G-CEALS: GAUSSIAN CLUSTER EMBEDDING IN AUTOENCODER LATENT SPACE FOR TABULAR DATA REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The latent space of autoencoders has been improved for clustering image data by jointly learning a t-distributed embedding with a clustering algorithm inspired by the neighborhood embedding concept proposed for data visualization. However, multivariate tabular data pose different challenges in representation learning than image data, where traditional machine learning is often superior to deep tabular data learning. In this paper, we address the challenges of learning tabular data in contrast to image data and present a novel Gaussian Cluster Embedding in Autoencoder Latent Space (G-CEALS) algorithm by replacing t-distributions with multivariate Gaussian clusters. Unlike current methods, the proposed method defines the Gaussian embedding and the target cluster distribution independently to accommodate any clustering algorithm in representation learning. A trained G-CEALS model is used to extract a quality embedding for unseen test data. Based on the embedding clustering accuracy, the average rank of the proposed G-CEALS method is 1.4 (0.7), which is superior to all eight baseline clustering and cluster embedding methods on seven tabular data sets. This paper shows one of the first algorithms to jointly learn embedding and clustering for improving the representation of multivariate tabular data in downstream clustering.

1 INTRODUCTION

Deep learning has replaced traditional machine learning in many data-intensive research and applications due to its ability to perform concurrent and efficient representation learning and classification. This concurrent learning approach outperforms traditional machine learning that requires *handcrafted features* to perform supervised classification (Lin et al., 2020; Alam et al., 2020). However, representation learning via supervisory signals from ground truth labels may be prone to overfitting Ying (2019) and adversarial attacks (Huang et al., 2017). Moreover, human annotations for supervised representation learning and classification may not be available in all data domains or for all data samples. To address these pitfalls, representation learning via unsupervised clustering algorithms may be a strong alternative to supervised learning methods.

The limitation of supervised representation learning may be overcome using self-supervision or pseudo labels that do not require human-annotated supervisory signals (Boubekki et al., 2021; Caron et al., 2018). In a self-supervised autoencoder, the objective is to preserve all information of input data in a low-dimensional embedding for data reconstruction. However, embeddings for data reconstruction do not emphasize representations essential for downstream classification or clustering tasks. Therefore, unsupervised methods have been proposed for jointly learning embedding with clustering to yield *clustering friendly* representations (Xie et al., 2016; Guo et al., 2017; Moradi Fard et al., 2020; Mrabah et al., 2020; Yang et al., 2017). The existing cluster embedding literature suggests several strict assumptions about clustering algorithms (k-means), cluster distributions (t-distribution), and data modality (image data). While deep representation learning of image data is well studied using convolutional neural networks (CNN), deep learning has not seen much success with structured tabular data. There is strong evidence in the literature that traditional machine learning still outperforms deep models in learning tabular data (Kohler et al., 2019; Smith et al., 2020; Borisov et al., 2021; Kadra et al., 2021; Shwartz-Ziv & Armon, 2022). In this paper, we review the assumptions made in the cluster embedding literature and revise those assumptions for the

representation learning of tabular data. Accordingly, a novel joint learning framework is proposed considering the architectural and algorithmic differences in learning image and tabular data.

The remainder of this manuscript is organized as follows. Section 2 provides a review of the state-of-the-art literature on deep cluster embedding. Section 3 introduces tabular data with some theoretical underpinnings of neighborhood embedding and cluster embedding in support of our proposed representation learning framework. Section 4 outlines the proposed joint cluster embedding framework to obtain a quality representation of tabular data for downstream clustering or classification. Section 5 summarizes the tabular data sets and experiments for evaluating the proposed joint learning framework. Section 6 provides the results following the experiments and compares our proposed method with similar methods in the literature. Section 7 summarizes the findings with additional insights into the results and limitations. The paper concludes in Section 8.

2 Related work

One of the earliest studies on cluster embedding, Deep Embedded Clustering (DEC) Xie et al. (2016), is inspired by the seminal work on t-distributed stochastic neighborhood embedding (t-SNE) (Van Der Maaten & Hinton, 2008). The DEC approach first trains a deep autoencoder by minimizing the data reconstruction loss. The trained encoder part (excluding the decoder) is then fine-tuned by minimizing the Kullback-Leibler (KL) divergence between a t-distributed cluster distribution (Q) on the embedding and a target distribution (P). The target distribution is obtained via a closed-form solution by taking the first derivative of the KL divergence loss between P and Q distributions with respect to P and equating it to zero. Therefore, the assumption of t-distribution holds for both Q and P distributions in similar work. The k-means clustering in the DEC approach is later replaced by spectral clustering to improve the quality of embedding in terms of clustering performance (Duan et al., 2019). The DEC approach is also enhanced by an improved DEC (IDEC) framework (Guo et al., 2017). In IDEC, the autoencoder reconstruction loss and the KL divergence loss are jointly minimized to update the weights of a deep autoencoder and produce the embedding. Similar approaches, including t-distributions, k-means clustering, and KL divergence loss, are adopted in joint embedding and cluster learning (JECL) for multimodal representation learning of text-image data pairs (Yang et al., 2020). The Deep Clustering via Joint Convolutional Autoencoder (DEPICT) approach learns image embedding via a de-noising autoencoder (Dizaji et al., 2017). The embedding is mapped to a softmax function to obtain a cluster distribution or likelihood (Q) instead of assuming a distribution. Following a series of mathematical derivations and assumptions, their final learning objective includes a cross-entropy loss involving P and Q distributions and an embedding reconstruction loss for each layer of the convolutional autoencoder.

A general trend in the cluster embedding literature shows that K-means is the most common clustering method (Xie et al., 2016; Guo et al., 2017; Mrabah et al., 2020; Moradi Fard et al., 2020; Zhang et al., 2019; Yang et al., 2020; Dizaji et al., 2017). The assumption of t-distributed cluster embedding made in the DEC method Xie et al. (2016) continues to appear in the literature Ren et al. (2019); Enguehard et al. (2019); Guo et al. (2017); Duan et al. (2019); Yang et al. (2020); Wu et al. (2022) without any alternatives. The assumption of t-distribution is originally made in the t-SNE algorithm for data visualization using neighborhood embedding maps (Van Der Maaten & Hinton, 2008). We argue that the assumptions of neighborhood embedding for data visualization are not aligned with the requirements of cluster embedding. Moreover, cluster embedding methods proposed in the literature are invariably evaluated on benchmark image data sets. The methods for image learning may not be optimal or even ready to learn tabular data representations. To the best of our knowledge, similar cluster embedding methods have not been studied on multivariate tabular data.

2.1 Contributions

This paper is one of the first to investigate the performance of joint cluster embedding methods on tabular data. The limitations of state-of-the-art joint cluster embedding methods are addressed to contribute a new cluster embedding algorithm as follows. First, we replace the current assumption of t-distributed embedding with a mixture of multivariate Gaussian distributions for multivariate tabular data by providing a theoretical underpinning for this choice. Second, a new cluster embedding algorithm is proposed using multivariate Gaussian distributions that can jointly learn distributions with any clustering algorithm. Third, we define the target cluster distribution on the tabular data

space instead of deriving it from the embedding because traditional machine learning of tabular data is still superior to deep learning and can add complementary benefits to the embedding learned via an autoencoder. Therefore, our embedding and target distributions are independent of each other to flexibly learn any target cluster distribution depending on the application domain.

3 THEORETICAL BACKGROUND

This section provides preliminaries on tabular data in contrast to image data. We draw multiple contrasts between *neighborhood embedding* proposed for data visualization and *cluster embedding* proposed for representation learning to underpin our proposed approach.

3.1 PRELIMINARIES

A tabular data set is represented in a matrix X $\in \Re^{n \times d}$ with n i.i.d samples in rows. Each sample (X_i) is represented by a d-dimensional feature vector, $X_i \in \Re^d = \{x_1, x_2, \ldots, x_d\}$, where i = $\{1, 2, \ldots, n\}$. Compared to a pixel distribution $P(I)$ of an image I, tabular data contain multivariate distributions $P(x_1, x_2, \ldots, x_d)$ of heterogeneous variables often in relatively lower dimensions. Although sequential (genomics) or time-series data can be high-dimensional and structured in data tables, those require special learning architectures. In general, tabular data in business, health records, and many domains fail to take advantage of deep convolutional learning due to the absence of sequential patterns or image-like spatial regularities. Therefore, tabular data sets are identified as the last "unconquered castle" for deep learning Kadra et al. (2021), where traditional machine learning methods are still competing strongly against advanced neural network architectures Kadra et al. (2021); Borisov et al. (2021). Unlike image learning, there is a need for robust tabular data learning methods to outperform the superior traditional machine learning or clustering methods.

3.2 NEIGHBORHOOD EMBEDDING

A neighborhood embedding is a low-dimensional map that preserves the similarity between data points (x_i and x_j) observed in a higher dimension. Maaten and Hinton propose a Student's t-distribution to model the similarity between samples in neighborhood embedding (z_i, z_j) of high-dimensional data points (x_i and x_j) for data visualization (Van Der Maaten & Hinton, 2008). First, the similarity between two sample points (x_i and x_j) in the high dimension is modeled by a Gaussian distribution, p_{ij} in Equation 1. Similar joint distribution can be defined for a pair of points in the low-dimensional embedding (z_i, z_j) as q_{ij} below.

$$p_{ij} = \frac{\exp(-||x_i - x_j||^2/2\sigma^2)}{\sum_{k \neq l} \exp(-||x_k - x_l||^2/2\sigma^2)}, \quad q_{ij} = \frac{\exp(-||z_i - z_j||^2/2\sigma^2}{\sum_{k \neq l} \exp(-||z_k - z_l||^2/2\sigma^2)} \tag{1}$$

The divergence between the target (p_{ij}) and embedding (q_{ij}) distributions is measured using a KL divergence loss, which is minimized to iteratively optimize the neighborhood embedding.

$$KL\,(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{2}$$

To facilitate high-dimensional data visualization in two dimensions (2D), the embedding distribution (q_{ij}) is modeled by a Student's t-distribution, as shown in Equation 3. One primary justification for t-distribution is its heavier tails compared to a Gaussian distribution. A heavier tail aids in an efficient mapping of outliers observed in high dimensional space to the 2D space for data visualization.

$$q_{ij} = \frac{(1 + ||z_i - z_j||)^{-1}}{\sum_{k \neq l} (1 + ||z_k - z_l||)^{-1}} \tag{3}$$

Therefore, data points placed at a moderate distance in high-dimension are pulled farther by a t-distribution to aid visualization in 2D space. In the context of cluster embedding, we argue that the additional separation between points in low dimensions may alter their cluster assignments. To illustrate this phenomenon, we project high-dimensional deep convolutional image features on 2D using 1) t-SNE and 2) two principal components, as shown in Figure 2 (Appendix A). The scattering of data points is evident in the t-SNE mapping, where one blue point appears on the left side of the figure leading to a wrong cluster assignment, unlike the PCA mapping (Figure 2 (b) in Appendix A). In general, the expectations of data visualization and clustering tasks are different, as highlighted in Table 4 (Appendix A), which should be considered in respective representation learning.

3.3 CLUSTER EMBEDDING

Cluster embedding is achieved by infusing cluster separation information into the low-dimensional latent space. While neighborhood embedding is initialized by sampling from a Gaussian distribution, cluster embedding methods use embedding learned from an autoencoder's latent space. However, the current cluster embedding methods use the same t-distribution (Equation 3) to define the embedding distribution (q_{ij}), similar to neighborhood embedding. The target distribution (p_{ij}) is derived as a function of q_{ij}, as shown below.

$$s_{ij} = \frac{q_{ij}^2}{\sum_i q_{ij}}, \;\; p_{ij} = \frac{s_{ij}}{\sum_j s_{ij}}. \tag{4}$$

While pair-wise sample distances in neighborhood embedding have a complexity of O (N^2), the distances from the centroids in embedding are O(N*K). Here, K is the number of clusters, which is much smaller than the number of samples (N). While an outlier point results in N large distances (extremely small p_{ij} values) in neighborhood embedding, there will be much fewer (K<<N) of those large distances in cluster embedding. Therefore, the effect of outliers on cluster embedding can be assumed to be much lower compared to the assumption in neighborhood embedding.

4 PROPOSED METHOD

We propose a novel cluster embedding method, Gaussian Cluster Embedding in Autoencoder Latent Space (G-CEALS), by replacing the t-distribution (Equation 3) with a multivariate Gaussian distribution and the target distribution (Equation 4) with the Gaussian likelihood of individual tabular data samples (X_i) belonging to a given cluster (C_j) as P ($X_i \mid C_j$) or p_{ij}. Two clustering algorithms are used separately in training and evaluating the proposed joint cluster embedding method: 1) k-means and 2) Gaussian mixture model (GMM). The clustering on tabular data space ($X_i \in \Re^d$) yields K cluster assignments for individual samples. Each cluster j is characterized by a centroid vector ($\mu_j \in \Re^d$) and a covariance matrix ($\Sigma_j \in \Re^{dxd}$). Because the dimensionality of tabular data is not as large as image data, Σ and μ parameters can be reasonably sized for computation. Therefore, the soft cluster assignment ($S_x(i,j)$) for individual samples can be obtained using a Gaussian kernel, which is the negative exponent of the Mahalanobis distance ($d_x(i,j)$) between the point (X_i) and the j-th cluster centroid vector, as shown in Equations 5 and 6.

$$d_x(i,j) \;\; = \;\; \sqrt{(X_i - \mu_j^x)\; \Sigma_x^{-1}\; (X_i - \mu_j)^T} \tag{5}$$

$$S_x(i,j) \;\; = \;\; \exp\; (-d_x^2(i,j)) \tag{6}$$

To ensure that the sum of all soft cluster assignments equals one for a given sample, we obtain a joint cluster distribution, P (x_i, μ_j) or p_{ij}, as shown in Equation 7. We set p_{ij}, obtained via superior traditional machine learning, as the target distribution to improve the autoencoder embedding (z_i) of tabular data. Similarly, the embedding distribution (q_{ij}) can be obtained using soft Gaussian cluster assignments (S_z) on the low-dimensional latent space (z_i), as shown in Equation 7.

$$p_{ij} = \frac{S_x(i,j)}{\sum_j S_x(i,j)}, \quad q_{ij} = \frac{S_z(i,j)}{\sum_j S_z(i,j)} \tag{7}$$

Therefore, our P and Q distributions are independent, unlike the current cluster embedding methods. Additionally, the covariance of the target (Σ_x) and embedding (Σ_z) distributions can regulate the scatter or compactness of data clusters, which is impossible with t-distributed embedding.

4.1 LOW-DIMENSIONAL EMBEDDING OPTIMIZATION

A single-layer autoencoder is trained to encode the input (X) to a latent space (Z), which is then decoded to reconstruct the original input ($\hat{X}_i$), as shown in Equation 8.

$$\mathcal{L}_{auto} = \underset{\theta,\Phi}{\operatorname{argmin}} \sum_{i=1}^{N} ||X_i - \hat{X}_i||_2^2. \tag{8}$$

Algorithm 1 Proposed G-CEALS Algorithm

Input: d-dimensional tabular data, $X \in \Re^{n \times d}$, where $X_i \in \Re^d$
Output: Tabular data embedding, $Z \in \Re^{n \times m}$, m<<d
$j-th$ cluster parameters, $\{\mu_j^x, \Sigma_j^x\} \leftarrow$ K-means or GMM clustering of X
$p_{ij} \leftarrow \{\mu_j^x, \Sigma_j^x\}$, in Equation 7
Initialize: $W^0 = \{W_{encoder}, W_{decoder}\}$
for t = 1 $\rightarrow$ n_epochs **do**
  $\{\hat{X}, Z^t\} \leftarrow$ Encoder $(X, W_{encoder}^t)$
  $\{\mu_j^z, q_{ij}\} \leftarrow$ K-means or GMM clustering of Z^t and using q_{ij} in Equation 7
  $\mathcal{L} \leftarrow \mathcal{L}_{auto} + \gamma * \mathcal{L}_{cluster}$, measure the loss terms in Equation 9
  $W^t \leftarrow$ AutoEncoder (W^{t-1}), update weights minimizing the joint loss in Equation 9
end for

Data set	Sample size	Dimensions	Classes	Domain
Breast Cancer	569	30	2	Diagnostic
Dermatology	358	34	6	Histopathological
E. coli	336	7	8	Protein cell
TUANDROMD	4465	241	2	Android malware
Mice Protein	552	78	8	Protein expression
Olive	572	10	3	Food & beverage
Vehicle	846	18	4	Silhouette features

Table 1: Summary of tabular data sets used for comparing clustering performance

Here, θ and Φ denote the trainable parameters of the encoder and decoder, respectively. The embedding obtained following each epoch of training is clustered using a clustering algorithm to obtain the cluster parameters (μ, Σ) and the embedding distribution (q_{ij}). Given the target distribution (p_{ij}) (Equation 7), one of the learning objectives of G-CEALS is to minimize the KL divergence between P and Q distributions ($\mathcal{L}_{cluster}$), as shown in Equation 9 below. The overall learning objective of G-CEALS is to update the autoencoder's weights by minimizing a joint cost function, the encoder reconstruction loss and the KL divergence loss, as below.

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{auto} + \gamma * \mathcal{L}_{cluster} \\ &= \underset{\theta,\Phi}{\operatorname{argmin}} \sum_{i=1}^{N} ||X_i - \hat{X}_i||_2^2 + \gamma * \sum_{i=1}^{N} \sum_{j=1}^{K} p_{ij} log \frac{p_{ij}}{q_{ij}} \end{aligned} \tag{9}$$

Here, γ is a trade-off hyperparameter to balance the contribution of cluster divergence $\mathcal{L}_{cluster}$ during representation learning. The G-CEALS algorithm is summarized in Algorithm 1.

5 EXPERIMENTS

All experimental steps and algorithms are implemented and evaluated in Python. The neural networks are built using the PyTorch package, and clustering modules are developed using the sci-kit-learn package[1] We evaluate the proposed and baseline methods on seven multi-domain and multivariate tabular data sets. A summary of these tabular data sets is provided in Table 1.

5.1 BASELINE METHODS

All existing cluster embedding methods are benchmarked on image data sets, mainly using 2D filters in deep CNN architectures. Compared to 2D image data, limited sample size and dimensionality of tabular feature vectors will require a simpler architecture due to overfitting and 1D filters instead of 2D image filters. For example, the latent space size is set to 256 for image learning, whereas the input dimension of tabular data can be as low as 10. We detail in Appendix B why existing methods for image learning are not suitable to apply directly on tabular data sets. Considering

[1]The source code will be shared publicly and kept private for anonymity during peer review.

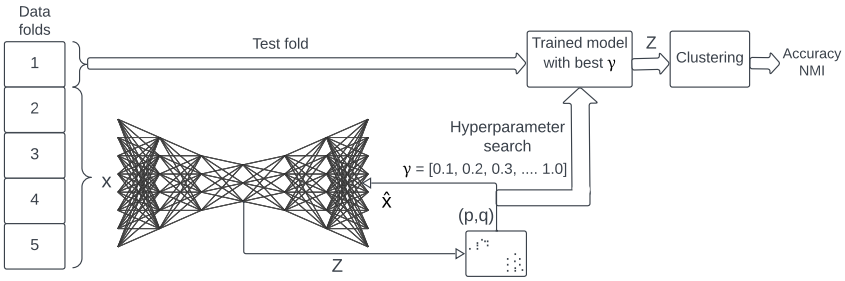


Figure 1: Five-fold cross-validation scheme for training and tuning the model with the best γ value (Equation 9). The clustering accuracy is reported on the embedding of the left-out test data fold.

these factors, we compare our method against four baseline methods. First, the k-means and GMM clustering are performed on input tabular data (X) because traditional machine learning methods are known to produce competitive results on tabular data, unlike image data. Second, a two-stage method is used: 1) the embedding (Z) extraction by training a single-layer autoencoder and then 2) perform clustering on Z (Peng et al., 2016; Tian et al., 2014). Third, embedding learning and clustering are performed jointly on tabular data. We compare fully-connected autoencoder (FC-AE) and CNN autoencoder (similar to the DEPICT method Dizaji et al. (2017)) in single-layer and three-layer settings to investigate the best learning architecture for tabular data. Fourth, despite the challenges in adapting image learning methods to tabular data learning (Appendix B), we use two pioneering methods for cluster embedding, DEC Xie et al. (2016), IDEC Guo et al. (2017), and a more recent method (deep k-means) DKM Moradi Fard et al. (2020) for tabular data to compare with our proposed method. Although deep and sophisticated autoencoder architectures proposed in the literature (stacked, variational, adversarial, convolutional) can be compared in an exhaustive search for the best method, it will introduce architectural bias in our claim for the best algorithm. Therefore, we use a common single-layer autoencoder to make a fair comparison between our and the baseline algorithms, especially considering the size and dimensionality of tabular data.

5.2 Evaluation

The proposed G-CEALS model training involves self-supervised data reconstruction and unsupervised clustering without requiring any ground truth. However, existing studies show that the hyperparameter (γ in Equation 9) value is data-dependent and is tuned based on clustering accuracy that requires ground truth labels. The quality of cluster embedding is evaluated in downstream clustering tasks using two standard metrics: clustering accuracy (ACC) Kuhn (1955) and normalized mutual information (NMI) Estévez et al. (2009). We follow the same metrics for hyperparameter tuning and cluster embedding evaluation. However, existing methods report the clustering accuracy on the same training data set because of the unsupervised nature of the problem. In contrast, we use a semi-supervised five-fold cross-validation scheme to obtain reproducible and transferable learning for downstream clustering or classification. As shown in Figure 1, four data folds are used in unsupervised training and supervised tuning, which is then used to obtain the embedding of a left-out test data fold. We report the average ACC and NMI scores across the five left-out data folds to compare the proposed and baseline methods. These metrics score between 0 (failure) and 1 (perfect clusters). For all evaluation purposes, the cluster number is set to the number of class labels for a given data set. The scores are multiplied by 100 to represent the numbers in percentage.

6 Results

All experiments are conducted on a Dell Precision 5820 workstation running Ubuntu 20.04 with 64GB RAM and an NVIDIA GeForce RTX 3080 GPU with 10GB memory. We standardize all tabular data using the mean and standard deviation of individual variables before training the autoencoder or performing clustering.

Data set		GMM on X	K-means on X	GMM on Z	K-means on Z	DEC	IDEC	DKM	G-CEALS GMM	G-CEALS K-means
Breast cancer	ACC	89.8 (4.6)	90.2 (4.3)	82.2 (7.4)	82.8 (5.7)	68.0 (3.0)	86.0 (3.6)	64.2 (3.9)	**91.2 (4.8)**	85.8 (3.3)
	NMI	55.4 (17.6)	56.2 (17.0)	40.6 (19.6)	39.2 (15.9)	9.2 (6.2)	44.4 (11.3)	2.7 (7.2)	59.2 (18.1)	43.8 (10.6)
Dermatology	ACC	76.8 (8.8)	76.2 (9.2)	72.2 (8.8)	63.0 (4.7)	50.4 (7.4)	76.6 (12.2)	23.2 (0.5)	**77.2 (9.8)**	76.4 (10.1)
	NMI	82.4 (4.2)	83.4 (5.0)	73.4 (5.0)	70.8 (4.4)	45.2 (6.9)	80.6 (7.2)	3.5 (0.3)	77.6 (5.8)	77.4 (5.4)
E. coli	ACC	29.4 (4.5)	29.2 (3.2)	30.6 (4.7)	29.0 (4.0)	26.2 (2.1)	32.6 (3.7)	**35.4 (2.9)**	32.2 (3.7)	31.6 (3.7)
	NMI	19.6 (6.6)	18.4 (5.6)	17.8 (6.5)	18.6 (6.3)	15.0 (5.3)	17.4 (6.3)	14.1 (2.7)	18.0 (6.8)	17.4 (3.1)
TUANDROMD	ACC	79.1 (1.7)	79.1 (1.7)	77.4 (2.6)	77.2 (3.6)	79.2 (1.5)	82.0 (4.1)	48.7 (11.3)	**83.4 (6.6)**	40.8 (4.0)
	NMI	0.5 (0.2)	0.6 (0.1)	1.6 (2.1)	0.5 (0.2)	0.8 (0.7)	13.8 (9.7)	6.8 (5.6)	18.8 (23.8)	36.0 (3.3)
Mice protein	ACC	40.8 (1.7)	40.2 (5.7)	36.4 (5.1)	35.0 (2.5)	34.8 (3.1)	35.6 (2.3)	17.2 (2.7)	**42.0 (1.8)**	40.6 (1.7)
	NMI	42.0 (3.2)	40.6 (4.5)	31.6 (3.8)	34.2 (4.2)	30.8 (6.2)	35.6 (3.6)	3.1 (9.1)	40.4 (3.0)	38.0 (3.9)
Olive	ACC	67.2 (11.5)	71.4 (10.8)	57.6 (11.2)	59.4 (12.7)	55.8 (5.5)	**77.2 (4.0)**	56.5 (0.0)	70.8 (7.9)	73.6 (6.4)
	NMI	39.4 (12.6)	43.2 (11.7)	30.0 (13.2)	32.6 (14.0)	25.4 (5.5)	49.4 (5.2)	0.0 (0.0)	42.6 (9.0)	47.0 (6.6)
Vehicle	ACC	39.4 (2.9)	37.2 (1.7)	39.0 (2.6)	39.0 (2.8)	41.4 (3.3)	42.8 (3.4)	31.1 (5.6)	41.2 (3.3)	**44.2 (3.5)**
	NMI	13.6 (1.0)	12.4 (1.6)	13.2 (1.5)	13.0 (1.7)	12.8 (1.5)	17.6 (3.8)	6.4 (6.3)	11.8 (2.8)	14.6 (2.2)

Table 2: Clustering accuracy (ACC) and normalized mutual information (NMI) scores of proposed and baseline methods. Z = autoencoder latent space without joint learning. X = tabular data space. The DKM method is used on tabular data set without customizing this image learning method for tabular data. Otherwise, a single-layer autoencoder without pre-training is used for all representation learning methods to avoid architectural bias in comparing the algorithms.

6.1 Learning architecture and model selection

We compare the performance of fully connected autoencoders (FC-AE) and convolutional autoencoders (CNN-AE) in single- and three-layer architectures for our tabular data sets. Table 5 in Appendix C clearly shows the superiority of single-layer FC-AE architecture over CNN and deeper architectures, which we select in our subsequent analysis. A single-layer autoencoder maps a d-dimensional tabular data sample to a five-dimensional autoencoder latent space (d>5), considering the range of dimensionality of our tabular data sets (seven to 241). For all experiments, the learning rate is set to 0.0001 with an Adam optimizer. The best autoencoder model jointly trained clustering is selected by searching the best epoch point and γ value while training it for a maximum of 5000 epochs. The best gamma value is searched from a range between 0.1 and 1.0. Table 6 in Appendix D shows example training epochs and gamma values that yield the best clustering accuracy on training data folds.

6.2 Clustering of tabular data versus latent space

Tables 2 and 3 show clustering scores and rank ordering for nine methods, respectively. Traditional clustering (K-means and GMM) on tabular data yields the top three scores for the dermatology, breast cancer, mice protein, and olive data sets. This finding is at odds with the previous finding that direct clustering of images in pixel space yields the worst performance. This is because tabular data sets have relatively lower dimensionality and the absence of regularity in patterns makes such data still suitable for traditional machine learning. For example, these clustering methods yield the worst (<1.0, max. 100) NMI scores for the highest dimensional (241) TUANDROMD data set. Alternatively, clustering methods (GMM, K-means) can be applied to the autoencoder's latent space (Z). A trained autoencoder is used to obtain the embedding on test data folds. The test data embedding is then clustered using GMM and K-means, which are presented as *GMM on Z* and *K-means on Z* in Table 2, respectively. Except for the E. coli data set, GMM on Z performs worse than GMM clustering of other data sets. Similarly, K-means clustering of tabular data yields substantially better accuracy than K-means clustering of Z, except for the vehicle data set.

6.3 Clustering of joint cluster embedding

The autoencoder latent space Z is jointly learned with data cluster distributions in this method. Our results on tabular data are reproduced using two pioneering cluster embedding methods: DEC Xie et al. (2016) and IDEC Guo et al. (2017). The DEC method appears to be among the worst of nine methods presented in Table 2, except for the vehicle data set. Therefore, a method proposed for image learning may not perform equally well on tabular data. However, the improved DEC (IDEC) method shows substantial improvement over the DEC method. The IDEC method always

Data set	GMM on X	K-means on X	GMM on Z	K-means on Z	DEC	IDEC	DKM	CNN GMM	Proposed G-CEALS
Breast cancer	4	3	7	6	8	5	9	1	2
Dermatology	2	4	6	7	8	3	9	5	1
E. coli	5	6	4	7	9	2	1	8	3
TUANDROMD	5	6	7	8	4	2	9	3	1
Mice protein	2	3	5	7	8	6	9	4	1
Olive	4	2	7	5	9	1	8	6	3
Vehicle	5	8	6	7	3	2	9	4	1
Average	3.9 (1.3)	4.6 (2.0)	6.0 (1.1)	6.7 (0.9)	7.0 (2.3)	3.0 (1.7)	7.7 (2.8)	4.4 (2.1)	1.4 (0.7)

Table 3: Data set-specific and average ranks of the proposed and baseline methods based on clustering accuracy. CNN-GMM is a single convolutional layer autoencoder trained jointly with GMM via the proposed algorithm. The DKM method is used without any customization or pretraining.

outperforms the GMM on Z except for the mice protein data set. K-means clustering on Z is always inferior to the IDEC method. The IDEC is the best of all methods for the olive data sets. Unlike DEC and IDEC methods, we do not modify the DKM method for the tabular data sets. Using the original image learning architecture (500-500-2000 neurons) on tabular data, the DKM method yields the worst of all clustering accuracy with a zero NMI score for the olive data set. However, it performs the best for the Ecoli data set with the lowest dimensionality (8) of all data sets.

6.4 Proposed G-CEALS method

Table 2 shows that our proposed G-CEALS method jointly trained with GMM clustering (ACC: 91.2) outperforms the second-best method, K-means clustering (ACC: 90.2), for the breast cancer data set. The proposed G-CEALS method with GMM clustering (ACC: 77.2) also outperforms the second-best method, GMM clustering (ACC: 76.8) for the dermatology data set. For the E. coli data set, the proposed method (ACC: 32.2) is on par with the second-best method, IDEC (ACC: 32.6). The G-CEALS method (ACC: 83.4) outperforms the second-best IDEC (ACC:82.0) method on the TUANDROMD data set. The proposed method with GMM clustering (ACC: 42.0) again outperforms the second-best method, GMM clustering (ACC: 40.8), on the mice protein data set. Only for the Olive data set, the IDEC method (ACC: 77.2) outperforms the proposed G-CEALS method with K-means clustering (ACC: 73.6). However, the G-CEALS method with K-means clustering (ACC: 44.2) outperforms the second best method, IDEC (ACC: 42.8) on the vehicle data set. Table 3 shows that the proposed G-CEALS method ranks the best method on four of the seven data sets. For the other three data sets, G-CEALS ranks among the top three. The average ranking reveals that our method (average rank: 1.4) substantially outperforms the second-best (IDEC, average rank: 3.0) and the third-best method: GMM clustering of tabular data (average rank: 3.9).

7 Discussion of results

The paper is one of the first studies to jointly learn embedding and clustering in an autoencoder latent space with tabular data. The findings of this article can be summarized as follows. First, traditional clustering on tabular data is competitive with clustering on the autoencoder latent space. Our method outperforms these superior methods for clustering. Second, joint embedding learning with a cluster distribution (IDEC and our method) shows improved data representation over all disjoint learning methods (DEC, K-means on Z, GMM on Z) and GMM/K-means clustering on tabular data. Including the DKM method, cluster embedding methods yield the best performance on all seven tabular data sets. Third, our assumption of Gaussian clusters and an independent target distribution have improved the clustering accuracy over the methods using t-distributed clusters and all other baselines on average. We elaborate on these findings in the following sections.

7.1 Image versus tabular data embedding

In computer vision, clustering images on high-dimensional image pixel space is ineffective, as reported in previous studies. On the other hand, deep learning methods simultaneously and efficiently achieve dimensionality reduction, feature learning via convolution operations, and classification in

an end-to-end framework. In contrast, tabular data sets appear with relatively smaller sample sizes and dimensionality than image data and do not contain regularity in features to leverage the benefit of convolutional feature extraction. Our results confirm that deep models with three hidden layers or convolutional networks or clustering on autoencoder embedding are not effective for tabular data. This finding confirms the superiority of traditional machine learning on tabular data over deep learning. However, our proposed representation learning method outperforms the superior clustering method to demonstrate that tabular data need special algorithms to perform effectively with neural network-based learning architectures. G-CEALS performs good with the GMM clustering algorithm may be due to the multivariate Gaussian modeling of the embedding distributions. Clustering on autoencoder embedding (Z) is unsatisfactory because the reconstruction loss may have altered the cluster distribution in the latent space. Therefore, setting the superior cluster distribution on tabular data space as the target may have helped in improving the quality of the cluster embedding. The G-CEALS method is designed for tabular data and may not be appropriate for images because the target distribution would be ineffective on pixel space. Overall, the cluster embedding of tabular data requires a learning algorithm and architecture distinct from those proposed for image data.

7.2 Trade-off between data reconstruction and cluster divergence loss

We observe that one of the first successful deep cluster embedding methods proposed for image clustering (DEC Xie et al. (2016)) performs the worst among all approaches with tabular data sets 3. The DEC method first pretrains an autoencoder and then separately finetunes the KL divergence loss after modeling a t-distributed embedding for clustering. There is no γ parameter because the $\mathcal{L}_{auto}$ and $\mathcal{L}_{cluster}$ optimizations happen separately in DEC. Conversely, the introduction of γ parameter in IDEC and our method has substantially improved the quality of cluster embedding. While the autoencoder reconstruction loss $\mathcal{L}_{auto}$ retains all information in the latent space, $\mathcal{L}_{cluster}$ (Equation 9) disrupts the latent space to learn the clusters. A disproportionate contribution of these two loss terms can collapse the cluster distribution in the embedding leading to poor cluster performance. Therefore, tuning the γ parameter is a crucial step in cluster embedding methods.

7.3 G-CEALS versus other methods

Our results show that the IDEC method performs better in cases where the clustering on tabular data space is substantially worse (E. coli and Olive data sets in Table 3). It may be because the IDEC method does not learn from the cluster distribution on the tabular data space, similarly to the proposed G-CEALS method. In IDEC, the target distribution is derived from the t-distributed embedding instead. Overall, finding an appropriate target distribution for optimizing the KL divergence loss remains an open problem for future work. Another common observation is that the Olive and E. coli data sets have the lowest dimensionality among the seven tabular data sets (Table 1) with only ten and seven variables, respectively. Therefore, it may be inferred that the IDEC method is superior to the proposed G-CEALS method when the dimensionality of tabular data is less than ten. For tabular data sets with the highest dimensionality (TUANDROMD (241) and mice protein (78)), the G-CEALS method outperforms the IDEC method as the best-performing method. This observation suggests that Gaussian embedding may be more suitable over its t-distributed counterparts when a tabular data set has higher dimensionality.

8 Conclusions

This paper presents a novel cluster embedding method in one of the first studies on tabular data sets. The superiority of our G-CEALS method suggests that multivariate Gaussian distribution is superior to the widely used t-distribution assumption for learning tabular data embedding. Our findings show that tabular data sets require learning algorithms and architectures distinct from those proposed for image learning. This data-centric learning approach may improve a deep model's performance on tabular data over its currently known superior machine learning counterparts. The proposed joint learning framework provides a promising representation learning of tabular data over its superior machine learning counterparts.

ACKNOWLEDGMENTS

Acknowledgments are kept hidden for anonymity in double-blind review.

REFERENCES

M Alam, M D Samad, L Vidyaratne, A Glandon, and K M Iftekharuddin. Survey on Deep Neural Networks in Speech and Vision Systems. *Neurocomputing*, 417:302–321, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.07.053.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep Neural Networks and Tabular Data: A Survey. *arXiv preprint arXiv:2110.01889*, oct 2021. URL http://arxiv.org/abs/2110.01889.

Ahcène Boubekki, Michael Kampffmeyer, Ulf Brefeld, and Robert Jenssen. *Joint optimization of an autoencoder for clustering and embedding*, volume 110. Springer US, 2021. ISBN 0123456789. doi: 10.1007/s10994-021-06015-5. URL https://doi.org/10.1007/s10994-021-06015-5.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11218 LNCS, pp. 139–156, jul 2018. ISBN 9783030012632. doi: 10.1007/978-3-030-01264-9_9. URL http://arxiv.org/abs/1807.05520.

Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep Clustering via Joint Convolutional Autoencoder Embedding and Relative Entropy Minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pp. 5747–5756, 2017. ISBN 9781538610329. doi: 10.1109/ICCV.2017.612.

Liang Duan, Charu Aggarwal, Shuai Ma, and Saket Sathe. Improving spectral clustering with deep embedding and cluster estimation. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2019-Novem(Icdm):170–179, 2019. ISSN 15504786. doi: 10.1109/ICDM.2019.00027.

Joseph Enguehard, Peter O'Halloran, and Ali Gholipour. Semi-supervised learning with deep embedded clustering for image classification and segmentation. *IEEE Access*, 7:11093–11104, 2019. ISSN 21693536. doi: 10.1109/ACCESS.2019.2891970.

Pablo A. Estévez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20:189–201, 2009. ISSN 10459227. doi: 10.1109/TNN.2008.2005601.

Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. *IJCAI International Joint Conference on Artificial Intelligence*, 0: 1753–1759, 2017. ISSN 10450823. doi: 10.24963/ijcai.2017/243.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned simple nets excel on tabular datasets. *Advances in neural information processing systems*, 34:23928–23941, 2021.

Niklas Kohler, Maren Buttner, and Fabian Theis. Deep learning does not outperform classical machine learning for cell-type annotation. *bioRxiv*, pp. 653907, 2019. doi: 10.1101/653907.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 3 1955. ISSN 00281441. doi: 10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/10.1002/nav.3800020109.

Lei Lin, Wencheng Wu, Zhongkai Shangguan, Safwan Wshah, Ramadan Elmoudi, and Beilei Xu. Hpt-rl: Calibrating power system models based on hierarchical parameter tuning and reinforcement learning. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1231–1237. IEEE, 2020.

Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. Deep k-Means: Jointly clustering with k-Means and learning representations. *Pattern Recognition Letters*, 138:185–192, 2020. ISSN 01678655. doi: 10.1016/j.patrec.2020.07.028. URL https://doi.org/10.1016/j.patrec.2020.07.028.

Nairouz Mrabah, Naimul Mefraz Khan, Riadh Ksantini, and Zied Lachiri. Deep clustering with a dynamic autoencoder: From reconstruction towards centroids construction. *Neural Networks*, 130:206–228, oct 2020. doi: 10.1016/j.neunet.2020.07.005. URL https://doi.org/10.1016%2Fj.neunet.2020.07.005.

Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi. Deep subspace clustering with sparsity prior. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 1925–1931. AAAI Press, 2016. ISBN 9781577357704.

Yazhou Ren, Kangrong Hu, Xinyi Dai, Lili Pan, Steven C.H. Hoi, and Zenglin Xu. Semi-supervised deep embedded clustering. *Neurocomputing*, 325:121–130, jan 2019. ISSN 18728286. doi: 10.1016/j.neucom.2018.10.016.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, may 2022. ISSN 15662535. doi: 10.1016/j.inffus.2021.11.011. URL https://linkinghub.elsevier.com/retrieve/pii/S1566253521002360.

Aaron M. Smith, Jonathan R. Walsh, John Long, Craig B. Davis, Peter Henstock, Martin R. Hodge, Mateusz Maciejewski, Xinmeng Jasmine Mu, Stephen Ra, Shanrong Zhao, Daniel Ziemek, and Charles K. Fisher. Standard machine learning approaches outperform deep representation learning on phenotype prediction from transcriptomics data. *BMC Bioinformatics*, 21(1):119, dec 2020. ISSN 1471-2105. doi: 10.1186/s12859-020-3427-8.

Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), Jun. 2014. doi: 10.1609/aaai.v28i1.8916. URL https://ojs.aaai.org/index.php/AAAI/article/view/8916.

Laurens Van Der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2625, 2008. ISSN 15324435.

Lirong Wu, Lifan Yuan, Guojiang Zhao, Haitao Lin, and Stan Z. Li. Deep Clustering and Visualization for End-to-End High-Dimensional Data Analysis. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022. ISSN 21622388. doi: 10.1109/TNNLS.2022.3151498.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *33rd International Conference on Machine Learning, ICML 2016*, volume 1, pp. 740–749, 2016. ISBN 9781510829008.

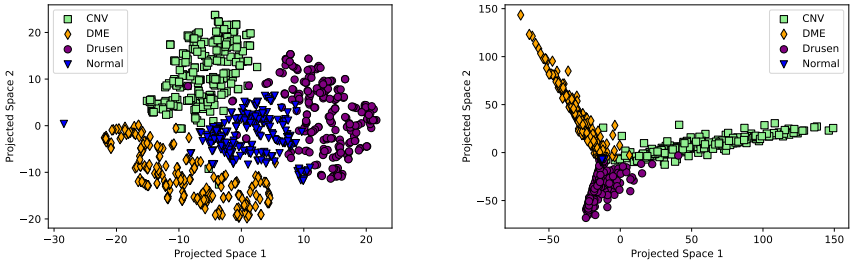
Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards K-means-friendly spaces: Simultaneous deep learning and clustering. In *34th International Conference on Machine Learning, ICML 2017*, volume 8, pp. 5888–5901, 2017. ISBN 9781510855144.

Sean T. Yang, Kuan Hao Huang, and Bill Howe. JECL: Joint embedding and cluster learning for image-text pairs. *Proceedings - International Conference on Pattern Recognition*, pp. 8344–8351, 2020. ISSN 10514651. doi: 10.1109/ICPR48806.2021.9412667.

Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, pp. 022022. IOP Publishing, 2019.

Rui Zhang, Yinglong Xia, Hanghang Tong, and Yada Zhu. Robust embedded deep K-means clustering. *International Conference on Information and Knowledge Management, Proceedings*, pp. 1181–1190, 2019. doi: 10.1145/3357384.3357985.

A CLUSTER EMBEDDING VERSUS NEIGHBORHOOD EMBEDDING



(a) t-SNE projected (b) PCA projected

Figure 2: Two-dimensional embeddings of high dimensional image features extracted from a deep convolutional neural network

	t-SNE Van Der Maaten & Hinton (2008)	DEC Xie et al. (2016)/ IDEC Guo et al. (2017)
Purpose	Neighborhood embedding	Cluster embedding
Low-dimensional embedding (z_i)	Sampled from Gaussian with low σ^2	Autoencoder latent space
Distance or similarity measure	Between sample points (x_i, x_j)	Between point & cluster centroid (x_i, μ_j)
Embedding distribution (q_{ij})	t-distribution, α = 1	t-distribution, α = 1
Target distribution (p_{ij})	Gaussian in high-dimensional space (x)	A function of t-distributed q_{ij}
Learning	$z_{i+1} = z_i + \frac{d\ KLD(p,q)}{d(z_i)}$	$w_{i+1} = w_i + \frac{d\ KLD(p,q)}{d(w)}$
Purpose	Visualization in d = 2	Clustering in d $>$ 2

Table 4: Comparison between neighborhood embedding proposed in t-SNE for data visualization Van Der Maaten & Hinton (2008) and cluster embedding proposed in DEC Xie et al. (2016) inspired by t-SNE. α = degrees of freedom of t-distribution, d = dimension of low-dimensional embedding. W represents the trainable parameter of an autoencoder.

B IMAGE VERSUS TABULAR DATA CLUSTER EMBEDDING METHODS

Caron et al. learned visual features from images using AlexNet and VGG-16 after Sobel filtering for color removal and contrast enhancement, which do not apply to tabular data (Caron et al., 2018). Their *deepCluster* architecture had five convolutional layers with up to 384 2D image filters to learn image texture. Transfer learning of tabular data, similarly to VGG-16 on images, is not intuitive because tabular datasets (e.g., from health records data to criminal justice data) do not share common transferable textures or patterns like images. Furthermore, these deep architectures can easily overfit data with limited sample size and dimensionality, similarly to tabular data sets. The DEPICT method used a convolutional denoising autoencoder for reconstructing original images from corrupted images (Dizaji et al., 2017). Because similar image corruption is not trivial on data tables, we used standard CNN autoencoders with single and three layers as baseline methods (Table 5 - Appendix C). The deep clustering network (DCN) method used a fully-connected deep neural network (FC-DNN) with 2000, 1000, 1000, 1000, and 50 neurons for learning high-dimensional image data (Yang et al., 2017), whereas tabular data can have as low as ten features. They avoided CNN architecture for image learning - *"to concentrate on their DCN algorithm to provide a proof-of-concept rather than exhausting the possibilities of combinations"*. However, they leveraged the architectural benefit of a stacked deep autoencoder instead of using a regular autoencoder model, which can mask the original contribution of the algorithm. Similarly, the deep k-means (DKM) method used FC-DNN instead of CNN for image learning (Moradi Fard et al., 2020). The authors of DKM compared their method only against the methods that used FC-DNN (excluding all CNN

based methods) to avoid architectural bias. However, we use the original DKM method to reproduce cluster embedding on tabular data. The suboptimal performance of DKM in Tables 2 and 3 suggests the need for customizing these architectures for learning tabular data. Recently, Mrabah et al., in their Dynamic Autoencoder (DynAE) method, used image augmentation (shifting and rotation), which is not intuitive with tabular data, and a 2D convolutional adversarial autoencoder for image data, which needs substantial customization of the adversarial network for learning 1D feature vectors in tabular format (Mrabah et al., 2020). One main limitation of the DynAE and DKM methods is that the latent dimension is restricted to the number of clusters, whereas our method is proposed for any latent dimension.

In general, the architectural benefits include a) stacked vs. variational vs. adversarial vs. convolutional autoencoders, b) CNN vs. DNN, c) with vs. without pretraining, d) deep vs. shallow are explored to exhaustively search the best model, which do not provide a fair ground to prove an algorithm's performance or the hypothesis regarding the proposed embedding distribution. Therefore, to focus on our hypothesis and algorithm contributions at this early stage of tabular data learning research, we avoid exhaustive model search using special autoencoders (stacked, variational, adversarial, convolutional), model pretraining, data augmentations, and deepening of the network. Therefore, we used the same single-layer standard autoencoder architecture to fairly compare against the baseline algorithms.

C COMPARISON BETWEEN LEARNING ARCHITECTURES.

Data set		FC-AE GMM	FC-AE K-means	CNN GMM	CNN K-means	CNN GMM	FC-AE GMM
	NHL	1	1	1	1	3	3
Breast cancer	ACC	91.2 (4.8)	85.8 (3.3)	**92.6 (3.3)**	89.8 (4.3)	62.7 (3.1)	85.4 (12.0)
	NMI	59.2 (18.1)	43.8 (10.6)	63.8 (11.7)	55.4 (11.2)	0.0 (0.0)	49.4 (24.0)
Dermatology	ACC	**77.2 (9.8)**	76.4 (10.1)	75.7 (5.4)	72.3 (4.9)	31.0 (3.8)	74.3 (6.6)
	NMI	77.6 (5.8)	77.4 (5.4)	77.8 (4.5)	76.1 (4.5)	0.0 (0.0)	82.9 (5.0)
E. coli	ACC	32.2 (3.7)	31.6 (3.7)	27.7 (2.3)	29.2 (2.5)	**34.2 (4.6)**	32.4 (1.5)
	NMI	18.0 (6.8)	17.4 (3.1)	17.7 (4.2)	17.1 (3.7)	13.3 (3.3)	15.6 (3.2)
TUANDROMD	ACC	**83.4 (6.6)**	40.8 (4.0)	79.4 (1.6)	79.6 (1.2)	79.4 (1.6)	80.6 (4.5)
	NMI	18.8 (23.8)	36.0 (3.3)	0.4 (0.2)	2.3 (1.0)	0.4 (0.2)	7.9 (14.6)
Mice protein	ACC	**42.0 (1.8)**	40.6 (1.7)	37.7 (2.2)	36.2 (3.0)	18.8 (2.4)	39.9 (2.1)
	NMI	40.4 (3.0)	38.0 (3.9)	36.4 (1.3)	33.3 (2.7)	0.0 (0.0)	40.8 (3.3)
Olive	ACC	70.8 (7.9)	**73.6 (6.4)**	58.9 (5.5)	71.2 (10.6)	56.5 (6.1)	66.8 (8.7)
	NMI	42.6 (9.0)	47.0 (6.6)	30.7 (6.9)	44.4 (10.2)	0.0 (0.0)	39.8 (10.4)
Vehicle	ACC	41.2 (3.3)	**44.2 (3.5)**	40.8 (2.2)	42.4 (4.6)	40.7 (4.3)	42.1 (4.2)
	NMI	11.8 (2.8)	14.6 (2.2)	11.5 (2.3)	13.5 (4.9)	12.7 (1.9)	14.8 (3.4)

Table 5: Comparing fully-connected autoencoder (FC-AE) with CNN-autoencoder for the proposed G-CEALS algorithm in single or three-layer settings. NHL = Number of hidden or convolutional layers.

D TUNING MODEL HYPERPARAMETERS

Data set	Method	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Best γ	IDEC	0.5	0.3	0.2	0.9	0.4
	G-CEALS with GMM	0.5	0.2	0.4	0.1	0.7
	G-CEALS with K-means	0.5	0.3	0.2	0.9	0.4
Best epoch point	IDEC	714	2911	1552	2319	3237
	G-CEALS with GMM	285	2231	135	3572	749
	G-CEALS with K-means	791	38	1549	2039	2315

Table 6: Hyperparameter values selected in five-fold cross-validation of the breast cancer data set.