Knowledge-Based Systems 196 (2020) 105753

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Layer-constrained variational autoencoding kernel density estimation model for anomaly detection ☆

Peng Lv [b], Yanwei Yu [a,b,*], Yangyang Fan [c], Xianfeng Tang [d], Xiangrong Tong [b]

[a] Department of Computer Science and Technology, Ocean University of China, Qingdao, Shandong 266100, China
[b] School of Computer and Control Engineering, Yantai University, Yantai, Shandong 264005, China
[c] Shanghai Key Lab of Advanced High-Temperature Materials and Precision Forming, Shanghai Jiao Tong University, Shanghai 200240, China
[d] College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802, USA

ARTICLE INFO

Article history:
Received 16 October 2019
Received in revised form 5 March 2020
Accepted 7 March 2020
Available online 10 March 2020

Keywords:
Anomaly detection
Variational autoencoder
Kernel density estimation
Layer constraint
Deep learning

ABSTRACT

Unsupervised techniques typically rely on the probability density distribution of the data to detect anomalies, where objects with low probability density are considered to be abnormal. However, modeling the density distribution of high dimensional data is known to be hard, making the problem of detecting anomalies from high-dimensional data challenging. The state-of-the-art methods solve this problem by first applying dimension reduction techniques to the data and then detecting anomalies in the low dimensional space. Unfortunately, the low dimensional space does not necessarily preserve the density distribution of the original high dimensional data. This jeopardizes the effectiveness of anomaly detection. In this work, we propose a novel high dimensional anomaly detection method called LAKE. The key idea of LAKE is to unify the representation learning capacity of layer-constrained variational autoencoder with the density estimation power of kernel density estimation (KDE). Then a probability density distribution of the high dimensional data can be learned, which is able to effectively separate the anomalies out. LAKE successfully consolidates the merits of the two worlds, namely layer-constrained variational autoencoder and KDE by using a probability density-aware strategy in the training process of the autoencoder. Extensive experiments on six public benchmark datasets demonstrate that our method significantly outperforms the state-of-the-art methods in detecting anomalies and achieves up to 37% improvement in F_1 score.

1. Introduction

Anomaly detection is a fundamental and hence well-studied problem in many areas, including cyber-security [1], manufacturing [2], system management [3], and medicine [4]. The core of anomaly detection is density estimation whether it is high-dimensional data or multi-dimensional data. In general, normal data is large and consistent with certain distribution, while abnormal data is small and discrete, therefore anomalies are residing in low density areas.

Although excellent progress have been achieved in anomaly detection in the past decades, anomaly detection of complex and high-dimensional data remains to be a challenge. It is hard to implement density estimation in original data space with the increasing of dimensionality, because as the data dimension increases, noise and extraneous features have a more negative impact on density estimation. But unfortunately for a real-world problem, the dimensionality of data could be very large, such as video surveillance [5], medical anomaly detection [6], and cyber-intrusion detection [7]. To address this issue, a two-step approach [4] is usually applied and has proved to be successful. It first reduces the dimensionality of data and then adopt density estimation in the latent low-dimensional space. Additionally, spectral anomaly detection [8–10] and alternative dimensionality reduction [11–13] techniques are implemented to find the lower dimensional representation of the original high-dimensional data, where anomalies and normal instances are expected to be separated from each other. However, the low dimensional space does not necessarily preserve the density distribution of the original data, and thus it is not able to effectively identify the anomalies in high-dimensional data by estimating the density in low-dimensional space.

Recently, deep learning has achieved great success in anomaly detection [7]. Autoencoder [14] and a range of variants have been

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105753.

* Correspondence to: 238 Songling RD, Laoshan District, Qingdao, Shandong 266100, China
E-mail addresses: lvpeng4869@outlook.com (P. Lv), yuyanwei@ouc.edu.cn (Y. Yu), yfan.mse@gmail.com (Y. Fan), xianfeng@ist.psu.edu (X. Tang), txr@ytu.edu.cn (X. Tong).

https://doi.org/10.1016/j.knosys.2020.105753
0950-7051/

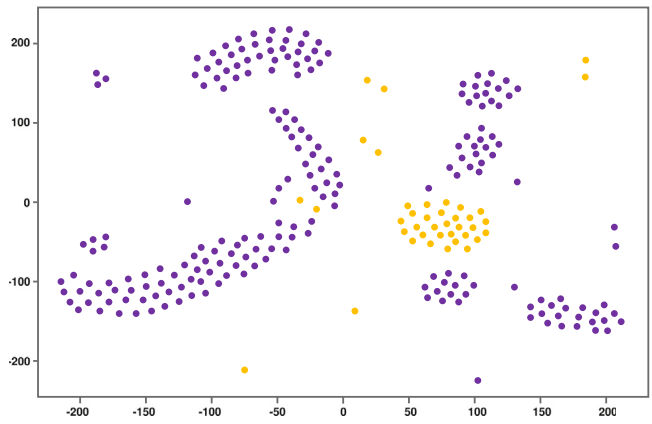


Fig. 1. An example of low-dimensional representation for samples from KDDCUP dataset: (1) purple/yellow points are normal/anomalous points; (2) use layer-constrained variational autoencoder to reduce the dimension and display them with t-SNE. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

widely used for unsupervised anomaly detection, such as deep autoencoder, variational autoencoder (VAE) [15], and adversarial autoencoder (AAE) [16]. The core idea of these methods is to encode input data into a low dimensional representation, and then decode the low dimensional representation into the original data space by minimizing the reconstruction error. In this process, the essential features of the original data are extracted in latent data space through training autoencoder, without noise and unnecessary features. Several recent studies have applied this structure into practical problems, but yet there remains largely unexplored. For example, the feature descriptor is to use an autoencoder to learn robust features for human appearance in the study of re-identification [17–19]. In the study of anomaly detection, AnoGan [6] uses adversarial autoencoder to detect anomaly in image data. But it only takes advantage of the reconstruction error and does not make full use of the low-dimensional representation. ALAD [20] considers both data distribution and reconstruction error based on bi-directional GANs, which derives adversarially learned features for the anomaly detection task. Nevertheless, ALAD still only uses reconstruction errors based on the adversarially learned features to determine if a data sample is anomalous. DAGMM [21] combines deep autoencoder and Gaussian mixture model (GMM) in anomaly detection. However, the real-world data may not only have high dimensions, but also is lacking of a clear predefined distribution. Manual parameter adjustment is also required in GMM when modeling the density distribution of input data, which has a serious impact on detection performance.

Furthermore, as the example shown in Fig. 1, although the anomalous points are separated from the normal points in the low-dimensional representation space using autoencoder model, the distribution of normal data may be arbitrary, rather than one kind of prior distribution (e.g., GMM). On the other hand, some anomaly data may show the distribution of dense clusters. This is an intractable problem both for neighbor-based and energy-based anomaly detection methods. Additionally, there are always some normal points discretely distributed near normal dense clusters in space. These factors also pose severe challenges for anomaly detecting from large-scale high-dimensional data.

In this paper, we propose a novel Layer-constrained variational Autoencoding Kernel density Estimation model (LAKE), a deep learning framework that addresses the aforementioned challenges in anomaly detection from high-dimensional datasets. LAKE is a *probability density-aware* model, which unifies the presentation learning capacity of layer-constrained variational autoencoder with the density estimation power of KDE to provide a probability density estimation of high-dimensional data for effectively identifying anomalies.

On the one hand, we propose a layer-constrained variational autoencoder to obtain a low-dimensional representation of the input data which contains the nature of input data. Different from the standard VAE, layer-constrained variational autoencoder considers the reconstruction errors on all corresponding layers of the encoder and decoder and keeps KL divergence unchanged. Since layer-constrained variational autoencoder takes account of both reconstruction error and the distribution of data in the latent data space, the density distribution of high dimensional data is preserved in low dimensional representation. On the other hand, LAKE uses KDE to estimate the probability density distribution of training data. Unlike DAGMM, which needs to manually specify the number of mixed Gaussian models, LAKE can model arbitrary distributed data sophisticatedly. We even flexibly choose kernel function in the KDE model to appropriately simulate the probability density distribution of data. As layer-constrained VAE encodes input data into low-dimensional representations while preserving the key features of input data, the one with a high density value is more likely to be a normal object, while the low one is considered to be an abnormal object.

However, as shown in Fig. 1, some abnormal objects may form a dense cluster due to their common anomalous characteristics. Such abnormal objects may not be detected by simply applying density estimation based on global data, because there are always some normal objects fall in the distribution margin discretely. But fortunately, for each individual abnormal object, it can be easily distinguished from the density distribution of sampled training data separately by estimating its density value in the trained KDE model. Therefore, we propose a *probability density estimation* strategy in the training and testing process. Specifically, we use sampled training data to learn a probability density distribution in LAKE. In terms of testing, we estimate the density value for each data object separately based on the trained probability density distribution.

Extensive experiments on six public benchmark datasets demonstrate that LAKE has superior performance compared to the state-of-the-art models, with up to 37% improvement in standard F_1 score for anomaly detection. It is worth noting that LAKE achieves better results with fewer training samples compared to existing methods based on deep learning.

To summarize, we make the following contributions:

- We propose a layer-constrained variational autoencoding kernel density estimation model for anomaly detection from high-dimensional datasets.
- We propose a *probability density-aware* strategy that learns a probability density distribution of the high-dimensional data in the training process that is able to effectively detect abnormal objects in the testing.
- We conduct extensive evaluations on six benchmark datasets. Experimental results demonstrate that our method significantly outperforms the state-of-the-art methods.

2. Related work

Varieties of research focus on anomaly detection in data mining and machine learning [22]. Distance-based anomaly detection [23] uses global density criterion to detect anomalies. Density-based methods [24,25] aim to detect local outliers, and thus they use local relative density as anomaly criterion. Several studies [26–28] apply KDE into density-based local outlier detection to improve the detection accuracy. However, such methods

rely on an appropriate distance metric, which is only feasible for handling low-dimensional data, but not for anomaly detection of high dimensional data. One-class classification approaches trained by normal data, such as one-class SVMs [11,29], are also widely used for anomaly detection. These methods use kernel methods to learn a decision boundary around the normal data. Another line of studies use fidelity of reconstruction to determine whether a sample is anomalous, such as conventional Principal Component Analysis (PCA), kernel PAC, and Robust PCA (RPCA) [12,30]. Several recent researches [31,32] apply model fitting to estimate model hypotheses for multistructure data with high outlier rates.

In recent years, varieties of anomaly detection methods based on deep neural networks are proposed to detect anomalies [7]. Autoencoder and its series of variants have been widely used in unsupervised anomaly detection, especially for high-dimensional data anomaly detection. Inspired by RPCA [12], Zhou et al. [14] design a Robust Deep Autoencoder (RDA), and use the reconstruction error to detect anomalies for high-dimensional data. The variational autoencoder is used directly for anomaly detection by using reconstruction error in [15]. With the rise of adversarial networks, more models have added adversarial networks to autoencoders. AnoGAN [33] uses a Generative Adversarial Network (GAN) [34] to detect anomalies in the context of medical images by reconstruction error. In a follow-up work, f-AnoGAN [35] introduces Wasserstein GAN (WGAN) [36] to improve AnoGAN to be adaptable to real-time anomaly detection applications. The above methods can be classified as reconstruction based anomaly detection method. However, the performance of reconstruction based methods is limited by the fact that they only consider reconstruction error as anomaly criterion.

Deep structured energy based model (DSEBM) [37] addresses the anomaly detection problem by directly modeling the data distribution with deep architectures. DSEBM integrates Energy-Based Models (EBMs) with different types of data such as static data, sequential data, and spatial data, and apply appropriate model architectures to adapt to the data structure. DSEBM has two decision criteria for performing anomaly detection: the energy score (DSEBM-e) and the reconstruction error (DSEBM-r). Deep Autoencoding Gaussian Mixture Model (DAGMM) [21] utilizes a deep autoencoder to generate a low-dimensional representation and reconstruction error for each input data point, which is further fed into a Gaussian Mixture Model. These methods use autoencoder to model the distribution of data, and then derive the criteria for determining anomalies through an energy model or a Gaussian mixture model. As GANs are able to model the complex high-dimensional distributions of real-world data, so that Adversarially Learned Anomaly Detection (ALAD) [20] derives adversarially learned features based on bi-directional GANs, and then uses reconstruction errors based on these learned features for the anomaly detection task.

Our proposed model is most related to DAGMM. However, unlike DAGMM, LAKE uses a layer-constrained variational autoencoder to extract useful features while preserving the data distribution of the original data for anomaly detection. And LAKE leverages KDE to model the arbitrary density distribution of training samples in the latent data space without parameter dependence, rather than a predefined GMM distribution. Most importantly, LAKE estimates the probability density value for each test sample based on the trained KDE model individually, and show a powerful ability of anomaly detection with few training samples.

3. The proposed LAKE model

3.1. Overview

Fig. 2 shows the architecture of the proposed layer-constrained variational autoencoding kernel density estimation model. LAKE is mainly composed of two parts: compression network and probability density estimation model. First, in the compression network, LAKE compresses the input data to obtain their low-dimensional representations in latent data space by a proposed layer-constrained variational autoencoder, and together with reconstruction errors they are fed to the probability density estimation model. Second, the estimation model takes the feeds and learns a probability density distribution using a Gaussian kernel density estimation.

3.2. Compression network

The compression network in LAKE is a layer-constrained variational autoencoder (LVAE). The low-dimensional representation of the original data in latent data space is derived from the layer-constrained variational autoencoder.

Variational autoencoder is a probabilistic graphical model that combines variational interference with deep learning [38,39], which includes an encoder and a decoder. For a given input data x, the variational autoencoder calculates its low-dimensional representation z as follows:

$$z = q(x, \theta), \tag{1}$$

$$\hat{x} = p(z, \phi), \tag{2}$$

where $q_\theta(z|x)$ denotes the encoder, $p_\phi(x|z)$ denotes the decoder, θ and ϕ are the network parameters of the encoder and decoder, and $\hat{x}$ is the reconstruction of original data.

The loss function of variational autoencoder is the negative log-likelihood with a regularizer. Since there are no global representations that are shared by all data points, we can decompose the loss function into only terms that depend on a single data point l_i. The total loss is then $\sum_{i=1}^{N} l_i$ for N total data points. The loss function l_i for data point x_i is:

$$l_i(\theta, \phi) = -E_{z\sim q_\theta(z|x_i)}\left[\log p_\phi\left(x_i|z\right)\right] + KL\left(q_\theta\left(z|x_i\right) \parallel p(z)\right), \tag{3}$$

where the first term is the reconstruction loss, or expected negative log-likelihood of the ith data point. The second term is the Kullback–Leibler (KL) divergence between the distribution $q_\theta(z|x)$ and $p(z)$. This divergence measures how much information is lost (in units of nats) when using q to represent p. It is one measure of how close q is to p.

To enhance the representation learning capacity of compression network, we propose a layer-constrained variational autoencoder (LVAE). Unlike the standard VAE that only reconstructs loss at the input and output layers, LVAE considers reconstruction losses on all corresponding layers of the encoder and decoder and keeps KL divergence unchanged. The advantages of our LVAE model are twofold: First, layer constraints enhance the reconstruction ability of compression network by minimizing the information losses of input data between each pair of corresponding layers, which retains the essential features of original data in the low-dimensional representation as much as possible. Second, our layer constrained model would make the reconstruction error of the training samples smaller, which makes it easier to distinguish anomalies with the reconstruction error in the testing process.

Let k denote the number of corresponding layers in the encoder and decoder, then our loss function is:

$$l_i(\theta, \phi) = -\sum_{j=1}^{k} E_{z\sim q_\theta\left(z|x_i^j\right)}\left[\log p_\phi\left(x_i^j|z\right)\right] + KL\left(q_\theta\left(z|x_i\right) \parallel p(z)\right), \tag{4}$$

where x_i^j is the representation of data point x_i in jth layer.

Although we use a LVAE as our compression network, we do not directly use the low-dimensional representation of LVAE.

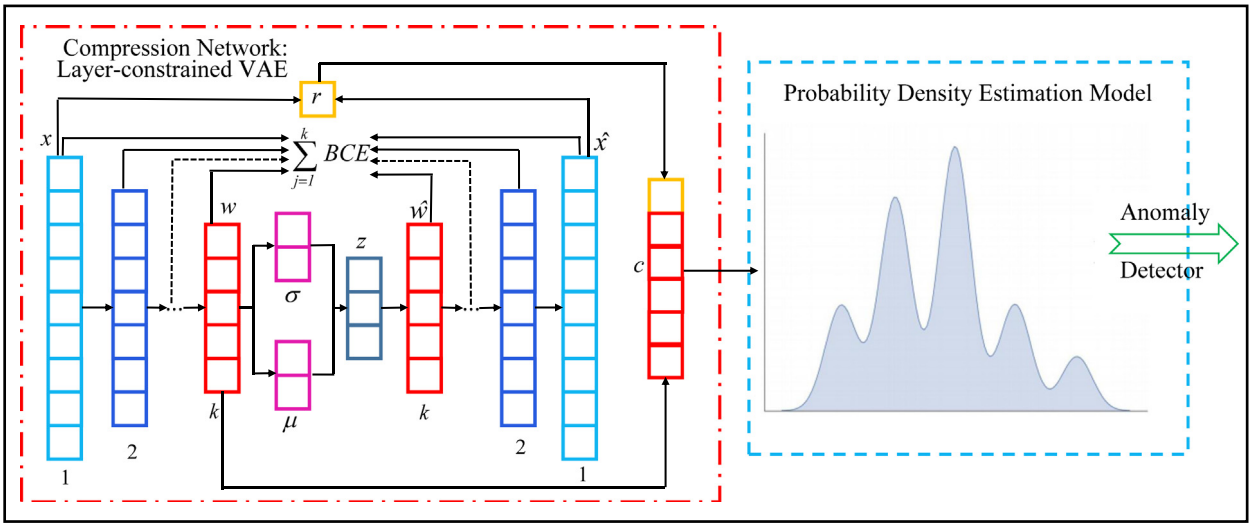


Fig. 2. An overview on layer-constrained variational autoencoding kernel density estimation model.

We further split the encoder $q_\theta(z|x)$ in LVAE into three steps. As shown in Eqs. (5)–(7), Eq. (5) first generates a low-dimensional representation w. Then Eq. (6) generates μ and σ, which describe the mean and variance of the potential state distribution. Eq. (7) samples from this distribution to obtain the potential representation z to reconstruct the original data. The above three formulas together constitute the encoder in LVAE, thus the parameters θ are shared.

$$w = f_e(x, \theta), \tag{5}$$

$$\mu, \sigma = f_r(w, \theta), \tag{6}$$

$$z = f_{re}(\mu, \sigma, \theta). \tag{7}$$

The general model uses z from Eq. (7) as the low-dimensional representation, but we use w from Eq. (5) as the low-dimensional representation. Since z is randomly sampled from the distribution obtained by Eq. (6), which is effective to enhance the generalization ability of LVAE model, but is not conducive to our next density estimation. w is the unique direct projection for the input data, which is more suited to being used for density estimation in the latent data space.

In addition to the low-dimensional representation of input data, we also consider the reconstruction error of LVAE for next anomaly detection. Therefore, the output of compression network is as follows:

$$c = [w, r], \tag{8}$$

$$r = [rec\_euclidean, rec\_cosine], \tag{9}$$

where r represents reconstruction error features, which can be multi-dimensional, considering multiple distance metrics such as Euclidean distance and cosine similarity. LAKE uses relative Euclidean distance and cosine similarity together as reconstruction features r.

3.3. Probability density estimation model

In probability density estimation model, we use the learned low-representation to model a probability density distribution of input data. We denote the low-dimensional representations obtained by Eq. (8) for the input data by $c_1, c_2, \ldots, c_n$. We next calculate their probability density distribution through KDE model.

Following the kernel density estimation model, for the feeds from compression network, the probability density distribution function of data is as follows:

$$\mathbf{f}_h(s) = \frac{1}{n}\sum_{i=1}^{n} K_h(s - c_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{s - c_i}{h}\right), \tag{10}$$

where s is the variable, and K is the kernel (a non-negative function) and h ($h > 0$) is a smoothing parameter called the bandwidth. In this paper, we adopt Gaussian kernel function in the KDE model.

3.4. Training process

This section mainly introduces the training process of proposed LAKE. Algorithm 1 shows the training process for our compression network and probability density estimation model. The training samples are denoted by $x_i (i = 0, 1, 2, \ldots, n)$.

Algorithm 1 LAKE training process

Input: Training dataset x_i ($x_1, x_2, \ldots, x_n$).
Output: LAKE model

$\theta, \phi \leftarrow$ Encoder and decoder parameters
for e from 1 to *epochs* **do** ▷ Compression network training
  for i from 1 to n **do**
    $w_i = f_e(x_i, \theta)$
    $\mu_i, \sigma_i = f_r(w_i, \theta)$
    $z_i = f_{re}(\mu_i, \sigma_i, \theta)$
    $\hat{x}_i = p(z_i, \phi)$
  KLD = $\frac{1}{2}\sum_{i=1}^{n}\left(1 + \log\left((\sigma_i)^2\right) - (\mu_i)^2 - (\sigma_i)^2\right)$
  BCE = $\sum_{i=1}^{n}(\sum_{j=1}^{k} binary\_cross\_entropy(\hat{x}_i^j, x_i^j))$
  loss function = KLD + BCE
  $\theta, \phi \leftarrow$ update parameters using gradients of SGD
θ, ϕ fixed
for i from 1 to n **do** ▷ Probability density estimation training
  $w_i = f_e(x_i, \theta)$
  $rec\_euclidean_i = relative\_euclidean\_distance(x_i, \hat{x}_i)$
  $rec\_cosine_i = cosine\_similarity(x_i, \hat{x}_i)$
  $r_i = [rec\_euclidean_i, rec\_cosine_i]$
  $c_i = [w_i, r_i]$
$\mathbf{f}_h(s) = \frac{1}{n}\sum_{i=1}^{n} K_h(s - c_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{s-c_i}{h}\right)$

First, we initialize the parameters θ and ϕ of the encoder and decoder. We denote the number of training rounds as epochs.

Specifically, in each epoch, for each input point, line 4 gets the low-dimensional representation from encoder; line 5 gets the mean and variance of the data distribution; line 6 gets the low-dimensional representation z; line 7 gets the decoder's reconstruction. KLD is the Kullback–Leibler divergence. Specifically, we use a binary_cross_entropy loss function as the reconstruction error for each pair of corresponding layers, and the sum of reconstruction errors of all layer pairs in LVAE constitutes the Binary Cross Entropy (BCE) loss (line 9). KLD and BCE together form the loss function, and we use the Stochastic Gradient Descent (SGD) to update the parameters θ and ϕ.

Our probability density estimation model is based on KDE model. We first use a trained LVAE to get a low dimensional representation c_i of the training data x_i (lines 13–18). Among them, we calculate the relative Euclidean distance between the input data x and the reconstruction $\hat{x}$ in line 15 and the cosine similarity between the input data x and the reconstruction $\hat{x}$ in line 16. Then, we get a probability density distribution function $\mathbf{f}_h(s)$ for simulating the distribution of training data in the latent space, as shown in line 19.

Our model can capture the arbitrary density distribution of training data through KDE model without tuning parameter.

3.5. Testing strategy

This section introduces the testing strategy for the proposed LAKE model. As showed in Fig. 1, even though abnormal objects are separated from normal data in low dimensional representation, some anomalies may form dense clusters due to their common anomalous characteristics preserved in the low-dimensional representation. On the other hand, there are always some normal objects that discretely distributed at the edges of the normal data dense regions. These edge normal objects may have a relative low density compared to the dense anomalies. Therefore, such abnormal objects and edge normal objects cannot be effectively detected and differentiated by the density estimation based on global data. But fortunately, we learn a probability density distribution of sampled training data in training LAKE model. For each individual abnormal object, it can be easily distinguished from the probability density distribution of training samples by estimating its density value in this probability density distribution. This is because: (1) The training set is used not only to train the parameters of LVAE, but also to generate a probability density distribution $\mathbf{f}_h(s)$; (2) The density of each test sample is not calculated based on the entire test data distribution, but based on the trained probability density distribution; (3) The vast majority of the training samples are normal samples, that is, the trained probability density distribution of training samples approximates the distribution of normal samples; (4) This probability density distribution takes into account not only the latent representation but also the reconstruction error features. The anomalous objects usually have a relatively large reconstruction error [6,21,37]. Therefore, when each test sample is independently mapped on this probability density distribution, the edge normal objects are closer to the distribution than the dense abnormal objects, that is, the edge normal samples can instead obtain higher density values in our probability density estimation model.

Algorithm 2 shows the pseudo-code of our testing strategy. First, we extract the low-dimensional representation of test data using the compression network (lines 1–6), and estimate the KDE value for each low-dimensional representation c_j using the trained probability density distribution $\mathbf{f}_h(s)$ of training samples (lines 7–8). Next, a threshold *thr* is calculated based on the abnormal ratio α and the sorted list of $\mathbf{f}_h(c)$ values in ascending order (lines 9). Here function $sort_{thr}()$ represents sorting all density values and determines the density threshold *thr* according to the abnormal ratio α. The test data points whose KDE values are higher than the threshold are normal data, otherwise they are abnormal data (lines 10–14).

Algorithm 2 LAKE testing process

Input: Testing dataset y_j(j=1, 2, 3, ..., m), abnormal ratio α.
Output: Anomalies
1: **for** j from 1 to m **do**
2: $w_j = f_e(y_j, \theta)$
3: $rec\_euclidean_j = relative\_euclidean\_distance(y_j, \hat{y}_j)$
4: $rec\_cosine_j = cosine\_similarity(y_j, \hat{y}_j)$
5: $r_j = [rec\_euclidean_j, rec\_cosine_j]$
6: $c_j = [w_j, r_j]$
7: **for** j from 1 to m **do**
8: $f(c_j) = \mathbf{f}_h(c_j)$
9: $thr = sort_{thr}\ (f(c), \alpha)$
10: **for** j from 1 to m **do**
11: **if** $f(c_j) < thr$ **then**
12: y_j is an anomaly
13: **else**
14: y_j is not an anomaly

4. Experiments

In this section, we use six public benchmark datasets to evaluate the effectiveness and robustness of our proposed model in anomaly detection. The code of the baseline methods is available at GitHub[1] released by ALAD. The source code of our proposed method is available at GitHub.[2]

4.1. Datasets

We use six well-known public benchmark datasets in the field of anomaly detection: KDDCUP, Thyroid, Arrhythmia, KDDCUP-Rev, SpamBase, and Cardiotocography, which are also used in [20,21,37].

- **KDDCUP**: The KDDCUP 10% dataset from UCI Machine Learning Repository[3] is a network intrusion dataset. We use one-hot representation to encode them, and eventually obtain a 118-dimensional dataset. As 20% of them are marked as "normal" and meanwhile others are marked as "attack", and "normal" samples constitute a small portion, therefore, we treat "normal" samples as anomalies in our experiment.
- **Thyroid**: Thyroid is from the UCI Machine Learning Repository thyroid disease classification dataset, which contains samples of 36 dimensions. There are 3 classes in original dataset. As hyperfunction is a minority class, we treat hyperfunction as anomaly class in our experiment.
- **Arrhythmia**: Arrhythmia dataset is also obtained from the UCI Machine Learning Repository, which contains 274 attributes. The smallest classes, including 3, 4, 5, 7, 8, 9, 14 and 15, are combined to form the anomaly class, and the rest of the classes are combined to form the normal class.
- **KDDCUP-Rev**: This dataset is an abbreviated version extracted from KDDCUP. We retain all "normal" data in this dataset, and randomly draw "attack" samples to keep the anomaly ratio as 0.2. As "attack" data is in minority part, we treat "attack" data as anomalies.

[1] https://github.com/houssamzenati/Adversarially-Learned-Anomaly-Detection.
[2] https://github.com/1246170471/LAKE.
[3] https://archive.ics.uci.edu/ml/.

Table 1
Statistics of the public benchmark datasets.

Dataset	#Dimensions	#Instances	Anomaly ratio (α)
KDDCUP	118	494,021	0.2
Thyroid	36	3772	0.025
Arrhythmia	274	432	0.15
KDDCUP-Rev	118	121,597	0.2
SpamBase	58	3485	0.2
Cardiotocography	22	2068	0.2

- **SpamBase**: SpamBase from UCI Machine Learning Repository includes 3485 emails classified as spam or non-spam. This dataset has 58 attributes. We treat spam as outliers. The anomaly ratio is 0.2.
- **Cardiotocography**: Cardiotocography data is also from UCI Machine Learning Repository and is related to heart diseases. This dataset contains 22 attributes. It describes 3 classes: normal, suspect, and pathological. Normal patients are treated as inliers and the remaining as outliers. The anomaly ratio as 0.2.

The details of the datasets are shown in Table 1.

4.2. Baseline methods

We compare our method with the following traditional and the state-of-the-art deep learning methods:

- **OC-SVM** [29]: One Class Support Vector Machines(OC-SVM) is a classic kernel method for novelty detection that only uses normal data to learn a decision boundary. We adopt the widely used radial basis function (RBF) kernel. In our experiments, we assume that the abnormal proportion is known. We set the parameter ν to the anomaly proportion, and set γ to $1/m$, where m is the number of input features.
- **DSEBM** [37]: Deep Structured Energy Based Models (DSEBM) is a deep learning method for anomaly detection, which contains two decision criteria for performing anomaly detection: the energy score (DSEBM-e) and the reconstruction error (DSEBM-r).
- **DAGMM** [21]: Deep Autoencoding Gaussian Mixture Model (DAGMM) is a state-of-the-art method for anomaly detection, which consists of two major components: a compression network and an estimation network. The compression network obtains low-dimensional representations, and feeds the representations to the subsequent estimation network to predicts their likelihood/energy in the framework of GMM.
- **AnoGAN** [6]: AnoGAN is a GAN-based method for anomaly detection. AnoGAN is trained with normal data, and uses both reconstruction error and discrimination components as the anomaly criterion. There are two approaches for the anomaly score in the original paper and we choose the best variant in our tasks.
- **ALAD** [20]: Adversarially Learned Anomaly Detection (ALAD) is also a state-of-the-art method based on bi-directional GANs, which derives adversarially learned features for the anomaly detection task. ALAD uses reconstruction errors based on the adversarially learned features to determine if a data sample is anomalous.

In addition to the above baseline methods, we also perform three variations of our model to demonstrate the advantages of LAKE as the compression network.

- **DAE-KDE**: In this variation, we use a deep autoencoder as compression network, and the KDE estimation model is unchanged.
- **AAE-KDE**: This variation uses a multi-layer adversarial autoencoder as our compression network, and the KDE model remains the same.
- **VAE-KDE**: VAE-KDE uses a standard variational autoencoder as our compression network, and the KDE estimation model is unchanged.

4.3. Experiment configuration

The network structure of LAKE used on each dataset is summarized in Tables A.8–A.12 in the Appendix. And LAKE uses relative Euclidean distance and cosine similarity together as reconstruction features r. The configurations of baselines used in experiments follows their original configurations.

We follow the setting in [21,37] with completely clean training data:

in each run, we take τ% of data by uniformly random sampling for training with the rest (1-τ%) reserved for testing, and only data samples from the normal data are used for training models. Each experiment is conducted repeatedly 20 runs using independent training data sampling, and the average results are reported. Specifically, for our LAKE method and three variations, we set $\tau = 10$ in KDDCUP, $\tau = 80$ in Thyroid, $\tau = 80$ in Arrhythmia, and $\tau = 10$ in KDDCUP-Rev.

4.4. Evaluation metrics

We use average precision, recall, and F_1 score to quantify the results. The precision and recall are defined as follows: $Precision = \frac{|G \cap R|}{|R|}$ and $Recall = \frac{|G \cap R|}{|G|}$, where G denotes the set of ground truth anomalies in the dataset, and R denotes the set of anomalies reported by the methods. F_1 score is defined as follows: $F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$. Based on the anomaly ratio α in Table 1, the threshold can be determined to identify anomalous objects.

4.5. Effectiveness evaluation

First, we valuate the overall effectiveness of our proposed model compared with all baseline methods on six benchmark datasets. Table 2 shows the average precision, recall, and F_1 score with their standard deviations for LAKE and all baselines in different datasets. For baselines, we follow the setting in [20] (i.e., 80% of the whole official dataset for training) and [21] (i.e., 50% of whole normal dataset for training).

We can see that LAKE significantly outperforms all baseline methods in terms of average precision, recall, and F_1 score on six datasets. LAKE achieves 4.5% improvement in standard F_1 score compared to the state-of-the-art ALAD on the classical KDDCUP dataset, reaching over 99% in all terms of precision, recall and F_1 score. In addition, LAKE significantly performs better than the state-of-the-art DSEBM, DAGMM and ALAD methods by over 35.4% and 37.3% improvement in standard F_1 score on Thyroid and Arrhythmia, respectively. On SpamBase and Cardiotocography, LAKE still surpass the second best DAGMM and OC-SVM by 15.5% and 3.9% improvement in standard F_1, respectively. Moreover, it can also be seen from the standard deviations that our proposed LAKE is also more stable than other methods. OC-SVM does not achieve good results for the anomaly detection in high-dimensional data. This is because the core idea of OC-SVM is to find a boundary in high-dimensional space by using normal data, but when there are too many attributes of data, irrelevant redundant attributes may have a great influence on the result of OC-SVM. Although DSEBM considers reconstruction error and energy error, it ignores the latent representation, which may be the main reason why our proposed model and DAGMM perform

Table 2
Average precision, recall and F_1 from LAKE and all baselines. For each metric, the best result is shown in bold.

Method	KDDCUP			Thyroid		
	Precision ± Std	Recall ± Std	F_1 ± Std	Precision ± Std	Recall ± Std	F_1 ± Std
OC-SVM	0.7457 ± 0.0157	0.8523 ± 0.0177	0.7954 ± 0.0166	0.3639 ± 0.0131	0.4239 ± 0.0156	0.3887 ± 0.0142
DSEBM-r	0.8744 ± 0.0607	0.8414 ± 0.0428	0.8575 ± 0.0516	0.0400 ± 0.0049	0.0403 ± 0.0043	0.0403 ± 0.0047
DSEBM-e	0.2151 ± 0.0757	0.2180 ± 0.0756	0.2170 ± 0.0755	0.1319 ± 0.0037	0.1319 ± 0.0059	0.1319 ± 0.0048
DAGMM	0.9297 ± 0.0103	0.9442 ± 0.0112	0.9369 ± 0.0107	0.4766 ± 0.0171	0.4834 ± 0.0101	0.4782 ± 0.0133
AnoGAN	0.8786 ± 0.0370	0.8297 ± 0.0160	0.8865 ± 0.0115	0.0412 ± 0.0119	0.0430 ± 0.0128	0.0421 ± 0.0123
ALAD	0.9427 ± 0.0060	0.9577 ± 0.0062	0.9501 ± 0.0061	0.3196 ± 0.0063	0.3333 ± 0.0094	0.3263 ± 0.0055
DAE-KDE	0.9840 ± 0.0033	0.9655 ± 0.0108	0.9710 ± 0.0086	0.7934 ± 0.0027	0.7849 ± 0.0095	0.7891 ± 0.0068
AAE-KDE	0.9842 ± 0.0045	0.9697 ± 0.0098	0.9754 ± 0.0065	0.7501 ± 0.0026	0.7419 ± 0.0010	0.7459 ± **0.0010**
VAE-KDE	0.9913 ± 0.0010	0.9912 ± 0.0085	0.9912 ± 0.0106	0.7548 ± 0.0110	0.7548 ± **0.0007**	0.7548 ± 0.0090
LAKE	**0.9985** ± **0.0002**	**0.9912** ± **0.0035**	**0.9948** ± **0.0010**	**0.8369** ± **0.0026**	**0.8279** ± 0.0010	**0.8324** ± 0.0019
Method	Arrhythmia			KDDCUP-Rev		
	Precision ± Std	Recall ± Std	F_1 ± Std	Precision ± Std	Recall ± Std	F_1 ± Std
OC-SVM	0.5397 ± 0.0058	0.4082 ± 0.0419	0.4581 ± 0.0040	0.7148 ± 0.0096	**0.9940** ± 0.0126	0.8316 ± 0.0109
DSEBM-r	0.4286 ± 0.0263	0.5000 ± 0.0300	0.4615 ± 0.0278	0.2036 ± 0.0110	0.2036 ± 0.0109	0.2036 ± 0.0110
DSEBM-e	0.4643 ± 0.0149	0.4645 ± 0.0379	0.4643 ± 0.0211	0.2212 ± 0.0219	0.2213 ± 0.0226	0.2213 ± 0.0211
DAGMM	0.4909 ± 0.0475	0.5078 ± 0.0349	0.4983 ± 0.0520	0.9370 ± 0.0079	0.9390 ± 0.0089	0.9380 ± 0.0084
AnoGAN	0.4118 ± 0.0293	0.4375 ± 0.0121	0.4242 ± 0.0206	0.8422 ± 0.0182	0.8305 ± 0.0004	0.8363 ± 0.0250
ALAD	0.5000 ± 0.0181	0.5313 ± 0.0096	0.5152 ± 0.0276	0.9547 ± 0.0074	0.9678 ± 0.0075	0.9612 ± 0.0075
DAE-KDE	0.8461 ± 0.0059	0.8333 ± 0.0083	0.8396 ± 0.0067	0.9890 ± **0.0021**	0.9889 ± 0.0072	0.9890 ± 0.0065
AAE-KDE	0.8553 ± 0.0034	0.8424 ± **0.0030**	0.8488 ± **0.0002**	0.9907 ± 0.0029	0.9906 ± 0.0101	0.9906 ± 0.0072
VAE-KDE	0.8461 ± 0.0060	0.8333 ± 0.0123	0.8396 ± 0.0091	**0.9936** ± 0.0054	0.9812 ± **0.0006**	0.9874 ± 0.0030
LAKE	**0.8953** ± **0.0026**	**0.8818** ± 0.0078	**0.8885** ± 0.0089	0.9914 ± 0.0023	0.9915 ± 0.0013	**0.9914** ± **0.0005**
Method	SpamBase			Cardiotocography		
	Precision ± Std	Recall ± Std	F_1 ± Std	Precision ± Std	Recall ± Std	F_1 ± Std
OC-SVM	0.7440 ± 0.0395	0.7972 ± 0.0159	0.7694 ± 0.0288	0.7366 ± 0.0148	0.6848 ± 0.0278	0.7051 ± 0.0146
DSEBM-r	0.4296 ± 0.0247	0.3085 ± 0.0209	0.3574 ± 0.0215	0.5584 ± 0.0202	0.5467 ± 0.0217	0.5365 ± 0.0250
DSEBM-e	0.4356 ± 0.0112	0.3185 ± 0.0169	0.3679 ± 0.0124	0.5564 ± 0.0218	0.5367 ± 0.0282	0.5515 ± 0.0221
DAGMM	0.9435 ± 0.0344	0.7233 ± 0.0221	0.7970 ± 0.0291	0.5024 ± 0.0250	0.4905 ± 0.0245	0.4964 ± 0.0248
AnoGAN	0.4963 ± 0.0368	0.5313 ± 0.0344	0.5132 ± 0.0178	0.4446 ± 0.0334	0.4360 ± 0.0337	0.4412 ± 0.0431
ALAD	0.5344 ± 0.0250	0.5206 ± 0.0293	0.5274 ± 0.0240	0.5983 ± 0.0138	0.5841 ± 0.0135	0.5911 ± 0.0137
DAE-KDE	0.9311 ± 0.0058	0.9282 ± **0.0011**	0.9230 ± **0.0011**	0.7170 ± 0.0358	0.3185 ± 0.1256	0.4296 ± 0.1146
AAE-KDE	0.9376 ± 0.0074	0.9282 ± 0.0073	0.9329 ± 0.0074	0.6502 ± 0.1081	0.4247 ± 0.1066	0.5011 ± 0.0969
VAE-KDE	0.9437 ± 0.0087	0.9335 ± 0.0092	0.9384 ± 0.0093	0.6914 ± 0.0805	0.5582 ± 0.1429	0.6117 ± 0.1127
LAKE	**0.9576** ± **0.0037**	**0.9480** ± 0.0036	**0.9528** ± 0.0036	**0.7483** ± **0.0110**	**0.7410** ± **0.0109**	**0.7446** ± **0.0109**

better than DSEBM. The reasons why LAKE is better than DAGMM may be attributed as: (1) LVAE is better than autoencoder in learning low-dimensional representation preserving the distribution of original data due to the existence of a variational structure and layer constraint; (2) LAKE adopts kernel density estimation to model the probability density distribution of data instead of Gaussian mixture model. KDE is superior to Gaussian mixture model, because GMM is a parameter estimation that refers to the process of using sample data to estimate the parameters of the selected distribution, while KDE is a nonparametric estimation that allows the functional form of a fit to data to be obtained in the absence of any guidance or constraints from theory. Additionally, GMM also needs to manually select the number of mixed Gaussian models, which is very tricky in the absence of domain knowledge. For AnoGAN, it adopts adversarial autoencoder to recover a latent representation for each input data, and uses both reconstruction error and discrimination components as the anomaly criterion, but AnoGAN does not make full use of the low-dimensional representation. Although ALAD can simulate the distribution of data well when the experimental data is large enough, it also ignores the consideration of latent representation. Another potential reason why our method is better than all baselines is that we adopt a novel probability density-aware strategy that only estimates density value of each input data with respect to the learned probability density distribution of normal training samples. This strategy helps our method to effectively separate densely distributed anomalies out in latent data space.

From Table 2, we also observe that LAKE outperforms its variations (i.e., DAE-KDE, AAE-KDE and VAE-KDE) on six datasets. In particular, LAKE significantly performs better than the three variations on four small datasets (i.e., Thyroid, Arrhythmia, SpamBase and Cardiotocography). This is because LVAE better preserves the key information during data dimension reduction. So that LVAE learns the distribution of data better in latent data space when using very few training data compared to DAE, AAE and VAE. But still, our variations are significantly better than all competitive baselines in terms of precision, recall and F_1 score.

4.6. Hypothesis testing

To further verify the superiority of our proposed method, we use Welch's t-test to statistically assess the proposed method compared with baselines over 6 datasets. More specifically, we use the code[4] from SciPy.org for statistical testing.

To compare our proposed method with each baseline (one vs. one), we evaluate the following null hypothesis $H0$ and the alternative hypothesis $H1$ for each pair of methods:

$$H0 : A \approx B,$$

$$H1 : A < B,$$

where B stands for the result of LAKE on a specific dataset, and A denotes the result of a specific baseline on the corresponding dataset. We calculate p-value for each test, and the hypothesis is checked at $p = 0.05$ significance level. The statistical assessment

[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html.

Table 3
p-values of Welch's t-test for precision.

Dataset	OC-SVM	DSEBM-r	DSEBM-e	DAGMM	AnoGAN	ALAD
KDDCUP	3.229e−12	2.816e−06	5.762e−16	4.766e−08	4.197e−06	9.073e−06
Thyroid	1.430e−12	7.901e−29	1.340e−22	9.334e−21	8.667e−25	8.399e−27
Arrhythmia	3.995e−16	8.030e−14	1.156e−12	2.526e−13	2.702e−17	1.132e−15
KDDCUP-Rev	8.913e−11	5.011e−14	3.435e−17	2.393e−10	4.773e−09	5.328e−08
SpamBase	7.251e−08	9.450e−17	5.318e−16	1.411e−02	1.812e−11	4.121e−15
Cardiotocography	2.806e−03	1.120e−14	9.638e−16	4.774e−14	5.367e−12	1.885e−14

Table 4
p-values of Welch's t-test for recall.

Dataset	OC-SVM	DSEBM-r	DSEBM-e	DAGMM	AnoGAN	ALAD
KDDCUP	1.033e−10	1.295e−05	3.359e−16	2.330e−07	6.057e−10	2.151e−07
Thyroid	1.767e−12	6.015e−30	1.104e−25	1.753e−20	8.839e−25	3.376e−28
Arrhythmia	8.821e−18	5.183e−12	1.149e−14	2.973e−13	1.730e−16	3.739e−14
KDDCUP-Rev	**5.301e−02**	2.373e−14	5.596e−17	5.005e−10	6.475e−10	5.976e−07
SpamBase	1.405e−10	7.174e−21	2.719e−20	3.802e−09	3.707e−11	2.065e−14
Cardiotocography	2.537e−07	9.638e−12	1.896e−17	6.077e−14	3.830e−12	3.165e−14

results of precision, recall and F_1 are shown in Tables 3, 4, and 5 respectively.

As shown in Tables 3, 4 and 5, Except for the recall of OC-SVM on KDDCUP-Rev, all Welch's t-test results of precision and recall are significant at $p = 0.05$. More importantly, Welch's t-test results on the more comprehensive F_1 score are all significant at $p = 0.05$. Thus we can reject the null hypothesis $H0$ and accept alternative Hypothesis $H1$. That is, our proposed LAKE performs significantly better than all baselines.

In summary, this experiment confirms that the improvement of our proposed method over the state-of-the-art methods in detecting anomalies is statistically significant.

4.7. Performance w.r.t. training set

In this experiment, we mainly study the performance of our method and baselines with respective to the number of training set. We use τ% of the normal dataset as the training set for all methods. Tables 6 and 7 show the average precision, recall, and F_1 score of LAKE and the competitive baselines on Arrhythmia and KDDCUP datasets, respectively.

As we can see, with only 30% and 10% training data, LAKE has already shown better performance than all baselines with the best performance in terms of precision, recall and F_1 score on Arrhythmia and KDDCUP respectively. As the training data increases, the performance of LAKE increases on both datasets, especially on Arrhythmia LAKE achieves a significant improvement. ALAD and AnoGAN basically keep stable in term of F_1 score when performing on KDDCUP, and have slight fluctuations when performing on Arrhythmia. This may be because ALAD and AnoGAN mainly use reconstruction error as the anomaly criterion, and as a consequence the results are influenced by the degree to which the models fit to the training samples. From Table 7, DSEBM-e that uses energy score as detection criterion is not suitable for KDDCUP, as the data distribution in KDDCUP is more complex than energy based models. The results of DSEBM-r are similar to those of ALAD and AnoGAN, because it also use reconstruction error as the criterion for anomaly detection. Although DAGMM has increased performance as the number of training set increases, LAKE is far superior to it, even using less training data.

In summary, this experiment confirms that the proposed LAKE can achieve better results with fewer training samples compared to the state-of-the-art methods.

4.8. Robustness evaluation

In this experiment, we evaluate the robustness of LAKE compared to the baselines on KDDCUP dataset. For LAKE, 10% of the normal data is selected as the training set, and meanwhile we mix c% of samples from the anomalous data into the training set. In terms of ALAD, DSEBM and DAGMM, 50% of the normal data is selected as the training set, while mixing c% of samples from anomaly data into their training set.

Our LAKE uses a Gaussian kernel function in KDE model to learn the probability density distribution, thus the bandwidth h determines how many nearest neighbors in the trained KDE are selected to estimate its density value for each test point. That is, adjusting the bandwidth can effectively enhance the stability of our method with respect to noise. This is because the larger the bandwidth, the more neighbors each test point needs to estimate its density in the trained KDE model. Even if the training set is doped with a small amount of noise, each test sample requires more nearest training objects to estimate its density value, and thus the abnormal data is also easily detected.

Fig. 3 shows the experimental results of LAKE and the baseline methods with different contaminated training data. From the results, we can see that LAKE is also affected by the data contamination when bandwidth $h = 0.01$. But we can increase the robustness of our model by adjusting the bandwidth. It can be seen that as the bandwidth increases, the robustness of our LAKE model is getting better, with limited damaging of the performance. LAKE is almost not affected by contaminated data when bandwidth $h \geq 0.05$. When the contamination ratio c% increases from 1% to 5%, the performance of DAGMM declines, but the impact on DSEBM-r and ALAD is not very significant. This may be because the GMM model in DAGMM is more sensitive to noise compared to the reconstruction error used in DSEBM-r and ALAD. However, LAKE with a large bandwidth is still significantly better than all baseline methods.

5. Conclusion

In this paper, we propose a layer-constrained variational autoencoding kernel density estimation model (LAKE) for anomaly detection from high-dimensional data. LAKE mainly consists of two parts: the compression network and the KDE model. The compression network obtains a low-dimensional representation while retaining the key features using a layer-constrained variational autoencoder. The KDE model takes the low-dimensional representation and reconstruction error features as feeds, and learns a probability density distribution of training samples. For

Table 5
p-values of Welch's t-test for F_1.

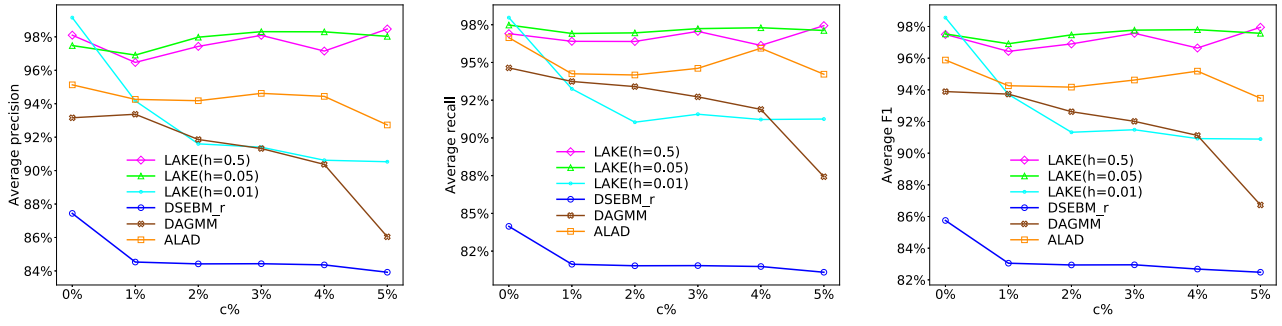
Dataset	OC-SVM	DSEBM-r	DSEBM-e	DAGMM	AnoGAN	ALAD
KDDCUP	1.042e−13	1.090e−05	1.764e−16	1.411e−07	3.060e−06	1.856e−07
Thyroid	7.901e−13	1.001e−30	5.814e−25	1.120e−21	1.212e−25	6.076e−28
Arrhythmia	1.547e−18	5.157e−12	2.383e−14	2.051e−11	5.168e−17	2.754e−14
KDDCUP-Rev	3.235e−12	1.247e−14	1.014e−17	6.296e−07	5.722e−10	1.046e−06
SpamBase	4.476e−08	2.009e−19	5.752e−16	4.900e−06	2.956e−11	9.938e−15
Cardiotocography	2.375e−04	2.318e−17	2.280e−17	2.399e−14	1.897e−12	1.644e−14

Table 6
Performance comparison w.r.t. training ratio on Arrhythmia.

Ratio τ%	LAKE			ALAD			DAGMM		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
30%	0.6046	0.5863	0.5953	0.4641	0.5250	0.4474	0.3750	0.4500	0.4091
40%	0.6651	0.6651	0.6651	0.4634	0.5278	0.4935	0.3902	0.4444	0.4156
50%	0.7138	0.7030	0.7083	0.5000	0.5312	0.5152	0.3824	0.4062	0.3939
60%	0.7890	0.7651	0.7769	0.4643	0.4643	0.4643	0.4643	0.4643	0.4643
70%	0.8484	0.8484	0.8484	0.3810	0.4000	0.3902	0.4286	0.4500	0.4390
80%	0.8953	0.8818	0.8885	0.3571	0.4167	0.3846	0.3571	0.4167	0.3846
Ratio τ%	DSEBM-e			DSEBM-r			AnoGAN		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
30%	0.4583	0.5500	0.5000	0.3542	0.4250	0.3864	0.2917	0.3500	0.3182
40%	0.4634	0.5278	0.4935	0.3902	0.4444	0.4156	0.3415	0.3889	0.3636
50%	0.5000	0.5312	0.5152	0.4118	0.4375	0.4242	0.3529	0.3750	0.3636
60%	0.4643	0.4643	0.4643	0.4286	0.4286	0.4286	0.4286	0.4286	0.4286
70%	0.4286	0.4500	0.4390	0.3810	0.4000	0.3902	0.4286	0.4500	0.4390
80%	0.4286	0.5000	0.4615	0.4286	0.5000	0.4615	0.3571	0.4167	0.3846

Table 7
Performance comparison w.r.t. training ratio on KDDCUP.

Ratio τ%	LAKE			ALAD			DAGMM		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
10%	0.9942	0.9873	0.9907	0.9576	0.9727	0.9651	0.9234	0.9382	0.9308
20%	0.9985	0.9913	0.9949	0.9554	0.9691	0.9622	0.9041	0.9171	0.9106
30%	0.9973	0.9932	0.9952	0.9513	0.9513	0.9513	0.9290	0.9437	0.9363
40%	0.9965	0.9945	0.9955	0.9466	0.9625	0.9545	0.9469	0.9628	0.9548
50%	0.9961	0.9952	0.9957	0.9513	0.9664	0.9588	0.9315	0.9464	0.9389
60%	0.9964	0.9956	0.9960	0.9502	0.9624	0.9563	0.9448	0.9570	0.9509
Ratio τ%	DSEBM-e			DSEBM-r			AnoGAN		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
10%	0.1121	0.1142	0.1131	0.8535	0.8233	0.8381	0.9166	0.8362	0.8667
20%	0.1322	0.1333	0.1332	0.8472	0.8166	0.8316	0.8590	0.8590	0.8590
30%	0.0830	0.0840	0.0830	0.8732	0.8403	0.8564	0.8344	0.8476	0.8409
40%	0.1311	0.1332	0.1321	0.8745	0.8422	0.8576	0.8343	0.8344	0.8344
50%	0.2151	0.2180	0.2170	0.8744	0.8414	0.8575	0.9472	0.8163	0.8630
60%	0.0401	0.0411	0.0410	0.8756	0.8399	0.8573	0.8496	0.8605	0.8550



(a) Precision (b) Recall (c) F_1 score

Fig. 3. Anomaly detection results on contaminated training data on KDDCUP.

each test data, its density value is estimated by the trained KDE model of training samples, and the objects with the lowest KDE values are reported as anomalies. Our experimental results on public benchmark datasets show that the proposed LAKE is significantly better than the state-of-the-art methods by up to 37% improvement on the standard F_1 score.

Table A.8
LAKE architecture and hyperparameters on KDDCUP and KDDCUP-Rev.

Operation	Units	Activation function
Encoder		
Dense	(118,90)	Tanh
Dense	(90,60)	Tanh
Dense	(60,25)	Tanh
mu	(25,20)	None
var	(25,20)	None
Decoder		
Dense	(20,25)	Tanh
Dense	(25,60)	Tanh
Dense	(60,90)	Tanh
Dense	(90,118)	Sigmoid
Optimizer		Adam(learning_rate = 0.00001)
Batch size		1000
Epochs		1000
Bandwidth		0.001

Table A.9
LAKE architecture and hyperparameters on Arrhythmia.

Operation	Units	Activation function
Encoder		
Dense	(274,130)	Tanh
Dense	(130,60)	Tanh
Dense	(60,25)	Tanh
Dense	(25,20)	Tanh
mu	(20,10)	None
var	(20,10)	None
Decoder		
Dense	(10,20)	Tanh
Dense	(20,25)	Tanh
Dense	(25,60)	Tanh
Dense	(60,130)	Tanh
Dense	(130,274)	Sigmoid
Optimizer		Adam(learning_rate = 0.00001)
Batch size		200
Epochs		1000
Bandwidth		0.001

Table A.10
LAKE architecture and hyperparameters on Thyroid.

Operation	Units	Activation function
Encoder		
Dense	(36,20)	Tanh
Dense	(20,11)	Tanh
mu	(11,10)	None
var	(11,10)	None
Decoder		
Dense	(10,11)	Tanh
Dense	(11,20)	Tanh
Dense	(20,36)	Sigmoid
Optimizer		Adam(learning_rate = 0.00001)
Batch size		200
Epochs		1000
Bandwidth		0.001

Table A.11
LAKE Architecture and hyperparameters on SpamBase.

Operation	Units	Activation function
Encoder		
Dense	(58,45)	Tanh
Dense	(45,35)	Tanh
mu	(35,30)	None
var	(35,30)	None
Decoder		
Dense	(30,35)	Tanh
Dense	(35,45)	Tanh
Dense	(45,58)	Sigmoid
Optimizer		Adam(learning_rate = 0.00001)
Batch size		200
Epochs		1000
Bandwidth		0.001

Table A.12
LAKE architecture and hyperparameters on Cardiotocography.

Operation	Units	Activation function
Encoder		
Dense	(22,20)	Tanh
Dense	(20,15)	Tanh
mu	(15,15)	None
var	(15,15)	None
Decoder		
Dense	(15,15)	Tanh
Dense	(15,20)	Tanh
Dense	(20,22)	Sigmoid
Optimizer		Adam(learning_rate = 0.00001)
Batch size		200
Epochs		1000
Bandwidth		0.001

In future work, we plan to investigate the effective semi-supervised anomaly detection method based on deep autoencoder variations in high-dimensional data. We also plan to explore the generalized autoencoder-based model for unsupervised anomaly detection in some domains.

CRediT authorship contribution statement

Peng Lv: Methodology, Software, Data curation, Visualization, Writing - original draft. **Yanwei Yu:** Conceptualization, Supervision, Writing - review & editing, Funding acquisition. **Yangyang Fan:** Data curation, Investigation, Writing - review & editing. **Xianfeng Tang:** Validation, Writing - review & editing. **Xiangrong Tong:** Resources, Funding acquisition.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. This work is partially supported by the National Natural Science Foundation of China under Grant Nos.: 61773331, 61572418 and 61403328. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

Appendix. LAKE Experimental Details

See Tables A.8–A.12.

References

[1] Swee Chuan Tan, Kai Ming Ting, Tony Fei Liu, Fast anomaly detection for streaming data, in: Twenty-Second International Joint Conference on Artificial Intelligence, 2011.
[2] Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 413–422.
[3] Fabian Keller, Emmanuel Muller, Klemens Bohm, HiCS: high contrast subspaces for density-based outlier ranking, in: 2012 IEEE 28th International Conference on Data Engineering, IEEE, 2012, pp. 1037–1048.
[4] Varun Chandola, Arindam Banerjee, Vipin Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 15.
[5] Waqas Sultani, Chen Chen, Mubarak Shah, Real-world anomaly detection in surveillance videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6479–6488.
[6] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, Georg Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: International Conference on Information Processing in Medical Imaging, Springer, 2017, pp. 146–157.

[7] Raghavendra Chalapathy, Sanjay Chawla, Deep learning for anomaly detection: A survey, 2019, arXiv:1901.03407.
[8] Tsuyoshi Idé, Hisashi Kashima, Eigenspace-based anomaly detection in computer systems, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004, pp. 440–449.
[9] Weiren Yu, Charu C. Aggarwal, Shuai Ma, Haixun Wang, On anomalous hotspot discovery in graph streams, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 1271–1276.
[10] Ryohei Fujimaki, Takehisa Yairi, Kazuo Machida, An approach to spacecraft anomaly detection problem using kernel feature space, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, ACM, 2005, pp. 401–410.
[11] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, Christopher Leckie, High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, Pattern Recognit. 58 (2016) 121–134.
[12] Emmanuel J. Candès, Xiaodong Li, Yi Ma, John Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 11.
[13] Tingquan Deng, Dongsheng Ye, Rong Ma, Hamido Fujita, Lvnan Xiong, Low-rank local tangent space embedding for subspace clustering, Inform. Sci. 508 (2020) 1–21.
[14] Chong Zhou, Randy C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 665–674.
[15] Jinwon An, Sungzoon Cho, Variational autoencoder based anomaly detection using reconstruction probability, Spec. Lect. IE 2 (2015) 1–18.
[16] Mahdyar Ravanbakhsh, Moin Nabi, Hossein Mousavi, Enver Sangineto, Nicu Sebe, Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 1689–1698.
[17] Cairong Zhao, Xuekuan Wang, Wai Keung Wong, Weishi Zheng, Jian Yang, Duoqian Miao, Multiple metric learning based on bar-shape descriptor for person re-identification, Pattern Recognit. 71 (2017) 218–234.
[18] Cairong Zhao, Xuekuan Wang, Duoqian Miao, Hanli Wang, Weishi Zheng, Yong Xu, David Zhang, Maximal granularity structure and generalized multi-view discriminant analysis for person re-identification, Pattern Recognit. 79 (2018) 79–96.
[19] Cairong Zhao, Xuekuan Wang, Wangmeng Zuo, Fumin Shen, Ling Shao, Duoqian Miao, Similarity learning with joint transfer constraints for person re-identification, Pattern Recognit. 97 (2020) 107014.
[20] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, Vijay Chandrasekhar, Adversarially learned anomaly detection, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 727–736.
[21] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, Haifeng Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018.
[22] Rémi Domingues, Maurizio Filippone, Pietro Michiardi, Jihane Zouaoui, A comparative evaluation of outlier detection algorithms: Experiments and analyses, Pattern Recognit. 74 (2018) 406–421.
[23] Edwin M. Knorr, Raymond T. Ng, Vladimir Tucakov, Distance-based outliers: algorithms and applications, VLDB J.—Int. J. Very Large Data Bases 8 (3–4) (2000) 237–253.
[24] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, Jörg Sander, LOF: identifying density-based local outliers, in: ACM Sigmod Record, Vol. 29, (2) ACM, 2000, pp. 93–104.
[25] Yizhou Yan, Lei Cao, Elke A. Rundensteiner, Scalable top-n local outlier detection, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 1235–1244.
[26] Rikard Laxhammar, Goran Falkman, Egils Sviestins, Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator, in: 2009 12th International Conference on Information Fusion, IEEE, 2009, pp. 756–763.
[27] Erich Schubert, Arthur Zimek, Hans-Peter Kriegel, Generalized outlier detection with flexible kernel density estimates, in: Proceedings of the 2014 SIAM International Conference on Data Mining, SIAM, 2014, pp. 542–550.
[28] Weiming Hu, Jun Gao, Bing Li, Ou Wu, Junping Du, Stephen John Maybank, Anomaly detection using local kernel density estimation and context-based regression, IEEE Trans. Knowl. Data Eng. (2018).
[29] Yunqiang Chen, Xiang Sean Zhou, Thomas S. Huang, One-class SVM for learning in image retrieval, in: ICIP (1), Citeseer, 2001, pp. 34–37.
[30] Simon Günter, Nicol N. Schraudolph, S.V.N. Vishwanathan, Fast iterative kernel principal component analysis, J. Mach. Learn. Res. 8 (Aug) (2007) 1893–1918.
[31] Taotao Lai, Riqing Chen, Changcai Yang, Qiming Li, Hamido Fujita, Alireza Sadri, Hanzi Wang, Efficient robust model fitting for multistructure data using global greedy search, IEEE Trans. Cybern. (2019).
[32] Taotao Lai, Hamido Fujita, Changcai Yang, Qiming Li, Riqing Chen, Robust model fitting based on greedy search and specified inlier threshold, IEEE Trans. Ind. Electron. 66 (10) (2018) 7956–7966.
[33] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, Nassir Navab, Deep autoencoding models for unsupervised anomaly segmentation in brain mr images, in: International MICCAI Brainlesion Workshop, Springer, 2018, pp. 161–169.
[34] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[35] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Georg Langs, Ursula Schmidt-Erfurth, F-anoGAN: Fast unsupervised anomaly detection with generative adversarial networks, Med. Image Anal. 54 (2019) 30–44.
[36] Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein generative adversarial networks, in: International Conference on Machine Learning, 2017, pp. 214–223.
[37] Shuangfei Zhai, Yu Cheng, Weining Lu, Zhongfei Zhang, Deep structured energy based models for anomaly detection, in: Proceedings of the 33rd International Conferernce on Machine Learing, Vol. 48, JMLR, 2016.
[38] Carl Doersch, Tutorial on variational autoencoders, 2016, arXiv preprint arXiv:1606.05908.
[39] Diederik P. Kingma, Max Welling, Auto-encoding variational bayes, in: Proceedings of 2nd International Conference on Learning Representations, 2014.