DRBM-ClustNet: A Deep Restricted Boltzmann-Kohonen Architecture for Data Clustering

J. Senthilnath, Senior Member, IEEE, Nagaraj G, Sumanth Simha C, Sushant Kulkarni, Meenakumari Thapa, Indiramma M and Jón Atli Benediktsson, Fellow, IEEE

***Abstract*—A Bayesian Deep Restricted Boltzmann-Kohonen architecture for data clustering termed as DRBM-ClustNet is proposed. This core-clustering engine consists of a Deep Restricted Boltzmann Machine (DRBM) for processing unlabeled data by creating new features that are uncorrelated and have large variance with each other. Next, the number of clusters are predicted using the Bayesian Information Criterion (BIC), followed by a Kohonen Network-based clustering layer. The processing of unlabeled data is done in three stages for efficient clustering of the non-linearly separable datasets. In the first stage, DRBM performs non-linear 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 Kohonen network, which performs clustering of the feature-extracted data obtained from the DRBM. This method overcomes the general disadvantages of clustering algorithms like the prior specification of the number of clusters, convergence to local optima and poor clustering accuracy on non-linear datasets. In this research we use two synthetic datasets, fifteen 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*—Data Clustering, Bayesian Information Criterion, Deep Restricted Boltzmann Machine, Kohonen Network.**

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]. Most popular partitional clustering algorithms are K-means and the Kohonen Network. In the case of model-based clustering, the expectation maximization (EM) Algorithm is most popular [14] [15] [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 Kohonen Network, 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 like 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 non-linearly separable data into linearly separable data. In the literature, the Restricted Boltzmann Machine (RBM) has overcome such issues [20] [21] [22]. The RBM is a non-linear feature extraction technique that is superior in comparison to linear feature extraction techniques such as Principal Component Analysis [23].

The first Boltzmann Machine (BM) model was developed in the 1980s and was inspired by statistical mechanics [24]. Smolensky in 1986 [25] implemented the Harmonium, which

Manuscript received August 09, 2020.

J. Senthilnath is with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), 138632 Singapore (e-mail: J_Senthilnath@i2r.a−star.edu.sg).

Nagaraj G is with the B.M.S. College of Engineering, Bengaluru 560019, India (e-mail: nagaraj170297@gmail.com).

Sumanth Simha C is with the B.M.S. College of Engineering, Bengaluru 560019, India (e-mail: ssimha152@gmail.com).

Sushant Kulkarni is with the Indian Institute of Technology, Chennai 600036, India (e-mail: ksushantk@gmail.com).

Meenakumari Thapa is with the Birla Institute of Technology and Science, Pilani 333031, India (e-mail: meenathapa31@gmail.com).

Indiramma M is with the B.M.S. College of Engineering, Bengaluru 560019, India (e-mail: indira.cse@bmsce.ac.in).

Jón Atli Benediktsson is with the Faculty of Electrical and Computer Engineering, University of Iceland, 107 Reykjavik, Iceland (e-mail: benedikt@hi.is).

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 implemented a modified version of BM, termed as RBM, for a fast and efficient way of learning [27]. 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, Hinton et al., [33] developed Deep Boltzmann Machines 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 Restricted Boltzmann Machine (DRBM) has been used as a generative model to learn the non-linear 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 semi-supervised 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 3D trajectories from 2D trajectories [41].

In this paper, 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 Kohonen Network. The proposed framework's performance is compared with six state-of-the-art clustering methods, namely, K-means [42], Self-Organizing Maps (SOM) [43], Expectation Maximization (EM) [44], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [45], Unsupervised Extreme Learning Machine (US-ELM) [46], and Bayesian Extreme Learning Machines Kohonen Network (BELMKN) [47]. The proposed DRBM-ClustNet and the state-of-the-art clustering algorithms are applied on 2 synthetic datasets, 15 benchmark datasets from the UCI Machine Learning Repository [48], and 4 image datasets. The performance evaluation is carried out using various statistical approaches.

The remaining content of the paper 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 2 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 4 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 which 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 Kohonen Network. 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 ($\boldsymbol{L_1}$) 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], [50]

$$F(\mathbf{x}_s) = \frac{0.9 - 0.1}{max(\mathbf{x}_i) - min(\mathbf{x}_i)}(\mathbf{x}_s - min(\mathbf{x}_i)) + 0.1, \quad (1)$$

where $\mathbf{x}_s$ is a dataset with $s = \{1, 2, ..., N\}$, $min(\mathbf{x}_i)$ and $max(\mathbf{x}_i)$ corresponds to the minimum and maximum value for a particular feature with $i = \{1, 2, ..., d\}$.

The normalized dataset is passed to $\boldsymbol{L_1}$ of the DRBM network but $\boldsymbol{L_1}$ consists of a visible layer and a hidden layer. The visible units v_i, $i \in 1$ to d are real-valued and the hidden units are given by h_j, $j \in 1$ to p . Let $W^1 \in \mathbb{R}^{d \times p}$ denote the weights between the hidden layer and visible layer units in $\boldsymbol{L_1}$ and w_{ij} indicate 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 are 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

$$P(v, h) = \frac{1}{Z} e^{-E(v,h)}, \quad (2)$$

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

$$E(v, h) = -\sum_{i=1}^{d}\sum_{j=1}^{p} w_{ij} v_i h_j - \sum_{i=1}^{d} b_i v_i - \sum_{j=1}^{p} c_j h_j, \quad (3)$$

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

$$Z = \sum_{i=1}^{d}\sum_{j=1}^{p} e^{-E(v_i, h_j)}. \quad (4)$$

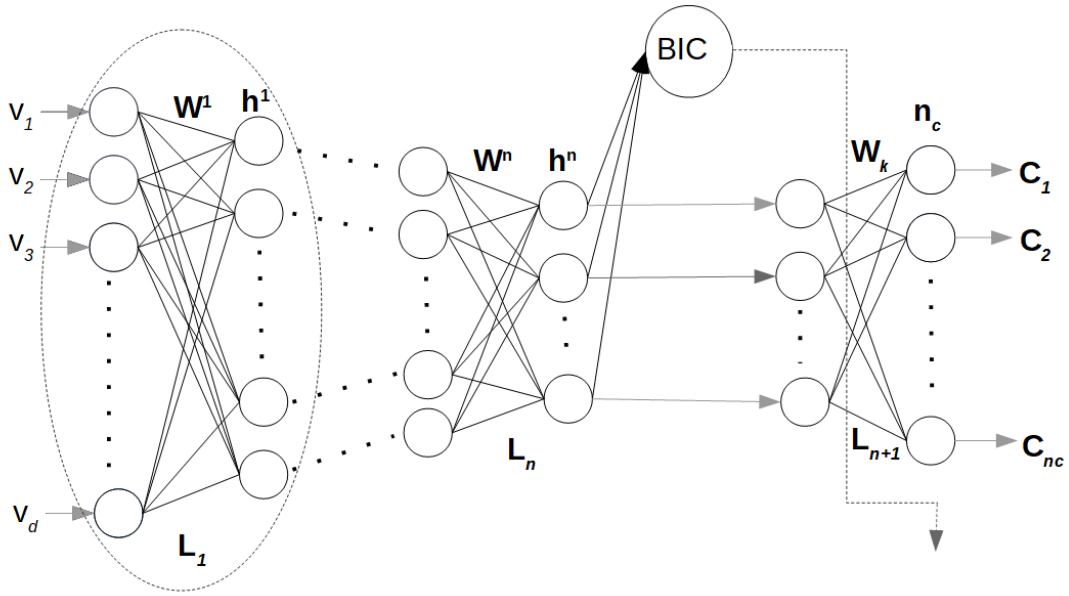

Fig. 1: DRBM-ClustNet architecture

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

$$P(v|h) = \prod_{i=1}^{d} P(v_i|h), \tag{5}$$

$$P(h|v) = \prod_{j=1}^{p} P(h_j|v). \tag{6}$$

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 Boltzmann Machine is defined as a Restricted Boltzmann Machine (RBM). The conditional probabilities from (5) and (6) are derived as a sigmoidal function given by

$$P(h_j|v) = s(\sum_{i=1}^{d} v_i w_{ij} + c_j), \tag{7}$$

$$P(v_i|h) = s(\sum_{j=1}^{p} h_j w_{ij} + b_i), \tag{8}$$

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

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 proposed the k-step Contrastive Divergence algorithm (CD) [29] 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 contrastive divergence and negative contrastive divergence followed by weight updating.

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

$$P(h|v) = s(v_{<0>}W + c) = h_{<0>}. \tag{9}$$

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

$$a^+ = v_{<0>}h_{<0>} = (vh)_{<0>}, \tag{10}$$

where a^+ denotes the positive associations.

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

$$P(v|h) = s(Wh_{<0>} + b) = v_{<1>}, \tag{11}$$

$$P(h|v) = s(Wv_{<1>} + c) = h_{<1>}. \tag{12}$$

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

$$a^- = v_{<1>}h_{<1>} = (vh)_{<1>}. \tag{13}$$

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 (ϵ)

$$\Delta w = \epsilon(a^+ - a^-). \tag{14}$$

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 ($\boldsymbol{L_1}$) RBM will not be able to capture the complete non-linearity in the dataset. As a result, we need to stack multiple RBM layers ($\boldsymbol{L_1}$, $\boldsymbol{L_2}$,..., $\boldsymbol{L_n}$) 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 the complex data representations. DRBM can efficiently learn a generative model of 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 ($\boldsymbol{L_1}$, $\boldsymbol{L_2}$,...,$\boldsymbol{L_n}$). Let h^l ($l \in 1$ to n) denote the l^{th} hidden layer, W^l denote the weights between the layers. The energy equation for the DRBM is given by

$$E(v, h^1, .., h^n; \theta) = -vW^1h^1 - h^1W^2h^2 - .. - h^{n-1}W^nh^n, \tag{15}$$

where $\theta = \{W^1, W^2, ..., W^n\}$ are the model parameters which 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 one-step contrastive divergence to obtain the reconstructions of the visible vectors. Then, the next layer RBM is trained using the first layer sampled hidden layer activations h^l obtained from $P(h^l|v; W^l)$ 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 $\theta = \{W^1, W^2, ..., W^n\}$ 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 ($\tilde{\mathbf{x}}_s^l$).

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, which are the mean and covariance matrices. These parameters are estimated using the EM algorithm. BIC for cluster prediction is defined as

$$BIC = ln(N)k - 2ln(\tilde{L}), \tag{16}$$

where N is the total number of samples, $\tilde{L}$ 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 $c = 1, 2, ..., N$ and the model value with minimum BIC value is selected and its corresponding number of clusters n_c are 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 Kohonen Network. The Kohonen Network (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 $W_k \in R^{n_i \times n_c}$, 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 Euclidean distance given by

$$d(j) = \sqrt{\sum_{i=1}^{n_i}(\tilde{x}_i - w_{ij})} \qquad j \in 1 \; to \; n_o. \tag{17}$$

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

$$h_{ci}(t) = \alpha(t) exp\left(\frac{-d_{ij}^2}{2\sigma(t)}\right), \tag{18}$$

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

$$\alpha(t) = \alpha_0 exp\left(\frac{-t}{T_1}\right), \tag{19}$$

$$\sigma(t) = \sigma_0 exp\left(\frac{-t}{T_2}\right), \tag{20}$$

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 are computed as follows:

$$\delta w_{ij} = h_{ci}(t)(y_i - w_{ij}). \tag{21}$$

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

$$\eta = \frac{\sum_{i=1}^{n_c} a_l}{N}, \tag{22}$$

where a_l 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 like 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 150 samples. The flame distribution and the moon distribution datasets

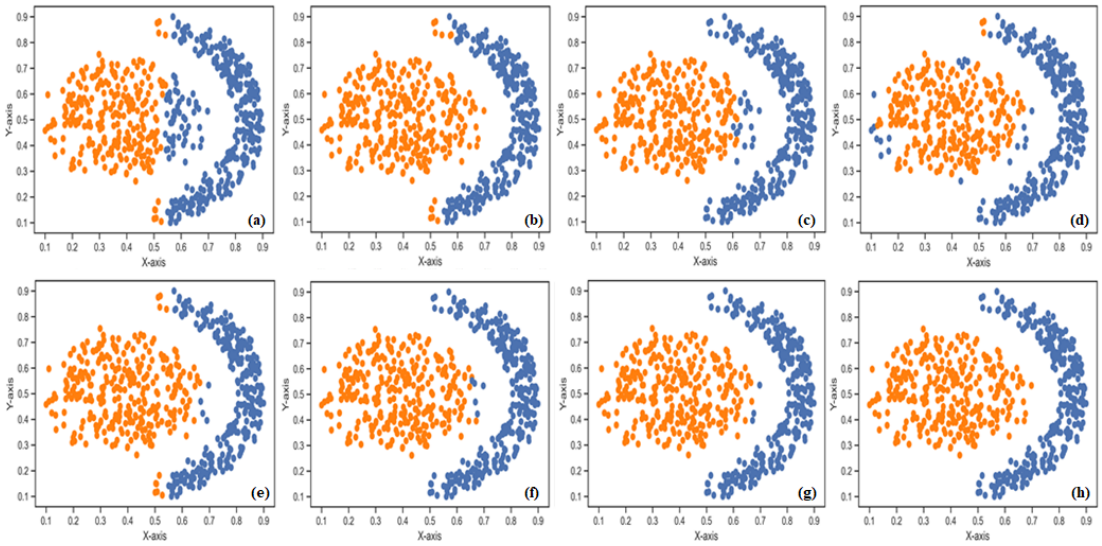

Fig. 2: Clustering of flame pattern distribution using (a) k-means; (b) SOM; (c) EM; (d) DBSCAN; (e) US-ELM; (f) BELMKN; (g) Single Layer RBM; and (h) DRBM-ClustNet

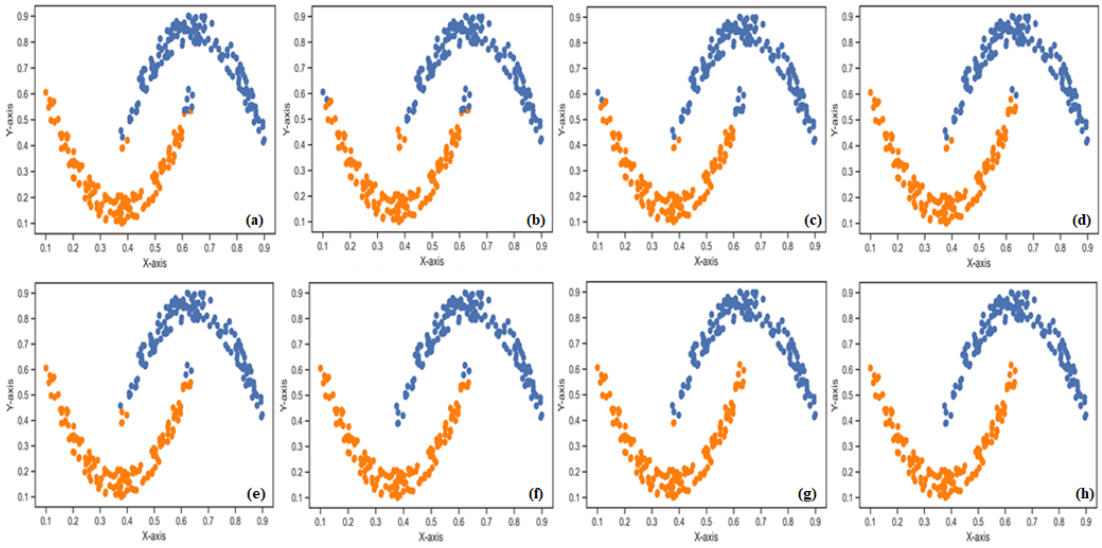

Fig. 3: Clustering of moon pattern distribution using (a) k-means; (b) SOM; (c) EM; (d) DBSCAN; (e) US-ELM; (f) BELMKN; (g) Single Layer RBM; and (h) DRBM-ClustNet

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 Kohonen Network 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 non-linearity 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 paper.

Similarly, we can observe from Figs. 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 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 which uses DRBM 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 efficient

Algorithm 1: DRBM-ClustNet Algorithm

Input: clustering dataset $\mathbf{X}=\{\mathbf{x}_s \in \mathbb{R}^d\}_{s=1}^{N}$, DRBM-ClustNet $(v_1, ..., v_d, h_1, ..., h_p)$, $\theta_1=\{W^l\}_{l=1}^{n}$, $\theta_2=\{W_K\}$, learning rate (ϵ, α).
Output: cluster prediction (n_c) and accuracy (η).
for x_s *in* X **do**
  $\mathbf{v}_i^s <- \mathbf{x}_s$;
  for $l = 1$ *to* n **do**
    Compute CD using (9)-(14) to obtain W^l;
    Perform feedforward pass using W^l;
    Extract features $\tilde{\mathbf{x}}_s^l$ from layer $\mathbf{h}^l$;
  Using extracted feature $\tilde{\mathbf{x}}_s^l$ predict n_c (16);
  Perform clustering with n_c on $\tilde{\mathbf{x}}_s^l$ using (17)-(21);
  Evaluate η using (22);
return: n_c, η.

clustering of non-linearly distributed datasets.

IV. Results and Discussion

This section discusses the results for the DRBM-ClustNet framework on fifteen different benchmark datasets from the UCI repository. Initially, we discuss the characteristics of the datasets shown in Table I. Next, the number of clusters are 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 is used for comparison against the BIC evaluated on the actual dataset. Finally, clustering performance of the DRBM-ClustNet algorithm is compared with other prominent clustering algorithms such as K-means [42], SOM [43], EM [44], DBSCAN [45], USELM [46] and BELMKN [47]. The algorithms were executed on a Core i3 processor, 4 GB RAM, Python 2.7 and Windows 10 OS.

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 is shown in Table I. The detailed description for the datasets are as follows:

Dataset 1: The Balance data classifies 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, the right distance, and 625 samples.

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

Dataset 3: The Cancer-Int data classifies a breast tumor either as malignant or benign. It has 699 observations and 9 attributes.

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

Dataset 5: The Dermatology data has different diagnosis details for the erythemato-squamous diseases in dermatology. It has 6 classes, 34 attributes and 366 observations.

Dataset 6: The Diabetes data is to diagnostically predict whether a patient has diabetes based on certain diagnostic measurements. It has 768 observations and 8 attributes.

Dataset 7: The E.Coli data has details of the proteins cellular localization sites. It contains 5 classes, 327 observations and 7 attributes.

Dataset 8: The Glass data has details for the oxide content and refractive index for different glass types. It consists of 6 classes, 214 observations and 9 attributes.

Dataset 9: The Heart data consists of details for the diagnosis of heart diseases. It consists of 2 classes, 270 observations and 13 attributes.

Dataset 10: The Horse data has the details for the different types of horses. It consists of 3 classes, 364 observations and 26 attributes.

Dataset 11: The Iris data classifies a sample into one of the three types of flower classes, namely, setosa or virginica or versicolor. It has 150 observations and 4 attributes, namely, width of petals, length of petals, the width of sepals and length of the sepals.

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

Dataset 13: The Thyroid data consists of three classes, namely, normal-functional, under-functional and over-functional. There are 215 observations and 5 attributes.

Dataset 14: The Vehicle data consists 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 lists different types of chemical analysis of wine. It consists of 3 classes, 178 observations and 13 attributes.

TABLE I: Properties of the Datasets

Sl. No	Dataset name	Number of samples	Input Dimensions	Number of Cluster
1	Balance	625	4	3
2	Cancer	569	30	2
3	Cancer Int	699	9	2
4	Credit	690	14	2
5	Dermatology	366	34	6
6	Diabetes	768	8	2
7	E.Coli	327	7	5
8	Glass	214	9	6
9	Heart	270	13	2
10	Horse	364	26	3
11	Iris	150	4	3
12	SECOM	1567	590	2
13	Thyroid	215	5	3
14	Vehicle	846	18	4
15	Wine	178	13	3

B. Parameters for Algorithm

All the datasets used in the paper are pre-processed 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 towards high-value features, all the features were normalized between 0.1 to 0.9 using min-max scaling. This also ensures a faster convergence for neural network-based approaches. It is necessary to tune the hyper-parameters for effective learning. Since our hyper-parameters are in a well-defined range, the standard grid search approach [52] has been used to find an optimal set of hyper-parameters for the data clustering. A single RBM had one hidden layer with 50 hidden units. The deep RBM has five hidden layers with 50 hidden units in each of the first four layers and 10 hidden units in the final layer. The maximum epochs and the learning rate are set to 50000 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 above-mentioned dataset. The feature-extracted dataset is used to predict the number of clusters with BIC. Furthermore, the obtained results is 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 on 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 when 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 behaviour is observed in Cancer and Cancer-Int datasets as they are linearly separable. By applying feature learning on these datasets, we are introducing additional non-linearity. As a result, feature learning techniques like 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 non-linear, hence, we get the actual number of clusters by applying the feature learning techniques on these two datasets. For the Balance, Horse and Vehicle datasets, the USELM, BELMKN 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 non-linear and the non-linearity 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 non-linearly distributed data.

D. Data visualization for DRBM

To analyze the behaviour 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 10 hidden neurons in the last layer. The Iris dataset has four features and the DRBM feature learning, obtained in the last layer, contains 10 dimensions. Similarly, the Wine dataset has 13 features reduced to 10 features using 10 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 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 Figs. 5(a) and 5(b) that there is a significant difference between the two scatter plots. Because of the non-linear dimensionality reduction from 13 to 10 dimensions in the DRBM feature extraction, linear separability is introduced to the dataset. Hence, by applying DRBM feature extraction, the number of clusters is well-predicted using BIC as shown in Table II.

Most machine learning algorithms involving sigmoidal activation functions or logistic regression functions, often exhibit poor performance if they have highly correlated input variables or feature vectors in the data. Fig. 6(a) shows the correlation matrix for the actual Iris dataset. From this figure, we can observe that the percentage of feature vector pairs having a high correlation (in the range 0.75-1.0) is 6 out of 12 i.e. 50%. In the case of DRBM feature learning, as shown in Fig. 6(b), we can observe that the percentage of attribute pairs having a high correlation (0.75-1.0) is 2 out of 12, i.e., 16.67%. 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 10 out of the 15 datasets, i.e., in 66.6% of the cases, whereas BIC applied on the actual dataset performs well for 7 out of 15 datasets, i.e., for 46.6% of the datasets. Both ELM and the single layer RBM perform well for 7 out of 15 datasets, i.e., in 46.6% of the datasets. Another observation from Table II is that DRBM with BIC predicts cluster consistently as the standard deviation is almost zero for most of the datasets whereas a large variation is

TABLE II: Cluster Prediction using BIC for 15 datasets

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	$\mathbf{3} \pm 0$
Cancer	2	3 ± 0.47	3 ± 0.26	3 ± 0	3 ± 0
Cancerint	2	$\mathbf{2} \pm 0.63$	3 ± 0.37	3 ± 0	3 ± 0
Credit	2	$\mathbf{2} \pm 0.39$	5 ± 0.31	4 ± 0	6 ± 0
Dermatology	6	5 ± 0.99	$\mathbf{6} \pm 0.87$	$\mathbf{6} \pm 0.44$	$\mathbf{6} \pm 0.39$
Diabetes	2	$\mathbf{2} \pm 0.93$	3 ± 0.79	5 ± 0.31	5 ± 0.22
E.Coli	5	4 ± 0.67	$\mathbf{5} \pm 0.61$	$\mathbf{5} \pm 0$	$\mathbf{5} \pm 0$
Glass	6	3 ± 0.81	$\mathbf{6} \pm 0.73$	$\mathbf{6} \pm 0$	$\mathbf{6} \pm 0$
Heart	2	$\mathbf{2} \pm 1.07$	3 ± 1.18	4 ± 0.94	4 ± 0.81
Horse	3	2 ± 0.75	2 ± 0.88	2 ± 0.18	$\mathbf{3} \pm 0.18$
Iris	3	$\mathbf{3} \pm 0.31$	$\mathbf{3} \pm 0.02$	$\mathbf{3} \pm 0$	$\mathbf{3} \pm 0$
SECOM	2	3 ± 0.15	$\mathbf{2} \pm 0.54$	$\mathbf{2} \pm 0.44$	$\mathbf{2} \pm 0.22$
Thyroid	3	$\mathbf{3} \pm 0.39$	$\mathbf{3} \pm 0.31$	$\mathbf{3} \pm 0$	$\mathbf{3} \pm 0$
Vehicle	4	3 ± 1.25	3 ± 1.10	3 ± 0.98	$\mathbf{4} \pm 0.65$
Wine	3	$\mathbf{3} \pm 0.49$	$\mathbf{3} \pm 0.23$	$\mathbf{3} \pm 0$	$\mathbf{3} \pm 0$

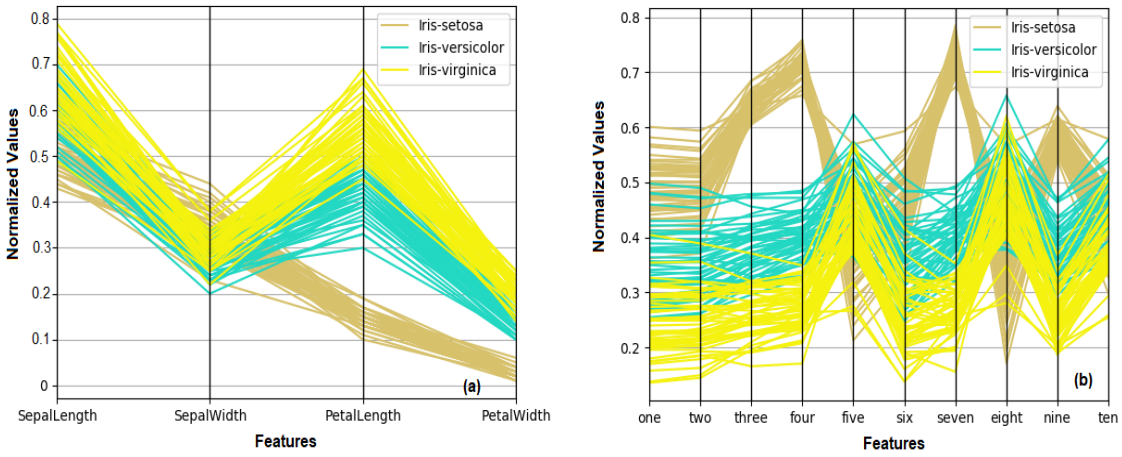

Fig. 4: Parallel coordinates plot for (a) Actual Iris dataset, (b) Iris dataset after DRBM feature extraction

present for other techniques. Hence, DRBM works effectively and efficiently for the non-linearly 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 Kohonen Network. This approach shows the best performance when compared to the traditional clustering algorithms such as K-means, EM, SOM, 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 towards 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 neighbourhood concept and iteratively computes the cluster centers. The main drawback of SOM is ineffective in clustering non-linearly 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 Kohonen Network as the clustering algorithms, respectively. As BELMKN uses the Kohonen Network, this clustering algorithm overcomes the stated drawbacks of K-means. As both USELM and BELMKN use ELM for feature learning, they suffer from the 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

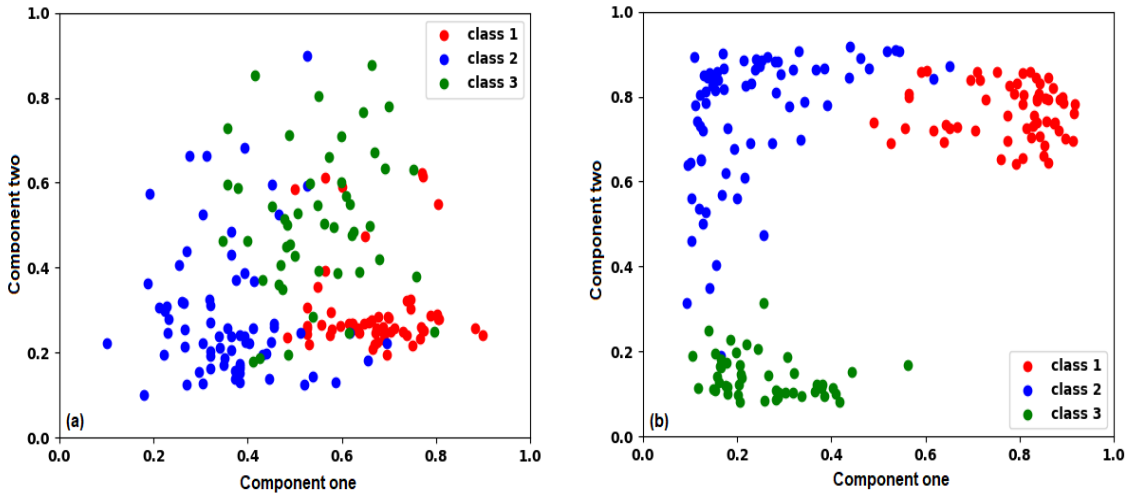

Fig. 5: Scatter plot for (a) Actual Wine dataset, (b) Wine dataset after DRBM feature extraction

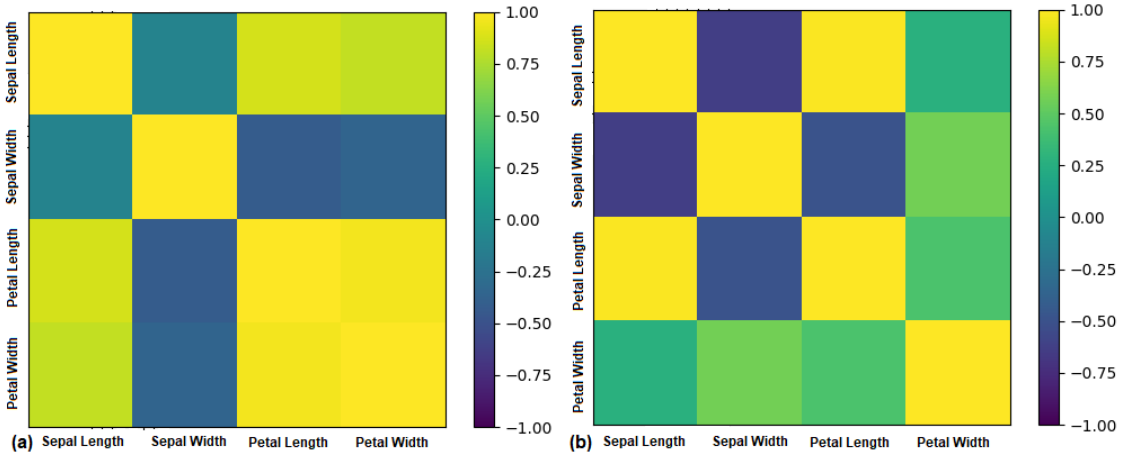

Fig. 6: Correlation matrix plot for (a) Original Iris dataset, (b) Iris dataset after RBM feature extraction

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 non-linear like Balance, Credit, Dermatology, Glass and Heart, the standard deviation is very high compared to the other datasets which indicate that the clustering results are not consistent for ELM. Hence, for USELM and BELMKN we need to run the program 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 like 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 modelled as bipartite graphical model to exploit the generative nature of data and to 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 contrastive divergence and the negative contrastive divergence, which is accomplished by Gibbs sampling [20]. Hence, the convergence to a global minimum is better guaranteed when compared to ELM-based feature learning. As a result, the clustering accuracy for the datasets using DRBM-ClustNet is higher when compared to USELM and BELMKN. In addition, there are higher accuracies obtained by DRBM-ClustNet for the non-linear datasets like Balance, Credit, Dermatology, Glass and Heart. As DRBM-ClustNet is less immune to the non-linearity in the dataset, it does not overfit the data.

The RBM network with one hidden layer is not sufficient to capture the non-linearity 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

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 (6)	52.5 ± 0.2 (5)	53.0 ± 0.6 (3)	45.9 ± 3.3 (8)	47.1 ± 3.4 (7)	52.5 ± 0.5 (4)	68.1 ± 0.16 (2)	**70.8** ± 0.06 (1)
Cancer	83.2 ± 1.9 (8)	86.0 ± 1.0 (6)	91.2 ± 1.6 (3)	83.8 ± 1.6 (7)	86.7 ± 2.0 (5)	88.8 ± 0.8 (4)	**95.0** ± 0.2 (1)	93.4 ± 0.2 (2)
Cancerint	95.8 ± 1.1 (3)	94.2 ± 0.4 (4)	93.6 ± 1.6 (5)	92.0 ± 1.5 (7)	92.5 ± 1.8 (6)	91.9 ± 0.9 (8)	96.1 ± 0.7 (2)	**97.0** ± 0.41 (1)
Credit	52.2 ± 2.8 (7)	54.8 ± 1.2 (6)	51.4 ± 1.8 (8)	58.5 ± 4.1 (5)	61.5 ± 3.8 (4)	67.1 ± 1.3 (3)	85.3 ± 0.13 (2)	**86.5** ± 0.26 (1)
Dermatology	24.2 ± 3.9 (8)	29.8 ± 1.5 (7)	67.7 ± 0.4 (5)	71.0 ± 3.9 (4)	76.6 ± 3.6 (3)	63.2 ± 0.9 (6)	92.8 ± 0.4 (2)	**95.5** ± 0.28 (1)
Diabetes	63.6 ± 1.5 (6)	64.8 ± 0.8 (5)	52.4 ± 0.5 (8)	65.0 ± 1.9 (4)	67.9 ± 2.4 (3)	61.7 ± 0.8 (7)	69.2 ± 0.23 (2)	**71.0** ± 0.25 (1)
E.Coli	53.8 ± 5.7 (7)	61.0 ± 3.8 (6)	82.0 ± 0.5 (2)	80.1 ± 2.7 (5)	80.7 ± 1.8 (4)	52.8 ± 4.2 (8)	80.9 ± 1.2 (3)	**83.1** ± 0.19 (1)
Glass	52.1 ± 3.8 (4)	53.8 ± 2.9 (3)	47.7 ± 0.4 (5)	40.2 ± 4.1 (7)	41.5 ± 3.5 (6)	39.8 ± 3.7 (8)	54.4 ± 0.21 (2)	**57.6** ± 0.26 (1)
Heart	58.2 ± 3.4 (7)	58.8 ± 2.2 (5)	52.6 ± 1.6 (8)	66.6 ± 2.2 (4)	71.4 ± 2.9 (3)	58.3 ± 2.0 (6)	79.2 ± 0.27 (2)	**82.4** ± 0.24 (1)
Horse	49.7 ± 4.4 (5)	48.3 ± 2.3 (6)	43.4 ± 1.9 (8)	58.6 ± 4.8 (4)	60.9 ± 4.1 (3)	47.6 ± 3.1 (7)	62.0 ± 0.56 (2)	**65.3** ± 0.26 (1)
Iris	88.6 ± 0 (6)	81.3 ± 0.8 (7)	90.0 ± 0.3 (5)	91.1 ± 5.3 (4)	92.3 ± 4.8 (2)	77.2 ± 1.8 (8)	91.5 ± 0.33 (3)	**93.8** ± 0.33 (1)
SECOM	62.2 ± 2.2 (6)	60.6 ± 0.3 (8)	66.1 ± 0.7 (4)	65.9 ± 1.2 (5)	69.4 ± 2.3 (3)	61.6 ± 0.9 (7)	73.1 ± 0.40 (2)	**73.5** ± 0.28 (1)
Thyroid	85.8 ± 2.2 (7)	86.2 ± 0.3 (6)	90.1 ± 0.7 (3)	86.5 ± 1.2 (5)	87.6 ± 2.3 (4)	85.6 ± 0.9 (8)	**94.1** ± 0.09 (1)	92.0 ± 0.09 (2)
Vehicle	44.0 ± 2.2 (4)	44.0 ± 1.5 (4)	45.0 ± 1.8 (3)	40.5 ± 3.8 (8)	42.0 ± 2.2 (7)	43.3 ± 1.5 (6)	46.0 ± 0.62 (2)	**48.4** ± 0.22 (1)
Wine	70.0 ± 3.4 (6)	75.0 ± 0.0 (8)	90.4 ± 0.3 (4)	91.9 ± 1.1 (5)	95.5 ± 1.7 (3)	74.8 ± 0.4 (7)	95.8 ± 0.08 (2)	**97.2** ± 0.08 (1)

TABLE IV: Average clustering accuracy and general ranking of the techniques for all datasets

Dataset	K-means	SOM	EM	USELM	BELMKN	DBSCAN	Single RBM	DRBM-ClustNet
Average	62.32	63.41	67.77	69.16	71.55	64.41	78.92	**80.59**
Rank	8	7	5	4	3	6	2	1

TABLE V: The sum of ranking of the techniques and general ranking based on total ranking

Dataset	K-means	SOM	EM	USELM	BELMKN	DBSCAN	Single RBM	DRBM-ClustNet
Average	92	84	75	81	63	97	30	**17**
Rank	7	6	4	5	3	8	2	1

of RBM layers are 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 when 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 clustering of high dimensional and highly imbalanced data. 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, for 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 are 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 when 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 when compared to all other clustering approaches.

Table IV shows the average clustering accuracy for each of the clustering approaches applied on 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. Amongst 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, 60000 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 the previous section. 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 a fractionally lower performance than fgraphDBN. This performance could be attributed to better subspace clustering by DRBM-ClustNet in the final stage, which uses the Kohonen network. Furthermore, it is also observed that the average number 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 number of clusters for COIL-20, YaleB, and MNIST dataset 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 non-linear 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 x 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.

TABLE VI: Clustering performance (NMI value) on the COIL-20, YaleB and MNIST datasets

Algorithms	COIL-20	YaleB	MNIST
SSC[1]	80.72 ± 0.88	63.42 ± 0.79	68.35 ± 0.02
LSC[1]	65.25 ± 2.66	41.96 ± 0.98	74.06 ± 2.75
SCC[1]	88.23 ± 1.37	34.39 ± 0.61	57.00 ± 0.04
LRR[1]	65.60 ± 1.02	10.19 ± 0.18	46.10 ± 0.01
LRSC[1]	67.55 ± 1.52	14.94 ± 0.39	50.96 ± 0.00
LSR1[1]	72.33 ± 0.88	66.93 ± 1.61	45.73 ± 0.01
LSR2[1]	72.18 ± 1.08	67.04 ± 1.62	45.72 ± 0.02
GGMM[1]	79.53 ± 1.26	12.37 ± 0.31	62.34 ± 1.69
GSC[1]	78.02 ± 3.58	26.55 ± 1.00	70.57 ± 2.95
GNMF[1]	72.00 ± 0.08	49.10 ± 1.37	72.00 ± 1.01
gSAE[1]	57.10 ± 2.88	16.53 ± 1.40	17.12 ± 0.54
RBM[1]	73.81 ± 1.79	18.53 ± 1.64	53.50 ± 1.61
DBN[1]	76.32 ± 2.63	21.77 ± 1.57	52.79 ± 2.44
DAE[1]	75.29 ± 1.73	18.37 ± 2.00	50.57 ± 1.31
SDAE[1]	77.72 ± 0.51	20.99 ± 1.84	52.81 ± 1.61
GraphRBM[1]	94.46 ± 2.14	73.03 ± 1.32	86.66 ± 1.23
mGraphRBM[1]	95.19 ± 1.98	73.36 ± 0.93	88.73 ± 1.49
fGraphRBM[1]	$\mathbf{95.54} \pm 1.43$	75.18 ± 1.29	89.97 ± 0.98
DRBM-ClustNet	95.06 ± 1.56	$\mathbf{76.06} \pm 1.11$	$\mathbf{90.11} \pm 1.03$

[1]Reproduced from [54]

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 6 and 8 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 85.21% and SOM clustering accuracies were 87.54% and 89.05%, respectively. However, the clustering accuracy 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 paper, 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 non-linearly 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 Kohonen Network to cluster the feature-extracted data from DRBM using BIC predicted

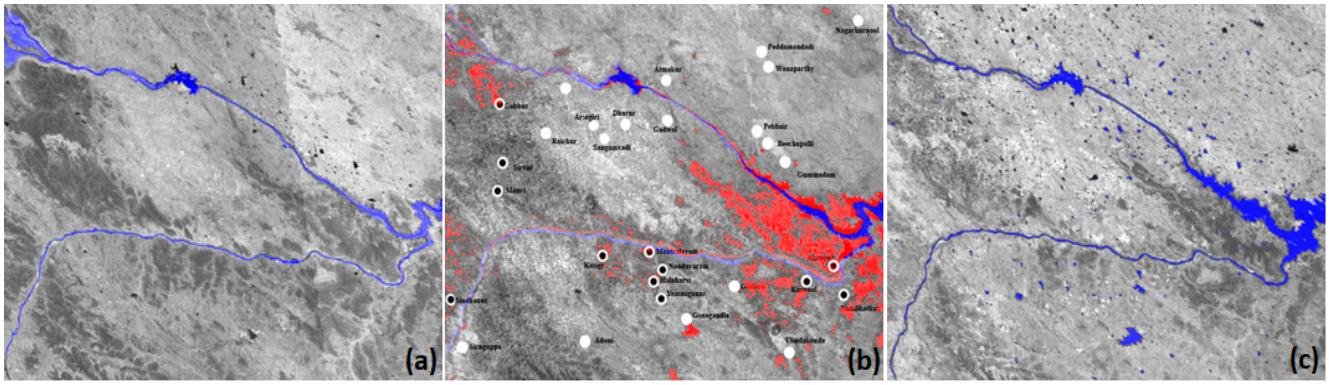

Fig. 7: Assessment flooded region: (a) before flood, (b) during flood, (c) after flood

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, fifteen 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 clustering of non-linearly separable datasets.

APPENDIX

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

$$P(v,h) = \frac{e^{-E(v,h)}}{\sum_v \sum_h e^{-E(v,h)}}, \tag{23}$$

where

$$E(v,h) = -\sum_{i,j} v_i h_j W_{ij} - \sum_i v_i b_i - \sum_j h_j c_j, \tag{24}$$

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)}, \tag{25}$$

where Z is called the partition function which is given by

$$Z = \sum_v \sum_h e^{-E(v,h)}. \tag{26}$$

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 to sample from.

Equation (24) can be expressed in matrix form as

$$E(v,h) = -b^T v - c^T h - v^T W h, \tag{27}$$

$$P(v,h) = \frac{1}{Z} e^{(b^T v + c^T h + v^T W h)}, \tag{28}$$

$$P(h|v) = \frac{P(h,v)}{P(v)}, \tag{29}$$

$$P(h|v) = \frac{1}{P(v)} \frac{1}{Z} e^{(b^T v + c^T h + v^T W h)}. \tag{30}$$

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

$$Z' = \frac{e^{(b^T v)}}{P(v) Z}. \tag{31}$$

Substituting (31) in (30), we get

$$P(h|v) = \frac{1}{Z'} \prod_{j=1}^{p} e^{(c_j h_j + v^T W_{ij} h_j)}. \tag{32}$$

Using Bayes theorem

$$P(h_j = 1|v) = \frac{P(h_j = 1, v)}{P(h_j = 0, v) + P(h_j = 1, v)}. \tag{33}$$

Substitute $h_j = 1$, in (32) to get

$$P(h_j = 1|v) = sigmoid(c_j + v^T W_{ij}). \tag{34}$$

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

$$P(v_i = 1|h) = sigmoid(b_i + W_{ij} h). \tag{35}$$

Therefore

$$P(h|v) = \prod_{j=1}^{p} sigmoid(c_j + v^T W_{ij}), \tag{36}$$

and

$$P(v|h) = \prod_{i=1}^{d} sigmoid(b_i + W_{ij} h). \tag{37}$$

Figs. 5(a) and 5(b) show 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 cluster which can also be expressed in terms of Mean Squared Error (MSS) as follows

$$BIC = ln(N)k - 2ln(\tilde{L}), \tag{38}$$

where $\tilde{L}$ is the maximum likelihood function

Assuming model errors are independent and identically distributed, according to the normal distribution theory

$$h_{ML} = \prod_{i=1}^{p} P(\frac{d_i}{p}), \tag{39}$$

where h_{ML} is the maximum likelihood hypothesis, p is the number of hidden layers and d_i is the output from DRBM

$$h_{ML} = \prod_{i=1}^{p} \frac{e^{-\frac{(d_i-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}, \tag{40}$$

where $d_i = f(x_i) + e_i$, μ and σ 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,

$$h_{ML} = \prod_{i=1}^{p} \frac{e^{-\frac{(d_i-f(x_i))^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}, \tag{41}$$

$$log(h_{ML}) = \sum_{i=1}^{p} ln(\frac{1}{\sqrt{2\pi\sigma^2}}) - \frac{(d_i - f(x_i))^2}{2\sigma^2}. \tag{42}$$

The first term in the above equation is independent of the function f and it can be discarded, therefore

$$log(h_{ML}) = \sum_{i=1}^{p} -\frac{(d_i - f(x_i))^2}{2\sigma^2}, \tag{43}$$

$$-log(h_{ML}) = -\sum_{i=1}^{p} -\frac{(d_i - f(x_i))^2}{2\sigma^2} = \frac{RSS}{n} = MSE. \tag{44}$$

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 Euclidean distance to find the winning neuron. Hence, SOM works well for datasets which are more linearly separable and for datasets where the clusters are non-overlapping. 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.

VI. Acknowledgement

This research is supported by the Accelerated Materials Development for Manufacturing Program at A*STAR via the AME Programmatic Fund by the Agency for Science, Technology and Research under Grant No. A1898b0043.

References

[1] D. Xu, and Y. 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] X. 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, Mar. 2019.
[5] M. Ahmed, A. Naser Mahmood, and J. Hu, "A Survey of Network Anomaly Detection Techniques," J Netw Comput Appl, vol. 60, no. C, pp. 19-31, Jan. 2016.
[6] L. Lv, D. Zhao, and Q. Deng, "Image clustering based on deep sparse representations," in Proc. IEEE Symposium Series on Computational Intelligence, 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 International Conference on Advances in Social Networks Analysis and Mining, 2013, 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. International Conference on Management Science and Engineering, 2006, pp. 121-124.
[9] J. Senthilnath, K. Harikumar, and S. Suresh, "Dynamic Area Coverage for Multi-UAV Using Distributed UGVs: A Two-Stage Density Estimation Approach," in Proc. IEEE International Conference on Robotic Computing, 2018, pp. 165-166.
[10] M. Yin, J. Gao, 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," Encycl. Semantic Comput. Robot. Intell., vol. 02, no. 01, pp. 1850008, May 2018.
[13] R. Xu, and D. 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] H. Li, K. Zhang, and T. Jiang, "The Regularized EM Algorithm," in Proc. National Conference on Artificial Intelligence, Vol. 2, 2005, pp. 807-812.
[17] J.C. Fort, P. Letremy, and M. Cottrell, "Advantages and drawbacks of the Batch Kohonen algorithm," in ESANN, 2002.
[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] Multimodel Inference: Understanding AIC and BIC in Model Selection - Kenneth P. Burnham, David R. Anderson, 2004. [Online]. Available: http://journals.sagepub.com/doi/abs/10.1177/0049124104268644. [Accessed: 29-Sep-2018].
[20] G.E. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines", in Neural Networks: Tricks of the trade, Springer Berlin Heidelberg, 2012, pp. 599-619.
[21] R. Salakhutdinov, and G. Hinton, "Deep Boltzmann Machines," in Artificial Intelligence and Statistics, 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," Chemom. Intell. Lab. Syst., vol. 2, no. 1, pp. 37-52, Aug. 1987.

[24] D.H. Ackley, G.E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," Cogn. Sci., vol. 9, no. 1, 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, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 194-281.

[26] J. Hopfield J., "Neural networks and physical systems with emergent collective computational abilities," in Proc. Natl. Acad. Sci., 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, pp. 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," Neurocomput., vol. 275, no. C, pp. 1186-1199, Jan. 2018.

[33] R. Salakhutdinov and G.E. 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. 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, "GeCo: 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. Liu, Z. Gu, M. Tan, Y. Wang, and Y. Li, "Spatial-Temporal Discriminative Restricted Boltzmann Machine for Event-Related Potential Detection and Analysis," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 27, no. 2, pp. 139-151, 2019.

[40] G. Wang, J. Qiao, J. Bi, Q.S. Jia, M.C. Zhou, "An Adaptive Deep Belief Network With Sparse Restricted Boltzmann Machines," IEEE Trans. Neural Netw. Learn. Syst, pp. 1-12, Dec. 2019.

[41] D.C. Mocanu, H. Bou 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, no. 1, 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. R. Stat. Soc. Ser. B, 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 transactions on image processing, vol. 25, no. 12, pp. 5933-5942, 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, S.C. Sumanth, G. Nagaraj, M. Thapa, and M. Indiramma, "BELMKN: Bayesian Extreme Learning Machines Kohonen Network," Algorithms, vol. 11, no. 5, pp. 56, Apr. 2018.

[48] C.L. Blake and C. Merz, "UCI Repository of machine learning databases," Univ. Calif. Dep. Inf. Comput. Sci., 1998.

[49] G.E. Cook, R.J. Barnett, K. Andersen and A.M. Strauss, "Weld modeling and control using artificial neural networks," IEEE Transactions on Industry Applications, vol. 31, no. 6, p. 1484-1491, 1995.

[50] V. Mani and S.N. Omkar, "Understanding weld modelling processes using a combination of trained neural networks," International journal of production research, vol. 40, no. 3, p.547-559, 2002.

[51] X. Guo, H. Huang, and J. Zhang, "Comparison of different variants of Restricted Boltzmann Machines," in Proc. of 2nd International Conference on Information Technology and Electronic Commerce, 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 ICML 2007, pp. 473-480.

[53] A. Inselberg and T. Avidan, "Classification and Visualization for High-dimensional Data," in Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 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, 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, 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," Advances in Space Research, vol. 50, no. 9, pp. 1274-1287, 2012.

J. Senthilnath is a Scientist in the Institute for Infocomm Research (I^2R) at the Agency for Science, Technology and Research (A*STAR), Singapore. He received the PhD degree in Aerospace Engineering from the Indian Institute of Science (IISc), India. His current research interests include artificial intelligence, multi-agent systems, generative models, online learning, reinforcement learning and optimization. He has published over 100 high-quality papers and won five best paper awards. He is a member of Artificial Intelligence, Analytics And Informatics (AI3), A*STAR. He has been serving as Guest Editor/organizing chair/co-chair/PC members 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. 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. 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 MTech degree from the Indian Institute of Technology (IIT), Chennai, India. He has a work experience of seven years in data science. His research interests are machine learning, NLP and optimization.

Meenakumari Thapa received the MTech degree in Computer Science in 2016 from BITS Pilani, India. She has work experience of more than 5 years in software development. Her research interests include data warehousing and data mining.

Indiramma M is currently working as Professor in Computer Science and Engineering Department and also heading Industry Institute Interaction Cell (IIIC) at B.M.S. College of Engineering, Bengaluru, India. She is in academics from past 30 years. Her research interests and area are Big Data Analytics, Artificial Intelligence, Machine learning, Deep learning. She is the author/co-author of over 60 technical papers in refereed international conferences and journals. She has chaired as Technical Program Committee for several IEEE conferences. Also serving as a reviewer, program committee member, and session chair of several International Conferences. She has also 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 received the Cand.Sci. degree in electrical engineering from the University of Iceland, Reykjavik, in 1984, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1987 and 1990, respectively. Since July 1, 2015 he is the President and Rector of the University of Iceland. From 2009 to 2015 he was the Pro Rector of Science and Academic Affairs and Professor of Electrical and Computer Engineering at the University of Iceland. His research interests are in remote sensing, pattern recognition, and image analysis. Prof. Benediktsson is a Highly Cited Researcher (Clarivate Analysis, 2018-2021). He was the 2011-2012 President of the IEEE Geoscience and and Remote Sensing Society (GRSS). He was Editor in Chief of the IEEE Transactions on Geoscience and Remote Sensing (TGRS) from 2003 to 2008 and has served as Associate Editor of TGRS since 1999, the IEEE Geoscience and Remote Sensing Letters since 2003 and IEEE Access since 2013. He is currently Senior Editor of the Proceedings of the IEEE, is on the International Editorial Board of the International Journal of Image and Data Fusion, and the Editorial Board of Remote Sensing. He is Fellow of IEEE and SPIE.