A Unified Unsupervised Gaussian Mixture Variational Autoencoder for High Dimensional Outlier Detection

Weixian Liao
Department of Computer and Information Sciences
Towson University
Towson, MD 20252
Email: wliao@towson.edu

Yifan Guo, Xuhui Chen, and Pan Li
Department of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, OH 44106
Email: {yxg383, xxc296, lipan}@case.edu

***Abstract*—Paradigm-shifting systems such as cyber-physical systems, collect data of high- or ultrahigh- dimensionality tremendously. Detecting outliers in this type of systems provides indicative understanding in wide-ranging domains such as system health monitoring, information security, etc. Previous dimensionality reduction based outlier detection methods suffer from the incapability of well preserving the critical information in the low-dimensional latent space, mainly because they generally assume an isotropic Gaussian distribution as prior and fail to mine the intrinsic multimodality in high dimensional data. Moreover, most of the schemes decouple the model learning process, resulting in suboptimal performance. To tackle these challenges, in this paper, we propose a unified Unsupervised Gaussian Mixture Variational Autoencoder for outlier detection. Specifically, a variational autoencoder firstly trains a generative distribution and extracts reconstruction based features. Then we adopt a deep brief network to estimate the component mixture probabilities by the latent distribution and extracted features, which is further used by the Gaussian mixture model to estimate sample densities with the Expectation-Maximization (EM) algorithm. The inference model is optimized jointly with the variational autoencoder, the deep brief network, and the Gaussian mixture model. Afterwards, the proposed detector identifies outliers when the estimated sample density exceeds a learned threshold. Extensive simulations on six public benchmark datasets show that the proposed framework outperforms state-of-the-art outlier detection schemes and achieves, on average, 27% improvements in F1 score.**

***Keywords*-Outlier detection, Gaussian mixture model, variational autoencoder.**

I. Introduction

Big data has become the key basis of intelligence and brought new possibilities to many domains such as human well-being [1], transportation system [2], education [3], and so on. Many big data applications rely on large amounts of data of high dimensionality, which need to be processed and synthesized efficiently [4]. However, it becomes extremely challenging in that disturbances or unusual events in this type of data lead to spurious patterns and adverse effects, which is known as outliers [5]. They manifest themselves in diverse ways, such as missing values, erroneous procedures, system failures, unexpected events, etc. For instance, outliers may turn into credit card theft, misuse, or unauthorized transactions [6]. In computer networks, anomalous patterns can be an action of sending out sensitive information to an unauthorized destination [7]. Therefore, it is in dire need to find effective and efficient methods for outlier detection in high dimensional data.

Outlier detection is the process of identifying anomalous behavior or events that represent significant deviations from normal patterns in system operations [8]. In recent years, the literature has made significant progress. For example, a line of research focuses on distance based approaches for outlier detection, including K nearest neighbor distance- and average K nearest neighbors' distance-based methods [9]. They are based on assessments of distances (sometimes indirectly by assuming certain distributions) in the fully dimensional data space. However, these approaches are bound to deteriorate due to the notorious "curse of dimensionality" [10]. In particular, when it comes to data with high- or even ultrahigh-dimensionality, the outlier detection becomes extremely challenging. The reasoning is that the frequently used concepts such as proximity, distance, or nearest neighbor become less meaningful as the number of dimensionality increases [11]. The high computational complexity of distance based schemes also hinder them from effective outlier detection in high dimensional data.

To account for this challenge, many existing works focus on dimensionality reduction based outlier detection. Camacho et al. [12] propose a Principal Component Analysis (PCA) based multivariate scheme for network anomaly detection. Sakurada et al. [13] present a nonlinear dimensionality reduction based autoencoder to detect anomalies. These schemes find lower dimensional representations by decreasing the number of variables in the data. However, most of them suffer from the incapability of well preserving the essentiality of data because they generally assume that data follows simple distributions like isotropic Gaussian distribution. As high dimensional data usually preserves the characteristic of multi-modality, simply applying isotropic Gaussian distribution to fit a multimodal distribution with unimodal model results in poor performance.

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

Motivated by this, in this paper, we try to preserve the multimodality of high dimensional data in the lower dimensional latent space by feeding a Gaussian mixture model into the training process. The intuition is as follows. The Gaussian distribution is the most widely used distribution for modeling the real-world unimodal data. Gaussian mixture model turns the multimodal high dimensional data into a mixture of many unimodal Gaussian distributions. Furthermore, it maintains many theoretical and computational benefits of Gaussian models, making it practical for efficiently modeling very large datasets [14]. Therefore, Gaussian mixture model becomes a natural alternative.

The widespread success of machine learning techniques inspires a variety of approaches in the literature to solve the outlier detection problem, which, based on whether labels are used in the training process, can be categorized as supervised, semi-supervised, and unsupervised learning techniques. In particular, unsupervised learning approaches are preferably used compared to semi-supervised and supervised learning approaches. The reasons are two-fold. First, high dimensional data is generally imbalanced. The anomalous instances or outliers are far fewer compared to the normal instances in the training data, inevitably raising the issues caused by imbalanced class distributions. Second, labeling is often manually conducted by domain experts. In most cases it is prohibitively expensive and cumbersome to obtain hand-labeled data which is accurate and represents all types of anomalous behaviors [15]. Therefore, tremendous efforts have been devoted to unsupervised outlier detection.

In this paper, we investigate the problem of unsupervised outlier detection on high dimensional data. By combining the power of deep learning and Gaussian mixture model, we are able to learn a Gaussian mixture generative model for outlier detection. In the literature, existing approaches train a one-class classifier with normal data. However, some classifiers fail to use the high-dimensional multimodal data [16]. Relying only on lower dimensional representation will lose critical information for outlier detection. Thus, we turn to propose a density estimation based detection scheme in which estimated sample density is used to detect outliers. The main challenge lies in the fusion of high-dimensional and heterogeneous modalities. Chandola et al. [15] present some works which apply autoencoder (AE) or variational autoencoder (VAE) to construct high dimensional non-anomalous training data. The intuition is that autoencoders cannot reconstruct unforeseen patterns of anomalous data very well compared to foreseen non-anomalous data. However, existing works on VAE may not approximate the original data distribution well, especially when input data distributions are strongly multi-modal, as they only assume a single Gaussian distribution as the prior in the data generative procedure. More importantly, the training process in these works require purely normal data, which requires formidable efforts to filter the outliers in the historical data, and does not really fall into the unsupervised fashion.

To tackle these challenges, we firstly employ a VAE to build a generative model with all normal and anomalous data. To capture the features derived from the reconstruction error in VAE, we use relative Euclidean distance to calculate reconstruction error features. The learned latent distribution from generative model and reconstruction error features are fed into a deep brief network (DBN) to estimate the component mixture probabilities by the latent distribution and extracted features, which is further used by the Gaussian mixture model to estimate sample densities based on the Expectation-Maximization (EM) algorithm [17]. This is different from existing Gaussian mixture model which takes only sample values. Feeding with richer information can potentially improve the detection performance. The inference model is finally optimized jointly with variational autoencoder, deep brief network, and Gaussian mixture model to avoid decoupled model learning and suboptimal performance. Afterwards, the proposed detector identifies outliers when the estimated sample density exceeds a learned threshold that is obtained during the training phase. We conduct extensive simulations and find that our proposed unsupervised scheme achieves best performance under different metrics by comparing with state-of-the-art unsupervised approaches.

The main contributions in this paper are summarized as follows:

- We design a unified unsupervised Gaussian mixture Variational Autoencoder that can perform outlier detection effectively on high dimensional data.
- We leverage the redesigned Gaussian mixture prior in the latent representation to preserve both the intrinsic multimodality in data and the reconstruction error feature.
- We jointly optimize the VAE, the DBN, with the Gaussian mixture model to avoid decoupled model learning.
- Experimental results on six public benchmark datasets show that the proposed scheme achieves, on average, 27% improvements in F1 score, compared with baseline models.

The rest of this paper is organized as follows. Related work on outlier detection is discussed in Section II. Section III introduces preliminaries for variational autoencoder and Gaussian mixture model. In Section IV, we provide the details of the proposed a unified unsupervised Gaussian mixture VAE for outlier detection on multidimensional data. Section V experimentally shows the performance of our proposed scheme by comparing with state-of-the-art schemes. Finally, we conclude the paper in Section VI.

II. Related Work

Outlier detection has been studied in the context of high dimensional data [15], [18]. We focus on the most related works that apply machine learning based techniques for outlier detection. Based on whether labels are used in the training process, it can be categorized into supervised, semi-supervised, and unsupervised outlier detection. Specifically, Gaddam et al. [19] utilize ID3 decision tree learning method to classify anomalies in computer networks. Abe et al. [20] transform the anomaly detection problem into a classification problem and propose a supervised active learning scheme. Although their proposed scheme is computationally efficient, the assumption of labelling in the training dataset makes it less practical for real world applications. Besides, Ashfaq et al. [21] present a fuzziness based semi-supervised learning approach by considering a large amount of unlabeled samples with labeled samples for intrusion detection. Li et al. [22] propose several methods for malicious code detection. However, it also requires human interference to distinguish between the actual intrusion and false positive ones. Tao et al. [23] combine Fisher score with autoencoder algorithm to build a data fusion scheme for network traffic classification. Salama et al. [24] use restricted Boltzmann machine (RBM) based deep belief network for intrusion detection. However, both supervised and semi-supervised learning techniques assume that at least parts of labels in the training dataset are available, which may not be available in real world problems.

On the other hand, unsupervised anomaly detection has received tremendous attention. Depending on how anomalies are detected, unsupervised schemes can be categorized into clustering based and reconstruction based approaches. In particular, clustering analysis, such as k-means, Gaussian Mixture Models (GMMs), is widely applied to anomaly detection. For example, Xiong et al. [25] categorize data clusters at both the instance level and the cluster level so that various types of group anomalies can be detected. However, these models cannot be directly applied to our problem because clustering based approaches have very high computational complexity and can hardly be directly applied in data of high dimensionality. Reconstruction based methods like in [26] assume that anomalies are incompressible and thus cannot be effectively reconstructed from lower-dimensional latent space projections. Zhou et al. [27] propose a deep autoencoder to detect anomalies. However, the performance of these methods is limited because they only analyze anomaly from reconstruction errors.

Similar to our proposed scheme, Nalisnick et al. [28] propose a DL-GMM that combines VAE and GMM together. It employs mixtures of Gaussian distribution to approximate only the posterior of VAE, which improves the capacity of the original VAE. However, it is not suitable for unsupervised outlier detection. Dilokthanakul et al. [29] use Gaussian mixture VAE on image clustering tasks. We focus on constructing a density estimation based outlier detection scheme. Johnson et al. [30] propose a structured VAE that combines a neural net likelihood with probabilistic graphical models and use recognition networks to conjugate graphical model potentials. Shu et al. [31] design a GM-CVAE, which also combines VAE with GMM together. However, the GMM in their framework model the transitions between video frames, which is not applicable in our problem setting. Our previous work [32] focuses on discovering the correlation and dependency in time series data for anomaly detection purposes. In this study, we propose a joint optimization framework where the VAE, the DBN and the Gaussian mixture model are optimized together to further improve the detection performance.

III. Preliminaries

A. Variational Autoencoder based Outlier Detection

Variational autoencoder is a probabilistic model which combines bayesian inference with the autoenoder framework. The main advantage of a VAE based anomaly detection model over an autoencoder based anomaly detection model is that it provides a probabilistic measure rather than a reconstruction error as the anomaly score. Compared with reconstruction errors, reconstruction probabilities are more principled and objective, and do not require to model specific thresholds for judging anomalies [33]. Particularly, the idea behind VAE is that many complex data distributions can actually be modeled by a smaller set of latent variables whose probability density distributions are easier to model. So the objective of VAE is to find a low dimensional representation of the latent variables of the input data. In a traditional VAE, the latent variables follow a certain type of underlying distribution, which is generally assumed to be the Gaussian distribution. Without loss of generality, we denote a vector of multi-dimensional input by $\mathbf{x} \in \mathbb{R}^D$ and the corresponding latent vector by $\mathbf{z} \in \mathcal{R}^K$, where D and K are the dimension of the input and that of the latent variables, respectively. We can present the generative process as:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z},$$

where $p(\cdot)$ is the probability distribution function. However, since the search space of $\mathbf{z}$ is continuous and combinatorially large, the marginalization is computationally intractable. [34] are the first to propose a computationally tractable method to train this model. The main idea is as follows. $\mathbf{z}$ is generated from a prior distribution $p(\mathbf{z})$, e.g., a normal Gaussian distribution. The posterior distribution, denoted by $q_\phi(\mathbf{z}|\mathbf{x})$, is learned in the encoder network, and the likelihood distribution, i.e., $p_\theta(\mathbf{x}|\mathbf{z})$, is learned in the decoder network so as to reconstruct the original input, $\mathbf{x}$. Note that ϕ and θ are the parameters of the encoder and decoder, respectively. Considering the scenario where the input data $\mathbf{x}$ is known and $\mathbf{z}$ is unknown, we hope that two distributions, i.e., $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z}|\mathbf{x})$, get as close as possible, then we have the following objective function:

$$\min_{\phi,\ \theta} \mathsf{D}_{\mathsf{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \ || \ p_\theta(\mathbf{z}|\mathbf{x})),$$

where D_{KL} is Kullback-Leibler divergence of the approximate from the true posterior. By statistical derivations, the marginal log-likelihood of the input data is obtained by:

$$\log p(\mathbf{x}) = \mathsf{D}_{\mathsf{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \ || \ p_\theta(\mathbf{z}|\mathbf{x})) + \mathcal{L}_{VAE}(\phi, \theta; \mathbf{x}),$$

where

$$\mathcal{L}_{VAE}(\phi, \theta; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[log\ p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[log\ q_\phi(\mathbf{z}|\mathbf{x})].$$

$\mathcal{L}_{VAE}(\phi, \theta; \mathbf{x})$ is called the variational lower bound. Recall that the distribution of the input is deterministic, and hence $\log p(\mathbf{x})$ is a constant. To minimize the KL divergence of the approximate from the true posterior is equivalent to maximize the variational lower bound, i.e., $\mathcal{L}_{VAE}(\phi, \theta; \mathbf{x})$. To this end, the VAE tries to optimize the parameters $\phi,\ \theta$ for a new objective function as follows:

$$\max_{\phi,\ \theta}\ \mathcal{L}_{VAE}(\phi, \theta; \mathbf{x}).$$

With further statistical derivations, we rewrite the variational lower bound as:

$$\begin{aligned}&\mathcal{L}_{VAE}(\phi, \theta; \mathbf{x})\\ &= -\mathsf{D}_{\mathsf{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}) \ ||\ p_{\theta}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}[log\ p_{\theta}(\mathbf{z}|\mathbf{x})].\end{aligned} \tag{1}$$

The first term in the right hand side of (1) is the regularization term. The goal is to minimize the difference between the posterior distribution $q_{\phi}(\mathbf{z}|\mathbf{x})$ and the latent the prior distribution $p_{\theta}(\mathbf{z})$. For simplicity, the prior distribution $p_{\theta}(\mathbf{z})$ is often set to $\mathcal{N}(0, 1)$. Thus, the optimization process of the regularization term is to make $q_{\phi}(\mathbf{z}|\mathbf{x})$ to be as close as possible to $\mathcal{N}(0, 1)$. The second term is the reconstruction term. Maximizing it is a maximum likelihood estimation process of input data, given the sampling from latent distribution, and can be modeled in a discriminative supervised way. If the input data is binary, binary cross entropy between input data and reconstructed data is used to approximate the reconstruction term. On the other hand, if the input data is continuous, we can use the mean squared error between input data and reconstructed data instead. To maximize $\mathcal{L}_{VAE}$, stochastic gradient descent methods [34] can be used.

```
Algorithm 1: Varational autoencoder based anomaly detection
Input : $\mathbf{X_{train}} = \{x^{(1)}, \ldots, x^{(N_{train})}\}$,
        $\mathbf{X_{test}} = \{x^{(1)}, \ldots, x^{(N_{test})}\}$, Reconstruction
        probability threshold $\alpha$
Output: Sequence of anomaly predictiosn $S$
$\theta, \phi \leftarrow$ Initialize parameters
$f_{\theta}, g_{\phi} \leftarrow$ Train the Variational Autoencoder
network using training data $\mathbf{X_{train}}$
for $i = 1$ to $N_{test}$ do
    $\mu_z[i], \sigma_z[i] = f_{\theta}(z|x^{(i)})$
    Draw $L$ samples from $Z \sim \mathcal{N}(\mu_z[i], \sigma_z[i])$
    for $l = 1$ to $L$ do
        $\mu_{\hat{x}}[i, l], \sigma_{\hat{x}}[i, l] = g_{\phi}(x|z^{[i,l]})$
    end
    Reconstruction Probability $RP(x|\hat{x})[i] =$
      $\frac{1}{L}\sum_{l=1}^{L} \mathcal{N}(x^{(i)}|\mu_{\hat{x}}[i, l], \sigma_{\hat{x}}[i, l])$
    if $RP(x|\hat{x})[i] < \alpha$ then
        $x^{(i)}$ is an anomaly, $S[i]$ = "Anomalous"
    end
    else
        $x^{(i)}$ is not an anomaly, $S[i]$ = "Normal"
    end
end
```

Algorithm 1 describes the process of the VAE based anomaly detection in a semi-unsupervised learning manner. The intuition of VAE based anomaly detection is to construct a latent distribution space, where the distribution of normal data can be represented in a low dimensional space while anomalous data follows an apparently different distribution. Thus, the reconstruction probabilities of normal data are relatively higher than those of anomalous data. The same as in autoencoder based anomaly detection, only normal data only is used in the training process. Then, in the testing phase, each data sample $x^{(i)}$ $(i = 1, \ldots, N_{test})$ is fed into the encoder side to get the corresponding mean vector $\mu_z[i]$ and standard deviation vector $\sigma_z[i]$ in the latent space. After that, the latent vector z will be sampled for L times by following a Gaussian distribution $\mathcal{N}(\mu_z[i], \sigma_z[i])$. For each sample $z^{(i,l)}$, which represents the lth generated latent vector for input data $x^{(i)}$, it will be fed into the decoder side to get the corresponding reconstructed mean vector $\mu_{\hat{x}}[i, l]$ and standard deviation vector $\sigma_{\hat{x}}[i, l]$. By fitting the input data sample $x^{(i)}$ into the the Gaussian distribution with the reconstructed mean vector and the reconstructed standard deviation vector, we can get the corresponding reconstruction probability $\mathcal{N}(x^{(i)}|\mu_{\hat{x}}(i, l), \sigma_{\hat{x}}(i, l))$ of the lth generated latent vector. After averaging over the L reconstruction probabilities, we can obtain the final reconstruction probability $RP(x|\hat{x})[i]$ for the input $x^{(i)}$. By comparing whether the reconstruction probability is smaller than a given threshold α, the system can determine whether the input data sample is anomalous.

B. Gaussian Mixture Models

Gaussian mixture models are Gaussian probabilistic models representing the presence of subpopulations within an overall population. Given only a limited number of observations, they are used to make statistical inference for characteristics of subpopulations, in which parameters are estimated using the iterative expectation-maximization (EM) algorithm [14]. In particular, a Gaussian mixture model has two types of parameters, namely, the mixture component probabilities, and component means and variances. Assuming that a Gaussian mixture model has K components, the mean and variance of k-th component are denoted by μ_k and σ_k, respectively. We denote ϕ_k as the mixture probability for component k. In order to normalize the total probability distribution, we have a constraint that $\sum_{i=1}^{K} \phi_i = 1$. Using one-dimensional Gaussian mixture model as an example, it is formulated as follows:

$$\begin{aligned}&p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x|\mu_i, \sigma_i)\\ &\mathcal{N}(x|\mu_i, \sigma_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp(-\frac{(x-\mu_i)^2}{2\sigma_i^2}),\\ &\sum_{i=1}^{K} \phi_i = 1.\end{aligned}$$

After initiating the training process, the Expectation-Maximization (EM) algorithm can be used to solve the one dimensional Gaussian mixture model. Due to the page limit, we omit the details and readers can refer to [14] for details.

In this paper, we study a high dimensional Gaussian mixture model to estimate the sample densities for outlier detection, which will be clear in next section.

IV. A Unified Unsupervised Gaussian Mixture Variational Autoencoder

A. System Architecture

The proposed unsupervised Gaussian mixture variational autoencoder works as follows. The VAE takes a batch of N high dimensional data, i.e., $\mathbf{x} \in \mathbb{R}^D$, as input and learns a generative model, i.e., $\mathbf{z}$, by minimizing the reconstruction error between the input and N sampled outputs from the generative distribution in the latent space, denoted as $\hat{\mathbf{x}}$. Mathematically, we get that

$$\mathbf{z} = f_\theta(\mathbf{x}), \quad \hat{\mathbf{x}} = g_\phi(\mathbf{z}), \quad \omega = h(\mathbf{x}, \hat{\mathbf{x}}),$$

where $\mathbf{z}$ denotes the latent distribution. $\hat{\mathbf{x}}$ is a vector of reconstructed samples, $f_\theta(\cdot)$ and $g_\phi(\cdot)$ are the variational encoder and variational decoder, respectively. ω is a vector of extracted reconstruction error features and $h(\cdot)$ denotes the process of calculating reconstruction error features, which can be relative Euclidean distance, absolute Euclidean distance, or cosine similarity [26].

Next, we feed the latent distribution $\mathbf{z}$ and the reconstruction error feature vector ω into Gaussian mixture model to learn its parameters and conduct density estimation for each sample. Existing variational autoencoder based Gaussian mixture models usually take $\mathbf{z}$ as the only input and largely neglect the extracted reconstructed error features, which results in suboptimal performance. By feeding the richer information in reconstruction error features in ω, the inference model should have better performance. On the other hand, in order to conduct density estimation in Gaussian mixture model, how to estimate the mixture probability for each component is of interest. However, existing approaches either randomize the mixture probability in the initiation step, or simply adopt average probability for each component. To tackle this challenge, we learn the mixture probability for each component by a deep belief network [35]. Deep belief network is rooted from Restricted Boltzmann Machine (RBM) and uses a number of neural layers as hidden units to probabilistically reconstruct inputs in an unsupervised fashion. It is efficient and effective to model the probability distribution. In the output layer of the deep belief network, a softmax function is used to generate the K-dimensional vector $\hat{\mathbf{\Gamma}}_i = [\hat{\gamma}_{i1}, \hat{\gamma}_{i2}, ..., \hat{\gamma}_{ik}, ..., \hat{\gamma}_{iK}]$ for each sample i, to estimate the mixture probability:

$$\hat{\mathbf{\Gamma}}_i = b_\lambda(\mathbf{z} + \omega)$$

where $b_\lambda(\cdot)$ denotes the deep brief network. Each element in vector $\hat{\mathbf{\Gamma}}_i$, i.e., $\hat{\gamma}_{ik}$, denotes the probability that the sample i is generated by component k in the Gaussian mixture model. Next, inspired by the maximization (M) step in Expectation-Maximization (EM) algorithm [17] , the inference model estimates the parameters in Gaussian mixture model with a batch of N samples and the corresponding mixture probabilities, $\hat{\mathbf{\Gamma}}_i$'s, as follows:

$$\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}, \hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} \mathbf{z}_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}, \hat{\sigma}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} (\mathbf{z}_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}, \tag{2}$$

where $\hat{\phi}_k$ is the component mixture probability for k's component. $\hat{\mu}_k$ and $\hat{\sigma}_k$ are the mean and covariance for component k, respectively. Now we can estimate the sample density, denoted by $\mathcal{E}(\mathbf{z}_i)$, by

$$\mathcal{E}(\mathbf{z}_i) = -\log\Big(\sum_{k=1}^{K} \hat{\phi}_k \frac{\exp(-\frac{1}{2}(\mathbf{z}_i - \hat{\mu}_k)^T \hat{\sigma}_k^{-1} (\mathbf{z}_i - \hat{\mu}_k))}{\sqrt{|2\pi\hat{\sigma}_k|}}\Big), \tag{3}$$

and $|\cdot|$ is the determinant of a matrix. In the testing phase, the sample density estimation works as the outlier detector which predicts outliers when the sample density is beyond a learned threshold, which can be obtained during the training phase.

B. Unifying the Variational Autoencoder with Deep Brief Network and Gaussian Mixture Model

Recall that decoupled learning suffers from suboptimal performance. In the proposed model learning framework, we unify the VAE, DBN, with the Gaussian mixture model. Specifically, we want to jointly optimize the entire system by adding the objective functions with reparameterization tricks, as follows:

$$\begin{aligned} \min \quad & \mathcal{J}(f_\theta, g_\phi, b_\lambda) = D_{KL}[q(\mathbf{z}, \Gamma|\mathbf{x}) || p(\mathbf{z}, \Gamma|\mathbf{x})] \\ & + \frac{\lambda_1}{N} \sum_{i=1}^{N} \mathcal{E}(\mathbf{z}_i) + \lambda_2 b_\lambda(\mathbf{z} + \omega) \end{aligned} \tag{4}$$

where $D_{KL}[q(\mathbf{z}, \Gamma|\mathbf{x}) || p(\mathbf{z}, \Gamma|\mathbf{x})]$ denotes the KL divergence between the posterior distribution $q(\mathbf{z}, \Gamma|\mathbf{x})$ and the likelihood distribution $p(\mathbf{z}, \Gamma|\mathbf{x})$, which will be clear in next section. $\mathcal{E}(\mathbf{z}_i)$ is the probability that we observe from the samples. Minimizing the sample density gives the best combination for variational autoencoder and Gaussian mixture model. λ_1 and λ_2 are hyperparameters to regularize the objective function and is set to be 0.1 and 0.001, respectively.

C. Analysis of Variational Lower Bound on Gaussian Mixture VAE

We want to model the network to maximize the likelihood of the high dimensional inputs $\mathbf{x}$. In the Gaussian mixture VAE, a mixture of Gaussian distribution is used as the prior in the latent space. Suppose there exist K components of Gaussian distribution. Γ is the mixture probabilities of each component in the Gaussian mixture model, which is learned by the deep belief network. Firstly, we sample the component based on the learned mixture probability distribution. Once the component is determined, the corresponding latent Gaussian distribution is determined as well. In the following, we use w to represent the random variable of the component's weight. Then, we can obtain the relationship between the log-likelihood of Gaussian mixture VAE and new variational lower bound as follows:

$$\begin{aligned}
&D_{KL}[q(\mathbf{z},\Gamma|\mathbf{x})||p(\mathbf{z},\Gamma|\mathbf{x})] \\
&= \sum_{\Gamma}\int_{\mathbf{z}} q(\mathbf{z},\Gamma|\mathbf{x}) \log\frac{q(\mathbf{z},\Gamma|\mathbf{x})}{p(\mathbf{z},\Gamma|\mathbf{x})} d\mathbf{z} \\
&= \sum_{\Gamma}\int_{\mathbf{z}} q(\mathbf{z},\Gamma|\mathbf{x}) \log\frac{q(\mathbf{z},\Gamma|\mathbf{x})}{p(\mathbf{z},\Gamma,\mathbf{x})} d\mathbf{z} \\
&\quad + \sum_{\Gamma}\int_{\mathbf{z}} q(\mathbf{z},\Gamma|\mathbf{x}) \log(p(\mathbf{x})) d\mathbf{z} \\
&= -\mathbf{E}_{q(\mathbf{z},\Gamma|\mathbf{x})} \log\frac{p(\mathbf{z},\Gamma,\mathbf{x})}{q(\mathbf{z},\Gamma|\mathbf{x})} + \log(p(\mathbf{x})) \\
&= -\mathcal{L}^{*}_{VAE} + \log(p(\mathbf{x}))
\end{aligned} \tag{5}$$

To minimize the KL divergence between the posterior distribution and the likelihood distribution, we maximize the new variational lower bound $\mathcal{L}^{*}_{VAE}$ under Gaussian mixture VAE. We assume $q(\mathbf{z},\Gamma|\mathbf{x})$ follows a mean-field distribution and can be factorized by:

$$q(\mathbf{z},\Gamma|\mathbf{x}) = q(\mathbf{z}|\mathbf{x})q(\Gamma|\mathbf{x}) \tag{6}$$

Then the new variational lower bound $\mathcal{L}^{*}_{VAE}$ is updated as follows:

$$\begin{aligned}
\mathcal{L}^{*}_{VAE} &= \mathbf{E}_{q(\mathbf{z},\Gamma|\mathbf{x})}[\log(p(\mathbf{z},\Gamma,\mathbf{x})) - \log(q(\mathbf{z},\Gamma|\mathbf{x}))] \\
&\overset{(6)}{=} \sum_{\Gamma}\int_{\mathbf{z}} q(\Gamma|\mathbf{x})q(\mathbf{z}|\mathbf{x})[\log(p(\mathbf{x}|\mathbf{z})) + \log(p(\mathbf{z}|\Gamma)) \\
&\quad + \log(p(\Gamma)) - \log(q(\mathbf{z}|\mathbf{x})) - \log(p(\Gamma|\mathbf{x}))]d\mathbf{z},
\end{aligned} \tag{7}$$

$\log(q(\mathbf{z}|\mathbf{x}))$ uses the encoder network to approximately model distribution. Then, the only unknown thing is how to formulate $\log(p(\Gamma|\mathbf{x}))$ to maximize the variational lower bound $\mathcal{L}^{*}_{VAE}$. We rewrite the function as follows:

$$\begin{aligned}
&\max_{q(\Gamma|\mathbf{x})} \mathcal{L}^{*}_{VAE} \\
&= \sum_{\Gamma}\int_{\mathbf{z}} q(\Gamma|\mathbf{x})q(\mathbf{z}|\mathbf{x})[\log(\frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}) + \log(\frac{p(\Gamma|\mathbf{z})}{q(\Gamma|\mathbf{x})})]d\mathbf{z} \\
&= \int_{\mathbf{z}} \{q(\mathbf{z}|\mathbf{x}) \log(\frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}) \\
&- q(\mathbf{z}|\mathbf{x})D_{KL}[q(\Gamma|\mathbf{x})||p(\Gamma|\mathbf{z})]\}d\mathbf{z}
\end{aligned} \tag{8}$$

As the first term in (8) is not relevant to Γ, we only need to consider minimizing the second term. That is to say, if $D_{KL}[q(\Gamma|\mathbf{x})||p(\Gamma|\mathbf{z})] = 0$ always holds, then $\mathcal{L}^{*}_{VAE}$ will reach its maximum with regard to $q(\Gamma|\mathbf{x})$, which corresponds to:

$$q(\Gamma|\mathbf{x}) = p(\Gamma|\mathbf{z}) = \frac{p(\Gamma)p(\mathbf{z}|\Gamma)}{\sum_{\Gamma} p(\Gamma)p(\mathbf{z}|\Gamma)} \tag{9}$$

In fact, (7) can be rewritten as:

$$\mathcal{L}^{*}_{VAE} = \mathbf{E}_{q(\mathbf{z},\Gamma|\mathbf{x})}[\log(p(\mathbf{x}|\mathbf{z}))] - D_{KL}(q(\mathbf{z},\Gamma|\mathbf{x})||p(\mathbf{z},\Gamma)) \tag{10}$$

The first term in (10) is the reconstruction term which helps reconstruct the input by considering both Γ and $\mathbf{z}$. The second term is the regularization term that makes the mixture of Gaussian prior be as close to variational posterior as possible.

V. Experiments

In this section, we use six public benchmark datasets to demonstrate the effectiveness of our model over baseline models for unsupervised outlier detection.

A. Dataset Description

Dataset	# Samples	# Dimensions	Outlier ratio
Arrhythmia	452	274	14.6%
KDDCUP (10%)	494,021	120	19.7%
KDDCUP	4,898,431	120	19.9%
MNIST	7,603	100	9.2%
Musk	3,602	166	3.2%
Thyroid	3,772	6	2.5%

TABLE I
Statistical description of public benchmark datasets

We employ six public benchmark datasets in the experiment. The detailed statistics of the datasets is listed in Table I and we elaborate the process of data pre-processing for each dataset as follows.

- **Arrhythmia.** The Arrhythmia dataset from the ODDS repository[1] is a multi-class classification dataset. The smallest classes, including 3, 4, 5, 7, 8, 9, 14, 15, are combined to form the outlier class and the rest of the classes are combined to form the normal class.
- **KDDCUP (10%).** The KDDCUP 10% dataset[2] originally contains samples of 41 features, where 34 of them are continuous and the rest 7 are discrete. We use one-hot representation to encode the features for categorical features, and eventually obtain a dataset with features of 120 dimensions. In this dataset, 19.7% of data samples are labeled as anomalies and the rest are labeled as normal samples.
- **KDDCUP.** The KDDCUP dataset is the full version of KDDCUP 10% dataset. We conduct the similar process of feature engineering as KDDCUP 10% dataset.
- **MNIST.** The original MNIST dataset of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. We obtain a subset of certain set from the ODDS repository. The digits have been size-normalized and centered in a fixed-size image. This dataset is converted for outlier detection as digit-zero class is considered as normal samples, while 700 images are sampled from digit-six class are labeled as the outliers. In addition, 100 features are randomly selected from total 784 features.
- **Musk.** The Musk dataset from ODDS repository contains several musk and non-musk classes. The non-musk classes j146, j147, and 252 are combined to form the normal class, while the musk classes 213 and 211 are labeled as anomalies without downsampling.

[1] http://odds.cs.stonybrook.edu/

[2] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99

- **Thyroid.** The Thyroid dataset is obtained from the ODDS repository. There are 3 classes in the original dataset. For this dataset, the hyperfunction class is labeled as outlier class and the other two classes are labeled as normal class, because hyperfunction is a clear minority class.

B. Baseline Models

We consider both traditional machine leaning and state-of-the-art deep learning methods as our baseline models. The detailed information for each baseline model is described as follows.

1) **OC-SVM.** One-class support vector machine (OC-SVM, [36]) is a popular kernel-based method used in outlier detection. We employ the radial basis function (RBF) kernel for all the tasks in the experiment,.
2) **DSEBM-e.** Deep structured energy based model (DSEBM-e, [37]) is a state-of-the-art deep learning method for unsupervised outlier detection. In DSEBM-e, sample energy is leveraged as the criterion to detect anomalies.
3) **VAE.** Variational Auto Encoder (VAE, [33]) is a popular deep generative model for unsupervised outlier detection. The reconstruction probability is used as the as the criterion to detect anomalies.
4) **DAGMM.** Deep Autoencoding Gaussian Mixture Model (DAGMM, [26]) is another cutting-edge deep learning method for unsupervised outlier detection by incorporating Gaussian Mixture model with deep autoencoding framework. In DAGMM, it adopts sample energy as the criterion to detect anomalies.

C. Performance Evaluation

We consider the following five metrics: accuracy, precision, recall, F1 score and area under the curve(AUC), for the system performance evaluation. We split each dataset into training part and testing part. 50% of data in each dataset will be randomly selected to form the training set and the rest 50% will be kept for testing. In the testing phase, we set the threshold to identify the abnormal samples based on the outlier ratio shown in Table I. For example, when our model performs on Arrhythmia dataset, we will mark top 14.6% samples of highest density as outliers.

Methods	Performance Evaluation @ Arrhythmia				
	Accuracy	Precision	Recall	F1	AUC
OC-SVM	0.8280	**0.4627**	0.3883	0.4222	0.6159
DSEBM-e	0.7743	0.3001	0.2792	0.2887	0.4957
VAE	0.7923	0.3328	0.3392	0.3360	0.5790
DAGMM	0.8097	0.3438	0.3333	0.3385	0.6123
Our Model	**0.8363**	0.4375	**0.4242**	**0.4308**	**0.6655**

TABLE II
PERFORMANCE EVALUATION ON BASELINE MODELS AND OUR MODEL. FOR EACH METRIC, THE BEST RESULT IS SHOWN IN BOLD.

Methods	Performance Evaluation @ KDDCUP (10%)				
	Accuracy	Precision	Recall	F1	AUC
OC-SVM	0.8095	0.7457	0.8523	0.7954	0.8088
DSEBM-e	0.7735	0.7369	0.7477	0.7423	0.7625
VAE	0.8990	0.7805	0.7903	0.7854	0.8830
DAGMM	0.9697	0.9409	0.9034	0.9218	0.9447
Our Model	**0.9739**	**0.9520**	**0.9141**	**0.9326**	**0.9513**

TABLE III
PERFORMANCE EVALUATION ON KDDCUP (10%). FOR EACH METRIC, THE BEST RESULT IS SHOWN IN BOLD.

Methods	Performance Evaluation @ KDDCUP				
	Accuracy	Precision	Recall	F1	AUC
OC-SVM	0.8185	0.7434	0.8433	0.7902	0.8021
DSEBM-e	0.7852	0.7183	0.7311	0.7246	0.7503
VAE	0.8389	0.7579	0.7710	0.7644	0.8230
DAGMM	0.9602	0.9180	0.8780	0.8975	0.9293
Our Model	**0.9686**	**0.9402**	**0.8992**	**0.9192**	**0.9425**

TABLE IV
PERFORMANCE EVALUATION ON KDDCUP. FOR EACH METRIC, THE BEST RESULT IS SHOWN IN BOLD.

We show in Table II - Table VII together the performance evaluation under five different metrics for both baseline models and our model on six real-world datasets. For each metric, the best result is shown in bold. The simulation results clearly demonstrate that our model achieves superior performance over the baseline models. In particular, compared the performance with the state-of-the-art methods, our model achieves, on average, 27% improvements on F1 score, which is encouraging. Our model even achieves 50% improvement at F1 score, compared with the best performance of baseline models on Thyroid dataset. Moreover, the improvement become more significant on datasets with relatively low outlier ratio. For instance, our model can still maintains good performance and achieves 33% improvement than baseline models in F1 score for datasets with less than 10% anomalies. By contrast, OC-SVM shows limited performance because of the well known “curse of dimensionality.” While DSEBM-e works reasonably well on multiple datasets, our model still outperforms. It is because we consider both the latent representation and reconstruction error features in density estimation phase and avoids decoupled model learning. VAE provides some acceptable performance on some certain datasets. However, its overall performance is still not as good as that of our model, simply because the assumption of unit Gaussian prior in the latent space limits the performance on datasets with multiple modality distribution, which can be alleviated by leveraging a mixture of Gaussian in the prior as proposed in our model. DAGMM, in most cases, reaches good performance among the baseline models. However, the performance, comparing with our model, is still limited when datasets has relatively low outlier ratio, because our models gives higher flexibility on the latent generative model by utilizing the variational

Methods	Performance Evaluation @ MNIST				
	Accuracy	Precision	Recall	F1	AUC
OC-SVM	0.6543	0.2875	0.2895	0.2885	0.4992
DSEBM-e	0.7795	0.3439	0.3239	0.3387	0.5859
VAE	0.7345	0.3319	0.3196	0.3251	0.5398
DAGMM	0.8848	0.3936	0.3699	0.3814	0.6547
Our Model	**0.9043**	**0.5015**	**0.4712**	**0.4859**	**0.7107**

TABLE V
PERFORMANCE EVALUATION ON MNIST. FOR EACH METRIC, THE BEST RESULT IS SHOWN IN BOLD.

Methods	Performance Evaluation @ Musk				
	Accuracy	Precision	Recall	F1	AUC
OC-SVM	0.9089	0.7998	0.8001	0.7999	0.8669
DSEBM-e	0.8993	0.7835	0.7817	0.7826	0.8476
VAE	0.9876	0.9290	0.9447	0.9368	0.9778
DAGMM	0.9941	0.8913	0.9111	0.9011	0.9539
Our Model	**0.9993**	**0.9783**	**1.0000**	**0.9890**	**0.9997**

TABLE VI
PERFORMANCE EVALUATION ON MUSK. FOR EACH METRIC, THE BEST RESULT IS SHOWN IN BOLD.

autoencoding framework. We can observe that our model maintain a robust performance on datasets with relatively low outlier ratio, like datasets in Musk and Thyroid.

D. Visualization of Latent Representation on Outlier Detection

In this section, we further demonstrate that our system achieves good performance on outlier detection from the learned low-dimensional representation. In lower dimensional space, we can better understand how the proposed model separates outliers from the normal samples.

Fig. 1 illustrates the learned low-dimensional representation after performing Principal Component Analysis (PCA) on latent vectors into 3D space on each dataset. The outlier ratio of datasets shown from fig. 1(a) to 1(f), is ranked from high to low. We can observe that the outlier samples are clearly separated out from the cluster of normal data for different datasets under different outlier ratio, which further demonstrates the robustness of our model on outlier detection. Moreover, we observe that when tackling datasets whose outlier ratio is relatively high, like KDDCUP (10%), the outlier samples gather into clusters and are distant to the normal clusters as shown in fig. 1(b). When tackling datasets whose outlier ratio is relatively low, like Thyroid, the outlier samples are identically and individually separated out from the normal clusters as shown in fig. 1(f). Generally, the distance between normal and outlier samples are not negligible from the perspective of learned low dimensional representation.

Methods	Performance Evaluation @ Thyroid				
	Accuracy	Precision	Recall	F1	AUC
OC-SVM	0.8380	0.3639	0.4239	0.3887	0.5659
DSEBM-e	0.3835	0.1319	0.1319	0.1319	0.3835
VAE	0.7227	0.3395	0.3592	0.3491	0.4890
DAGMM	0.9740	0.4737	0.3830	0.4235	0.6861
Our Model	**0.9836**	**0.7105**	**0.5745**	**0.6353**	**0.7842**

TABLE VII
PERFORMANCE EVALUATION ON THYROID. FOR EACH METRIC, THE BEST RESULT IS SHOWN IN BOLD.

E. Visualization on Histogram of Density Distribution

In this section, we further demonstrate the effectiveness of the learned density function for distinguishing outliers from the normal ones. Fig. 2 illustrates the histogram of density distributions for both normal samples and outlier samples on each dataset. Considering that samples in dataset Arrhythmia is limited and hard to visualize on histogram plot clearly, we only present visualization figures of density distribution on the five rest datasets. From the histogram of density distribution in Fig. 2 we can observe that the density distribution of normal samples and outlier samples are clearly distinguished from each other. Although there are small overlaps in Fig. 2(c) and Fig. 2(e), the density distribution of normals and outliers are remarkably distinguished in most of datasets with good performance (with high F1 score), i.e., KDDCUP and Musk, which further demonstrates the effectiveness of our scheme on outlier detection with our learned density function.

VI. CONCLUSION

In this paper, we have proposed a unified unsupervised Gaussian mixture variational autoencoder for outlier detection in high dimensional data. In particular, we have trained a deep latent embedding as a data generative model. Instead of assuming single Gaussian distribution as prior in the data generative procedure, we have devised a Gaussian mixture model in which a deep brief network is trained with both the samples from the generative model and distance based features of reconstruction error to learn the mixture probabilities. The inference model jointly optimizes the variational autoencoder, the deep brief network, and the Gaussian mixture model to avoid the decoupled model learning. We have conducted extensive experiments and found that the performance of the proposed framework achieves significant improvements compared to state-of-the-art outlier detection schemes.

ACKNOWLEDGMENT

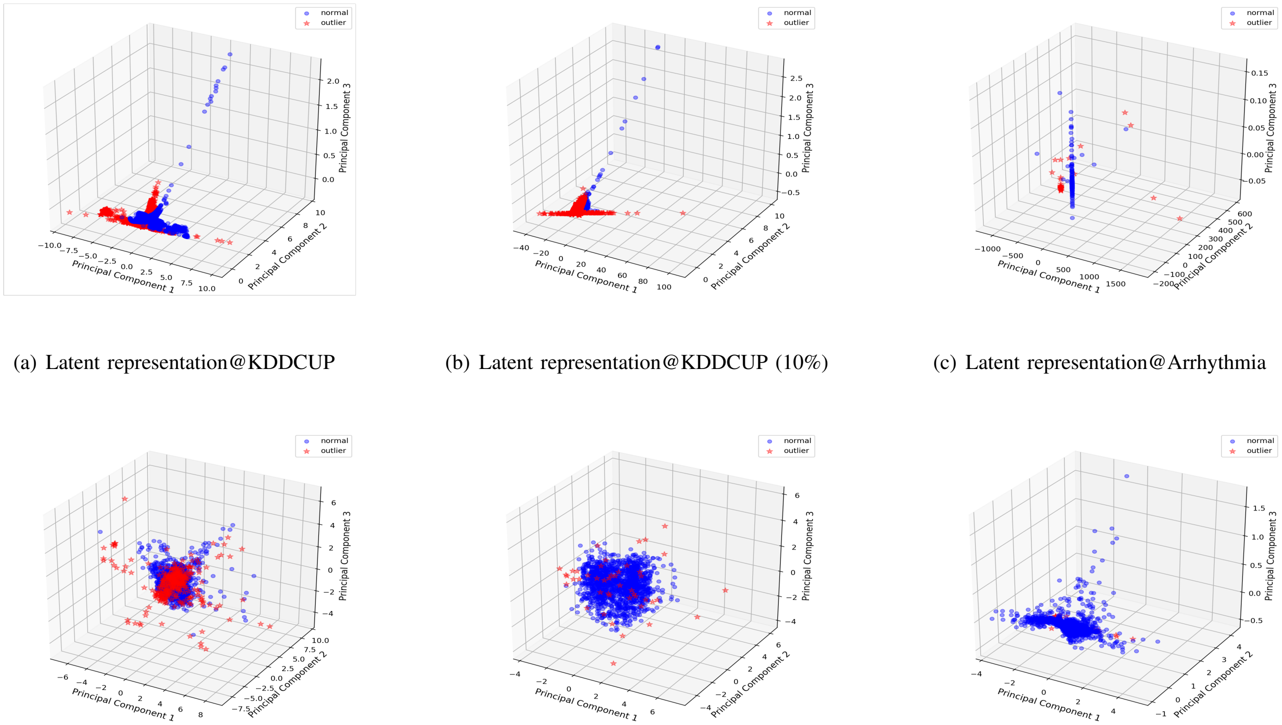
This work was partially supported by the US National Science Foundation under grants CNS-1602172 and CNS-1566479.

REFERENCES

[1] V. R. Jakkula and D. J. Cook, “Detecting anomalous sensor events in smart home data for enhancing the living experience.” *Artificial intelligence and smarter living*, vol. 11, no. 201, p. 1, 2011.

[2] A. Agovic, A. Banerjee, A. R. Ganguly, and V. Protopopescu, “Anomaly detection in transportation corridors using manifold embedding,” in *Proceedings of the 1st International Workshop on Knowledge Discovery from Sensor Data*, 2007, pp. 435–455.

[3] L. Akoglu, H. Tong, and D. Koutra, “Graph based anomaly detection and description: a survey,” *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.



(a) Latent representation@KDDCUP

(b) Latent representation@KDDCUP (10%)

(c) Latent representation@Arrhythmia

(d) Latent representation@MNIST

(e) Latent representation@Musk

(f) Latent representation@Thyroid

Fig. 1. Visualization on the learned latent representation on each dataset in 3D space. The blue points denote the normal samples, and the red star points indicate the outliers. In lower dimensional space, we can better observe that our model successfully separates the outliers from the normal samples in each dataset.

[4] W. Liao, C. Luo, S. Salinas, and P. Li, "Efficient secure outsourcing of large-scale convex separable programming for big data," *IEEE Transactions on Big Data*, 2017.

[5] S. Basu and M. Meckesheimer, "Automatic outlier detection for time series: an application to sensor data," *Knowledge and Information Systems*, vol. 11, no. 2, pp. 137–154, 2007.

[6] P. H. Tran, K. P. Tran, T. T. Huong, C. Heuchenne, P. HienTran, and T. M. H. Le, "Real time data-driven approaches for credit card fraud detection," in *Proceedings of the 2018 International Conference on E-Business and Applications*. ACM, 2018, pp. 6–9.

[7] A. G. Tartakovsky, A. S. Polunchenko, and G. Sokolov, "Efficient computer network anomaly detection by changepoint detection methods," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 4–11, 2013.

[8] N. Stojanovic, M. Dinic, and L. Stojanovic, "A data-driven approach for multivariate contextualized anomaly detection: Industry use case," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1560–1569.

[9] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927, 2016.

[10] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," *arXiv preprint arXiv:1806.04808*, 2018.

[11] H.-P. Kriegel, A. Zimek *et al.*, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 444–452.

[12] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "Pca-based multivariate statistical network monitoring for anomaly detection," *Computers & Security*, vol. 59, pp. 118–137, 2016.

[13] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. ACM, 2014, p. 4.

[14] D. Reynolds, "Gaussian mixture models," *Encyclopedia of biometrics*, pp. 827–832, 2015.

[15] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[16] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[18] X. Chen, J. Ji, K. Loparo, and P. Li, "Real-time personalized cardiac arrhythmia detection and diagnosis: A cloud computing architecture," in *Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on*. IEEE, 2017, pp. 201–204.

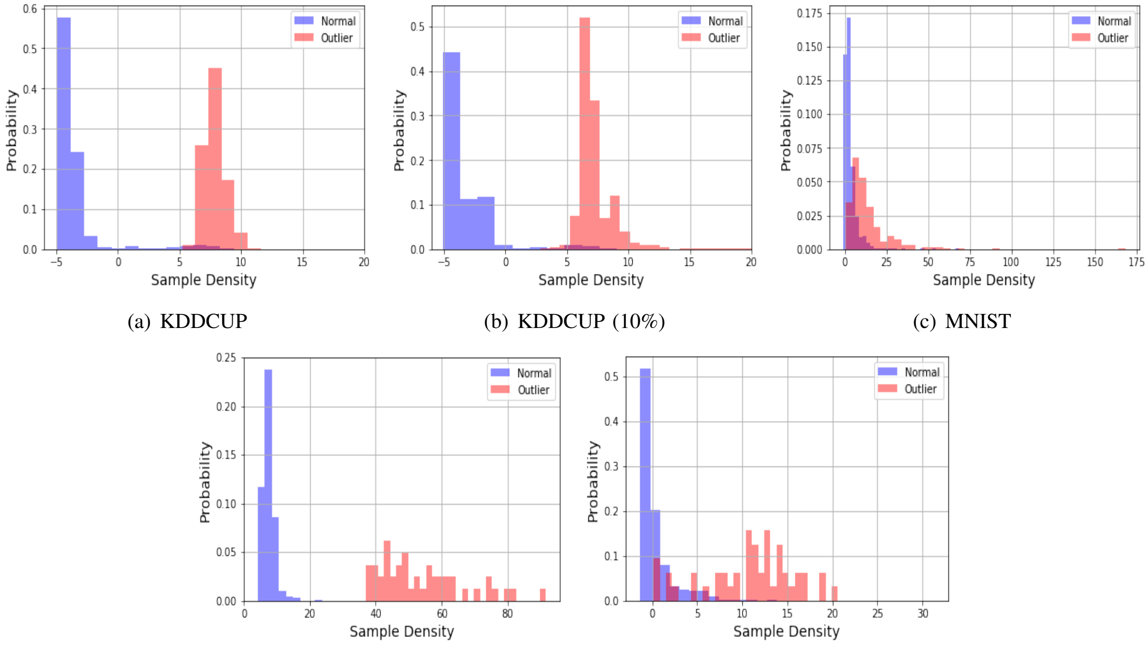
[19] S. R. Gaddam, V. V. Phoha, and K. S. Balagani, "K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 345–354, 2007.

[20] N. Abe, B. Zadrozny, and J. Langford, "Outlier detection by active learning," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 504–509.

[21] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484–497, 2017.

[22] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *methods*, vol. 9, no. 5, 2015.

[23] X. Tao, D. Kong, Y. Wei, and Y. Wang, "A big network traffic data



(a) KDDCUP

(b) KDDCUP (10%)

(c) MNIST

(d) Musk

(e) Thyroid

Fig. 2. Histogram of density distribution on each dataset. The horizontal axis represents the density space, and the vertical axis represents the corresponding histogram of distribution for both normal and outlier samples. From the histogram of density distribution, we can observe that the density distribution of normal samples and outlier samples are clearly distinguished from each other.

fusion approach based on fisher and deep auto-encoder," *Information*, vol. 7, no. 2, p. 20, 2016.

[24] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," in *Soft computing in industrial applications*. Springer, 2011, pp. 293–303.

[25] L. Xiong, B. Póczos, and J. G. Schneider, "Group anomaly detection using flexible genre models," in *Advances in neural information processing systems*, 2011, pp. 1071–1079.

[26] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," 2018.

[27] C. Zhou and R. 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.

[28] E. Nalisnick, L. Hertel, and P. Smyth, "Approximate inference for deep latent gaussian mixtures," in *NIPS Workshop on Bayesian Deep Learning*, vol. 2, 2016.

[29] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *arXiv preprint arXiv:1611.02648*, 2016.

[30] M. Johnson, D. K. Duvenaud, A. Wiltschko, R. P. Adams, and S. R. Datta, "Composing graphical models with neural networks for structured representations and fast inference," in *Advances in neural information processing systems*, 2016, pp. 2946–2954.

[31] R. Shu, J. Brofos, F. Zhang, H. H. Bui, M. Ghavamzadeh, and M. Kochenderfer, "Stochastic video prediction with conditional density estimation," in *ECCV Workshop on Action and Anticipation for Visual Learning*, vol. 2, 2016.

[32] Y. Guo, W. Liao, Q. Wang, L. Yu, T. Ji, and P. Li, "Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach," in *Proceedings of the 10th Asian Conference on Machine Learning (ACML18)*, Beijing, China, Nov. 14-16, 2018.

[33] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, pp. 1–18, 2015.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[35] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[36] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class svm for learning in image retrieval," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1. IEEE, 2001, pp. 34–37.

[37] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep structured energy based models for anomaly detection," *arXiv preprint arXiv:1605.07717*, 2016.