOM, are able to identify only six and eight cities with false positives being 4 and 5, respectively. These numbers reflect the inability to identify the majority of flooded regions accurately. In the case of before the flood and after the flood, K-means clustering accuracies were 83.87% and

**Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.**
85.21%, and SOM clustering accuracies were 87.54% and 89.05%, respectively. However, the clustering accuracies of DRBM-ClustNet for before the flood and after the flood were 97.3% and 98%, respectively. Overall, the DRBM-ClustNet extracted images show a better performance than the other considered approaches. Therefore, DRBM-ClustNet can be further explored for other practical image clustering tasks.

V. CONCLUSION

In this article, DRBM-ClustNet is proposed, a Bayesian deep restricted Boltzmann–Kohonen network based on clustering. DRBM-ClustNet uses three stages to maximize the clustering efficiency, particularly for nonlinearly separable datasets. The first stage uses the DRBM, a generative model built by stacking multiple layers of RBM for feature learning. The second stage uses BIC to calculate the number of actual clusters in the dataset. The final stage uses the KN to cluster the feature-extracted data from DRBM using BIC predicted clusters. The parameters of the DRBM-ClustNet are set by the grid search approach, i.e., to determine the number of hidden layers and the number of hidden neurons to obtain a better clustering accuracy.

The data clustering task was successfully accomplished with DRBM-ClustNet on two synthetic datasets, 15 UCI repository benchmark data, and four image datasets. The results demonstrate that the proposed DRBM-ClustNet approach outperforms the other clustering techniques used in experiments. DRBM-ClustNet is also able to accurately predict the number of clusters for various datasets, giving it a further edge over other techniques. The obtained results show that DRBM-ClustNet is a very consistent, efficient, and reliable algorithm for the clustering of nonlinearly separable datasets.

APPENDIX

Recall, in the DRBM-ClustNet framework, stage 1 consists of DRBM; here, the joint probability of the visible and hidden units is given by

$$P(v, h) = \frac{1}{Z} e^{-E(v, h)}$$

(25)

where $Z$ is called the partition function, which is given by

$$Z = \sum_v \sum_h e^{-E(v, h)}.$$  

(26)

The partition function $Z$ is intractable, and the joint probability $P(v, h)$ is also intractable. Hence, we derive the conditional probability distribution $P(h|v)$ and $P(v|h)$ from the joint probability distribution $P(v, h)$, which is easy to compute and sample from.

Equation (24) can be expressed in the matrix form as

$$E(v, h) = -b^T v - c^T h - v^T W h$$

(27)

$$P(v, h) = \frac{1}{Z} e^{(b^T v + c^T h + v^T W h)}$$

(28)

$$P(h|v) = \frac{P(h, v)}{P(v)}$$

(29)

$$P(h|v) = \frac{1}{P(v)} \frac{1}{Z} e^{(b^T v + c^T h + v^T W h)}.$$  

(30)

The derivative $Z'$ using (28) can be expressed in terms of $Z$ as

$$Z' = \frac{e^{(b^T v)}}{P(v)} Z.$$  

(31)

Substituting (31) into (30), we get

$$P(h|v) = \frac{1}{Z} \prod_{j=1}^{p} e^{(c_j + v^T W_{ij} h_j)}.$$  

(32)

Using the Bayes theorem

$$P(h_j = 1|v) = \frac{P(h_j = 1, v)}{P(h_j = 0, v) + P(h_j = 1, v)}.$$  

(33)

Substitute $h_j = 1$ into (32) to get

$$P(h_j = 1|v) = \text{sigmoid}(c_j + v^T W_{ij}).$$  

(34)

Similarly, for $P(v|h)$, we obtain

$$P(v_i = 1|h) = \text{sigmoid}(b_i + W_{ij} h).$$  

(35)

Therefore,

$$P(h|v) = \prod_{j=1}^{p} \text{sigmoid}(c_j + v^T W_{ij})$$  

(36)
and
\[
P(v|h) = \prod_{i=1}^{d} \text{sigmoid}(b_i + W_{ij}h), \quad (37)
\]

Fig. 5(a) and (b) shows the cluster separation for the Wine dataset before applying DRBM and after applying DRBM, respectively, using the conditional probability distributions as mentioned above.

We can observe that, in Fig. 5(a), the clusters are overlapping before applying DRBM, and the clusters are not so obvious, making it difficult for BIC to predict the number of clusters and also for SOM to calculate the cluster centroids. In Fig. 5(b), the clusters are well separated, which makes it easy for BIC and SOM to predict the number of clusters and calculate the cluster centroids, respectively. The same behavior is also observed for the other datasets used for comparing the clustering accuracy.

BIC uses the maximum likelihood parameter to determine the number of clusters, which can also be expressed in terms of mean square error (mse) as follows:
\[
\text{BIC} = \ln(N)k - 2\ln(\hat{L}) \quad (38)
\]
where \(\hat{L}\) is the maximum likelihood function.

Assume that model errors are independent and identically distributed according to the normal distribution theory
\[
h_{\text{ML}} = \frac{1}{p} \prod_{i=1}^{p} \sum_{j=1}^{p} \text{exp} \left( -\frac{(x_i - \mu)^2}{2\sigma^2} \right) \quad (39)
\]
where \(h_{\text{ML}}\) is the maximum likelihood hypothesis, \(p\) is the number of hidden layers, and \(d_i\) is the output from DRBM
\[
h_{\text{ML}} = \frac{1}{p} \prod_{i=1}^{p} \frac{e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}}{2\pi \sigma^2} \quad (40)
\]
where \(d_i = f(x_i) + e_i\), \(\mu\) and \(\sigma\) are the mean and variance of normal distribution, \(f(x_i)\) is a function of \(x_i\), and \(e_i\) is random variable representing noise. Hence,
\[
h_{\text{ML}} = \frac{1}{p} \prod_{i=1}^{p} \text{exp} \left( -\frac{(x_i - \mu)^2}{2\sigma^2} \right) \quad (41)
\]
\[
\log(h_{\text{ML}}) = \sum_{i=1}^{p} \ln \left( \frac{1}{\sqrt{2\pi \sigma^2}} \right) - \frac{(d_i - f(x_i))^2}{2\sigma^2} \quad (42)
\]
The first term in the above equation is independent of the function \(f\), and it can be discarded; therefore,
\[
\log(h_{\text{ML}}) = \sum_{i=1}^{p} -\frac{(d_i - f(x_i))^2}{2\sigma^2} \quad (43)
\]
\[
-\log(h_{\text{ML}}) = -\sum_{i=1}^{p} \frac{(d_i - f(x_i))^2}{2\sigma^2} = \frac{\text{RSS}}{n} = \text{mse}. \quad (44)
\]

As BIC is expressed in terms of mse as described above and as DRBM makes the cluster separation linearly separable, as observed in Fig. 5(b), it becomes easy for BIC to predict the number of clusters. SOM uses the Gaussian neighborhood function to find the nearest neighbors and uses the Euclidean distance to find the winning neuron. Hence, SOM works well for datasets that are more linearly separable and for datasets where the clusters are nonoverlapping. Hence, applying a DRBM transformation on the existing dataset facilitates SOM to calculate the cluster centroids more accurately, which, in turn, increases SOM clustering efficiency.

REFERENCES

[1] D. Xu and Y. A. Tian “A comprehensive survey of clustering algorithms,” Ann. Data Sci., vol. 2, no. 2, pp. 165–193, Jun. 2015.
[2] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” ACM Comput. Surv., vol. 31, no. 3, pp. 264–323, Sep. 1999.
[3] J. Chang, G. Meng, L. Wang, S. Xiang, and C. Pan, “Deep self-evolution clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 4, pp. 809–823, Apr. 2020.
[4] P. Pei, C. Chen, and W. Gong, “Concept factorization with adaptive neighbors for document clustering,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 2, pp. 343–352, Feb. 2018.
[5] M. Ahmed, A. N. Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” J. Netw. Comput. Appl., vol. 60, pp. 19–31, Jan. 2016.
[6] L. L. Xv, D. Zhao, and Q. Deng, “Image clustering based on deep sparse representations,” in Proc. IEEE Symp. Comput. Intell. (SSCI), Dec. 2016, pp. 1–6.
[7] K. Alfalahi, Y. Atif, and S. Harous, “Community detection in social networks through similarity virtual networks,” in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining, Aug. 2015, pp. 1116–1123.
[8] W. Yu, G. Qiang, and L. Xiao-Li, “A kernel aggregate clustering approach for mixed data set and its application in customer segmentation,” in Proc. Int. Conf. Manage. Sci. Eng., 2006, pp. 121–124.
[9] S. Jayavelu, H. Kandath, and S. Sundaram, “Dynamic area coverage for multi-UAV using distributed UGVs: A two-stage density estimation approach,” in Proc. 2nd IEEE Int. Conf. Robotic Comput. (IRC), Jan. 2018, pp. 165–166.
[10] M. Yin, J. Guo, S. Xie, and Y. Guo, “Multiview subspace clustering via tensorial T-product representation,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 3, pp. 851–864, Mar. 2019.
[11] S. Al-Doobani and D. Wunsch, “Model order reduction based on agglomerative hierarchical clustering,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 6, pp. 657–669, Jun. 2019.
[12] J. Senthilnath, K. Harikumar, and S. Suresh, “Efficient resource allocation for decentralized heterogeneous system using density estimation approach,” Encyclopedia Semantic Comput. Robotic Intell., vol. 2, no. 1, Jun. 2018, Art. no. 1850008.
[13] R. Xu and D. C. Wunsch, “Survey of clustering algorithms,” IEEE Trans. Neural Netw., vol. 16, no. 3, pp. 645–678, May 2005.
[14] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient K-means clustering algorithm: Analysis and implementation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 881–892, Jul. 2002.
[15] T. Kohonen, “Essentials of the self-organizing map,” Neural Netw., vol. 37, pp. 52–65, Jan. 2013.
[16] X. Li, K. Zhang, and T. Jiang, “The regularized EM algorithm,” in Proc. Nat. Conf. Artif. Intell., vol. 2, 2005, pp. 807–812.
[17] J. C. Fort, P. Letremy, and M. Cottrell, “Advantages and drawbacks of the batch Kohonen algorithm,” in Proc. ESANN, vol. 2, 2002, pp. 223–230.
[18] P. Mangiameli, S. K. Chen, and D. West, “A comparison of SOM neural network and hierarchical clustering methods,” Eur. J. Oper. Res., vol. 93, no. 2, pp. 402–417, Sep. 1996.
[19] K. P. Burnham and D. R. Anderson, “Multimodel inference: Understanding AIC and BIC in model selection,” Sociol. Methods Res., vol. 33, no. 2, pp. 261–304, 2004.
[20] G. E. Hinton, “A practical guide to training restricted Boltzmann machines,” in Neural Networks: Tricks Trade. Berlin, Germany: Springer, 2012, pp. 599–619.
[21] R. Salakhutdinov and G. Hinton, “Deep Boltzmann machines,” in Proc. 12th Int. Conf. Artif. Intell. Statist., vol. 5, PMLR, 2009, pp. 448–455.
[22] A. Fischer and C. Igel, “Training restricted Boltzmann machines: An introduction,” Pattern Recognit., vol. 47, no. 1, pp. 25–39, Jan. 2014.
[23] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” Chemometrics Intell. Lab. Syst., vol. 2, nos. 1–3, pp. 37–52, Aug. 1987.
[24] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, “A learning algorithm for Boltzmann machines,” Cognit. Sci., vol. 9, pp. 147–169, Jan. 1985.

[25] P. Smolensky, “Information processing in dynamical systems: Foundations of harmony theory,” in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Cambridge, MA, USA: MIT Press, 1986, pp. 194–281.

[26] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” Proc. Nat. Acad. Sci. USA, vol. 79, no. 8, pp. 2554–2558, 1982.

[27] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural Comput., vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[28] G. E. Hinton, “Deep belief networks,” Scholarpedia, vol. 4, no. 5, p. 5947, May 2009.

[29] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural Comput., vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[30] Y. Bengio, “Learning deep architectures for AI,” Found. Trends Mach. Learn., vol. 2, no. 1, pp. 1–127, Jan. 2009.

[31] S. Bu, L. Wang, P. Han, Z. Liu, and K. Li, “3D shape recognition and retrieval based on multi-modality deep learning,” Neurocomputing, vol. 259, pp. 183–193, Oct. 2017.

[32] N. Zhang, S. Ding, J. Zhang, and Y. Xue, “An overview on Restricted Boltzmann Machines,” Neurocomputing, vol. 275, pp. 1186–1199, Jan. 2018.

[33] R. Salakhutdinov and G. Hinton, “An efficient learning procedure for deep Boltzmann machines,” Neural Comput., vol. 24, no. 8, pp. 1967–2006, Aug. 2012.

[34] L. Gu, J. Huang, and L. Yang, “On the representational power of restricted Boltzmann machines for symmetric functions and Boolean functions,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 5, pp. 1335–1347, May 2019.

[35] L.-W. Kim, “DeepX: Deep learning accelerator for restricted Boltzmann machine artificial neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 5, pp. 1441–1453, May 2018.

[36] W. Yi, J. Park, and J.-J. Kim, “GeoC: Classification restricted Boltzmann machine hardware for on-chip semisupervised learning and Bayesian inference,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 1, pp. 53–65, Jan. 2020.

[37] A. Sankaran, G. Goswami, M. Vatsa, R. Singh, and A. Majumdar, “Class sparsity signature based restricted Boltzmann machine,” Pattern Recognit., vol. 61, pp. 674–685, Jan. 2017.

[38] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang, and Q. Sun, “Deep learning for image-based cancer detection and diagnosis—a survey,” Pattern Recognit., vol. 83, pp. 134–149, Nov. 2018.

[39] J. Li, Z. L. Yu, Z. Gu, M. Tan, Y. Wang, and Y. Li, “Spatial–temporal discriminative restricted Boltzmann machine for event-related potential detection and analysis,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 27, no. 2, pp. 139–151, Feb. 2019.

[40] G. Wang, J. Qiao, J. Bi, Q. S. Jia, and M. C. Zhou, “An adaptive deep belief network with sparse restricted Boltzmann machines,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 10, pp. 1–12, Dec. 2019.

[41] D. C. Mocanu, H. B. Ammar, L. Puig, E. Eaton, and A. Liotta, “Estimating 3D trajectories from 2D projections via disjunctive factored four-way conditional restricted Boltzmann machines,” Pattern Recognit., vol. 69, pp. 325–335, Sep. 2017.

[42] A. K. Jain, “Data clustering: 50 years beyond K-means,” Pattern Recognit. Lett., vol. 31, no. 8, pp. 651–666, Jun. 2010.

[43] T. Kohonen, “The self-organizing map,” Neurocomputing, vol. 21, nos. 1–3, pp. 1–6, Nov. 1998.

[44] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” J. Roy. Statist. Soc., B, Methodol., vol. 39, no. 1, pp. 1–38, 1977.

[45] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao, “Real-time superpixel segmentation by DBSCAN clustering algorithm,” IEEE Trans. Image Process., vol. 25, no. 12, pp. 5933–5942, Dec. 2016.

[46] 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.

[47] J. Senthilnath, C. Simha, G. Nagaraj, M. Thapa, and M. Indiramma, “BELMKN: Bayesian extreme learning machines Kohonen network,” Algorithms, vol. 11, no. 5, pp. 56, Apr. 2018.

[48] D. Dua and C. Graff, “UCI machine learning repository,” School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2019. [Online]. Available: http://archive.ics.uci.edu/ml

[49] G. E. Cook, R. J. Barnett, K. Andersen, and A. M. Strauss, “Weld modeling and control using artificial neural networks,” IEEE Trans. Ind. Appl., vol. 31, no. 6, pp. 1484–1491, Nov. 1995.

[50] V. Mani and S. N. Omkar, “Understanding weld modelling processes using a combination of trained neural networks,” Int. J. Prod. Res., vol. 40, no. 3, pp. 547–559, Jan. 2002.

[51] X. Guo, H. Huang, and J. Zhang, “Comparison of different variants of restricted Boltzmann machines,” in Proc. 2nd Int. Conf. Inf. Technol. Electron. Commerce, Dec. 2014, pp. 239–242.

[52] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, “An empirical evaluation of deep architectures on problems with many factors of variation,” in Proc. 24th Int. Conf. Mach. Learn. (ICML), 2007, pp. 473–480.

[53] A. Insolberg and T. Avidan, “Classification and visualization for high-dimensional data,” in Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2000, pp. 370–374.

[54] D. Chen, J. Lv, and Z. Yi, “Graph regularized restricted Boltzmann machine,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 6, pp. 2651–2659, Jun. 2018.

[55] D. Cai, X. He, and J. Han, “Document clustering using locality preserving indexing,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

[56] J. Senthilnath, S. Bajpai, S. N. Omkar, P. G. Diwakar, and V. Mani, “An approach to multi-temporal MODIS image analysis using image classification and segmentation,” Adv. Space Res., vol. 50, no. 9, pp. 1274–1287, 2012.

J. Senthilnath (Senior Member, IEEE) received the Ph.D. degree in aerospace engineering from the Indian Institute of Science (IISc), Bengaluru, India, in 2014.

He is currently a Scientist with the Institute for Infocomm Research (I-FR), Agency for Science, Technology and Research (A*STAR), Singapore. He is also a member of Artificial Intelligence, Analytics and Informatics (AI3), A*STAR. He has published over 100 high-quality papers. His current research interests include artificial intelligence, multigait systems, generative models, online learning, and reinforcement learning and optimization.

Dr. Senthilnath won five best paper awards. He has been serving as a guest editor/organizing chair/co-chair/PC member in leading AI and data analytics journals and conferences.

Nagaraj G. received the B.E. degree in computer science and engineering from the B.M.S. College of Engineering, Bengaluru, India, in 2019.

He has three years of work experience in software development. His areas of interest are pattern recognition and natural language processing.

Sumanth Simha C. received the B.E. degree in computer science and engineering from the B.M.S. College of Engineering, Bengaluru, India, in 2019.

He has three years of work experience in software development. His areas of interest are deep learning, image processing, and convex optimization.
Sushant Kulkarni received the M.Tech. degree from IIT Madras, Chennai, India, in 2016. He has a work experience of seven years in data science. His research interests are machine learning, natural language processing (NLP), and optimization.

Meenakumari Thapa received the M.Tech. degree in computer science from the Birla Institute of Technology and Science (BITS), Pilani, India, in 2016. She has work experience of more than five years in software development. Her research interests include data warehousing and data mining.

Indiramma M. is currently a Professor with the Computer Science and Engineering Department and also heading the Industry Institute Interaction Cell (IIIC), B.M.S. College of Engineering, Bengaluru, India. She is in academics for the past 30 years. She is the author/coauthor of over 60 technical papers in refereed international conferences and journals. Her research interests and area are big data analytics, artificial intelligence, machine learning, and deep learning. Dr. M. has chaired the technical program committee for several IEEE conferences. She is also serving as a reviewer, a program committee member, and the session chair of several international conferences. She has served as a reviewer for several reputed international journals. She has been invited as a resource person/expert for many faculty development programs and workshops.

Jón Atli Benediktsson (Fellow, IEEE) received the Cand.Sc. degree in electrical engineering from the University of Iceland, Reykjavik, Iceland, in 1984, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1987 and 1990, respectively. From 2009 to 2015, he was the Pro-Rector of science and academic affairs and a Professor of electrical and computer engineering with the University of Iceland. Since July 2015, he has been the President and a Rector with the University of Iceland. His research interests are in remote sensing, pattern recognition, and image analysis.

Prof. Benediktsson is also a fellow of Society of Photo-Optical Instrumentation Engineers (SPIE). He was a Highly Cited Researcher (Clarivate Analysis) from 2018 to 2021. He was the 2011–2012 President of the IEEE Geoscience and Remote Sensing Society (GRSS). He was the Editor-in-Chief of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS) from 2003 to 2008. He has been serving as an Associate Editor for TGRS since 1999, the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS since 2003, and IEEE ACCESS since 2013. He is also a Senior Editor of the PROCEEDINGS OF THE IEEE. He is on the International Editorial Board of the International Journal of Image and Data Fusion and Remote Sensing.