Hierarchical One-Class Classifier With Within-Class Scatter-Based Autoencoders

Tianlei Wang, Jiuwen Cao, *Member, IEEE*, Xiaoping Lai, *Member, IEEE*, and Q. M. Jonathan Wu, *Senior Member, IEEE*

***Abstract*— Autoencoding is a vital branch of representation learning in deep neural networks (DNNs). The extreme learning machine-based autoencoder (ELM-AE) has been recently developed and has gained popularity for its fast learning speed and ease of implementation. However, the ELM-AE uses random hidden node parameters without tuning, which may generate meaningless encoded features. In this brief, we first propose a within-class scatter information constraint-based AE (WSI-AE) that minimizes both the reconstruction error and the within-class scatter of the encoded features. We then build stacked WSI-AEs into a one-class classification (OCC) algorithm based on the hierarchical regularized least-squared method. The effectiveness of our approach was experimentally demonstrated in comparisons with several state-of-the-art AEs and OCC algorithms. The evaluations were performed on several benchmark data sets.**

***Index Terms*— Autoencoder (AE), extreme learning machine (ELM), one-class classification (OCC), scatter matrix.**

I. Introduction

Autoencoders (AEs), which learn data representations by replacing the output with the input and mapping the inputs to the outputs with the least possible number of distortions, have been developed for representation learning in deep neural networks (DNNs) [1]–[3]. The extreme learning machine-based autoencoder (ELM-AE) [4]–[6], built on a single hidden-layer feedforward network with random hidden parameters [7]–[13], has been developed for feature learning. The ELM-AE is formulated as $\mathbf{H}\boldsymbol{\beta} = \mathbf{X}$, where $\mathbf{X}$ and $\mathbf{H}$ are the input and hidden-layer output matrices of the AE, respectively, and the output weight $\boldsymbol{\beta}$ needs to be optimized for feature representation. Because ELM-AE uses random parameters without tuning, the optimization problem of the output weight becomes a quadratic minimization problem, which can be analytically solved by the least-squares method. ELM-AE is advantaged by fast learning speed, ease of implementation, and fewer parameters requiring human intervention than many other methods. Since the multilayer neural network with stacked ELM-AEs (ML-ELM) was developed for large and complex data learning [4], researchers have proposed the hierarchical learning framework using ℓ_1-norm penalty-based ELM-AE, which exploits the sparse and discriminative features [5], and the kernel ELM-AE-based multilayer ELM [6]. A novel sparse-feature encoding ELM-AE using a Johnson–Lindenstrauss lemma-based random projection has also been developed [14]. Li *et al.* [15] proposed an ELM-AE based on double random hidden layers and fixed the outputs of the stacked AEs at the original $\mathbf{X}$. In [16], the reconstruction in ELM-AE was improved by manifold regularization. In [17], self-reconstruction was replaced by a graph-weighted reconstruction of all samples in the ELM-AE.

The superiority of ELM-AE over backpropagation-based AE [2] has been demonstrated in evaluations of learning speed and generalization performance [4]. However, because ELM-AE randomly selects the hidden parameters without tuning, and no constraints are imposed on the encoded features, the algorithm may generate meaningless features that degrade the representation performance [18]. In this brief, we show that under certain assumptions, an ELM-AE-encoded feature is merely equivalent to a random projection. To improve this situation, we develop a novel within-class scatter information-based ELM-AE (WSI-AE) for feature learning. Instead of merely minimizing the norm of the output weight matrix, this method constrains the AE. The imposed constraint is based on the within-class scatter information that regularizes the encoded feature distribution. As stated in [19]–[22], applications, such as anomaly/fault detection and network intrusion, are better handled by one-class classification (OCC) algorithms than by conventional multiclass classification methods. To enhance the learning capability on complex and high-dimensional data, this brief develops a novel hierarchical regularized least-square-based OCC (HLS-OC) algorithm, in which the proposed WSI-AEs are stacked in parallel for feature learning. When trained on the discriminative features extracted by the proposed WSI-AEs, HLS-OC outperforms the shallow OCC algorithms. Moreover, because WSI-AE imposes an explicit constraint on the within-class scatter, HLC-OC can learn more compact encoded features and thus obtain a more promising data representation.

II. Brief Review

A. ELM

Consider a training set with N samples and M classes $(\mathbf{x}_i, t_i)$, where $\mathbf{x}_i \in \mathbb{R}^D$ is the input vector and $t_i \in \{c_1, \ldots, c_M\}$ is its associated class label. For randomly generated input weights $\mathbf{W}$ and biases $\mathbf{b}$, the ELM attempts to minimize the training error

$$\min_{\boldsymbol{\beta}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_F^2. \tag{1}$$

Here, $\|\cdot\|_F$ is the Frobenius norm, $\boldsymbol{\beta}_{L\times M}$ is the output weight matrix, and $\mathbf{T} = [\ \hat{\mathbf{t}}_1 \ \cdots \ \hat{\mathbf{t}}_N]^T$ is the desired output matrix, where $\hat{\mathbf{t}}_i$ is an M-dimensional vector constructed by transforming $t_i = c_j$ to a vector with 1 in its jth element and 0s in all other elements. $\mathbf{H} = \{\mathbf{h}(\mathbf{x}_i)\}$, where $\mathbf{h}(\mathbf{x}_i) = g(\mathbf{W}\mathbf{x}_i + \mathbf{b})^T$, $g(\cdot)$ is an activation function, and $\mathbf{h}(\mathbf{x}_i) \in \mathbf{R}^{1\times L}$ represents the hidden-layer output vector of $\mathbf{x}_i$. The ELM calculates $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{2}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of $\mathbf{H}$.

Manuscript received July 29, 2019; revised February 14, 2020 and June 17, 2020; accepted August 8, 2020. Date of publication August 21, 2020; date of current version August 4, 2021. This work was supported by the National Natural Science Foundation of China under Grant U1909209, Grant 61503104, Grant 61573123, and Grant 91648208. *(Corresponding author: Jiuwen Cao.)*

Tianlei Wang, Jiuwen Cao, and Xiaoping Lai are with the Artificial Intelligence Institute, Hangzhou Dianzi University, Zhejiang 310018, China, and also with the Key Laboratory for IOT and Information Fusion Technology of Zhejiang, Hangzhou Dianzi University, Zhejiang 310018, China (e-mail: tianlei.wang.cn@gmail.com; jwcao@hdu.edu.cn; laixp@hdu.edu.cn).

Q. M. Jonathan Wu is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: jwu@uwindsor.ca).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3015860

B. ELM-AE

ELM-AE learns the data representation by setting the input $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_N]$ as the desired output. The ℓ_2-norm regularized based ELM-AE uses orthogonal hidden-layer parameters [4]. The network is trained by the following minimization problem:

$$\min \frac{1}{2}C\left\|\mathbf{H}\boldsymbol{\beta} - \mathbf{X}^T\right\|_F^2 + \frac{1}{2}\|\boldsymbol{\beta}\|_F^2$$
$$\text{s. t.} \begin{cases} \mathbf{W}^T\mathbf{W} = \mathbf{I},\ \mathbf{b}^T\mathbf{b} = \mathbf{I} & L \geq D \\ \mathbf{W}\mathbf{W}^T = \mathbf{I},\ \mathbf{b}^T\mathbf{b} = \mathbf{I} & L < D \end{cases} \tag{3}$$

where C is the regularization parameter. The encoded outputs of the ELM-AEs are obtained by solving

$$\mathbf{Y} = \boldsymbol{\beta}\mathbf{X}. \tag{4}$$

To reduce the feature density in the ELM-AE, Tang *et al.* [5] imposed an ℓ_1-norm constraint on the output weight $\boldsymbol{\beta}$ of sparse feature encoding-based ELM (ELM-SAE), which obtains sparse encoded features.

III. Proposed Method

A. Problem Statement

The assumptions in [23], which state that nonregularized ELM-AE has a linear activation function and zero bias, are followed here. Kasun *et al.* [23] demonstrated that when the number of input neurons exceeds the number of hidden neurons ($L < D$) and $\mathbf{W}\mathbf{W}^T = \mathbf{I}$, then $\boldsymbol{\beta} = \mathbf{W}\mathbf{V}\mathbf{V}^T$, where $\mathbf{V}$ denotes the eigenvectors of the covariance matrix $\mathbf{X}\mathbf{X}^T$ and $\mathbf{V}\mathbf{V}^T = \mathbf{I}$. Thus, we have $\boldsymbol{\beta} = \mathbf{W}$ and the encoded features are $\boldsymbol{\beta}\mathbf{X} = \mathbf{W}\mathbf{X} = \mathbf{H}^T$.

Under the same assumptions, when $L \geq D$, $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ as shown in (3), we have

$$\mathbf{H}\mathbf{H}^T = \mathbf{X}^T\mathbf{W}^T\mathbf{W}\mathbf{X} = \mathbf{X}^T\mathbf{X}. \tag{5}$$

The nonregularized ELM-AE is required to solve the objective function

$$\min_{\boldsymbol{\beta}} \left\|\mathbf{H}\boldsymbol{\beta} - \mathbf{X}^T\right\|_F^2. \tag{6}$$

The solution of (6) is computed as

$$\boldsymbol{\beta} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{X}^T. \tag{7}$$

The encoded features are then derived as

$$\boldsymbol{\beta}\mathbf{X} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{X}^T\mathbf{X} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{H}\mathbf{H}^T \tag{8}$$

that is $\boldsymbol{\beta}\mathbf{X} = \mathbf{H}^T$. Thus, the encoded features of a nonregularized linear ELM-AE are merely the hidden-layer output $\mathbf{H}$, which is a random projection of the input.

B. WSI-AE

To address the aforementioned problem of meaningless encoded features, we impose the within-class scatter information of the encoded features on the ELM-AE. The resultant WSI-AE can then regularize the feature distribution. Let N_j be the number of samples belonging to class c_j and $\mathbf{X}_j$ be the associated data set. The mean feature vector of class c_j can be computed as

$$\boldsymbol{\mu}_j = \frac{1}{N_j}\sum_{\mathbf{x}\in\mathbf{X}_j}\mathbf{x}. \tag{9}$$

The sample-based estimated covariance matrix of class c_j is

$$\mathbf{D}_j = \sum_{\mathbf{x}\in\mathbf{X}_j}\left(\mathbf{x} - \boldsymbol{\mu}_j\right)\left(\mathbf{x} - \boldsymbol{\mu}_j\right)^T. \tag{10}$$

The within-class scatter matrix is

$$\mathbf{S}_w = \sum_{j=1}^{M}\mathbf{D}_j. \tag{11}$$

From (4) and (10), the within-class scatter matrix of the encoded features of an ELM-AE is derived as $\boldsymbol{\beta}\mathbf{S}_w\boldsymbol{\beta}^T$. The proposed WSI-AE, which constrains the encoded features by the within-class scatter matrix, is then expressed as

$$\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) = \frac{1}{2}C\left\|\mathbf{H}\boldsymbol{\beta} - \mathbf{X}^T\right\|_F^2 + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\beta}\mathbf{S}_{\mathrm{w}}\boldsymbol{\beta}^{\mathrm{T}}\right). \tag{12}$$

Here, $\mathrm{tr}(\cdot)$ is the trace of a matrix. WSI-AE aims to minimize both the reconstruction errors and the within-class scatter information of the encoded features. By imposing a penalty on the underlying data distribution of the encoded features, we restrict the solution space and thereby avoid the meaningless encoded results.

Unlike the formula of conventional ELM-AE, (12) cannot be solved analytically. However, as the hidden node parameters are randomly generated, the optimization of (12) becomes a quadratic convex problem, which can be solved by general convex optimization methods. This brief employs the stochastic gradient descent (SGD) method, which determines the gradient of (12) with respect to $\boldsymbol{\beta}$ as

$$\begin{aligned} \nabla J(\boldsymbol{\beta}) &= C\mathbf{H}^T\mathbf{H}\boldsymbol{\beta} - C\mathbf{H}^T\mathbf{X}^T + \boldsymbol{\beta}\boldsymbol{S}_w \\ &= C\mathbf{H}^T\left(\mathbf{H}\boldsymbol{\beta} - \mathbf{X}^T\right) + \boldsymbol{\beta}\boldsymbol{S}_w \\ &= C\sum_{n=1}^{N}\mathbf{h}_n^T\left(\mathbf{h}_n\boldsymbol{\beta} - \mathbf{x}_n^T\right) + \boldsymbol{\beta}\boldsymbol{S}_w. \end{aligned} \tag{13}$$

Then, $\boldsymbol{\beta}$ can be iteratively updated as

$$\begin{aligned} \mathbf{v}_i &= \gamma\,\mathbf{v}_{i-1} + \frac{\nabla J\left(\boldsymbol{\beta}_i\right)}{\left\|\nabla J\left(\boldsymbol{\beta}_i\right)\right\|_\infty} \\ \boldsymbol{\beta}_{i+1} &= \boldsymbol{\beta}_i - \alpha_i\mathbf{v}_i. \end{aligned} \tag{14}$$

Here, γ is a tradeoff parameter, $\mathbf{v}_i$ denotes the momentum, and α_i is the learning rate. The SGD algorithm approximates the gradient from a single or minibatch of training samples at each iteration. This approximation usually performs faster and better than the standard gradient descent algorithms. Assuming a minibatch with N_s samples randomly chosen at each iteration, the gradient $\nabla J(\boldsymbol{\beta})$ in (13) can be approximated as

$$\nabla J(\boldsymbol{\beta})_S = C\sum_{n=1}^{N_s}\mathbf{h}_n^T\left(\mathbf{h}_n\boldsymbol{\beta} - \mathbf{x}_n^T\right) + \boldsymbol{\beta}\boldsymbol{S}_w. \tag{15}$$

Note that minibatch samples are required only for the error term of the cost in (12). For a data set obeying a certain distribution with fixed mean and correlation, the within-class scatter matrix can be calculated using the correlation matrix. Meanwhile, $\mathbf{S}_w$ is fixed in SGD, and can be estimated in advance. Algorithm 1 gives the pseudo code of WSI-AE.

Algorithm 1 WSI-AE

Given:

Input data $\mathbf{X}$, the number of hidden nodes L, the regularization parameter C, the activation function $\mathrm{g}(\cdot)$, the mini-batch size and the number of epochs.

Steps:

1) Compute the within-class scatter matrix by (11)
2) Generate $\mathbf{W}$ and $\mathbf{b}$ and then calculate $\mathbf{H}$
3) Optimize $\boldsymbol{\beta}$ iteratively with (14) and (15)
4) Derive the encoded result with (4)

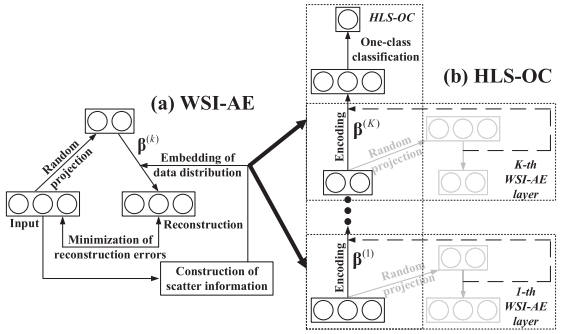

Fig. 1. Architectures of the WSI-AE and HLS-OC.

C. HLS-OC

The OCC algorithms, which learn the data set of one target, are more suitable than conventional algorithms for applications, such as anomaly detection and fault diagnosis. The proposed WSI-AE constrains the within-class scatter information of a target class, so can be embedded into OCC for feature learning. Inspired by this idea, we developed our novel HLS-OC algorithm using stacked WSI-AEs for feature learning (see Fig. 1). Assuming that K WSI-AEs are stacked and letting $\mathbf{Y}^{(k-1)}$ be the output of the $(k-1)$th layer with $\mathbf{Y}^{(0)} = \mathbf{X}$, the output $\mathbf{Y}^{(k)}$ of the kth layer is determined as

$$\mathbf{Y}^{(k)} = \mathrm{g}\big(\boldsymbol{\beta}^{(k)}\mathbf{Y}^{(k-1)}\big), \quad k = 1, \ldots, K \tag{16}$$

where $\boldsymbol{\beta}^{(k)}$ is optimized by Algorithm 1 and $\mathrm{g}(\cdot)$ is the activation function. The last layer of HLS-OC is an ELM-based OCC. This layer is trained as

$$\min_{R,\boldsymbol{\beta}} \quad \frac{1}{2}C_1 \max\Big(0 \quad \Big\|\big(\mathbf{Y}^{(K)}\big)^T\boldsymbol{\beta} - \mathbf{t}\Big\|_\infty - R\Big) + \frac{1}{2}C_2R^2 + \frac{1}{2}\|\boldsymbol{\beta}\|_F^2 \tag{17}$$

where $\|\cdot\|_\infty$ is the infinity norm, $\mathbf{t} = [t, \ldots, t]^T \in \mathbb{R}^N$ is the desired output vector, and $C_1 > 0$ and $C_2 > 0$ are two tradeoff parameters. Unlike the multiclass problem, the OCC in (17) seeks the minimum-radius (i.e., R) hypersphere centered at t that encloses the training data. To show the effectiveness of the WSI-AE (the main objective of this brief), we employ a simplified version of (17) as follows [22]:

$$\min_{\boldsymbol{\beta}} \frac{1}{2}C\Big\|\big(\mathbf{Y}^{(K)}\big)^T\boldsymbol{\beta} - \mathbf{t}\Big\|_F^2 + \frac{1}{2}\|\boldsymbol{\beta}\|_F^2. \tag{18}$$

Equation (18) is solved as

$$\begin{aligned}\boldsymbol{\beta} &= \mathbf{Y}^{(K)}\left(\frac{\mathbf{I}}{C} + \big(\mathbf{Y}^{(K)}\big)^T\mathbf{Y}^{(K)}\right)^{-1}\mathbf{t}, \quad \text{if } N < L^{(K)} \\ \boldsymbol{\beta} &= \left(\frac{\mathbf{I}}{C} + \mathbf{Y}^{(K)}\big(\mathbf{Y}^{(K)}\big)^T\right)^{-1}\mathbf{Y}^{(K)}\mathbf{t}, \quad \text{if } N \geq L^{(K)}\end{aligned} \tag{19}$$

where $L^{(K)}$ is the number of hidden neurons in the Kth AE. Note that (18) uses only the least-squared loss to cluster the data at the center t. The radius of the cluster hypersphere can be manually determined by thresholding

$$\varepsilon(\mathbf{x}_i) = \left|\left(\mathbf{y}_i^{(K)}\right)^T\boldsymbol{\beta} - t\right|. \tag{20}$$

Here, $\mathbf{y}_i^{(K)}$ denotes the encoded results of sample $\mathbf{x}_i$. The threshold η is usually determined by choosing a percentage of training samples as outliers in the training error sequence. The HLS-OC is summarized in Algorithm 2.

Algorithm 2 HLS-OC

Given:

Input data $\mathbf{X}$, the regularization parameter and hidden nodes $(C^{(k)}, L^{(k)}), k = 1, \ldots, K$,

Training stage:

1) for $k = 1 : K$
   a) Compute $\boldsymbol{\beta}^{(k)}$ of the k-th WSI-AE by Algorithm 1
   b) Obtain the encoded result $\mathbf{Y}^{(k)}$ with (4)
2) Calculate the output weight by (19)
3) Derive training errors by (20) and select the threshold η

Testing stage:

A testing sample $\mathbf{x}_p$

1) for $k = 1 : K$,
   $\mathbf{y}_p^{(k)} = \mathrm{g}(\boldsymbol{\beta}^{(k)}\mathbf{y}_p^{(k-1)})$
2) Compute the output $o_p = \mathrm{g}((\mathbf{y}_p^{(K)})^T\boldsymbol{\beta})$
3) Derive $\varepsilon(\mathbf{x}_p) = |o_p - t|$ and perform the classification

$$\begin{cases}\varepsilon\big(\mathbf{x}_p\big) \leq \eta \rightarrow \text{target} \\ \varepsilon\big(\mathbf{x}_p\big) > \eta \rightarrow \text{outlier.}\end{cases}$$

IV. Experiments

This section demonstrates the effectiveness of the proposed method on various benchmark data sets. The data set specifications are shown in Table I. For the first 12 data sets, the first class was set as the target and the remaining classes were considered as outliers. For the USPS data set, the fifth class was assigned as the target. For the HAR,[1] MNIST,[2] Fashion,[3] and NORB[4] data sets, the performance was evaluated in different scenarios (see Table I for details). Each target class was indexed by a number following the "–" symbol. For example, by assigning classes 2–5 as the target classes in the Fashion data set, we formed the Fashion-2, Fashion-3, Fashion-4, and Fashion-5 data sets, respectively. As an additional test, we merged classes 5, 7, and 9 as the target class in the Fashion data set, forming Fashion-579. The NORB data set was processed by the method described in [5], which employs the zero-phase component analysis whitening [24] for noise filtering and feature reduction.

We compared the performances of our method and several state-of-the-art existing AEs, namely, the original ELM-AE [4], the ELM-SAE [5], the random sparse matrix-based AE (SMA) [14], the weight-decay regularization-based AE (WD-AE) [2], and the denoising AE with binary and Gaussian masking noises (DAE^B and DAE^G, respectively) [3]. For all AEs, the hidden nodes L and the regularized parameter C were optimized on the grid $\{10, 100, 500, 1000, 2000\} \times \{10^{-3}, 10^0, 10^3\}$. The number of epochs and the minibatch size in the SGD were both set to 100, and the initial learning rate was 0.01. The learning rate was attenuated by 0.1 per 10 epochs. The tradeoff parameter γ was empirically set as 0.9. The input corruption rate in the DAE^B and the Gaussian variance in the DAE^G were both set to 0.5. The ℓ_1-norm optimization in ELM-SAE was iterated 100 times, and a Gaussian-distributed random sparse matrix was generated for feature mapping in SMA. In each case, two AEs were embedded as a stack in the HLS-OC for feature learning. The regularized parameter in the last OCC layer of HLS-OC was optimized on the grid $\{10^0, 10^3, 10^5\}$, and the threshold η was set to exclude the 10% of samples with the largest training error as outliers.

[1] http://archive.ics.uci.edu/ml/index.php
[2] http://yann.lecun.com/exdb/mnist/
[3] https://github.com/zalandoresearch/fashion-mnist
[4] https://cs.nyu.edu/ ylclab/data/norb-v1.0/

TABLE I
SPECIFICATIONS OF THE BENCHMARK DATA SETS

Dataset	Training Data / target	Testing Data / target	No. of features	Class
Arrhythmia	225 / 122	227 / 123	274	2
Breast	341 / 222	342 / 222	9	2
g50c	440 / 220	110 / 55	50	2
optdigits	3823 / 376	1797 / 178	64	10
Sonar	103 / 55	105 / 56	60	2
Spectf heart	133 / 27	134 / 28	44	2
abalone	2088 / 703	2089 / 704	8	2
BASEHOCK	1195 / 596	798 / 398	4862	2
PCMAC	1166 / 589	777 / 393	3289	2
pendigits	7494 / 780	3498 / 363	16	10
RELATHE	856 / 467	571 / 312	4322	2
sat	3217 / 760	3218 / 773	36	7
USPS	7291 / 556	2007 / 160	256	10
HAR-4	7352 / 1286	2947 / 491	561	6
HAR-5	7352 / 1374	2947 / 532	561	6
HAR-6	7352 / 1407	2947 / 537	561	6
MNIST-2	60000 / 6742	10000 / 1135	10	10
MNIST-4	60000 / 6131	10000 / 1010	10	10
MNIST-5	60000 / 5842	10000 / 982	10	10
NORB-1/2/3	24300 / 4860	24300 / 4860	2048	5
Fashion-2/3/4/5	60000 / 6000	10000 / 1000	784	10
Fashion-579	60000 / 18000	10000 / 3000	784	10

The performance of each algorithm was quantified by the G-mean [25], which is computed as

$$\text{G-mean} = \sqrt{\frac{\text{TP}}{\text{TP}+\text{FN}} \times \frac{\text{TN}}{\text{TN}+\text{FP}}} \tag{21}$$

where TP, TN, FP, and FN are the true positive, true negative, false positive, and false negative, respectively. The results are reported as the average G-means of ten independent trials.

A. Comparisons of WSI-AE and ELM-AE

The performances of WSI-AE and ELM-AE [4] were compared on 13 small data sets, and the results are given in Table II. Here, a single AE was applied in HLS-OC, and the best results are highlighted in bold font. The proposed WSI-AE outperformed ELM-AE on 11 out of the 13 data sets and provided the same G-mean as ELM-AE on the sat data set. On some data sets, the G-mean was significantly higher in WSI-AE than in ELM-AE. Specifically, the increment of WSI-AE over ELM-AE was 4.2% on Arrhythmia, 20.64% on BASEHOCK, 11.52% on PCMAC, and 7.8% on USPS.

Fig. 2 shows a 2-D plot of the feature classifications (target versus outliers) obtained by ELM-AE and WSI-AE on the BASEHOCK and pendigits data sets. These plots were produced by principal component analysis (PCA). On the BASEHOCK data set, PCA could not find the discriminative features among the features encoded by ELM-AE but separated those encoded by the proposed WSI-AE [see Fig. 2 (c)]. On the pendigits data set, the features were better discriminated by ELM-AE than by WSI-AE [see Fig. 2(b) and (d)]. However, it is worth noting that the features encoded by ELM-AE were dispersed around the outliers, but those encoded by WSI-AE tended to be clustered as the target, mainly because the WSI-AE algorithm minimizes the within-class scatter. In contrast, the OCC seeks the minimum-radius hypersphere that encloses the target class [see (17) and (18)]. Thus, the G-means of the proposed WSI-AE and ELM-AE were comparable to this data set.

To further demonstrate the effectiveness of WSI-AE, we computed the ratio RW of the within-class scatter among the encoded features of the target samples to that of the outliers, and the ratio RB of the target within-class scatter to the between-class scatter of the encoded features

$$\text{RW} = \frac{\text{tr}\left(\mathbf{S}_T^{(k)}\right)}{\text{tr}\left(\mathbf{S}_O^{(k)}\right)}, \quad \text{RB} = \frac{\text{tr}\left(\mathbf{S}_T^{(k)}\right)}{\text{tr}\left(\mathbf{S}_b^{(k)}\right)} \tag{22}$$

where $\mathbf{S}_T^{(k)}$ and $\mathbf{S}_O^{(k)}$ are the target and outlier within-class scatter matrices, respectively, $\mathbf{S}_b^{(k)}$ is the between-class scatter matrix between the target and outlier classes, and k represents the kth AE. Obviously, the smaller the values of RW and RB, the more discriminative are the encoded features. Fig. 3 compares the RW and RB values of WSI-AE and ELM-AE on the BASEHOCK and pendigits data sets. In both comparisons, the proposed WSI-AE generally acquired smaller RW and RB values than ELM-AE.

TABLE II
G-MEAN (%) COMPARISONS ON SMALL-SIZE DATA SETS

Dataset	ELM-AE [4]	WSI-AE
Arrhythmia	63.92 ± 1.06	**68.12 ± 1.86**
Breast	95.39 ± 0.00	**95.72 ± 0.32**
g50c	71.82 ± 0.48	**74.15 ± 0.44**
optdigits	95.02 ± 1.08	**95.59 ± 0.51**
Sonar	57.28 ± 4.76	**60.01 ± 3.09**
Spectf heart	72.85 ± 0.00	**73.60 ± 1.37**
abalone	74.71 ± 0.02	**75.97 ± 0.78**
BASEHOCK	47.07 ± 1.57	**67.71 ± 3.00**
PCMAC	48.07 ± 0.95	**59.59 ± 1.23**
pendigits	**90.49 ± 0.00**	90.45 ± 0.26
RELATHE	55.21 ± 0.65	**56.51 ± 1.39**
sat	95.78 ± 0.00	95.78 ± 0.19
USPS	58.37 ± 2.57	**66.17 ± 0.23**

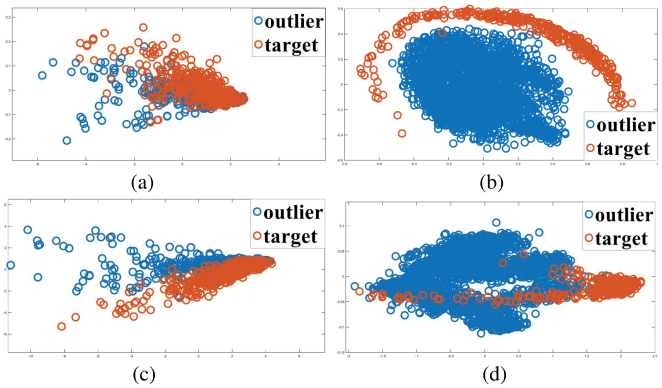

Fig. 2. Visualization of the 2-D data manifolds on the BASEHOCK (left) and pendigits (right) data sets, obtained by PCA in ELM-AE (top) and WSI-AE (bottom). (a) ELM-AE on BASEHOCK. (b) ELM-AE on pendigits. (c) WSI-AE on BASEHOCK. (d) WSI-AE on pendigits.

B. Comparisons of WSI-AE and State-of-the-Art AEs

Table III compares the G-means obtained by the ELM-AE [4], ELM-SAE [5], SMA [14], and the proposed WSI-AE algorithms. All methods were applied to the HLS-OC framework with two stacked AEs. As highlighted, the WSI-AE obtained the highest G-mean on 13 out of 14 data sets. Fig. 4 shows the RW and RB as functions of the number of AEs, obtained by the different methods on the HAR-4, NORB-1, and Fashion-2 data sets. The curves are plotted on a log-linear scale for clarity. The features encoded by WSI-AE generally achieved the lowest RW and RB on all three data sets. Fig. 5 shows the 2-D data manifolds obtained by PCA. The features encoded by WSI-AE were more discriminative than those encoded by the other AEs.

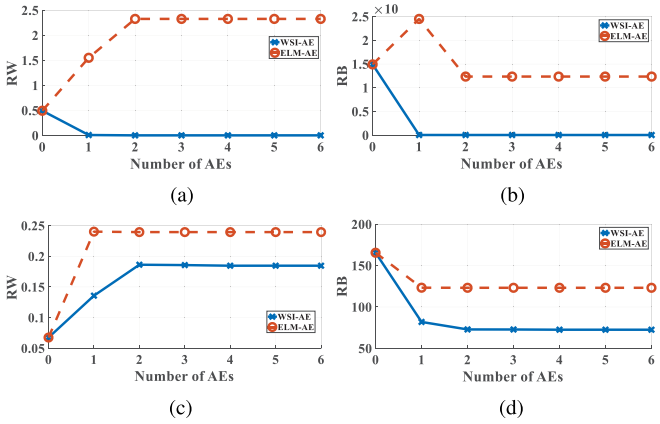

Fig. 3. Scatter comparisons of WSI-AE (blue) and ELM-AE (orange) on (a) and (b) BASEHOCK and (c) and (d) pendigits (bottom) data sets.

TABLE III
G-MEAN (%) COMPARISONS OF WSI-AE AND ELM-BASED AES

Dataset	ELM-AE [4]	ELM-SAE [5]	SMA [14]	WSI-AE
HAR-4	85.86 ± 0.52	86.15 ± 1.01	83.20 ± 1.26	**86.51 ± 0.12**
HAR-5	86.62 ± 0.22	84.85 ± 0.53	86.17 ± 1.66	**87.77 ± 0.23**
HAR-6	92.57 ± 0.54	78.95 ± 1.79	92.77 ± 0.64	**93.08 ± 0.08**
MNIST-2	88.58 ± 0.53	84.63 ± 1.21	86.78 ± 0.30	**90.63 ± 2.20**
MNIST-4	65.32 ± 5.65	57.39 ± 4.83	59.05 ± 4.89	**66.33 ± 0.05**
MNIST-5	69.79 ± 2.41	62.18 ± 1.15	72.94 ± 3.81	**77.00 ± 3.53**
NORB-1	79.72 ± 0.04	78.93± 0.64	79.40 ± 0.50	**81.55 ± 0.10**
NORB-2	92.48 ± 0.03	88.44 ± 1.14	90.52 ± 0.46	**92.98 ± 0.14**
NORB-3	66.5 ± 1.44	67.01 ± 0.56	**67.41 ± 1.84**	67.03 ± 1.56
Fashion-2	74.49 ± 1.76	63.7 ± 1.00	70.19 ± 2.67	**75.41 ± 0.08**
Fashion-3	76.37 ± 2.46	65.3 ± 7.56	72.90 ± 1.85	**77.2 ± 1.45**
Fashion-4	78.77 ± 0.40	66.51 ± 1.60	73.19 ± 0.81	**80.82 ± 0.09**
Fashion-5	71.15 ± 3.79	66.9 ± 1.00	71.16 ± 3.76	**72.67 ± 2.85**
Fashion-579	79.95 ± 2.17	83.1 ± 1.93	81.03 ± 2.62	**88.59 ± 0.77**

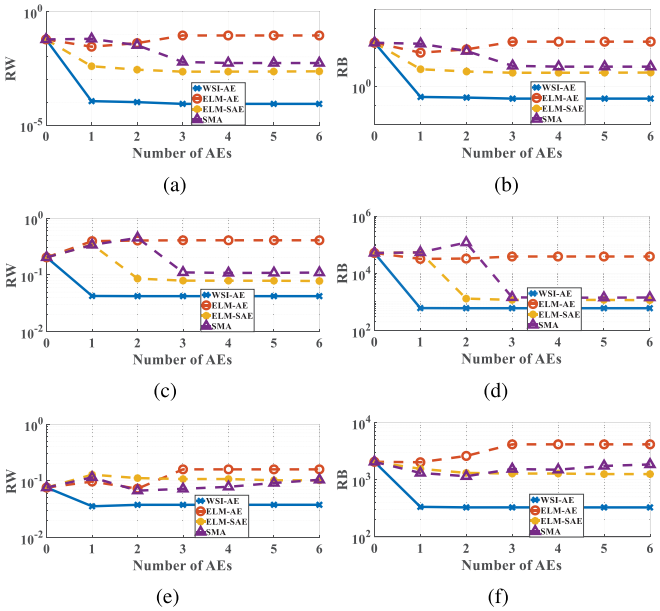

Fig. 4. Scatter comparisons of various methods on (a) and (b) HAR-4, (c) and (d) NORB-1, and (e) and (f) Fashion-2.

Next, the G-means of WSI-AE were compared with those of three state-of-the-art AEs, namely, WD-AE [2], DAE^B, and DAE^G [3]. The average G-means of the competitive AEs are presented in Fig. 6. The

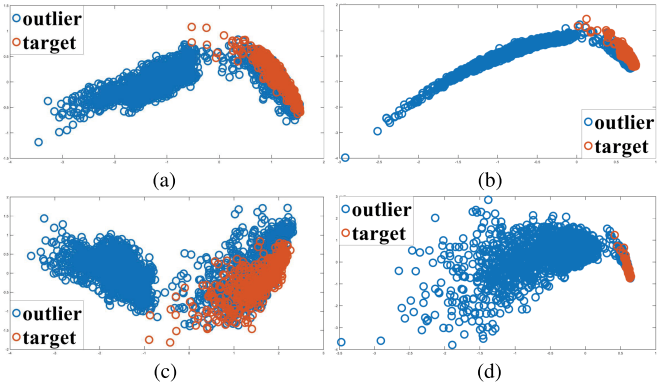

Fig. 5. Visualization of the 2-D data manifolds on the HAR-4 data set, obtained by PCA in (a) ELM-AE, (b) ELM-SAE, (c) SMA, and (d) WSI-AE algorithms.

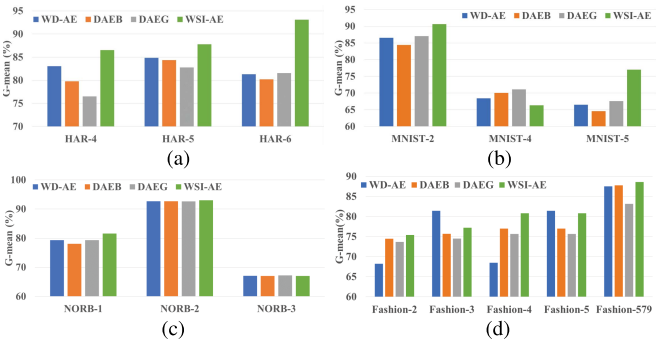

Fig. 6. Performance comparisons of WSI-AE and other state-of-the-art AE algorithms. (a) HAR. (b) MNIST. (c) NORB. (d) Fashion.

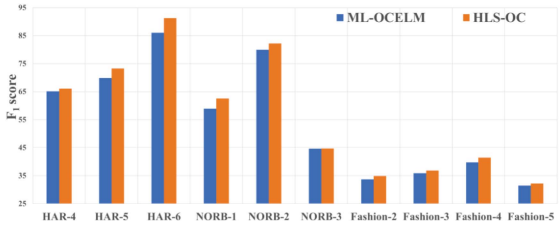

Fig. 7. $\mathbf{F}_1$ score comparisons of ML-OCELM and HLS-OC on various data sets.

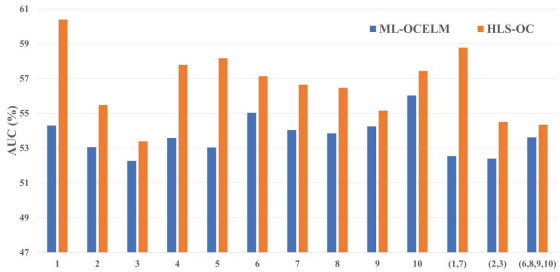

Fig. 8. AUC comparisons of ML-OCELM and HLS-OC on the SVHN data set.

WSI-AE algorithm generally outperformed the other AEs and offered the best performance on 10 out of the 14 data sets.

C. Comparisons of HLS-OC and State-of-the-Art OCC Algorithms

This section compares the performances of HLS-OC and other OCC algorithms. Table IV first shows the comparisons with the classical OCC algorithms. For a fair comparison, we adopted the

TABLE IV
$\mathbf{F}_1$ SCORE COMPARISONS OF WSI-AE AND CLASSICAL OCC ALGORITHMS

Dataset Classifier	Spectfheart	Arrhythmia	Sonar	Liver	Ecoli	Diabetes	Breast	Abalone
Naive Parzen	41.7 ± 4.2	61.8 ± 1.1	46.8 ± 2.2	41.5 ± 0.9	71.7 ± 7.0	68.7 ± 1.1	82.4 ± 3.2	51.8 ± 0.3
Parzen	39.3 ± 1.7	63.7 ± 1.2	49.8 ± 2.9	40.7 ± 1.4	75.1 ± 5.7	67.7 ± 1.4	80.1 ± 7.7	49.0 ± 1.0
k-means	38.3 ± 4.7	63.7 ± 1.7	53.2 ± 3.2	41.7 ± 1.4	54.6 ± 15.2	68.9 ± 1.1	58.8 ± 17.2	45.2 ± 1.3
1-NN	31.8 ± 2.6	59.2 ± 1.5	60.4 ± 2.2	41.3 ± 1.3	21.2 ± 3.9	64.8 ± 0.9	35.3 ± 5.8	35.7 ± 1.0
k-NN	34.7 ± 1.2	62.4 ± 0.9	55.3 ± 1.3	42.0 ± 1.2	43.9 ± 14.2	68.8 ± 1.0	34.9 ± 7.5	49.8 ± 1.4
Autoencoder	39.3 ± 3.4	64.8 ± 1.6	50.6 ± 2.4	42.2 ± 1.7	53.5 ± 15.8	66.9 ± 1.3	37.9 ± 10.9	48.7 ± 2.7
PCA	NaN	26.3 ± 5.3	37.2 ± 8.3	41.1 ± 1.3	33.7 ± 15.4	65.5 ± 1.9	31.1 ± 1.0	46.0 ± 0.5
MST	33.7 ± 1.7	62.4 ± 0.8	56.7 ± 1.8	42.1 ± 1.1	36.3 ± 12.9	67.5 ± 0.7	34.4 ± 3.4	47.3 ± 0.9
k-centers	36.4 ± 2.9	62.8 ± 1.2	53.3 ± 2.3	41.6 ± 1.3	38.8 ± 6.5	67.7 ± 1.1	49.4 ± 22.9	42.5 ± 2.6
SVDD	38.9 ± 4.7	60.5 ± 4.8	51.2 ± 5.8	40.6 ± 3.1	50.9 ± 10.6	61.1 ± 2.5	68.0 ± 9.1	44.5 ± 3.3
MPM	31.1 ± 8.7	51.9 ± 5.0	44.6 ± 6.3	40.7 ± 2.0	38.8 ± 11.8	63.5 ± 1.7	71.7 ± 6.5	44.9 ± 1.9
LPDD	38.3 ± 3.9	63.8 ± 2.0	52.2 ± 4.2	40.7 ± 1.6	67.8 ± 11.0	66.6 ± 0.8	79.6 ± 7.5	45.8 ± 2.0
OCSVM	38.1 ± 6.4	63.4 ± 1.9	53.6 ± 3.1	40.5 ± 2.4	57.2 ± 12.8	66.6 ± 1.1	83.1 ± 3.0	46.2 ± 1.2
OCELM	42.6 ± 1.8	63.6 ± 1.6	54.2 ± 3.5	43.0 ± 1.6	**77.1 ± 4.8**	69.1 ± 1.1	80.1 ± 5.1	53.0 ± 0.7
ML-OCELM	52.5 ± 0.0	**76.4 ± 0.2**	61.0 ± 4.0	50.0 ± 0.2	67.94 ± 6.15	71.9 ± 1.8	95.3 ± 0.0	67.9 ± 0.0
HLS-OC	**53.1 ± 1.3**	**76.4 ± 1.0**	**66.6 ± 8.0**	**57.0 ± 1.3**	47.50 ± 4.45	**77.4 ± 0.8**	**95.6 ± 0.3**	**69.0 ± 0.7**

TABLE V
AUC COMPARISONS OF HLS-OC AND STATE-OF-THE-ART OCC ALGORITHMS ON THE CIFAR-10 DATA SET

Target Class	Deep SVDD	LSAR	OCGAN	HLS-OC
AIRPLANE	61.7	73.5	**75.7**	73.7
AUTOMOBILE	65.9	58.0	53.1	**74.4**
BIRD	50.8	**69.0**	64.0	60.9
CAT	59.1	54.2	62.0	**63.9**
DEER	60.9	**76.1**	72.3	71.0
DOG	**65.7**	54.6	62.0	64.1
FROG	67.8	75.1	72.3	**78.8**
HORSE	67.3	53.5	57.5	**70.6**
SHIP	75.9	71.7	**82.0**	81.8
TRUCK	73.1	54.8	55.4	**79.0**

experimental setup and evaluation metric ($\mathbf{F}_1$ score) in [26]. The multilayer one-class ELM (ML-OCELM) [22] and the proposed HLS-OC each employed a single AE. As confirmed by the $\mathbf{F}_1$ scores in Table IV, the HLS-OC with the proposed WSI-AE generally achieved the best performance. The same result is observed in Fig. 7, which visually compares the $\mathbf{F}_1$ scores obtained by the proposed HLS-OC and ML-OCELM on the HAR, NORB, and Fashion data sets. Subsequently, the performances of HLS-OC and ML-OCELM were compared on the SVHN[5] data set of cropped digits, in which 531 131 extra samples are employed as the training data. As the targets, we assigned each class in the SVHN data set and the merged (1, 7), (2, 3), and (6, 8, 9, 10) classes. The number of samples for OCC training is up to 177 325. The performances were evaluated by the area under the curve (AUC) of the receiver operating characteristic, and the comparison results are presented in Fig. 8. The HLS-OC clearly outperformed ML-OCELM.

To further demonstrate its performance, the HLS-OC algorithm has competed against the Deep SVDD [19], LSAR [20] and OCGAN [21] OCC algorithms on the CIFAR-10[6] data set. The experimental setups and results were those in [19]–[21], and three WSI-AEs were stacked for feature learning in HLS-OC. The proposed HLS-OC algorithm obtained the highest AUC value in five out of ten categories (see Table V). Especially, HLS-OC outperformed Deep SVDD [19] on all data sets except the DOG class.

V. CONCLUSION

To enhance the feature discriminability and reduce the meaningless encoded features in standard AEs, we proposed a novel WSI-AE. We then developed a hierarchical OCC algorithm using stacked WSI-AEs for feature learning, followed by an OCC for target classification. The effectiveness of the resultant HLS-OC algorithm was demonstrated in experiments on 19 benchmark data sets, and its accuracy was compared with those of several representative AEs and state-of-the-art OCC algorithms. The comparisons demonstrated the superiority of the proposed WSI-AE and HLS-OC. In future work, we will introduce our highly parallel alternating direction method of multipliers algorithm [13] to WSI-AE, which will reduce the memory cost and training time of the present WSI-AE. The WSI-AE will also be improved for multiclass problems by introducing the between-class scatter information.

[5]http://ufldl.stanford.edu/housenumbers/

[6]https://www.cs.toronto.edu/ kriz/cifar.html

REFERENCES

[1] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proc. ICML Workshop Unsupervised Transf. Learn.*, 2012, pp. 37–49.

[2] D. Charte, F. Charte, S. García, M. J. del Jesus, and F. Herrera, "A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines," *Inf. Fusion*, vol. 44, pp. 78–96, Nov. 2018.

[3] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[4] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Nov. 2013.

[5] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2016.

[6] C. M. Wong, C. M. Vong, P. K. Wong, and J. Cao, "Kernel-based multilayer extreme learning machines for representation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 757–762, Mar. 2018.

[7] W. F. Schmidt *et al.*, "Feed forward neural networks with random weights," in *Proc. Int. Conf. Pattern Recognit. Methodol. Syst.*, vol. 2, 1992, pp. 1–4.

[8] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[9] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[10] L. Zhang and D. Zhang, "Evolutionary cost-sensitive extreme learning machine," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 3045–3060, Dec. 2017.

[11] H. Yu, X. Yang, S. Zheng, and C. Sun, "Active learning from imbalanced data: A solution of online weighted extreme learning machine," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1088–1103, Apr. 2019.

[12] J. Cao, K. Zhang, H. Yong, X. Lai, B. Chen, and Z. Lin, "Extreme learning machine with affine transformation inputs in an activation function," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2093–2107, Jul. 2019.

[13] X. Lai, J. Cao, X. Huang, T. Wang, and Z. Lin, "A maximally split and relaxed ADMM for regularized extreme learning machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 1899–1913, Jun. 2020.

[14] T. Wang, X. Lai, J. Cao, C.-M. Vong, and B. Chen, "An enhanced hierarchical extreme learning machine with random sparse matrix based autoencoder," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3817–3821.

[15] R. Li, X. Wang, L. Lei, and C. Wu, "Representation learning by hierarchical elm auto-encoder with double random hidden layers," *IET Comput. Vis.*, vol. 13, no. 4, pp. 355–368, 2019.

[16] K. Sun, J. Zhang, C. Zhang, and J. Hu, "Generalized extreme learning machine autoencoder and a new deep neural network," *Neurocomputing*, vol. 230, pp. 374–381, Mar. 2017.

[17] L. Yang, S. Song, S. Li, Y. Chen, and G. Huang, "Graph embedding-based dimension reduction with extreme learning machine," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 22, 2019, doi: 10.1109/TSMC.2019.2931003.

[18] Y. Yang, Q. M. J. Wu, and Y. Wang, "Autoencoder with invertible functions for dimension reduction and image reconstruction," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 7, pp. 1065–1079, Jul. 2018.

[19] L. Ruff *et al.*, "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4390–4399.

[20] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 481–490.

[21] P. Perera, R. Nallapati, and B. Xiang, "OCGAN: One-class novelty detection using GANs with constrained latent representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2898–2906.

[22] H. Dai, J. Cao, T. Wang, M. Deng, and Z. Yang, "Multilayer one-class extreme learning machine," *Neural Netw.*, vol. 115, pp. 11–22, Jul. 2019.

[23] L. L. C. Kasun, Y. Yang, G.-B. Huang, and Z. Zhang, "Dimension reduction with extreme learning machine," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3906–3918, Aug. 2016.

[24] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vis. Res.*, vol. 37, no. 23, pp. 3327–3338, 1997.

[25] T. Wang, J. Cao, X. Lai, and B. Chen, "Deep weighted extreme learning machine," *Cognit. Comput.*, vol. 10, no. 6, pp. 890–907, Dec. 2018.

[26] Q. Leng, H. Qi, J. Miao, W. Zhu, and G. Su, "One-class classification with extreme learning machine," *Math. Problems Eng.*, vol. 2015, pp. 1–11, May 2015.