Computing and Informatics, Vol. 40, 2021, 1–28, doi: 10.31577/cai_2021_1_1

UNSUPERVISED ADAPTATION FOR HIGH-DIMENSIONAL WITH LIMITED-SAMPLE DATA CLASSIFICATION USING VARIATIONAL AUTOENCODER

Mohammad Sultan Mahmud, Joshua Zhexue Huang*
Xianghua Fu, Rukhsana Ruby, Kaishun Wu

College of Computer Science and Software Engineering
Shenzhen University, Shenzhen 518060, China
&
National Engineering Laboratory for Big Data System Computing Technology
Shenzhen University, Shenzhen 518060, China
e-mail: {sultan, zx.huang, fuxh, ruby, Wu}@szu.edu.cn

Abstract. High-dimensional with limited-sample size (HDLSS) datasets exhibit two critical problems: (1) Due to the insufficiently small-sample size, there is a lack of enough samples to build classification models. Classification models with a limited-sample may lead to overfitting and produce erroneous or meaningless results. (2) The 'curse of dimensionality' phenomena is often an obstacle to the use of many methods for solving the high-dimensional with limited-sample size problem and reduces classification accuracy. This study proposes an unsupervised framework for high-dimensional limited-sample size data classification using dimension reduction based on variational autoencoder (VAE). First, the deep learning method variational autoencoder is applied to project high-dimensional data onto lower-dimensional space. Then, clustering is applied to the obtained latent-space of VAE to find the data groups and classify input data. The method is validated by comparing the clustering results with actual labels using purity, rand index, and normalized mutual information. Moreover, to evaluate the proposed model strength, we analyzed 14 datasets from the Arizona State University Digital Repository. Also, an empirical comparison of dimensionality reduction techniques shown to conclude their applicability in the high-dimensional with limited-sample size data settings. Experimental results demonstrate that variational autoencoder can achieve more accuracy than traditional dimensionality reduction techniques in high-dimensional with limited-sample-size data analysis.

* Corresponding author

Keywords: HDLSS problem, dimensionality reduction, unsupervised framework, variational autoencoder, deep learning

Mathematics Subject Classification 2010: 68-T99

1 INTRODUCTION

By essence, in many domains, including computational biology, bioinformatics, ecology, geology, neuroscience datasets are characterized by a small number of samples N (records), but a large number of features p (dimensions). These datasets are called the high-dimensional limited-sample size (HDLSS) dataset (*aka* 'fat' dataset), often written as $p \gg N$. HDLSS data classification and clustering both are crucial and challenging tasks in data mining and machine learning. High variance and bias are the main concern for HDLSS data analysis. As a result, simple and highly-regularized classification and regression techniques often become the method of choice [1].

The caution of insufficiently small-sample size has been flagged, especially dangerous to draw conclusions from the limited-sample dataset [2, 3, 4]. In HDLSS datasets, typically sample size is too small to allow for the split into train-test testing or k-fold cross-validation. However, data miners train a classifier model and estimate the classification accuracy. It can be challenging to build a stable and reliable classifier and draw a conclusion from such limited-samples.

The difficulty occurs when dealing with high-dimensional data, where the accuracy of classifiers or clustering algorithms tends to deteriorate are often referred to as the curse of dimensionality [5, 6]. Consequently, dimensionality reduction (DR) is an innovative and important tool in the fields of data analysis, data mining, and machine learning. Several techniques have been proposed for DR such as principal components analysis (PCA) [7, 8, 9], independent components analysis (ICA) [10], factor analysis (FA) [11], multidimensional scaling (MDS) [12], and non-negative matrix factorization (NMF) [13]. Traditional methods like PCA, ICA, FA, and classical MDS suffer from being based on linear models.

However, recently, some nonlinear dimensionality reduction (NLDR) (*aka* 'manifold learning') methods have been developed and have become a popular topic. Traditional DR methods PCA, ICA, FA, and TSVD usually require sufficient data, otherwise, they might be less effective. In the context of $N > p$, there is a relatively large application of PCA, ICA, FA, and TSVD. In the case of $p \gg N$, transformed lower-dimension (d) is lower than or equal to sample size ($d \leq N$), there is difficulty to preserved information about the original data in such too lower-dimensional space. Therefore, for the HDLSS problem ($p \gg N$), it is clear that the basic formulation of PCA, ICA, FA, and TSVD does not work.

Over the decades, deep learning (DL) has succeeded in a variety of fields to extract information from high-dimensional data such as image, speech, text, and

vision [14, 15]. The limitation of DL is getting a large number of training data to ensure learning accuracy. Different types of deep learning architecture have been proposed to solve the problem of insufficient samples [16, 17]. Recently, unsupervised deep learning models such as generative adversarial net (GAN) and variational autoencoder (VAE) have shown the modeling power without the labels. VAEs harness to generate 'blurry' data compared with other generative models, also more stable to train [18]. Moreover, unlike many existing techniques (e.g., PCA, ICA, FA), VAE also capable of reducing the dimension as necessary from the high-dimensional space.

Recently, we examined that VAE based dimensionality reduction outperforms PCA, fastICA, FA, NMF, and LDA in HDLSS data classification in a supervised model [48]. It is also addressed that classifiers and obtained reduced dimensions show inconsistent behavior w.r.t classification accuracy and vary considerably. This discrepancy raises the supervised framework applicability to HDLSS data analysis, yet critical. Although there are varieties of classification algorithms, the challenge is an appropriate selection in the application of the limited-sample domain. Hence, it is more advantageous to use an unsupervised framework. We favored an idea of the unsupervised model, in [19]. It is noteworthy to mention that this paper is an extension of our work reported at the 4th International Conference on Advanced Robotics and Mechatronics [19].

This study manifests an extensive empirical analysis of traditional DR techniques and the effectiveness of the approach we proposed in [19]. The proposed DR approach can maintain a reasonable size of dimensions even after the reduction, unlike many existing methods that often reduce the dimensions too heavily. The problem with a huge number of dimensions is known as the curse of dimensionality, but we argue that there is a blessing of dimensions as well in the sense that we often need a reasonable size of dimensions for useful data analysis. The proposed DR approach can maintain a reasonable-size of dimensions and utilize the blessing of dimensions in the HDLSS setting.

The contribution of this study is to present an unsupervised framework for HDLSS data classification. In particular, we employed the deep learning technique variational autoencoder for dimensionality reduction, and the clustering is applied on the obtained latent variables (low-dimensional space) to group data, and then validated clustering results with the original class labels. We tested the effectiveness of the proposed framework in varieties types of fourteen HDLSS datasets, such as biology, image, mass, and spectrometry, and comparisons with various reduced dimensions in classification are also shown. Moreover, we provided an empirical comparison of different dimensionality reduction methods, compare their performances on a wide range of challenging HDLSS datasets, and conclude their applicability to HDLSS application.

The paper is organized as follows. Section 2 surveyed related works on dimensionality reduction of HDLSS data analysis. In Section 3, the idea of the proposed method is described. Empirical comparisons and concluding remarks are in Sections 4 and 5, respectively.

2 RELATED WORK

2.1 State-of-the-Art Data Dimensionality Reduction Techniques

HDLSS data analysis is vital for scientific discoveries in many areas. When dealing with HDLSS data, the overfitting and high-variance gradients are the main challenges in majority models. In the past, significant work has been done on HDLSS asymptotic theory, where the sample size N is fixed or $N/p \to 0$ as the data dimension $p \to \infty$ [20, 6]. In the HDLSS context, Jung and Marron explored several types of geometric representations and showed inconsistent properties of the sample eigenvalues and eigenvectors [21].

In past decades, numerous dimensionality reduction (DR) techniques, including PCA [7, 8, 9], ICA [10], FA [11], MDS [12], NMF [13] proposed. PCA is perhaps one of the oldest and best-known DR methods in high-dimensional data processing and mining. Traditional methods like PCA, ICA, FA, and classical MDS suffer from being (based on) linear models. Recently, to discover the intrinsic manifold structure of the data, nonlinear DR algorithms are developed, such as locally linear embedding (LLE) [22], kernel PCA (KPCA) [23], sparse PCA (SPCA) [24], and spectral embedding (SE) [25]. DR methods can be roughly categorized into supervised and unsupervised. Semi-supervised DR is recognized as a new issue in semi-supervised learning, which learns from a combination of both labeled and unlabeled data. Table 1 presents a summary of canonical DR methods to clarify their characteristic in HDLSS ($p \gg N$) settings.

Supervised or classification methods are often used for HDLSS data analysis. Most achievements in the supervised model show that more samples and lower-dimension can improve the performance of classifiers. However, sufficient large-samples are essential to building a classification model with good generalization ability, expected that perform equally well on the training and independent testing dataset. Consequently, the classification technique does not suit with small-sample size dataset, to avoid overfitting (training and validation data), the unsupervised (that is, clustering) methods also applied for HDLSS analysis.

Many researchers considered PCA in the classification and clustering of biological data in the context of HDLSS, among them are [26, 27, 28, 29]. In fact, PCA reduces the dimensionality of the data linearly, and it may not extract some nonlinear relationships of the data. In the same vein, [30, 31] pointed that though many researchers considered PCA as a DR method, it is even more useful for data visualization in high-dimensional contexts. NMF is another widely used tool for high-dimensional data analysis. NMF has also been applied for gene clustering, microarray and protein sequence data analysis, and recognition [32, 33]. PCA is deterministic while NMF is stochastic, so NMF appears to be more suitable for HDLSS data analysis than PCA. In [48], explored that PCA, ICA, FA, LDA, MBDL, and NMF are not efficient for dimensionality reduction in HDLSS data classification.

For decades, deep learning (DL) techniques have achieved state-of-the-art performance with large-sample sizes in many domains. Nevertheless, recently, few efforts

have been devoted to applying DL to the HDLSS settings by [34, 53, 15]. DLs also suffer overfitting on HDLSS problems. The 'Dropout' method was proposed to prevent overfitting by reducing the parameters of the full-connection layer, for detail see [35, 16]. Also, a transfer learning-based deep convolutional neural network (CNN) has been developed to solve the problem of the small-sample dataset [17]. In the last few years, a variety of supervised and semi-supervised deep learning models has blossomed in the context of natural language processing (NLP). Recently, there have few efforts to develop unsupervised learning techniques by building upon variational autoencoders [36, 37, 38].

Algorithm	Method	Degrees of Freedom
Principal components analysis (PCA) [7, 8, 9]	LDR	$d \leq N$
Independent components analysis (fastICA) [10]	LDR	$d \leq N$
Factor analysis (FA) [11]	LDR	$d \leq N$
Truncated SVD (*aks* LSA) [39]	LDR	$d \leq N$
Latent Dirichlet allocation (LDA) [40, 41]	NLDR	$d < p \star$
Mini-batch dictionary learning (MBDL) [42]	NLDR	$d < p \star$
Non-negative matrix factorization (NMF) [13]	NLDR	$d < p \star$
Kernel PCA (KPCA) [23]	NLDR	$d \leq N$
Sparse PCA (SPCA) [24]	NLDR	$d < p \star$
Locally linear embedding (LLE) [22]	NLDR	$d < N$
Spectral embedding (SE) [25]	NLDR	$d < N$
Multidimensional scaling (MDS) [12]	NLDR/LDR	$d < p \star$
Autoencoder (AE) [43, 45]	NLDR/LDR	$d < p \star$

Degrees of freedom is possible computed number of latent variables

$\star$ indicates succeed at most are desired to keep dimension

Table 1. A summary of most known and used dimensionality reduction techniques in the HDLSS setting. N: number of the samples, d: dimensionality of the latent space, p: dimensionality of the data space, LDR: linear dimensionality reduction; NLDR: nonlinear dimensionality reduction.

2.2 Variational Autoencoder (VAE) Model

Kingma and Welling [43] introduced the VAE, which is based on the autoencoding framework as a latent variable generative model (see Figure 1). VAE can discover nonlinear explanatory features through data compression and nonlinear activation functions. A traditional autoencoder (AE) consists of an encoding and a decoding phase where input data is projected into lower-dimensions and then reconstructed. AE is deterministic and trained by minimizing reconstruction error. In contrast, VAE is stochastic and learns the distribution of explanatory features over samples. VAE achieves these properties by learning two distinct latent representations: a mean (μ) and standard deviation (σ) vector encoding. The model adds a Kullback-Leibler (KL) divergence term to the reconstruction error, which also regularizes

weights by constraining the latent vectors to match a Gaussian distribution [44]. In a VAE, these two representations are learned concurrently through the use of a reparameterization trick that permits a backpropagated gradient. Importantly, projected data onto an existing VAE feature space enabling new data to be assessed. In this, we aim to build a VAE that compresses high-dimensional features and reveals a relevant latent space.

A VAE performs density estimation on $p(x, z)$ where z are latent variables, to maximize the likelihood of the observed data x, where $x_i \in X \subset \mathbb{R}^m$ is the i^{th} observation: $\log p(X) = \sum_{i=1}^{N} \log p(x_i)$.

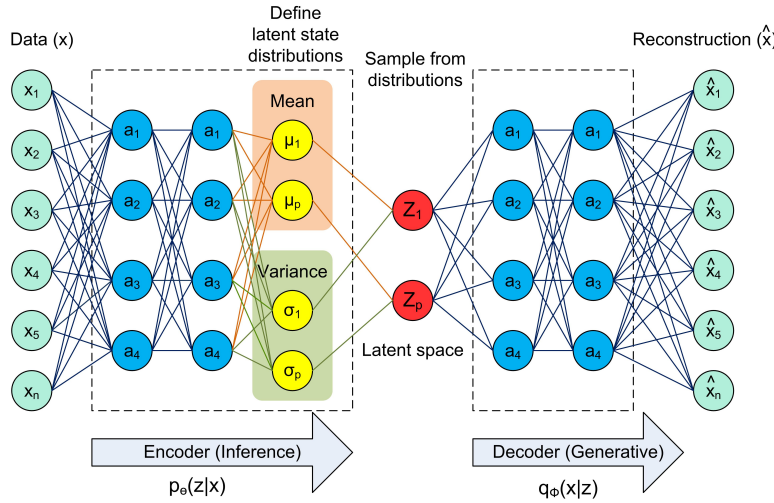

Figure 1. Variational autoencoder (VAE) framework [43, 45]

A VAE consists of an encoder, a decoder, and a loss function unit.

The encoder is a neural network, compresses data x into a latent space z. Encoder's transformed representation is d-dimensional, which is much small than the original p-dimensions. The lower-dimensional space is stochastic; encoder output parameter is $p_\theta(z|x)$, which is a Gaussian probability density. Encoder weight and bias parameter is θ.

The decoder is another neural network, gets input as latent representation z and output the parameters of a probability distribution of the data. Its weight and bias parameter is ϕ. Decoder reconstructs the data is denoted by $q_\phi(x|z)$. It goes from a smaller to a larger dimension. Information loss computed using the reconstruction log-likelihood $\log q_\phi(x|z)$. This measure states how effectively decodes the z into N real-valued numbers. VAE uses a decoder-based generative model as

$$p(x, z) = p(x|z)p(z),$$
$$p(z) = \mathcal{N}(z; 0, 1).$$

The loss function of the VAE is the negative log-likelihood with a regularizer. Since the marginal likelihood is difficult to work with directly for non-trivial models, instead a parametric inference model $p(x|z)$ is used to optimize the variational lower-bound on the marginal log-likelihood

$$\mathcal{L}(x, \theta, \phi) = -\mathbb{E}_{q(z|x_i)}[\log q_\theta(x_i|z)] + \mathbf{KL}(p_\theta(z|x_i)\|p((z)). \tag{1}$$

In Equation (1), the first term of $\mathcal{L}$ is reconstruction error or expected negative log-likelihood of the i^{th} data point of the decoder. The second term $\mathbf{KL}(.\|.)$ is a regularizer, the KL-divergence between the encoder and decoder distribution, to minimize the KL-divergence from a chosen prior distribution.

3 METHOD

This study proposes an unsupervised adaptation for HDLSS data classification, which aims to exclusively apply a generative model variational autoencoder (VAE) to investigate dimensionality reduction ability on the HDLSS dataset. In this framework, we divided an unlabeled HDLSS dataset into groups based on the hidden properties of the data. However, conventional classification techniques cannot cope with this HDLSS dataset due to insufficient sample size to build and test a classifier or cross-validation. The proposed unsupervised scheme for HDLSS data classification is illustrated in Figure 2.

Figure 2. Proposed framework

Consider $\mathbb{D} = [X, Y] = [(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k), \ldots, (x_N, y_N)]$ be a $N \times p$ data matrix, where $p \gg N$; where p and N are the number of features and samples, respectively. $x_i \in X$ is the i^{th} observation and the class label is $y_i \in Y$ belonging to C classes. X is mapped a choice of p-dimensional onto a d-dimensional representations $\mathbf{Z}$, $z_i = (z_1, z_2, \ldots, z_d)$, where $d < p$, such that the transformed lower-dimensional representations $z_i = \mathbf{Z}^T x_i$ can preserve the information of the original data. The key aspects of the framework are as follows:

Dimensionality reduction: The first step of the proposed framework is the dimensionality reduction, which receives the HDLSS dataset as input, and then class labels are removed from the dataset. Consequently, the deep learning model VAE is applied to the unlabeled data to project desirable high-dimensional data onto lower-dimensional space. The deep learning technique VAE empowers the method to avoid overfitting.

Clustering: Then, the clustering technique is applied to the obtained transformed low-dimensional space to find the data groups. The exploratory and unsupervised learning nature of the clustering demands efficient to use that would benefit from the combination in the strength of the framework. A clustering technique groups similar data in a cluster, whereas dissimilar data in different clusters. K-means is widely used and one of the prominent data mining techniques for its simplicity. In this study, simple K-means clustering is used. Determining the number of clusters in a dataset is fundamental in K-means clustering, which requires the user to specify the number of clusters K to be generated. There are different methods for identifying the optimal number of clusters in a dataset, including DBSCAN, Xmeans, I-Nice, Elbow, Silhouette, Gap statistic.

Decision making: The last step of the proposed framework is decision making, which validated the clustering results with the original class labels. Assume that sample points from one class form clustered in the same group.

4 EXPERIMENT AND DISCUSSIONS

4.1 Datasets for Experiments

The experiments were conducted on 14 high-dimensional limited-sample size $p \gg N$ datasets obtained from the Arizona State University repository[1]. Table 2 presents the detail of the datasets.

4.2 Experiment Settings

Experiments were designed for the empirical study of DR techniques on the HDLSS dataset. We applied two types of experiments:

1. without dimensionality reduction (WDR), which ensures that all the original features were used for classification, and
2. with dimensionality reduction (DR), where original data space mapped into a new space with a much smaller number of dimensions were used for classification.

In this study, different choices of latent-space were investigated (i.e., 2, 10, 20, 50, 100, 150, 200, 250, ..., 500) to see how the dimensionality of the projected space affects the performance. To evaluate the effectiveness of dimensionality reduction various DR methods were applied, such as VAE, AE, PCA, Kernel PCA, LLE, MDS, Sparse PCA, NMF, Truncated SVD, SE.

Computations were performed using machines with x64-based processor Intel(R) core i7-7700, CPU 3.60 Hz, and 8.0 GB memory. VAE code implementation using the CPU based on Tensorflow and Keras libraries.

[1] `http://featureselection.asu.edu/`

ID	Dataset	Abbrev.	N	p	c	Type
1	ALLAML	ALL	72	7 129	2	continuous, binary
2	CARCINOM	CAR	174	9 182	11	continuous, multi-class
3	CLL_SUB_111	CLL	111	11 340	3	continuous, multi-class
4	GLI_85	GLI	85	22 283	2	continuous, binary
5	GLIOMA	GMA	50	4 434	4	continuous, multi-class
6	NCI9	NCI	60	9 712	9	discrete, multi-class
7	PROSTATE_GE	PROS	102	5 966	2	continuous, binary
8	SMK_CAN_187	SMK	187	19 993	2	continuous, binary
9	TOX_171	TOX	171	5 748	4	continuous, binary
10	ORLRAWS10P	ORL	100	10 304	10	continuous, multi-class
11	PIXRAW10P	PIX	100	10 000	10	continuous, multi-class
12	WARPAR10P	WPAR	130	2 400	10	continuous, multi-class
13	WARPPIE10P	WPIE	210	2 420	10	continuous, multi-class
14	ARCENE	ARC	200	10 000	2	continuous, binary

Table 2. Characteristics of the datasets. ID 1–9 are biological, 10–13 are face image, and 14 is mass-spectrometry dataset (N: number of samples, p: number of features, and c: number of classes).

4.3 VAE Design

For the structure of the VAE, we exhaustedly investigated the best setting, such as the number of intermediate layers, the size of each intermediate layer, batch size, and learning rates. It is found that the network structure of VAE also affects the performance of the feature extraction. In the experiment, VAE is performed on the single intermediate layer (encode) with the following architecture: input encoded onto d-dimensional latent space ($d = z = 2, 10, 20, 50, 100, 200, \ldots, 500$) and reconstructed back to the original dimension. We kept the intermediate dimension as 10 % of the original data space. The network parameter optimized with an 'adam' optimizer, included 'rectified linear units' and batch normalization in the encoding stage, and 'sigmoid' activation in the decoding stage. A parameter scope is performed on batch size 50, 100, 150, and 200; epochs 100, 200, and 300; learning rates 0.005, 0.001, 0.0015, and 0.0025; and warmups (k) 0.01, 0.05, 0.001, and 0.0005. k controls how much the KL-divergence loss contributes to learning. In general, training was relatively stable for many parameter combinations. Ultimately, the best parameter combination based on validation was batch size 100, learning rate 0.0005, and epochs 200. Training stabilized after about 120 epochs.

4.4 Determining Number of Clusters in Dataset

A large variety of clustering methods has been proposed to discover the inherent cluster structure in data. DBSCAN [46], Xmeans [47], and I-nice [49] are popular methods for determining the number of clusters, K. We use these three methods to

determine the K-value for the clustering of this study. Table 3 presents the obtained number of clusters of the algorithms. Results showed that the DBSCAN is inefficient when applied to large-dimensional data. To determine the K value for K-means, we assumed that the number of classes is equal to the number of clusters.

4.5 Evaluation Criteria

The attained results were analyzed in terms of three external cluster evaluation measures: purity [50], rand index (RI) [51], and normalized mutual information (NMI)[52]. Purity is the percent of the total number of objects classified correctly, it is calculated as follows:

$$\text{Purity} = \frac{1}{N}\sum_{1}^{K} \mathbf{max}_j |C_i \cap Y_j| \tag{2}$$

where N is number of objects in the dataset, K is number of clusters, C_i is a cluster in C, and Y_j is the classification which has the **max** count for cluster C_i.

Rand index (RI) is another popular cluster validation index, measures the percentage of correct decisions, it can be defined as Equation (3).

$$\text{RI} = \frac{TP + TN}{TP + FP + FN + TN} \tag{3}$$

where TP and FP are the numbers of true positive and false positive, whereas TN and FN are the numbers of true and false negative, respectively.

Normalized mutual information (NMI) is the mutual information between the clustering and the classification on the shared object membership, with a scaling factor corresponding to the number of object in the respective clusters, can define by Equation (4).

$$\text{NMI} = \frac{I(C_i, Y_i)}{[H(Y_j) + H(C_i)]/2} \tag{4}$$

where $I(C_i, Y_i)$ denotes the mutual information between true assigned class and obtained cluster label, and $H(C_i)$ is the entropy of cluster C_i while information about Y_j classes is available. The range of NMI is between $[0, 1]$. A higher value indicates a better quality of clustering.

4.6 Analysis and Discussions

The attained results were analyzed in terms of the average on different used dimensions. The results of each selected dimension were obtained by the best of 5 runs. Tables 4, 5, and 6 show the summarized achieved averaged values of different dimensionality reduction (DR) techniques for each experimental dataset's purity, RI, and NMI, respectively.

	DBSCAN										I-nice										Xmeans									
Dataset	AE	KPCA	LLE	MDS	SE	SPCA	TSVD	NMF	PCA	VAE	AE	KPCA	LLE	MDS	SE	SPCA	TSVD	NMF	PCA	VAE	AE	KPCA	LLE	MDS	SE	SPCA	TSVD	NMF	PCA	VAE
ALL	1/2	1/1	1/1	1/1	1/3	1/1	1/1	1/2	1/2	1/2	2/3	2/3	2/2	2/2	2/3	2/4	2/3	2/2	2/3	2/3	2/2	2/2	2/3	2/3	2/3	2/3	2/2	2/3	2/2	2/2
CAR	2/8	1/7	1/4	1/2	2/5	1/8	1/7	3/10	2/8	2/8	7/10	7/10	4/8	2/5	5/11	8/12	7/9	10/12	8/11	8/10	6/10	3/5	8/12	2/2	4/12	3/4	2/3	4/11	2/3	6/9
CLL	1/2	1/2	1/4	1/1	1/2	1/2	1/2	1/2	1/2	1/2	2/4	3/5	2/4	2/4	2/5	2/5	2/5	2/4	2/3	2/3	2/4	2/3	2/5	2/2	3/4	2/3	2/2	2/5	2/3	2/3
GLI	1/3	1/3	1/4	1/1	2/6	1/2	1/1	2/3	1/3	2/3	2/4	2/3	2/3	2/4	2/4	2/3	2/4	2/3	2/3	2/3	2/3	2/2	2/2	2/2	2/4	2/2	2/2	2/5	2/2	2/4
GMA	1/2	1/1	1/3	1/2	1/4	1/2	1/1	1/2	1/1	1/2	3/4	2/3	2/4	3/5	3/5	3/5	2/4	2/4	3/4	3/5	2/5	2/2	2/3	2/5	2/7	2/2	2/2	2/3	2/3	2/5
NCI	1/3	1/2	1/2	1/2	1/3	1/2	1/2	1/2	1/2	1/3	6/9	5/8	6/8	7/9	8/10	6/8	7/9	6/9	6/8	6/10	2/5	2/3	2/4	2/2	2/4	2/2	2/2	2/4	2/4	2/5
PROS	1/2	1/1	2/6	1/2	1/3	1/3	1/2	1/2	1/1	1/2	2/4	2/4	2/4	2/5	2/5	2/4	2/5	2/3	3/3	3/4	2/4	2/4	2/6	2/2	2/9	2/4	2/3	2/6	2/3	3/5
SMK	1/2	1/1	1/2	1/5	1/5	1/5	1/1	1/2	1/1	1/2	2/4	2/4	2/4	2/4	2/3	2/4	3/5	2/3	2/4	2/4	2/4	2/3	2/4	2/3	2/5	3/4	2/3	2/5	2/3	3/6
TOX	1/2	1/6	1/1	1/1	1/7	1/4	1/7	1/8	1/6	1/2	3/5	3/5	4/6	2/4	4/6	3/7	3/5	3/5	3/5	3/5	2/5	2/3	3/5	2/2	2/3	3/4	2/2	2/5	2/2	2/3
ORL	2/9	1/7	1/12	1/7	1/8	2/11	3/7	8/9	1/7	3/10	7/11	7/9	7/12	6/11	7/11	8/12	6/11	8/11	7/11	8/11	2/6	2/3	2/10	2/2	6/16	5/7	2/3	2/5	2/3	5/8
PIX	3/10	3/10	2/6	1/8	3/10	7/11	3/10	6/10	4/10	4/8	6/11	6/11	5/8	6/9	6/11	6/12	6/11	7/11	7/11	7/10	4/12	2/9	2/13	3/4	3/4	7/14	3/10	2/14	4/7	7/13
WPAR	2/8	1/9	1/12	1/6	2/9	7/8	1/9	7/10	2/10	4/7	6/9	5/8	5/8	5/9	5/9	5/9	5/9	6/9	5/8	6/9	2/5	2/3	2/3	2/2	2/5	2/3	2/2	2/2	2/2	2/4
WPIE	2/8	1/17	1/17	1/5	1/7	1/6	7/16	1/3	1/8	3/9	7/9	6/9	7/11	4/6	6/8	6/9	6/11	8/9	6/9	8/9	2/8	2/3	2/2	3/5	6/17	3/5	2/3	3/17	2/5	2/7
ARC	1/2	1/5	3/6	1/3	3/4	1/5	1/4	1/7	1/5	1/2	2/4	3/5	3/5	3/5	3/4	3/5	3/5	3/3	3/5	2/3	2/7	3/7	3/16	3/3	3/13	3/8	2/9	7/12	4/7	2/8

Table 3. Summarized results of three cluster number determination methods on each dataset. The values shown in the table are minimum and maximum number of clusters obtained over the applied different number of dimensions, i.e., 2, 10, 20, 50, 100, 200, 300, 400, 500 (min/max).

Table 4 shows the achieved average purity of different DR techniques for each experimental dataset; it is observable that VAE outperformed others. Considering all the experiments, VAE (1.4), AE (3.2), and SPCA (3.6) were ranked from first to third, respectively. KPCA (5.1) and TSVD (5.3) were ranked as fourth and fifth ranks subsequently. It reveals the strength of VAE, preserve more information in lower-dimensional space in the context of HDLSS.

Besides, the average RI of each algorithm over all experiments is listed in Table 5. Among the techniques, VAE (1.3), AE (2.6), and SPCA (3.9) were ranked from first to third in terms of the correct decision. Nonetheless, the superior RI of VAE shows that not only it copes with the HDLSS but also outperforms traditional DR techniques such as LLE, MDS, PCA, KPCA, NMF, SE, TSVD. Based on the reported results in Table 6, VAE (2.3) is ranked as the most normalized mutual information measure. AE (3.1) is ranked second, and the third rank is assigned to SPCA (3.9).

Based on the observations from Tables 4, 5, and 6, it can be seen VAE is robust against the HDLSS dataset. AE, SPCA, and KPCA also perform well, while traditional DR techniques PCA, NMF, LLF, MDS, TSVD, and SE provide quite poor performance.

To assess the importance of the projected space size in the HDLSS problem, we can examine in a little more detail the performances of the 14 datasets with different numbers of dimensions, as shown in Appendix A (Tables A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14). From observation, it is therefore of interest to note that the performance gained with the raise of dimension size. It is impressive that VAE almost always achieved the highest accuracy in used different reduced dimensions. Moreover, it seems that SPCA and MDS are affected by the size of dimensionality and stable for a wide range of dimensions, while other methods (i.e., PCA, KPCA, LLE, SE) typically require relatively more dimensions to obtain good accuracy. It is worthwhile to mention that the use of VAE, AE, SPCA, and KPCA is advantageous compared to other techniques, they can preserve more information in possible higher-dimensional space. Though, analyzing in lower-dimensional space is much easier than in a higher-dimensional space. Noted that 7 out of 14 datasets (i.e., 1, 2, 4, 5, 9, 10, and 11) were provided the best results where respective dimensions $d > N$. So, it is reasonable to try to more latent space concerning the preservation of information that increases the chances of obtaining useful results. Thus, it can be concluded that VAE is providing the best DR for the unsupervised HDLSS data classification in this study; also, AE, SPCA, KPCA can be a competitive choice.

One major limitation of this framework is that when the data has a complex distribution (each class has different distribution). For instance, we assumed the number of clusters is equal to the number of classes, the assumption is not valid when distinctive mini-clusters exist.

Dataset	WDR	AE	KPCA	LLE	MDS	SE	SPCA	TSVD	NMF	PCA	VAE
ALL	70.8	73.4 (3)	72.6 (4)	64.2 (9)	67.4 (8)	56.3 (10)	73.6 (2)	70.8 (5)	68.1 (7)	69.8 (6)	86.1 (1)
CAR	66.7	56.7 (3)	55.1 (5)	47.8 (8)	50.9 (7)	43.6 (9)	57.2 (2)	54.6 (6)	37.2 (10)	56.6 (4)	71.4 (1)
CLL	53.2	55.2 (4)	55.1 (5)	54.1 (8)	53.0 (9)	55.3 (3)	55.7 (2)	54.4 (7)	50.5 (10)	54.6 (6)	61.0 (1)
GLI	64.7	69.8 (5)	69.7 (6)	67.1 (7)	70.5 (4)	65.9 (9)	71.0 (2)	70.9 (3)	61.6 (10)	66.8 (8)	71.4 (1)
GMA	60.0	62.6 (2)	57.0 (6)	46.0 (10)	55.6 (7)	48.0 (9)	60.7 (4)	62.0 (3)	52.7 (8)	58.5 (5)	65.6 (1)
NCI	43.3	50.1 (2)	42.1 (4.5)	42.1 (4.5)	40.0 (8)	37.1 (9)	44.6 (3)	41.7 (6)	35.2 (10)	40.8 (7)	50.9 (1)
PROS	57.8	58.5 (4)	58.4 (6.5)	57.8 (8)	58.5 (4)	57.1 (9)	58.5 (4)	58.4 (6.5)	55.8 (10)	58.6 (2)	59.7 (1)
SMK	51.9	55.1 (8)	55.7 (3)	58.2 (1)	56.4 (2)	52.6 (10)	55.6 (4)	55.4 (5.5)	54.3 (9)	55.4 (5.5)	55.3 (7)
TOX	44.4	51.3 (2)	45.1 (5.5)	39.8 (9)	43.8 (7)	40.5 (8)	46.3 (4)	48.2 (3)	36.4 (10)	45.1 (5.5)	56.9 (1)
ORL	76.0	76.1 (2)	72.4 (4.5)	60.5 (8)	72.4 (4.5)	59.5 (9)	73.7 (3)	72.0 (6)	47.0 (10)	68.2 (7)	78.7 (1)
PIX	81.0	86.4 (3)	77.4 (7)	57.3 (9)	81.4 (6)	59.8 (8)	86.8 (2)	81.8 (4.5)	54.3 (10)	81.8 (4.5)	88.0 (1)
WPAR	32.3	37.4 (2)	27.5 (6)	31.2 (4)	27.2 (7)	27.1 (8.5)	27.8 (5)	23.8 (10)	31.6 (3)	27.1 (8.5)	39.1 (1)
WPIE	31.0	58.8 (2)	30.1 (7)	36.7 (4)	29.8 (9)	27.4 (10)	29.9 (8)	30.6 (5)	38.9 (3)	30.2 (6)	62.2 (1)
ARC	34.0	64.8 (3)	64.9 (2)	55.3 (9)	62.9 (7)	55.0 (10)	63.2 (6)	63.8 (4)	59.6 (8)	63.6 (5)	66.3 (1)

Table 4. Average purity (in %) of different dimensions of different techniques on datasets. Higher value is better and values in parentheses indicate the rank of algorithm.

Dataset	WDR	AE	KPCA	LLE	MDS	SE	SPCA	TSVD	NMF	PCA	VAE
ALL	58.1	63.9 (2)	59.7 (4)	56.3 (7)	55.9 (9)	50.5 (10)	60.6 (3)	58.4 (5)	56.1 (8)	57.7 (6)	76.8 (1)
CAR	91.2	89.3 (3)	87.7 (6)	80.3 (8)	87.0 (7)	76.5 (10)	89.4 (2)	88.0 (5)	78.4 (9)	89.0 (4)	93.3 (1)
CLL	55.3	57.6 (3.5)	57.2 (5.5)	52.2 (9)	56.6 (7)	52.4 (8)	57.6 (3.5)	57.2 (5.5)	47.5 (10)	57.7 (2)	58.7 (1)
GLI	53.8	68.2 (1)	57.3 (7)	56.1 (8)	58.7 (3)	54.6 (10)	58.5 (4)	58.3 (5)	57.4 (6)	55.7 (9)	58.8 (2)
GMA	73.1	74.6 (3.5)	73.7 (6)	57.8 (10)	70.1 (7)	59.5 (9)	74.6 (3.5)	73.8 (5)	64.8 (8)	74.7 (2)	75.4 (1)
NCI	80.7	82.9 (2)	80.9 (6)	79.8 (8)	81.0 (5)	76.6 (9)	82.7 (3)	82.6 (4)	56.4 (10)	80.0 (7)	85.1 (1)
PROS	50.7	51.0 (5)	51.0 (5)	51.2 (2)	51.0 (5)	50.9 (8.5)	51.0 (5)	50.9 (8.5)	50.4 (10)	51.0 (5)	51.6 (1)
SMK	49.8	50.7 (2)	50.4 (6.5)	51.5 (1)	50.6 (3)	50.0 (10)	50.4 (6.5)	50.3 (8.5)	50.5 (4.5)	50.3 (8.5)	50.5 (4.5)
TOX	67.9	69.1 (2)	68.2 (3)	57.8 (9)	66.0 (7)	59.1 (8)	68.1 (4)	67.8 (5)	46.2 (10)	67.7 (6)	72.2 (1)
ORL	93.6	93.4 (3.5)	93.4 (3.3)	85.9 (9)	92.7 (6)	86.5 (8)	93.5 (2)	92.8 (5)	80.7 (10)	91.9 (7)	94.3 (1)
PIX	95.1	96.4 (3)	94.2 (7)	83.0 (10)	95.2 (6)	85.6 (8)	96.5 (2)	95.4 (5)	84.7 (9)	95.5 (4)	96.8 (1)
WPAR	83.9	83.3 (2)	82.6 (4)	72.0 (10)	82.7 (3)	78.0 (8)	82.1 (6.5)	82.1 (6.5)	74.3 (9)	82.3 (5)	85.2 (1)
WPIE	82.3	87.3 (2)	83.2 (3.5)	78.2 (8)	83.2 (3.5)	76.6 (9)	82.8 (5)	82.4 (7)	74.6 (10)	82.6 (6)	90.1 (1)
ARC	54.9	53.8 (3)	54.2 (2)	50.5 (9)	53.3 (7)	50.3 (10)	53.4 (6)	53.7 (4)	51.9 (8)	53.6 (5)	55.1 (1)

Table 5. Average RI (in %) of different dimensions of different techniques on datasets. Higher value is better and values in parentheses indicate the rank of algorithm.

4.7 Run-Time

The average run-time of different applied dimensions of each dimensionality reduction method on each dataset is provided in Table 7. It clearly illustrates that AE and VAE are slower and computationally expensive than the corresponding methods for training the network, which lasted more than 2× to 3×. It can be seen that VAE is faster than AE to achieve selected (suitable) dimensionality reduction. Besides, KPCA, LLE, MDS, SE, SPCA, TSVD, NMF, and PCA were not much different in running time. Furthermore, VAE and AE consumed more run-time compared to other methods but was provided the best performance.

Dataset	WDR	AE	KPCA	LLE	MDS	SE	SPCA	TSVD	NMF	PCA	VAE
ALL	.090	.148 (3)	.139 (4)	.118 (6)	.078 (9)	.052 (10)	.149 (2)	.114 (7)	.081 (8)	.119 (5)	.446 (1)
CAR	.322	.590 (3.5)	.574 (6)	.504 (8)	.523 (7)	.457 (9)	.593 (2)	.576 (5)	.341 (10)	.590 (3.5)	.723 (1)
CLL	.187	.258 (3)	.253 (5)	.145 (9)	.231 (7)	.181 (8)	.260 (2)	.245 (6)	.132 (10)	.272 (1)	.256 (4)
GLI	.197	.147 (5)	.205 (2)	.090 (7)	.129 (6)	.033 (10)	.174 (3)	.217 (1)	.074 (8)	.049 (9)	.159 (4)
GMA	.491	.525 (2)	.540 (1)	.287 (10)	.456 (7)	.332 (9)	.517 (4)	.506 (5)	.378 (8)	.520 (3)	.487 (6)
NCI	.435	.492 (2)	.419 (6)	.424 (5)	.396 (9)	.404 (8)	.459 (3)	.457 (4)	.359 (10)	.417 (7)	.520 (1)
PROS	.019	.040 (5)	.023 (8.5)	.041 (4)	.023 (8.5)	.060 (1)	.023 (8.5)	.023 (8.5)	.043 (3)	.024 (6)	.049 (2)
SMK	.001	.010 (5.5)	.009 (7)	.033 (1)	.012 (3)	.010 (5.5)	.008 (9)	.008 (9)	.019 (2)	.008 (9)	.011 (4)
TOX	.164	.318 (2)	.239 (6)	.195 (8)	.242 (5)	.152 (9)	.259 (4)	.272 (3)	.133 (10)	.235 (7)	.355 (1)
ORL	.849	.820 (4)	.822 (2.5)	.708 (8)	.803 (5)	.645 (9)	.829 (1)	.798 (6)	.544 (10)	.781 (7)	.822 (2)
PIX	.902	.904 (2)	.863 (7)	.696 (9)	.871 (5)	.709 (8)	.910 (1)	.870 (6)	.637 (10)	.877 (4)	.901 (3)
WPAR	.288	.372 (2)	.230 (9)	.302 (4)	.241 (6)	.231 (8)	.246 (5)	.206 (10)	.306 (3)	.236 (7)	.415 (1)
WPIE	.328	.514 (2)	.316 (5.5)	.405 (3)	.310 (8)	.245 (10)	.316 (5.5)	.308 (9)	.386 (4)	.314 (7)	.659 (1)
ARC	.091	.073 (3)	.080 (2)	.022 (9)	.059 (7)	.011 (10)	.063 (5.5)	.069 (4)	0.038 (8)	.063 (5.5)	.090 (1)

WDR: without dimensionality reduction; AE: autoencoder; KPCA: kernel PCA; LLE: locally linear embedding; MDS: multi-dimensional scaling; SE: spectral embedding; SPCA: sparse PCA; TSVD: truncated singular value decomposition; NMF: non-negative matrix factorization; PCA: principal component analysis; VAE: variational autoencoder

Table 6. Average NMI of different dimensions of different techniques on datasets. Higher value is better and values in parentheses indicate the rank of algorithm.

4.8 Statistical Analysis

In this section, we examined two statistical significance tests deemed most appropriate for the multiple-methods evaluation. We carried the nonparametric sign test and Friedman test for hypothesis testing.

Dataset	AE	KPCA	LLE	MDS	SE	SPCA	TSVD	NMF	PCA	VAE
ALL	66.1	11.4	11.0	10.5	11.5	13.0	11.7	11.8	12.6	32.0
CAR	71.4	14.6	12.4	12.3	13.6	15.1	12.4	12.3	13.6	38.4
CLL	75.0	18.3	16.1	22.8	22.4	22.5	18.5	18.1	18.4	42.0
GLI	77.9	25.4	25.1	24.6	24.7	24.4	26.8	25.7	24.7	44.9
GMA	56.7	8.4	9.6	11.3	10.9	10.6	9.2	8.5	10.4	27.7
NCI	62.4	13.4	13.2	13.4	13.5	13.9	14.0	14.5	13.3	33.4
PROS	58.5	11.7	10.3	10.8	11.6	12.6	11.6	11.5	12.6	27.5
SMK	76.4	24.5	23.2	24.4	24.3	24.3	25.1	23.3	23.3	45.4
TOX	57.4	11.1	10.1	10.2	11.9	12.3	11.2	11.6	12.4	29.4
ORL	64.2	19.9	17.7	21.8	21.4	20.7	18.5	17.5	17.7	40.6
PIX	61.1	18.9	14.4	17.8	16.5	19.6	14.6	14.7	14.9	40.1
WPAR	45.3	9.0	10.6	10.4	12.7	12.1	12.9	8.5	9.1	28.3
WPIE	49.4	9.3	11.3	11.3	13.6	13.3	12.3	9.5	9.4	30.4
ARC	71.4	22.6	23.0	23.5	21.9	22.8	21.3	21.8	21.6	47.4

Table 7. Average run-time of different reduced dimensions of different methods for each dataset (in seconds). The values shown in the table are the average of applied different dimensions, i.e., 2, 10, 20, 50, 100, 200, 300, 400, and 500.

4.8.1 Sign Test

Figure 3 illustrates a statistical comparison of VAE over state-of-the-art techniques. The nonparametric test, right-tailed sign test is carried out in the significance level $\alpha = 0.05$ (i.e., 95 % significance level). In the figure, for each metric, the first ten bars exhibit the z-value (test statistic value) for VAE against other techniques, whereas the eleventh bar presents the **z-ref** value. If the calculated z-value is greater than the **z-ref** value, then it indicates that the observed performance of VAE against the corresponding technique is statistically significant. From Figure 3, it is clear that the results obtained by VAE are significantly better than without DR and traditional DR techniques, although NMI seems quite poor for WDR and SPCA.

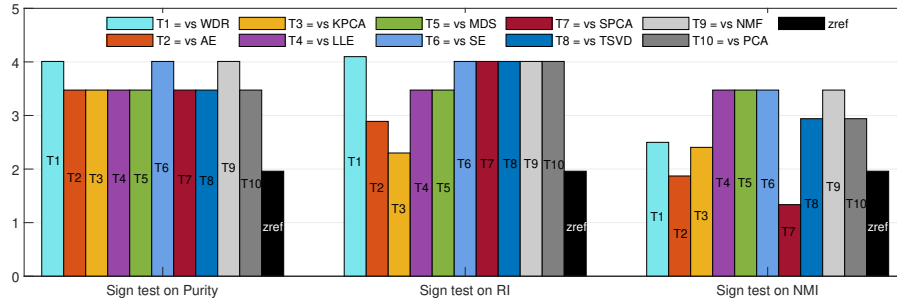

Figure 3. Sign test of VAE on the used 14 experimental datasets

4.8.2 Friedman Test

The Friedman test is used to assess there are any statistically significant differences between the distributions of methods. The p-values (**.000** $<$ .05) for this test are very small. Therefore, it is plausible that the eleven methods are significantly different. From Table 8, we have sufficient evidence to conclude a statistically significant difference between VAE and methods. For pairwise comparisons, we observed there were no significant differences between VAE and AE, SPCA.

5 CONCLUSION

This paper motivates the necessity of adopting moderate-dimensionality reduction and an unsupervised framework for high-dimension limited-sample size (HDLSS) data analysis. It proposes an unsupervised framework to deal with the classification of HDLSS data. The proposed method attempts to project the high-dimensional

Hypothesis Test Summary						Pairwise comparisons				
Friedman test on	**Null Hypothesis**	**Test Statistic**	**Kendall's W**	**Sig.**	**Decision**	**Sample Pair**	**Test Statistic**	**Std. Test Statistic**	**Sig.**	**Adj. Sig.**
Purity	The distributions of WDR, AE, KPCA, LLE, MDS, SE, SPCA, TSVD, NMF, PCA, and VAE are the same.	74.995	0.536	**.000**	Reject the Null hypothesis	T1	5.071	4.046	.000	**.003**
						T2	1.857	1.994	.046	1.000
						T3	4.214	3.362	.001	**.043**
						T4	6.214	4.957	.000	**.000**
						T5	5.607	4.473	.000	**.000**
						T6	7.964	6.353	.000	**.000**
						T7	2.500	1.994	.046	1.000
						T8	4.286	3.149	.001	**.035**
						T9	7.786	6.211	.000	**.000**
						T10	4.786	3.818	.000	**.007**
RI	The distributions of WDR, AE, KPCA, LLE, MDS, SE, SPCA, TSVD, NMF, PCA, and VAE are the same.	86.706	0.619	**.000**	Reject the Null hypothesis	T1	4.964	3.960	.000	**.004**
						T2	1.643	1.311	.190	1.000
						T3	3.964	3.162	.002	.086
						T4	7.179	5.727	.000	**.000**
						T5	4.786	3.818	.000	**.007**
						T6	8.429	6.724	.000	**.000**
						T7	2.964	2.365	.018	.993
						T8	4.679	0.541	.000	**.012**
						T9	8.214	6.553	.000	**.000**
						T10	4.643	3.704	.000	**.012**
NMI	The distributions of WDR, AE, KPCA, LLE, MDS, SE, SPCA, TSVD, NMF, PCA, and VAE are the same.	47.282	0.338	**.000**	Reject the Null hypothesis	T1	3.536	2.821	.005	**.264**
						T2	0.679	0.541	.588	1.000
						T3	2.857	2.279	.023	1.000
						T4	4.321	3.447	.001	**.031**
						T5	4.571	3.647	.000	**.015**
						T6	6.286	5.014	.000	**.000**
						T7	1.643	1.311	.190	1.000
						T8	3.643	2.906	.004	.201
						T9	5.393	4.302	.000	**.001**
						T10	3.607	2.878	.004	.220

Asymptotic significances (2-sided tests) are displayed. The significance level is 0.05.

Table 8. Friedman test results for different methods. T1 = VAE vs. WDR; T2 = VAE vs. AE; T3 = VAE vs. KPCA; T4 = VAE vs. LLE; T5 = VAE vs. MDS; T6 = VAE vs. SE; T7 = VAE vs. SPCA; T8 = VAE vs. TSVD; T9 = VAE vs. NMF; T10 = VAE vs. PCA.

data onto lower-dimensional space using variational autoencoder (VAE), then clustering is applied to the obtained lower-dimensional latent-space to find the groups and classify input data. The deep learning approach VAE enables the framework to avoid overfitting. To evaluate the method fourteen HDLSS datasets and three evaluation criteria were applied. Also, an empirical comparison is shown between VAE and state-of-the-art DR techniques. The results of the experiment demonstrated the effectiveness of the approach. In particular, experimentally investigated that dimension reduction of VAE is better than traditional techniques in the context of HDLSS data classification.

An effective and efficient DR method is essential for HDLSS data analysis. HDLSS data classification severe overfitting and high-variance gradients, whereas an unsupervised framework proved to be a good alternative. In contrast to the traditional DR method while use VAE can reduce the dimension as suitable from the HDLSS data that enhances the performance. This study combines the advantages of both unsupervised DR and unsupervised classification. A future line of this research is to study what kind of encoders and decoders are best suited for the

HDLSS problem. Another interesting future line of research will be finding an efficient dimension selection method (determining moderate d from p). Also, we are interested in designing a general framework that works on both unsupervised and semi-supervised settings. Finally, the reliability of the HDLSS data classification can increase in the meta or ensemble model.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 61972261, 61473194).

APPENDIX

Detail performances of the 14 datasets with different numbers of dimensions in Tables A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14.

REFERENCES

[1] Hastie, T.—Tibshirani, R.—Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition. Springer Series in Statistics, 2009, doi: 10.1007/978-0-387-84858-7.

[2] Li, Y.—Li, T.—Liu, H.: Recent Advances in Feature Selection and Its Applications. Knowledge and Information Systems, Vol. 53, 2017, No. 3, pp. 551–577, doi: 10.1007/s10115-017-1059-8.

[3] Kuncheva, L. I.—Rodríguez, J. J.: On Feature Selection Protocols for Very Low-Sample-Size Data. Pattern Recognition, Vol. 81, 2018, pp. 660–673, doi: 10.1016/j.patcog.2018.03.012.

[4] Kuncheva, L. I.—Matthews, C. E.—Arnaiz-González, Á.—Rodríguez, J. J.: Feature Selection from High-Dimensional Data with Very Low Sample Size: A Cautionary Tale. 2020, arXiv: 2008.12025v1.

[5] Köppen, M.: The Curse of Dimensionality. 5$^{\text{th}}$ Online World Conference on Soft Computing in Industrial Applications (WSC5), 2000.

[6] Lv, J.: Impacts of High Dimensionality in Finite Samples. The Annals of Statistics, Vol. 41, 2013, No. 4, pp. 2236–2262, doi: 10.1214/13-aos1149.

[7] Pearson, K.: On Lines and Planes of Closest Fit to Systems of Points in Space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, Vol. 2, 1901, No. 11, pp. 559–572, doi: 10.1080/14786440109462720.

[8] Hotelling, H.: Analysis of a Complex of Statistical Variables into Principal Components. Journal of Educational Psychology, Vol. 24, 1933, No. 6, pp. 417–441, doi: 10.1037/h0070888.

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	70.8	72.2	72.2	72.2	73.6	74.8	76.4	75.0	73.6	54.0	60.6	57.0	70.0	63.4	62.0	73.7	73.7	60.6	.135	.124	.130	.142	.135	.131	.212	.204	.121
KPCA	**73.6**	73.6	72.2	70.8	x	x	x	x	x	**60.6**	60.6	59.3	58.1	x	x	x	x	x	**.155**	.137	.139	.125	x	x	x	x	x
LLE	55.6	**84.7**	59.7	56.9	x	x	x	x	x	49.9	**73.7**	51.2	50.3	x	x	x	x	x	.020	**.348**	.018	.087	x	x	x	x	x
MDS	58.3	70.8	70.8	70.8	63.9	65.3	75.0	66.7	65.3	50.7	58.1	58.1	58.1	53.2	54.0	62.0	54.9	54.0	.017	.125	.125	.125	.027	.039	.162	.001	.079
SE	54.2	61.1	59.7	50.0	x	x	x	x	x	49.6	51.8	51.2	49.3	x	x	x	x	x	.016	.055	.002	.138	x	x	x	x	x
SPCA	72.2	73.6	73.6	73.6	75.0	73.6	73.6	75.0	72.2	59.3	60.6	60.6	60.6	62.0	60.6	60.6	62.0	59.3	.130	.145	.155	.155	.171	.155	.145	.171	.115
TSVD	72.2	76.4	68.1	66.7	x	x	x	x	x	59.3	63.4	55.9	54.9	x	x	x	x	x	.139	.190	.059	.068	x	x	x	x	x
NMF	72.2	66.7	75.0	66.7	66.7	69.4	66.7	65.3	63.9	59.3	54.9	62.0	54.9	54.9	57.0	54.9	54.0	53.2	.139	.068	.163	.068	.068	.067	.068	.039	.047
PCA	**73.6**	65.3	75.0	65.3	x	x	x	x	x	**60.6**	54.0	62.0	54.0	x	x	x	x	x	**.155**	.079	.162	.079	x	x	x	x	x
VAE	72.2	81.9	**76.4**	**91.7**	**91.7**	**88.9**	**93.1**	**93.1**	**86.1**	59.3	70.0	**63.4**	**84.5**	**84.5**	**80.0**	**86.9**	**86.9**	**75.7**	.151	.299	**.190**	**.564**	**.573**	**.554**	**.675**	**.615**	**.388**

Table A1. Results of different dimensions of different techniques for the ALLAML dataset (72 observations, 7 129 features, 2 classes)

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	43.1	50.6	51.7	53.4	64.4	62.1	62.1	62.1	60.9	85.1	88.1	89.4	90.7	90.3	90.1	90.1	91.2	89.1	.356	.582	.448	.633	.574	.712	.673	.623	.711
KPCA	32.8	55.7	68.4	58.0	60.3	x	x	x	x	84.0	88.9	89.1	87.9	88.6	x	x	x	x	.354	.637	.658	.609	.610	x	x	x	x
LLE	**44.8**	**70.1**	55.2	37.9	31.0	x	x	x	x	85.1	92.0	81.6	69.7	72.9	x	x	x	x	**.498**	**.724**	.611	.400	.290	x	x	x	x
MDS	39.1	51.7	51.1	49.4	50.6	49.4	54.0	51.7	60.9	86.2	88.6	87.9	86.0	87.9	82.9	88.1	86.7	89.0	.349	.509	.532	.498	.570	.519	.567	.571	.595
SE	43.1	57.5	44.8	39.7	32.8	x	x	x	x	85.2	90.3	78.5	72.9	55.5	x	x	x	x	.494	.643	.467	.409	.273	x	x	x	x
SPCA	34.5	57.5	53.4	60.3	62.1	63.8	62.1	63.8	57.5	84.3	90.2	87.8	90.7	91.3	91.1	90.6	89.7	88.9	.348	.600	.579	.619	.653	.662	.649	.631	.595
TSVD	29.9	57.5	69.0	57.5	59.2	x	x	x	x	80.8	89.4	91.5	89.5	89.0	x	x	x	x	.296	.622	.695	.634	.632	x	x	x	x
NMF	28.7	64.9	64.9	53.4	27.6	22.4	23.0	25.9	24.1	82.9	**92.3**	91.2	83.7	58.3	72.4	70.1	76.4	78.5	.279	.685	**.721**	.542	.226	.122	.156	.167	.176
PCA	33.9	66.7	57.5	62.6	62.1	x	x	x	x	83.3	91.8	88.8	90.4	90.6	x	x	x	x	.372	.676	.613	.660	.631	x	x	x	x
VAE	**44.8**	64.4	**70.7**	**74.1**	**80.5**	**81.0**	**75.9**	**77.0**	**74.1**	**86.3**	90.0	**93.5**	**93.9**	**95.9**	**95.8**	**94.6**	**95.4**	**94.4**	.486	.630	.705	**.741**	**.826**	**.792**	**.769**	**.785**	**.776**

Table A2. Results of different dimensions of different techniques for the CARCINOM dataset (174 observations, 9 182 features, 11 classes)

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	52.3	50.1	51.4	55.0	56.8	**60.4**	60.4	55.9	54.1	57.4	58.2	57.7	57.6	58.2	58.2	56.9	55.2	59.4	.265	.298	.216	.289	.298	.159	**.287**	.216	.292
PCA	52.3	61.3	54.1	53.2	55.0	x	x	x	x	57.4	58.8	57.5	56.9	55.6	x	x	x	x	.213	.319	.248	.192	.296	x	x	x	x
LLE	57.7	51.4	55.9	51.4	54.1	x	x	x	x	57.2	52.6	51.1	52.3	47.9	x	x	x	x	.255	.250	.118	.046	.054	x	x	x	x
MDS	40.5	49.5	51.4	52.3	56.8	55.9	60.4	55.0	55.0	53.0	55.8	56.7	55.2	58.2	**58.5**	57.3	57.6	57.2	.020	.146	.149	**.379**	.277	**.378**	.268	**.250**	.215
SE	**58.6**	**67.6**	49.5	55.0	45.9	x	x	x	x	57.7	**60.6**	44.9	57.6	41.4	x	x	x	x	.272	.254	.082	.250	.046	x	x	x	x
SPCA	56.8	55.9	55.0	55.0	58.6	55.0	55.0	55.0	55.0	**58.1**	58.0	57.6	57.6	56.9	57.6	57.8	57.6	57.6	.272	.267	.250	.250	.285	.250	.263	**.250**	.250
TSVDAE	56.8	51.4	55.0	54.1	55.0	x	x	x	x	57.6	55.3	57.6	57.7	57.7	x	x	x	x	.248	.184	.250	.260	.281	x	x	x	x
NMF	56.8	55.0	52.3	50.5	45.9	45.9	50.5	47.7	49.5	57.6	56.7	55.2	44.3	41.6	42.0	43.2	43.3	44.1	.248	.267	**.328**	.068	.046	.046	.067	.046	.077
PCA	56.8	55.0	54.1	51.4	55.9	x	x	x	x	57.6	57.6	57.4	57.1	58.9	x	x	x	x	.248	.250	.222	.196	**.443**	x	x	x	x
VAE	55.0	61.3	**62.2**	**62.2**	**63.1**	59.5	**61.3**	**60.4**	**64.0**	57.6	59.4	**60.4**	**59.7**	**59.8**	53.4	**58.9**	**58.1**	**60.7**	**.275**	**.322**	.220	.260	.319	.083	.285	.229	**.311**

Table A3. Results of different dimensions of different techniques for the CLL_SUB dataset (111 observations, 11 340 features, 3 classes)

Bold, shade, and x indicate best result of individual reduced dimensions, best among the applied reduced dimensions, and algorithms failed to provide reduced dimensions, respectively.

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	71.8	**72.9**	58.0	69.4	**70.6**	70.6	**72.9**	**72.9**	69.4	62.4	**68.6**	**76.2**	60.1	**68.2**	**78.9**	**69.8**	**63.6**	**66.2**	.181	.216	.219	.140	.109	.123	.083	.105	.148
KPCA	69.4	69.4	68.2	71.8	x	x	x	x	x	57.0	57.0	56.1	59.0	x	x	x	x	x	.239	.239	.228	.112	x	x	x	x	x
LLE	70.6	57.6	65.9	**74.1**	x	x	x	x	x	58.0	50.6	54.5	**61.2**	x	x	x	x	x	.101	.089	.000	.171	x	x	x	x	x
MDS	75.3	54.1	**72.9**	69.4	**70.6**	**78.8**	69.4	71.8	71.8	62.4	49.7	60.1	57.0	58.0	66.2	57.0	59.0	59.0	**.302**	.004	.145	.061	**.251**	**.217**	.008	.095	.076
SE	68.2	63.5	63.5	68.2	x	x	x	x	x	56.1	53.1	53.1	56.1	x	x	x	x	x	.066	.001	.060	.003	x	x	x	x	x
SPCA	68.2	**72.9**	68.2	68.2	69.4	76.5	70.6	68.2	**76.5**	56.1	60.1	56.1	56.1	57.0	63.6	58.0	56.1	63.6	.047	.173	.047	.228	.239	.316	.059	**.228**	**.231**
TSVD	69.4	70.6	70.6	72.9	x	x	x	x	x	57.0	58.0	58.0	60.1	x	x	x	x	x	.239	**.251**	.101	**.275**	x	x	x	x	x
NMF	72.9	67.1	**72.9**	56.1	58.0	56.1	56.1	57.0	58.0	60.1	55.3	60.1	56.1	58.0	56.1	56.1	57.0	58.0	.213	.183	.083	.022	.033	.033	.022	.008	.071
PCA	57.6	68.2	70.6	70.6	x	x	x	x	x	50.6	56.1	58.0	58.0	x	x	x	x	x	.020	.047	.059	.071	x	x	x	x	x
VAE	**76.5**	70.6	70.6	67.1	**70.6**	70.6	**72.9**	70.6	72.9	**63.6**	58.0	58.0	55.3	58.0	58.0	60.1	58.0	60.1	.213	**.251**	**.251**	.217	.035	.071	**.107**	.071	.213

Table A4. Results of different dimensions of different techniques for the GLI_85 dataset (85 observations, 22 283 features, 2 classes)

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	60.0	62.0	60.0	64.0	62.0	64.0	66.0	66.0	60.0	73.6	74.2	73.1	77.2	74.3	75.6	72.7	75.6	74.9	**.556**	.564	.495	.517	.508	**.584**	.499	.511	.490
KPCA	58.0	52.0	54.0	64.0	x	x	x	x	x	**75.6**	71.5	73.3	74.2	x	x	x	x	x	.501	**.576**	**.573**	.512	x	x	x	x	x
LLE	54.0	40.0	44.0	x	x	x	x	x	x	71.7	54.9	46.8	x	x	x	x	x	x	.476	.213	.172	x	x	x	x	x	x
MDS	46.0	54.0	42.0	56.0	60.0	62.0	58.0	68.0	54.0	71.3	72.3	48.7	71.1	73.1	73.6	72.7	75.5	72.3	.397	.467	.255	.479	.485	.494	**.524**	.490	**.518**
SE	56.0	46.0	42.0	x	x	x	x	x	x	73.2	52.5	52.8	x	x	x	x	x	x	.497	.277	.221	x	x	x	x	x	x
SPCA	60.0	56.0	56.0	58.0	**64.0**	**66.0**	66.0	54.0	**66.0**	73.1	74.9	72.7	75.6	74.4	75.4	**75.8**	74.3	**75.4**	.500	.489	.568	.501	**.509**	.524	.511	**.544**	.509
TSVD	60.0	60.0	**66.0**	62.0	x	x	x	x	x	73.6	73.6	**74.5**	73.6	x	x	x	x	x	.491	.484	.539	.508	x	x	x	x	x
NMF	56.0	60.0	50.0	34.0	52.0	54.0	54.0	54.0	60.0	64.9	70.4	63.8	31.5	65.1	71.4	71.1	71.8	73.1	.245	.537	.271	.125	.436	.482	.414	.429	.464
PCA	56.0	62.0	58.0	58.0	x	x	x	x	x	74.9	73.6	74.2	75.9	x	x	x	x	x	.489	.538	.538	.514	x	x	x	x	x
VAE	**62.0**	**68.0**	60.0	**72.0**	62.0	**66.0**	**68.0**	**70.0**	62.0	74.1	**75.0**	73.6	**80.1**	**74.8**	**76.7**	73.6	**77.2**	73.8	.508	.493	.458	**.545**	.465	.469	.424	.535	.486

Table A5. Results of different dimensions of different techniques for the GLIOMA dataset (50 observations, 4 434 features, 4 classes)

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	41.7	46.7	51.7	**51.7**	53.3	55.0	**53.3**	50.0	48.3	81.7	84.3	81.3	84.3	82.9	81.9	82.7	82.9	84.3	.436	.462	.502	.523	.486	.553	.481	**.542**	.447
KPCA	36.7	41.7	48.3	41.7	x	x	x	x	x	82.4	80.5	81.9	78.9	x	x	x	x	x	.396	.407	.471	.404	x	x	x	x	x
LLE	41.7	**51.7**	41.7	33.3	x	x	x	x	x	82.7	84.0	82.9	69.4	x	x	x	x	x	.426	.542	.387	.343	x	x	x	x	x
MDS	28.3	31.7	43.3	36.7	46.7	40.0	41.7	46.7	45.0	81.7	79.7	81.7	80.8	84.3	81.9	80.4	80.7	78.3	.298	.314	.436	.353	.493	.427	.398	.441	.404
SE	38.3	38.3	40.0	31.7	x	x	x	x	x	82.0	81.9	76.4	66.1	x	x	x	x	x	.421	.467	.438	.291	x	x	x	x	x
SPCA	35.0	41.7	43.3	43.3	43.3	48.3	**53.3**	48.3	45.0	81.4	84.3	80.7	81.3	82.1	82.4	84.0	84.4	83.7	.385	.482	.451	.412	.440	.449	.512	.523	.476
TSVD	38.3	43.3	43.3	41.7	x	x	x	x	x	**82.8**	83.1	81.5	82.8	x	x	x	x	x	.411	.502	.465	.451	x	x	x	x	x
NMF	**43.3**	**51.7**	**58.3**	28.3	26.7	26.7	30.0	26.7	25.0	82.2	81.5	83.3	37.5	33.0	34.7	47.6	58.0	49.9	**.462**	**.553**	**.600**	.303	.267	.243	.285	.266	.255
PCA	35.0	48.3	45.0	35.0	x	x	x	x	x	81.9	81.2	79.0	77.7	x	x	x	x	x	.379	.476	.447	.366	x	x	x	x	x
VAE	**43.3**	48.3	48.3	**51.7**	**55.0**	**56.7**	**53.3**	**51.7**	**50.0**	80.9	**85.5**	**85.0**	**85.9**	**85.5**	**86.9**	**85.5**	**85.7**	**85.3**	.448	.519	.514	**.543**	**.532**	**.588**	**.523**	.533	**.479**

Table A6. Results of different dimensions of different techniques for the NCI9 dataset (60 observations, 9 712 features, 9 classes)

Bold, shade, and x indicate best result of individual reduced dimensions, best among the applied reduced dimensions, and algorithms failed to provide reduced dimensions, respectively.

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	53.9	55.9	57.8	59.8	**62.7**	58.8	59.8	61.7	55.9	50.7	49.6	51.1	51.4	50.5	50.7	51.4	**52.8**	51.1	.007	.019	.010	.060	.034	.097	.055	.059	.018
KPCA	**58.8**	58.8	56.9	58.8	58.8	x	x	x	x	**51.1**	51.1	50.5	51.1	51.1	x	x	x	x	.026	.026	.014	.026	.026	x	x	x	x
LLE	55.9	**66.7**	**58.8**	53.9	53.9	x	x	x	x	50.2	**55.1**	**51.1**	49.8	49.8	x	x	x	x	.010	**.114**	.026	.047	.007	x	x	x	x
MDS	57.8	57.8	**58.8**	58.8	58.8	58.8	58.8	58.8	57.8	50.7	50.7	**51.1**	51.1	51.1	51.1	51.1	51.1	50.7	.018	.018	.026	.026	.026	.027	.026	.027	.019
SE	**58.8**	64.7	55.9	53.9	52.0	x	x	x	x	**51.1**	53.9	50.2	49.8	49.6	x	x	x	x	.055	.169	.029	.009	.036	x	x	x	x
SPCA	**58.8**	57.8	57.8	58.8	58.8	57.8	58.8	58.8	58.8	**51.1**	50.7	50.7	51.1	51.1	50.7	51.1	51.1	51.1	.026	.018	.018	.026	.026	.019	.026	.026	.026
TSVD	**58.8**	58.8	57.8	58.8	57.8	x	x	x	x	**51.1**	51.1	50.7	51.1	50.7	x	x	x	x	.026	.026	.019	.026	.019	x	x	x	x
NMF	**58.8**	57.8	52.0	50.0	52.0	53.9	58.8	58.8	**59.8**	**51.1**	50.7	49.6	49.5	49.6	49.8	51.1	51.1	**51.4**	**.027**	.057	**.036**	.034	.036	.004	.056	.056	**.078**
PCA	**58.8**	58.8	**58.8**	58.8	57.8	x	x	x	x	**51.1**	51.1	**51.1**	51.1	50.7	x	x	x	x	.026	.026	.026	.026	.019	x	x	x	x
VAE	55.9	56.9	55.9	**62.7**	**62.7**	**59.8**	**61.8**	**62.7**	58.8	50.2	50.5	50.2	**52.8**	**52.8**	**51.4**	**52.3**	**52.8**	51.1	.010	.013	.010	**.076**	**.051**	**.113**	**.067**	**.076**	.026

Table A7. Results of different dimensions of different techniques for the PROSTATE_GE dataset (102 observations, 5 966 features, 2 classes)

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	56.7	58.8	55.6	56.7	56.7	56.1	52.4	51.9	50.8	50.6	51.1	50.6	51.3	50.6	**50.6**	50.0	**50.5**	50.5	.008	.019	.010	.008	.012	.010	.002	.008	.010
KPCA	56.1	52.9	56.7	56.1	56.7	x	x	x	x	50.5	49.9	50.6	50.5	50.6	x	x	x	x	.010	.002	.012	.010	.012	x	x	x	x
LLE	**65.2**	52.4	54.0	**60.4**	**58.8**	x	x	x	x	**54.4**	49.8	50.1	**51.9**	**51.3**	x	x	x	x	**.076**	.015	.010	**.037**	**.029**	x	x	x	x
MDS	58.3	56.7	56.7	56.7	54.0	**56.7**	**56.1**	**56.1**	**56.7**	51.1	50.6	50.6	50.6	50.1	**50.6**	**50.5**	**50.5**	**50.6**	.019	.012	.012	.012	.007	.012	**.010**	**.010**	**.012**
SE	50.3	52.9	50.8	52.4	56.7	x	x	x	x	49.7	49.9	49.7	49.8	50.6	x	x	x	x	.000	.020	.005	.005	.018	x	x	x	x
SPCA	55.6	53.5	55.6	55.6	55.6	56.1	**56.1**	**56.1**	55.6	50.4	50.0	50.4	50.4	50.4	50.5	**50.5**	**50.5**	50.4	.008	.003	.008	.008	.008	.010	**.010**	**.010**	.008
TSVD	56.1	56.1	55.6	52.9	56.1	x	x	x	x	50.5	50.5	50.4	49.9	50.5	x	x	x	x	.010	.010	.008	.002	.010	x	x	x	x
NMF	56.7	56.7	**64.7**	51.9	52.4	51.3	51.3	52.4	51.3	50.6	50.6	**54.1**	49.8	49.8	49.8	49.8	49.8	49.8	.012	.012	**.066**	.023	.026	.023	.003	.005	.000
PCA	53.5	56.1	55.6	55.6	56.1	x	x	x	x	50.0	50.5	50.4	50.4	50.5	x	x	x	x	.003	.010	.008	.008	.010	x	x	x	x
VAE	58.8	**61.0**	57.2	57.8	54.0	52.4	52.9	51.9	51.9	51.3	**52.1**	50.8	50.9	50.1	49.8	49.9	49.8	49.8	.022	**.034**	.015	.012	.004	.001	.002	.006	.000

Table A8. Results of different dimensions of different techniques for the SMK_CAN_187 dataset (187 observations, 19 993 features, 2 classes)

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	44.4	57.3	52.6	52.6	48.5	48.0	57.3	57.3	44.4	66.3	68.0	72.7	69.2	70.1	69.2	70.9	68.0	67.4	.212	.418	.361	.293	.296	.290	.294	.449	.252
KPCA	42.7	42.1	46.8	46.2	48.0	x	x	x	x	66.4	66.2	70.1	69.3	69.2	x	x	x	x	.216	.261	.253	.231	.233	x	x	x	x
LLE	48.0	45.6	41.5	33.3	30.4	x	x	x	x	65.7	68.9	65.2	51.8	37.5	x	x	x	x	**.346**	.324	.187	.064	.055	x	x	x	x
MDS	39.8	42.1	43.9	42.7	43.9	44.4	45.0	47.4	45.0	64.6	68.5	67.4	60.4	66.2	66.3	63.2	69.7	67.6	.081	.212	.228	.295	.295	.296	.280	.254	.241
SE	48.0	41.5	40.4	37.4	35.1	x	x	x	x	**68.4**	62.0	61.2	54.3	49.7	x	x	x	x	.283	.141	.161	.100	.075	x	x	x	x
SPCA	43.3	44.4	46.2	45.6	47.4	46.2	46.8	48.5	**48.5**	67.9	67.0	66.9	67.2	69.4	68.0	69.4	69.2	**67.7**	.287	.277	.253	.274	.231	.267	.232	.259	.255
TSVD	**48.5**	45.0	46.2	48.5	52.6	x	x	x	x	69.9	66.9	65.6	68.3	68.3	x	x	x	x	.246	.241	.277	.298	.297	x	x	x	x
NMF	44.4	47.4	49.7	33.9	30.4	30.4	28.7	29.8	32.7	66.3	68.1	65.4	32.7	32.2	32.7	27.7	41.1	49.8	.241	.259	.190	.164	.114	.102	.055	.036	.037
PCA	43.9	46.2	46.8	43.9	45.0	x	x	x	x	66.3	69.2	67.8	67.7	67.8	x	x	x	x .234	.228	.232	.239	.241	x	x	x	x	
VAE	46.2	**60.8**	**59.6**	**57.3**	**57.9**	**57.3**	**64.3**	**63.2**	45.6	67.2	**75.9**	**74.2**	**72.0**	**72.7**	**70.9**	**74.7**	**77.2**	64.9	.196	**.459**	**.393**	**.310**	**.313**	**.312**	**.396**	**.552**	**.264**

Table A9. Results of different dimensions of different techniques for the TOX_171 dataset (171 observations, 5 748 features, 4 classes)

Bold, shade, and x indicate best result of individual reduced dimensions, best among the applied reduced dimensions, and algorithms failed to provide reduced dimensions, respectively.

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	61.0	75.0	77.0	73.0	78.0	82.0	78.0	77.0	84.0	87.8	93.9	94.7	92.3	95.0	95.2	93.7	93.5	94.0	.754	.870	.820	.847	.790	.800	.837	.795	.867
KPCA	**70.0**	71.0	69.0	74.0	78.0	x	x	x	x	92.8	93.2	92.3	94.3	94.3	x	x	x	x	.825	.797	.782	**.855**	.853	x	x	x	x
LLE	60.0	**80.0**	61.0	41.0	x	x	x	x	x	86.7	95.5	85.3	75.9	x	x	x	x	x	**.832**	**.893**	.681	.426	x	x	x	x	x
MDS	68.0	69.0	77.0	69.0	75.0	70.0	75.0	72.0	77.0	91.8	92.3	94.6	93.1	93.0	89.7	93.5	91.1	95.0	.742	.761	.829	.782	.836	.800	.797	.805	.871
SE	59.0	77.0	62.0	40.0	x	x	x	x	x	87.8	93.3	86.8	78.1	x	x	x	x	x	.730	.800	.660	.390	x	x	x	x	x
SPCA	69.0	71.0	72.0	71.0	79.0	73.0	**82.0**	73.0	73.0	**93.1**	92.8	91.8	93.5	95.2	93.6	**94.7**	93.9	93.2	.828	.825	.795	.809	**.880**	.829	.845	.830	.820
TSVD	58.0	68.0	**80.0**	**78.0**	76.0	x	x	x	x	90.2	91.7	**94.9**	93.5	93.7	x	x	x	x	.727	.774	**.846**	.816	.830	x	x	x	x
NMF	54.0	74.0	73.0	50.0	26.0	30.0	43.0	40.0	33.0	89.8	92.8	92.3	83.1	59.7	72.7	76.7	81.1	77.8	.695	.783	.844	.622	.252	.332	.525	.439	.408
PCA	61.0	63.0	72.0	**78.0**	67.0	x	x	x	x	91.2	90.1	92.2	**94.6**	91.6	x	x	x	x	.767	.761	.795	.830	.754	x	x	x	x
VAE	58.0	78.0	**80.0**	**78.0**	**83.0**	**84.0**	**82.0**	**78.0**	**87.0**	89.4	**94.9**	94.0	93.5	**95.4**	**95.9**	94.6	**94.8**	**96.3**	.635	.843	.822	.839	.793	**.866**	**.853**	**.853**	**.895**

Table A10. Results of different dimensions of different techniques for the ORLRAWS10P dataset (100 observations, 10 304 features, 10 classes)

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	82.0	84.0	89.0	84.0	83.0	94.0	82.0	91.0	89.0	95.	9 95.6	96.6	95.6	96.6	98.1	94.8	97.1	96.9	**.902**	.887	.896	.905	.890	.911	**.915**	.890	.938
KPCA	82.0	**86.0**	72.0	80.0	67.0	x	x	x	x	95.5	96.1	93.5	94.9	91.2	x	x	x	x	.877	.891	.836	.867	.845	x	x	x	x
LLE	56.0	72.0	61.0	40.0	x	x	x	x	x	88.1	90.9	83.6	69.3	x	x	x	x	x	.763	.855	.707	.457	x	x	x	x	x
MDS	71.0	79.0	85.0	81.0	78.0	89.0	**84.0**	82.0	84.0	93.0	94.8	96.3	95.0	94.0	96.5	**96.4**	95.4	95.2	.790	.851	.901	.867	.858	.912	.902	.896	.863
SE	56.0	82.0	57.0	44.0	x	x	x	x	x	87.6	95.9	85.4	73.4	x	x	x	x	x	.731	.909	.724	.475	x	x	x	x	x
SPCA	79.0	**86.0**	**91.0**	**94.0**	84.0	91.0	81.0	81.0	**94.0**	93.7	96.1	**97.1**	**98.1**	96.9	97.1	95.6	95.7	**98.1**	.839	.891	**.922**	**.950**	**.938**	.922	.876	**.905**	**.950**
TSVD	74.0	84.0	82.0	82.0	**87.0**	x	x	x	x	93.4	96.0	95.5	95.3	**96.8**	x	x	x	x	.802	.893	.887	.854	.914	x	x	x	x
NMF	75.0	81.0	72.0	56.0	45.0	40.0	37.0	44.0	39.0	94.0	95.5	91.4	86.1	79.7	77.0	75.8	82.9	80.1	.825	.890	.780	.653	.638	.507	.495	.536	.410
PCA	82.0	84.0	89.0	73.0	81.0	x	x	x	x	95.3	**96.6**	96.6	93.4	95.5	x	x	x	x	.864	**.920**	.913	.824	.864	x	x	x	x
VAE	**83.0**	**86.0**	90.0	89.0	**87.0**	**98.0**	81.0	**94.0**	84.0	**96.2**	95.7	96.6	96.5	96.5	**99.3**	95.2	**98.1**	96.7	.899	.878	.897	.912	.897	**.972**	.844	.891	.915

Table A11. Results of different dimensions of different techniques for the PIXRAW10P dataset (100 observations, 10 000 features, 10 classes)

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	23.8	42.3	39.2	46.2	38.5	40.8	40.0	30.8	34.6	80.7	82.6	83.5	85.5	83.3	83.1	84.4	83.5	83.3	.141	.352	.379	.451	**.453**	.395	.483	.296	.384
KPCA	23.1	24.6	30.8	31.5	27.7	x	x	x	x	82.3	81.8	83.8	83.2	81.9	x	x	x	x	.180	.174	.269	.285	.239	x	x	x	x
LLE	23.8	34.6	37.7	33.8	26.2	x	x	x	x	79.3	84.0	68.5	75.2	52.9	x	x	x	x	.174	.419	.354	.312	.248	x	x	x	x
MDS	23.8	27.7	28.5	24.6	29.2	28.5	29.2	26.2	26.9	**83.1**	83.2	81.8	80.7	83.3	82.9	82.9	83.3	82.6	.216	.276	.246	.189	.236	.269	.263	.230	.243
SE	23.8	28.5	30.0	28.5	24.6	x	x	x	x	82.3	83.5	81.7	76.6	66.0	x	x	x	x	.182	.248	.294	.225	.207	x	x	x	x
SPCA	20.8	29.2	30.0	26.2	30.8	27.7	29.2	29.2	26.9	80.7	83.7	81.6	81.9	81.9	80.9	83.2	81.9	82.6	.160	.247	.252	.256	.294	.230	.240	.268	.265
TSVD	21.5	22.3	25.4	26.2	23.8	x	x	x	x	82.6	80.3	82.9	82.9	81.8	x	x	x	x	.194	.131	.232	.224	.251	x	x	x	x
NMF	**24.6**	38.5	**54.6**	40.0	26.9	23.8	26.9	23.8	25.4	82.0	85.5	87.3	75.8	53.9	70.1	65.5	70.5	78.3	**.226**	.399	.571	.421	.263	.227	.238	.194	.213
PCA	**24.6**	25.4	27.7	29.2	28.5	x	x	x	x	82.3	83.1	82.3	83.0	80.6	x	x	x	x	.195	.208	.234	.280	.262	x	x	x	x
VAE	21.5	**46.2**	40.0	**50.0**	**39.2**	**42.3**	**40.8**	**33.8**	**37.7**	81.5	**86.5**	**85.7**	**87.5**	**84.4**	**85.6**	**86.5**	**84.2**	**84.6**	.150	**.463**	**.388**	**.517**	.446	**.478**	**.502**	**.370**	**.419**

Table A12. Results of different dimensions of different techniques for the WARPAR_10P dataset (130 observations, 2 400 features, 10 classes)

Bold, shade, and x indicate best result of individual reduced dimensions, best among the applied reduced dimensions, and algorithms failed to provide reduced dimensions, respectively.

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	34.3	58.6	54.7	86.9	91.8	68.4	48.7	42.8	42.8	83.1	88.3	87.4	88.6	98.4	88.6	84.1	83.5	83.9	.302	.610	.506	.739	.825	.591	.365	.307	.380
KPCA	22.4	29.5	30.5	35.7	32.4	30.0	x	x	x	82.4	83.1	84.0	82.7	83.4	83.7	x	x	x	.148	.342	.337	.382	.355	.329	x	x	x
LLE	26.7	51.4	54.3	34.8	29.5	23.8	x	x	x	81.8	86.7	87.9	66.9	77.6	68.2	x	x	x	.344	.616	**.698**	.329	.267	.175	x	x	x
MDS	24.3	29.0	31.4	29.5	32.9	31.9	31.0	29.0	29.5	82.3	83.8	83.5	82.8	83.6	83.9	83.4	83.7	82.1	.209	.321	.332	.308	.347	.333	.320	.345	.272
SE	21.9	29.5	30.5	32.9	28.6	21.0	x	x	x	80.9	82.7	80.1	80.4	75.6	59.8	x	x	x	.182	.285	.350	.291	.220	.141	x	x	x
SPCA	24.3	28.1	29.5	30.0	31.4	30.0	32.9	30.5	32.4	82.5	82.9	82.6	83.4	80.9	83.3	83.5	82.3	82.6	.226	.286	.300	.329	.356	.337	.341	.320	.352
TSVD	21.9	34.3	33.3	31.0	32.4	31.0	x	x	x	82.5	82.9	82.9	82.6	82.0	81.3	x	x	x	.156	.380	.360	.318	.329	.304	x	x	x
NMF	25.2	48.6	**67.1**	54.8	42.9	31.4	29.5	24.3	26.2	82.4	86.9	88.6	81.7	64.7	59.1	73.9	63.3	71.2	.238	.556	.689	.570	.464	.307	.235	.213	.204
PCA	23.3	30.5	33.3	31.4	30.5	32.4	x	x	x	82.4	82.3	84.2	80.8	83.4	82.4	x	x	x	.205	.300	.372	.335	.335	.335	x	x	x
VAE	**34.8**	**63.8**	58.6	**91.0**	**100.0**	**70.5**	**51.4**	**46.7**	**43.3**	**84.3**	**91.0**	**89.4**	**97.4**	**100.0**	**92.5**	**87.3**	**84.1**	**84.7**	**.369**	**.719**	.621	**.918**	**1.000**	**.782**	**.582**	**.474**	**.465**

Table A13. Result of different dimensions of different techniques for the WARPPIE dataset (210 observations, 2 420 features, 10 classes)

	Purity(%)									RI(%)									NMI (rate0-1)								
Algorithm	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500	2	10	20	50	100	200	300	400	500
AE	64.5	64.5	63.0	65.0	67.0	64.5	66.0	64.0	65.0	53.7	53.1	54.0	54.0	54.3	54.0	54.3	53.1	54.0	.053	.067	.081	.073	.090	.075	.073	.081	.081
KPCA	64.5	65.0	65.0	65.0	65.0	65.0	x	x	x	54.0	54.3	54.3	54.3	54.3	54.3	x	x	x	.077	.081	.081	.081	.977	.081	x	x	x
LLE	59.0	57.5	50.5	53.5	56.0	x	x	x	x	51.4	50.9	49.8	50.0	50.5	x	x	x	x	.016	.006	.057	.021	.008	x	x	x	x
MDS	56.5	64.5	65.0	63.0	64.5	64.5	64.5	59.0	65.0	50.6	54.0	54.3	53.1	54.0	54.0	54.0	51.4	54.3	.009	.077	.081	.039	.077	**.077**	**.077**	.016	.081
SE	59.0	53.5	54.0	55.0	53.5	x	x	x	x	51.4	50.0	50.1	50.3	50.0	x	x	x	x	.016	.007	.001	.008	.021	x	x	x	x
SPCA	64.0	64.5	64.5	64.5	64.5	64.5	59.0	64.5	59.0	53.7	54.0	54.0	54.0	54.0	54.0	51.4	54.0	51.4	.073	.077	.077	.077	.077	.077	.016	.077	.016
TSVD	64.5	59.0	65.0	65.0	65.0	64.5	x	x	x	54.0	51.4	54.3	54.3	54.3	54.0	x	x	x	.077	.016	.081	.081	.081	.077	x	x	x
NMF	**65.0**	**66.0**	63.0	58.5	56.5	55.5	56.5	58.5	56.5	**54.3**	**54.9**	53.1	51.2	50.6	50.4	50.6	51.2	50.6	**.081**	**.090**	.039	.015	.028	.020	.028	.013	.028
PCA	63.0	59.0	65.0	65.0	65.0	64.5	x	x	x	53.1	51.4	54.3	54.3	54.3	54.0	x	x	x	.039	.016	.081	.082	.082	.078	x	x	x
VAE	**65.0**	**66.0**	**65.5**	**66.0**	**69.0**	**66.0**	**66.5**	**66.0**	**67.0**	**54.3**	**54.9**	**54.6**	**54.9**	**57.0**	**54.9**	**55.2**	**54.9**	**55.6**	**.081**	**.090**	**.102**	**.091**	**.102**	**.091**	.076	**.091**	**.100**

Table A14. Results of different dimensions of different techniques for the ARCENE dataset (200 observations, 10 000 features, 2 classes)

Bold, shade, and x indicate best result of individual reduced dimensions, best among the applied reduced dimensions, and algorithms failed to provide reduced dimensions, respectively.

[9] Tipping, M. E.—Bishop, C. M.: Mixtures of Probabilistic Principal Component Analyzers. Neural Computation, Vol. 11, 1999, No. 2, pp. 443–482, doi: 10.1162/089976699300016728.

[10] Hyvärinen, A.—Oja, E.: Independent Component Analysis: Algorithms and Applications. Neural Networks, Vol. 13, 2000, No. 4-5, pp. 411–430, doi: 10.1016/s0893-6080(00)00026-5.

[11] Barber, D.: Bayesian Reasoning and Machine Learning. Cambridge University Press, New York, NY, United States, 2012, doi: 10.1017/CBO9780511804779.

[12] Cox, T.—Cox, M.: Multidimensional Scaling. In: Chen, C. H., Härdle, W. K., Unwin, A. (Eds.): Handbook of Data Visualization. Springer, Berlin, Heidelberg, Springer Handbooks Comp. Statistics, 2008, pp. 315–347, doi: 10.1007/978-3-540-33037-0_14.

[13] Cichocki, A.—Phan, A. H.: Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E92.A, 2009, No. 3, pp. 708–721, doi: 10.1587/transfun.e92.a.708.

[14] Wang, Y.—Yao, H.—Zhao, S.: Auto-Encoder Based Dimensionality Reduction. Neurocomputing, Vol. 184, 2016, pp. 232–242, doi: 10.1016/j.neucom.2015.08.104.

[15] Yue, T.—Wang, H.: Deep Learning for Genomics: A Concise Overview. 2018, arXiv: 1802.00810.

[16] Srivastava, N.—Hinton, G.—Krizhevsky, A.—Sutskever, I.—Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, Vol. 15, 2014, pp. 1929–1958.

[17] Zhao, W.: Research on the Deep Learning of the Small Sample Data Based on Transfer Learning. AIP Conference Proceedings, Vol. 1864, 2017, No. 1, Art. No. 020018, doi: 10.1063/1.4992835.

[18] Lamb, A.—Dumoulin, V.—Courville, A.: Discriminative Regularization for Generative Models. 2016, arXiv: 1602.03220.

[19] Mahmud, M. S.—Fu, X.: Unsupervised Classification of High-Dimension and Low-Sample Data with Variational Autoencoder Based Dimensionality Reduction. 2019 IEEE 4$^{\text{th}}$ International Conference on Advanced Robotics and Mechatronics (ICARM), Toyonaka, Japan, 2019, doi: 10.1109/icarm.2019.8834333.

[20] Hall, P.—Marron, J. S.—Neeman, A.: Geometric Representation of High Dimension, Low Sample Size Data. Journal of the Royal Statistical Society, Series B, Vol. 67, 2005, No. 3, pp. 427–444, doi: 10.1111/j.1467-9868.2005.00510.x.

[21] Jung, S.—Marron, J. S.: PCA Consistency in High Dimension, Low Sample Size Context. The Annals of Statistics, Vol. 37, 2009, No. 6B, pp. 4104–4130, doi: 10.1214/09-aos709.

[22] Li, X.—Lin, S.—Yan, S.—Xu, D.: Discriminant Locally Linear Embedding with High-Order Tensor Data. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 38, 2008, No. 2, pp. 342–352, doi: 10.1109/tsmcb.2007.911536.

[23] Schölkopf, B.—Smola, A. J.—Müller, K.-R.: Kernel Principal Component Analysis. In: Burges, C. J. C., Schölkopf, B., Smola, A. J. (Eds.): Advances in Kernel Methods. MIT Press, Cambridge, MA, USA, 1999, pp. 327–352.

[24] Zou, H.—Hastie, T.—Tibshirani, R.: Sparse Principal Component Analysis. Journal of Computational and Graphical Statistics, Vol. 15, 2006, No. 2, pp. 262–286, doi: 10.1198/106186006x113430.

[25] Ng, A. Y.—Jordan, M. I.—Weiss, Y.: On Spectral Clustering: Analysis and an Algorithm. In: Dietterich, T. G., Becker, S., Ghahramani, Z. (Eds.): Advances in Neural Information Processing Systems 14 (NIPS 2001), MIT Press, 2001, pp. 849–856.

[26] Yeung, K. Y.—Ruzzo, W. L.: Principal Component Analysis for Clustering Gene Expression Data. Bioinformatics, Vol. 17, 2001, No. 9, pp. 763–774, doi: 10.1093/bioinformatics/17.9.763.

[27] Mishra, D.—Dash, R.—Rath, A. R.—Acharya, M.: Feature Selection in Gene Expression Data Using Principal Component Analysis and Rough Set Theory. In: Arabnia, H., Tran, Q. N. (Eds.): Software Tools and Algorithms for Biological Systems. Springer, New York, Advances in Experimental Medicine and Biology, Vol. 696, 2011, pp. 91–100, doi: 10.1007/978-1-4419-7046-6_10.

[28] Sato-Ilic, K.: Structural Classification Based Correlation and Its Application to Principal Component Analysis for High-Dimension Low-Sample Size Data. 2012 IEEE International Conference on Fuzzy Systems, 2012, doi: 10.1109/fuzz-ieee.2012.6251200.

[29] Yata, K.—Aoshima, M.: Effective PCA for High-Dimension, Low-Sample-Size Data with Noise Reduction via Geometric Representations. Journal of Multivariate Analysis, Vol. 105, 2012, No. 1, pp. 193–215, doi: 10.1016/j.jmva.2011.09.002.

[30] Ramsay, J. O.—Silverman, B. W.: Applied Functional Data Analysis: Methods and Case Studies. Springer, New York, Springer Series in Statistics, 2002, doi: 10.1007/b98886.

[31] Shen, D.—Shen, H.—Zhu, H.—Marron, J. S.: The Statistics and Mathematics of High Dimension Low Sample Size Asymptotics. Statistica Sinica, Vol. 26, 2016, No. 4, pp. 1747–1770, doi: 10.5705/ss.202015.0088.

[32] Pascual-Montano, A.—Carmona-Saez, P.—Chagoyen, M.—Tirado, F.—Carazo, J. M.—Pascual-Marqui, R. D.: bioNMF: A Versatile Tool for Non-Negative Matrix Factorization in Biology. BMC Bioinformatics, Vol. 7, 2006, Art. No. 366, doi: 10.1186/1471-2105-7-366.

[33] Berry, M. W.—Browne, M.—Langville, A. N.—Pauca, V. P.—Plemmons, R. J.: Algorithms and Applications for Approximate Nonnegative Matrix Factorization. Computational Statistics and Data Analysis, Vol. 52, 2007, No. 1, pp. 155–173, doi: 10.1016/j.csda.2006.11.006.

[34] Danaee, P.—Ghaeini, R.—Hendrix, D. A.: A Deep Learning Approach for Cancer Detection and Relevant Gene Identification. In: Altman, R. B., Dunker, A. K., Hunter, L., Ritchie, M. D., Murray, T. A., Klein, T. E. (Eds.): Proceedings of the Pacific Symposium on Biocomputing 2017 (Biocomputing 2017), 2017, pp. 219–229, doi: 10.1142/9789813207813_0022.

[35] Hinton, G. E.—Srivastava, N.—Krizhevsky, A.—Sutskever, I.—Salakhutdinov, R. R.: Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. 2012, arXiv: 1207.0580.

[36] Zhang, D.—Zhou, Z. H.—Chen, S.: Semi-Supervised Dimensionality Reduction. Proceedings of the 2007 SIAM International Conference on Data Mining (SDM 2007), 2007, pp. 629-634, doi: 10.1137/1.9781611972771.73.

[37] Rasmus, A.—Valpola, H.—Honkala, M.—Berglund, M.—Raiko, T.: Semi-Supervised Learning with Ladder Network. 2015, arXiv: 1507.02672.

[38] Quint, E.—Wirka, G.—Williams, J.—Scott, S.—Vinodchandran, N. V.: Interpretable Classification via Supervised Variational Autoencoders and Differentiable Decision Trees. 6th International Conference on Learning Representations (ICLR 2018), 2018.

[39] Halko, N.—Martinsson, P. G.—Tropp, J. A.: Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. SIAM Review, Vol. 53, 2011, No. 2, pp. 217–288, doi: 10.1137/090771806.

[40] Hoffman, M. D.—Blei, D. M.—Bach, F.: Online Learning for Latent Dirichlet Allocation. In: Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., Culotta, A. (Eds.): Advances in Neural Information Processing Systems 23 (NIPS 2010), 2010, pp. 856–864.

[41] Hoffman, M. D.—Blei, D. M.—Wang, C.—Paisley, J.: Stochastic Variational Inference. Journal of Machine Learning Research, Vol. 14, 2013, No. 4, pp. 1303–1347.

[42] Mairal, J.—Bach, F.—Ponce, J.—Sapiro, G.: Online Dictionary Learning for Sparse Coding. Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09), 2009, pp. 689–696, doi: 10.1145/1553374.1553463.

[43] Kingma, D. P.—Welling, M.: Auto-Encoding Variational Bayes. International Conference on Learning Representations, 2014, arXiv: 1312.6114, doi: 10.1561/2200000056.

[44] Kullback, S.—Liebler, R. A.: On Information and Sufficiency. The Annals of Mathematical Statistics, Vol. 22, 1951, No. 1, pp. 79–86, doi: 10.1214/aoms/1177729694.

[45] Rezende, D. J.—Mohamed, S.—Wierstra, D.: Stochastic Backpropagation and Approximate Inference in Deep Generative Models. Proceedings of the 31st International Conference on Machine Learning, Proceedings of Machine Learning Research, Vol. 32, 2014, No. 2, pp. 1278–1286, arXiv: 1401.4082.

[46] Ester, M.—Kriegel, H.-P.—Sander, J.—Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI Press, 1996, pp. 226–231.

[47] Pelleg, D.—Moore, A. W.: X-Means: Extending K-Means with Efficient Estimation of the Number of Clusters. Proceedings of the 17th International Conference on Machine Learning (ICML '00), 2000, pp. 727–734.

[48] Mahmud, M. S.—Huang, J. Z.—Fu, X.: Variational Autoencoder-Based Dimensionality Reduction for High-Dimensional Small-Sample Data Classification. Interna-

tional Journal of Computational Intelligence and Applications, Vol. 19, 2020, No. 1, Art. No. 2050002, doi: 10.1142/S1469026820500029.

[49] Masud, M. A.—Huang, J. Z.—Wei, C.—Wang, J.—Khan, I.—Zhong, M.: I-Nice: A New Approach for Identifying the Number of Clusters and Initial Cluster Centres. Information Sciences, Vol. 466, 2018, pp. 129–151, doi: 10.1016/j.ins.2018.07.034.

[50] Manning, C. D.—Raghavan, P.—Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, 2008.

[51] Rand, W. M.: Objective Criteria for the Evaluation of Clustering Methods. Journal of the American Statistical Association, Vol. 66, 1971, No. 336, pp. 846–850, doi: 10.1080/01621459.1971.10482356.

[52] Strehl, A.—Ghosh, J.: Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. Journal of Machine Learning Research, Vol. 3, 2002, pp. 583–617.

[53] Liu, B.—Wei, Y.—Zhang, Y.—Yang, Q.: Deep Neural Networks for High Dimension, Low Sample Size Data. Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17), Melbourne, Australia, 2017, pp. 2287–2293, doi: 10.24963/ijcai.2017/318.

Mohammad Sultan Mahmud is currently Ph.D. candidate at the Shenzhen University, China. He received his Master's degree from the King Mongkut's University of Technology North Bangkok, Thailand, in 2014. He was awarded the Outstanding Doctoral Student of Shenzhen University in 2017 and Shenzhen Universiade International Scholarship in 2018. His current research focuses on big data mining and distributed and parallel computing.

Joshua Zhexue Huang received his Ph.D. degree from the Royal Institute of Technology, Sweden, in 1993. He is Distinguished Professor of the College of Computer Science and Software Engineering at Shenzhen University. Also, he is the Director of Big Data Institute and the Deputy Director of the National Engineering Laboratory for Big Data System Computing Technology. His main research interests include big data technology and applications. He has published over 200 research papers in conferences and journals. In 2006, he received the most influential paper award in the First Pacific-Asia Conference on Knowledge Discovery and Data Mining. He is known for his contributions to the development of a series of k-means type clustering algorithms in data mining, such as k-modes, fuzzy k-modes, k-prototypes, and w-k-means, that are widely cited and used, and some of which have been included in commercial software. He has extensive industry expertise in business intelligence and data mining, and has been involved in numerous consulting projects in Australia and China.

Xianghua Fu received his Ph.D. degree in computer science and technology from the Xi'an Jiaotong University, China, in 2005 and his M.Sc. degree from the Northwest A & F University, China, in 2002. Currently, he is Professor at the College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China. He led a project of the National Natural Science Foundation, hosts a project of the Natural Science Foundation of Guangdong Province, and several projects of the Science and Technology Foundation of Shenzhen City. His research interests include machine learning, information retrieval, and natural language processing.

Rukhsana Ruby received her Master's degree from the University of Victoria, Canada, in 2009, and her Ph.D. degree from The University of British Columbia, Canada, in 2015. She has authored nearly 60 technical papers of well-recognized journals and conferences. Her research interest includes the management and optimization of next-generation wireless networks. She was a recipient of several awards or honors, notable among which are the Wait-listed for Canadian NSERC Postdoctoral Fellowship, the IEEE Exemplary Certificate (IEEE Communications Letters in 2018 and IEEE Wireless Communications Letters in 2018).

Kaishun Wu received his Ph.D. degree from the Hong Kong University of Science and Technology (HKUST), in 2011. From 2013, he is Distinguish Professor at the Shenzhen University, China. He has co-authored 2 books and published over 100 research papers in international journals and conferences, such as the IEEE Transactions on Mobile Computing, the IEEE Transactions on Parallel and Distributed Systems, ACM MobiCom, and IEEE INFOCOM. He holds 6 U.S. patents and has over 90 Chinese pending patents. He was a recipient of the 2012 Hong Kong Young Scientist Award and 2014 Hong Kong ICT awards and the 2014 IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award. He is Fellow of the IET and IEEE Senior Member.