Multidimensional Time Series Anomaly Detection: A GRU-based Gaussian Mixture Variational Autoencoder Approach

Yifan Guo* YXG383@CASE.EDU
Weixian Liao+ WLIAO@TOWSON.EDU
Qianlong Wang* QXW204@CASE.EDU
Lixing Yu* LXY257@CASE.EDU
Tianxi Ji* TXJ116@CASE.EDU
Pan Li* PXL288@CASE.EDU
*Department of EECS, Case Western Reserve University, Cleveland, OH 44106, USA
+Department of Computer and Information Sciences, Towson University, Towson, MD 21252, USA

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

Unsupervised anomaly detection on multidimensional time series data is a very important problem due to its wide applications in many systems such as cyber-physical systems, the Internet of Things. Some existing works use traditional variational autoencoder (VAE) for anomaly detection. They generally assume a single-modal Gaussian distribution as prior in the data generative procedure. However, because of the intrinsic multimodality in time series data, previous works cannot effectively learn the complex data distribution, and hence cannot make accurate detections. To tackle this challenge, in this paper, we propose a GRU-based Gaussian Mixture VAE system for anomaly detection, called GGM-VAE. In particular, Gated Recurrent Unit (GRU) cells are employed to discover the correlations among time sequences. Then we use Gaussian Mixture priors in the latent space to characterize multimodal data. The proposed detector reports an anomaly when the reconstruction probability is below a certain threshold. We conduct extensive simulations on real world datasets and find that our proposed scheme outperforms the state-of-the-art anomaly detection schemes and achieves up to 5.7% and 7.2% improvements in accuracy and F1 score, respectively, compared with existing methods.

Keywords: Anomaly detection, gated recurrent unit (GRU), Gaussian Mixture model, variational autoencoder (VAE).

1. Introduction

Anomalies, also referred to as outliers, are defined as observations which deviate so much from the other observations as to arise suspicions that they were generated by different mechanisms. Anomaly detection has been a widely researched problem in machine learning and is of paramount importance in many areas such as intrusion detection (Portnoy et al. (2001)), fraud detection (Kou et al. (2004)), health monitoring (Chen et al. (2017)). The importance of anomaly detection lies in the fact that anomalies in data translate to significant (and often critical) information in a wide variety of application domains (Chandola et al. (2007), Liao et al. (2017)). For instance, in computer networks, anomalous patterns can

indicate an action of sending out sensitive information to an unauthorized destination. In fraud detection, outliers can mean credit card theft, misuse, or unauthorized transactions.

With their widespread success in numerous machine learning tasks, there have been quite a few deep learning approaches in the literature proposed for anomaly detection, which, based on whether data labels are used in the training process, can be categorized into supervised, semi-supervised, and unsupervised learning techniques. In particular, unsupervised learning approaches are preferably used for anomaly detection compared with semi-supervised and supervised learning approaches. The reasons are as follows. First, training data is usually imbalanced. Anomalous instances are far fewer than normal instances, which inevitably raises the issues caused by imbalanced class distributions. Second, labeling is often conducted manually by human experts with domain knowledge. In many cases it is prohibitively expensive and cumbersome to obtain hand-labeled data which is accurate and represents all types of anomalous behaviors (Chandola et al. (2009)). Therefore, tremendous efforts have been devoted to unsupervised anomaly detection.

In this study, we investigate the problem of unsupervised anomaly detection on multimodal sensory data. A common approach in the literature trains a one-class classifier with normal data. However, many previous schemes (Park et al. (2018)) cannot well deal with high-dimensional multimodal sensory data, because relying only on lower dimensional representation can easily lose critical information for anomaly detection. Particularly, Chandola et al. (2009) present some works which employ an autoencoder (AE) or a variational autoencoder (VAE). The idea behind this is that autoencoders can reconstruct normal data with small errors, while the reconstruction errors of anomalous data are usually much larger. Unfortunately, most previous works cannot well characterize the original data distribution, especially when it is strongly multi-modal, as they generally only assume a single Gaussian distribution as the prior in the data generative procedure.

To tackle these challenges, in this paper we propose an unsupervised GRU-based Gaussian Mixture VAE, called GGM-VAE, for anomaly detection. In particular, Gated Recurrent Unit (GRU) cells are employed to discover the correlations among time sequences. Then we use Gaussian Mixture prior in the latent space to characterize the multimodal data. The VAE infers the latent embedding and reconstruction probability in a variational manner by optimizing the variational lower bound. The proposed detector reports an anomaly when the reconstruction probability is below a certain threshold. We conduct extensive simulations and find that our proposed unsupervised scheme achieves the best performance under different metrics compared with the state-of-the-art unsupervised approaches.

Our main contributions in this paper are summarized as follows:

- We devise an unsupervised Gaussian Mixture VAE, called GGM-VAE, that can effectively perform anomaly detection on multidimensional time series data.
- We leverage the Gaussian Mixture prior in the latent representation to characterize the intrinsic multimodality in time series data.
- Gated Recurrent Unit (GRU) cells are employed under the VAE framework to discover the correlations among the time series data.
- Experimental results on real world datasets show that the proposed scheme outperforms the state-of-the-art schemes.

2. Related Work

Anomaly detection has been studied for decades. We focus on the most related works that apply machine learning techniques to anomaly detection. Based on whether the labels are used in the training process, they can be categorized into supervised, semi-supervised, and unsupervised anomaly detection.

Specifically, Gaddam et al. (2007) utilize a supervised ID3 decision tree to detect anomalies in computer networks. Abe et al. (2006) consider the anomaly detection problem as a classification problem and propose a supervised active learning scheme. Besides, Ashfaq et al. (2017) present a fuzzy theory based semi-supervised learning approach for intrusion detection. Li et al. (2015) propose several methods for malicious code detection which requires human interference to distinguish between the actual intrusion and false positive ones. However, such supervised and semi-supervised learning techniques assume that labels are available for partial or all the training dataset, which is both time and efforts consuming and may even be impractical in many 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. (2011) 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 we deal with time series data. Besides, due to very high computational complexity, clustering based approaches can hardly be directly applied in data of high dimensionality. Reconstruction based methods like in Zong et al. (2018) assume that anomalies are incompressible and thus cannot be effectively reconstructed from lower-dimensional latent space projections. Zhou and Paffenroth (2017) propose a deep autoencoder to detect anomalies. However, the performance of these methods is limited because they only analyze anomaly from reconstruction errors.

Related to our proposed scheme, Yamanishi et al. (2004) propose an expectation-maximization (EM) algorithm based GMM model for online unsupervised outlier detection. The system performance is limited due to the low convergence rate when applying EM algorithms. Johnson et al. (2016) incorporate probabilistic graphical models to improve the traditional VAE sturcture. Nalisnick et al. (2016) propose an architecture that combines VAE and GMM together. It employs a Gaussian Mixture latent space to improve the capacity of the original VAE. Moreover, Dilokthanakul et al. (2016) further study the Gaussian Mixture VAE to relieve the problem of over-regularization. Shu et al. (2016) design a new system called GM-CVAE, which integrate Conditional Variational Autoencoder(CVAE) with Gaussian Mixture prior to model the transition images between video frames. Note that these models are mostly intended for image clustering tasks, where each input is an individual image, and cannot be directly employed to process time sequence data as in this study. The main challenge here lies in discovering the temporal correlations in time sequences. In this study, we propose to use GRU cells in both the encoder and the decoder to discover the data correlation and dependency.

3. Preliminaries

3.1. Autoencoder based Anomaly Detection

An autoencoder is an artificial neural network that consists of sequentially connected encoder and decoder networks. The encoder learns a compressed representation, i.e., latent variables of the input data, which is fed into the decoder network to reconstruct the input. This network tries to minimize the reconstruction error, which is defined as the difference between the output of the decoder and the original input. The traditional autoencoder based anomaly detection method is a deviation based anomaly detection method in a semi-supervised learning fashion (An and Cho (2015)). The reconstruction error is set as the anomaly score, while samples with high reconstruction errors are considered as anomalies. In the training phase, only normal data will be used to train the autoencoder, aiming to minimize the reconstruction error, so that the autoencoder can recognize the characteristics of normal data. In the testing phase, the learned autoencoder will be able to reconstruct normal data will small reconstruction errors, but fail with anomalous data which the autoencoder has not encountered before and thus have relatively higher reconstruction errors compared with normal data. Thus, by comparing whether the anomaly score is above a predefined threshold, an autoencoder can determine whether the tested data is anomalous.

3.2. Variational Autoencoder based Anomaly 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 (An and Cho (2015)). 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. Kingma and Welling (2013) 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 ϕ, θ 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:

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

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 (Kingma and Welling (2013)) can be used.

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

Algorithm 1 Varational auto encoder 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 α

Output: Sequence of anomaly predictiosn S

$\theta, \phi \leftarrow$ Initialize parameters
$f_\theta, g_\phi, \alpha \leftarrow$ 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

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.

4. A GRU-based Gaussian Mixture Variational Autoencoder

Traditional VAE uses single-modal Gaussian distribution as the prior in the latent space because this allows easy inference and learning. However, such an assumption is over-simplified because usually a single-modal latent distribution cannot approximate the original data distribution well, especially when input data distributions are strongly multimodal. Compared with traditional VAE, Gaussian Mixture VAE uses a mixture of Gaussians as prior in the latent space. In so doing, Gaussian Mixture VAE could learn complex and informative hidden distributions and better approximate the original data distribution. In this section, we first present a novel model for anomaly detection by integrating GRU cells with Gaussian Mixture VAE, which is called GGM-VAE. Then, the analysis on the variational lower bound of Gaussian Mixture VAE and the description of the GGM-VAE based anomaly detection algorithm are demonstrated subsequently.

4.1. System architecture

We consider a system with D sensors. The system status at time t is denoted by $\mathbf{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \ldots, x_s^{(t)}, \ldots, x_D^{(t)}]$, where $x_s^{(t)}$ represents the value of sensor s ($s = 1, 2, ..., D$) at time t. In practice, the reading of a sensor can be a scalar or a vector. Without loss of generality, we consider $x_s^{(t)}$ as a scalar in the following, and the proposed system can be easily extended to vector-valued sensors. We group the time series system statuses into time windows of size T and each group of time series data, say the ith, is a $D \times T$ matrix denoted as $\mathbf{X}^{(i)} = [\mathbf{x}^{(i,1)}; \mathbf{x}^{(i,2)}; \ldots; \mathbf{x}^{(i,T)}]_{D \times T}$, where $\mathbf{x}^{(i,t)} = \mathbf{x}^{((i-1)\times T+t)}$. Figure 1 shows the system architecture, where GRU cells are introduced in both the encoder and the decoder of a Gaussian Mixture VAE to mine the data dependency in the time domain and among different sensors. Long Short Term Memory (LSTM) Network, a representative of Recurrent Neural Networks, has been shown to be capable of discovering long-term dependency among sequence data (Gers et al. (1999)), while GRU network is a variation on the LSTM Network (Cho et al. (2014)). Chung et al. (2014) have demonstrated that GRU network can achieve better performance in discovering the correlations among sequence data over traditional LSTM network with even smaller datasets due to its fewer parameters. Therefore, in our system we utilize GRU cells in both the encoder and the decoder.

The system works as follows. We first feed $\mathbf{X}^{(i)}$ into the GRU-based encoder, where the internal equations of GRU cell are:

$$
\begin{aligned}
\mathbf{z}^{(i,t)} &= \sigma(\mathbf{W}_{xz}\mathbf{x}^{(i,t)} + \mathbf{U}_{hz}\mathbf{h}^{(i,t-1)}) \\
\mathbf{r}^{(i,t)} &= \sigma(\mathbf{W}_{xr}\mathbf{x}^{(i,t)} + \mathbf{U}_{hr}\mathbf{h}^{(i,t-1)}) \\
\tilde{\mathbf{h}}^{(i,t)} &= tanh(\mathbf{W}_{xh}\mathbf{x}^{(i,t)} + \mathbf{U}_{rh}(\mathbf{r}^{(i,t)} \otimes \mathbf{h}^{(i,t-1)})) \\
\mathbf{h}^{(i,t)} &= (1 - \mathbf{z}^{(i,t)}) \otimes \mathbf{h}^{(t,t-1)} + \mathbf{z}^{(i,t)} \otimes \tilde{\mathbf{h}}^{(i,t)},
\end{aligned}
$$

where σ is the sigmoid function, $\otimes$ is an element-wise multiplication operator, $\mathbf{z}^{(i,t)}$, $\mathbf{r}^{(i,t)}$, $\tilde{\mathbf{h}}^{(i,t)}$, and $\mathbf{h}^{(i,t)}$ are the update gate, reset gate, candidate activation and output vectors, respectively, at the t hidden state of the ith input sebsequence, and $\mathbf{W}_{xz}$, $\mathbf{W}_{xr}$, $\mathbf{W}_{xh}$, $\mathbf{U}_{hz}$, $\mathbf{U}_{hr}$, and $\mathbf{U}_{rh}$ are the learned weight matrices.

Afterwards, the output of the GRU-based encoder will be mapped to the Gaussian Mixture latent space. The corresponding output will be further transported to the GRU-based decoder part to reconstruct the original input. The loss function measures the difference of the reconstructed data from the original input.

4.2. Analysis of variational lower bound of Gaussian Mixture VAE

Next, we conduct theoretical analysis of variational lower bound of Gaussian Mixture VAE. Specifically, in the Gaussian Mixture VAE, a mixture of Gaussian distributions are used as the prior in the latent space. Suppose there exist K components in the Gaussian Mixture, following a categorical prior distribution $\text{Cat}(\boldsymbol{\pi})$ where $\boldsymbol{\pi}$ represents the distribution parameters. Particularly, $w_k^{(t)}$ is the prior probability of the kth component ($k = 1, 2, \ldots, K$). As we will see later in Algorithm 2, for each data sample, we first select one component based on the categorical prior distribution among K components. Then, once the component is determined, the corresponding latent Gaussian distribution is determined as well. In the

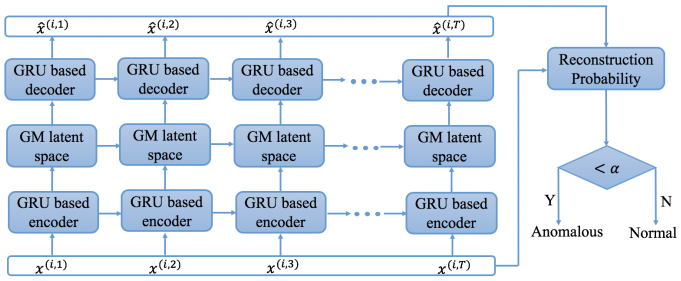


Figure 1: System architecture for GRU-based Gaussian Mixture VAE on anomaly detection

following, we use $w^{(t)}$ instead of $w_k^{(t)}$, $p(\cdot)$ instead of $p_\theta(\cdot)$, and $q(\cdot)$ instead of $q_\phi(\cdot)$, for simplicity. Then, we can caculate the Kullback-Leibler divergence of the approximate from the true posterior as:

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

To minimize the KL divergence, we need to maximize the variational lower bound $\mathcal{L}^*_{VAE}$ under Gaussian Mixture VAE. We assume that $q(\mathbf{z}^{(t)}, w^{(t)}|\mathbf{x}^{(t)})$ follows a mean-field distribution, i.e., the variables can be partitioned and they are independent. Thus, we get

$$
q(\mathbf{z}^{(t)}, w^{(t)}|\mathbf{x}^{(t)}) = q(\mathbf{z}^{(t)}|\mathbf{x}^{(t)}) q(w^{(t)}|\mathbf{x}^{(t)}). \tag{2}
$$

Then the variational lower bound $\mathcal{L}^*_{VAE}$ can be calculated as:

$$
\begin{aligned}
\mathcal{L}^*_{VAE} &= \mathbf{E}_{q(\mathbf{z}^{(t)}, w^{(t)}|\mathbf{x}^{(t)})}[\log p(\mathbf{z}^{(t)}, w^{(t)}, \mathbf{x}^{(t)}) - \log q(\mathbf{z}^{(t)}, w^{(t)}|\mathbf{x}^{(t)})] \\
&\overset{(2)}{=} \sum_{w^{(t)}} \int_{\mathbf{z}^{(t)}} q(w^{(t)}|\mathbf{x}^{(t)}) q(\mathbf{z}^{(t)}|\mathbf{x}^{(t)}) \Big[\log p(\mathbf{x}^{(t)}|\mathbf{z}^{(t)}) + \log p(\mathbf{z}^{(t)}|w^{(t)}) \\
&\quad + \log p(w^{(t)}) - \log q(\mathbf{z}^{(t)}|\mathbf{x}^{(t)}) - \log q(w^{(t)}|\mathbf{x}^{(t)}) \Big] d\mathbf{z}^{(t)}.
\end{aligned} \tag{3}
$$

To maximize $\mathcal{L}^*_{VAE}$, we may follow a similar process to that in regular VAE. Thus, we can rewrite equation (3) like the following:

$$
\mathcal{L}^*_{VAE} = \mathbf{E}_{q(\mathbf{z}^{(t)}, w^{(t)}|\mathbf{x}^{(t)})}[\log p(\mathbf{x}^{(t)}|\mathbf{z}^{(t)})] - D_{KL}(q(\mathbf{z}^{(t)}, w^{(t)}|\mathbf{x}^{(t)})||p(\mathbf{z}^{(t)}, w^{(t)})). \tag{4}
$$

The first term in (4) is the reconstruction term which helps reconstruct the input by considering both $w^{(t)}$ and $\mathbf{z}^{(t)}$, while the second term is the regularization term that makes the mixture of Gaussian prior as close to the variational posterior as possible. However, we find that we cannot maximize (4) directly because it is hard to have the analytical expression due to the involvement of the Gaussian mixture. Let us take a closer look at (3). In fact, the only unknown distribution is $\log q(w^{(t)}|\mathbf{x}^{(t)})$. To find this distribution, we can rewrite (3) as follows:

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

As the first term in (5) is not relevant to $w^{(t)}$, in order to maximize $\mathcal{L}^*_{VAE}$, we only need to minimize the second term. Therefore, if $D_{KL}(q(w^{(t)}|\mathbf{x}^{(t)})||p(w^{(t)}|\mathbf{z}^{(t)})) = 0$ always holds, then $\mathcal{L}^*_{VAE}$ achieves its maximum with regard to $q(w^{(t)}|\mathbf{x}^{(t)})$. Consequently, by having

$$q(w^{(t)}|\mathbf{x}^{(t)}) = p(w^{(t)}|\mathbf{z}^{(t)}) = \frac{p(w^{(t)}) p(\mathbf{z}^{(t)}|w^{(t)})}{\sum_{w^{(t)}} p(w^{(t)}) p(\mathbf{z}^{(t)}|w^{(t)})},$$

we can get an analytical expression for $\mathcal{L}^*_{VAE}$, and hence maximize it using methods like Stochastic Gradient Descent (Kingma and Welling (2013)).

4.3. GGM-VAE based anomaly detection algorithm

In this section, we demonstrate how our GGM-VAE model can be used to detect anomalies with anomaly detection algorithm in detail. Algorithm 2 describes the GGM-VAE based anomaly detection algorithm in an unsupervised learning fashion.

Specifically, we first train the GGM-VAE model with unlabeled training data, which can include both normal and abnormal data samples. After the training process, the system will learn the parameters in the encoder and the decoder, and the reconstruction probabilities corresponding to the training data. Let the anomaly ratio of the training data, which can be collected after the training process, be denoted by r ($r \in [0, 1]$). We choose the (100*r)-th percentile as the threshold α for testing by ranking the reconstruction probabilities of the training samples in descending order[1].

Then, in the testing phase, each data sample $\mathbf{x}^{(i,t)}$ is fed into the encoder to get the corresponding mean and standard deviation vectors in the latent space. Different from regular VAE where there is only one set of mean and standard deviation vectors, in Gaussian Mixture VAE, there are K sets of mean and standard deviation vectors, i.e., $\mu_{\mathbf{z}|w_k}[i, t]$ and

1. In the literatures, there are a few works like An and Cho (2015) and Malhotra et al. (2016), discussing how to determine the reconstruction probability threshold, which is out of the scope of this paper.

Algorithm 2 GGM-VAE based anomaly detection algorithm

Input : $\mathbf{X_{train}} = \{\mathbf{x}^{(1,1)}, \ldots, \mathbf{x}^{(1,T)}, \ldots, \mathbf{x}^{(N_{train},1)}, \ldots, \mathbf{x}^{(N_{train},T)}\}$, $\mathbf{X_{test}} = \{\mathbf{x}^{(1,1)}, \ldots, \mathbf{x}^{(1,T)}, \ldots, \mathbf{x}^{(N_{test},1)}, \ldots, \mathbf{x}^{(N_{test},T)}\}$, Time window T, Number of components K, Weight prior distribution $W \sim \text{Cat}(\boldsymbol{\pi})$

Output: Sequence of anomaly prediction S

$\theta, \phi \leftarrow$ Initialize parameters
$f_\theta, g_\phi, \alpha \leftarrow$ Train the Gaussian Mixture VAE network using the raining sequence $\mathbf{X_{train}}$
for $i = 1$ **to** N_{test} **do**
  for $t = 1$ **to** T **do**
    $\mu_{\mathbf{z}|w_k}[i,t], \sigma_{\mathbf{z}|w_k}[i,t] = f_\theta(\mathbf{z}|\mathbf{x}^{(i,t)}, w_k)$ for each component k $(k = 1, \ldots, K)$
    for $l = 1$ **to** L **do**
      Sample a component k^* based on prior distribution W
      Draw a sample from $\mathbf{z} \sim \mathcal{N}(\mu_{\mathbf{z}|w_{k^*}}[i,t], \sigma_{\mathbf{z}|w_{k^*}}[i,t])$
      $\mu_{\hat{\mathbf{x}}|\mathbf{z},w_{k^*}}[i,t,l], \sigma_{\hat{\mathbf{x}}|\mathbf{z},w_{k^*}}[i,t,l] = g_\phi(\mathbf{x}|\mathbf{z}^{(i,t,l)}, w_{k^*})$ for the component k^*
    end
    Reconstruction Probability
    $RP(\mathbf{x}|\hat{\mathbf{x}})[i,t] = \frac{1}{L}\sum_{l=1}^{L} \mathcal{N}(\mathbf{x}^{(i,t)}|\mu_{\hat{\mathbf{x}}|\mathbf{z},w_{k^*}}[i,t,l], \sigma_{\hat{\mathbf{x}}|\mathbf{z},w_{k^*}}[i,t,l])$
    if $RP(\mathbf{x}|\hat{\mathbf{x}})[i,t] < \alpha$ **then**
      $\mathbf{x}^{(i,t)}$ is an anomaly, $S[i,t] =$ "Anomalous"
    end
    else
      $\mathbf{x}^{(i,t)}$ is not an anomaly, $S[i,t] =$ "Normal"
    end
  end
end

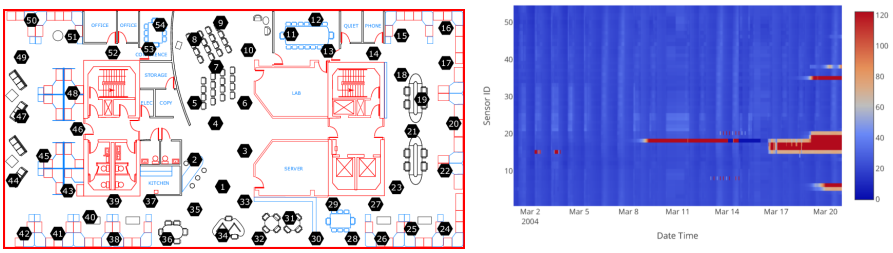
$\sigma_{\mathbf{z}|w_k}[i,t]$ $(k \in [K])$ in the latent space. The system chooses certain set of mean and the corresponding standard deviation vector, say the k^*-th, by following the categorical prior distribution $W \sim \text{Cat}(\boldsymbol{\pi})$. After that, the latent vector $\mathbf{z}$ will be sampled on the Gaussian distribution with the selected mean and standard deviation vectors, i.e., $\mu_{\mathbf{z}|w_{k^*}}[i,t]$ and $\sigma_{\mathbf{z}|w_{k^*}}[i,t]$. This process repeats L times to generate L samples. Each sample $\mathbf{z}^{(i,t,l)}$, which represents the lth latent vector for input data $\mathbf{x}^{(i,t)}$, is fed into the decoder to get the corresponding reconstructed mean vector $\mu_{\hat{\mathbf{x}}|\mathbf{z},w_{k^*}}[i,t,l]$ and reconstructed standard deviation vector $\sigma_{\hat{\mathbf{x}}|\mathbf{z},w_{k^*}}[i,t,l]$. By fitting the input data $\mathbf{x}^{(i,t)}$ into the the multivariate Gaussian distribution with the learned reconstructed mean vector and reconstructed standard deviation vector, we can get the corresponding reconstruction probability $\mathcal{N}(\mathbf{x}^{(i,t)}|\mu_{\hat{\mathbf{x}}|\mathbf{z},w_{k^*}}[i,t,l], \sigma_{\hat{\mathbf{x}}|\mathbf{z},w_{k^*}}[i,t,l])$ for the lth generated latent vector. After averaging over all the L reconstruction probabilities, we can get the final reconstruction probability $RP(\mathbf{x}|\hat{\mathbf{x}})[i,t]$ for each input $\mathbf{x}^{(i,t)}$. Therefore, the system can determine whether the data sample is anomalous by checking if the reconstruction probability is smaller than the learned threshold α.

5. Experiments

5.1. Case Study I: Intel Berkeley Research Lab Dataset

5.1.1. Dataset Description

This dataset is collected from 54 sensors deployed in the Intel Berkeley Research lab between Feb. 28th and Apr. 5th, 2004 (int). It contains timestamped topology information, humidity, temperature, light and voltage values in every 31 seconds. Figure 2(a) shows the physical location for each sensor at the lab. Here, we focus on temperature recordings between March 1st and March 20th for our case study.



(a) Physical map of physical location for each sensor at the lab

(b) Heat map of temperature recordings on each sensor

Figure 2: Physical map and heat map of the sensory data

5.1.2. Data Pre-processing

There exist some missing values at certain timestamps for different sensors. First, we use the linear interpretation method to fill the missing entries. Afterwards, we downsample it every 20 minutes and use the average as inputs. Figure 2(b) depicts a heat map of temperature recordings on each sensor after downsampling. In order to avoid outlier values influencing the system performance, we drop the data of sensor 15 and 18. Meanwhile, we normalize the input data. After preprocessing, the sequence length is 1440 timestamps. For each timestamp, the system status is composed of 52 sensory values.

5.1.3. Quantitative Comparison with State-of-the-art

Considering that there is no ground truth for this dataset, we need to manually label the data for performance evaluation purpose. Particularly, we consider that if the entry value does not lie between the mean minus three times of standard deviation and the mean plus three times of standard deviation for any sensor, it is labeled as anomalous data. The dataset is split into training data and testing data. In the training phase, we use the first 70% training data without labels to train a latent representation in the form of Gaussian Mixture. In the testing phase, we use the rest 30% data to fit the learned Gaussian Mixture model for unsupervised clustering. The time window is set to 3. We compare our proposed GGM-VAE with EM-GMM (Dempster et al. (1977)), GRU (Chung et al. (2014)), GRU-AE (Malhotra et al. (2016)), and GRU-VAE (An and Cho (2015)). Table 1 presents the performance comparison regarding accuracy, precision, recall, F1, and AUC (area under

ROC curve). We find that the proposed method GGM-VAE outperforms the state-of-the-art methods. In particular, compared with the best existing method, our scheme achieves improvement of 5.7% and 7.2% in accuracy and F1 score, respectively.

Methods	Performance Evaluation				
	Accuracy	Precision	Recall	F1	AUC
EM-GMM	0.5659	0.5627	0.5910	0.5765	0.5659
GRU	0.7835	0.8481	0.6907	0.7614	0.7835
GRU-AE	0.8190	0.9982	0.6392	0.7794	0.8190
GRU-VAE	0.8883	**0.9986**	0.7778	0.8744	0.8883
GGM-VAE	**0.9387**	0.9592	**0.9164**	**0.9373**	**0.9387**

Table 1: Performance evaluation on different models for Case Study I.

5.2. Case Study II: Yahoo anomaly detection dataset

5.2.1. Dataset Description

Yahoo dataset consists of real and synthetic time-series with tagged anomaly points. In particular, the dataset includes the real traffic to some of the Yahoo's properties. It consists of time-series with varying trend, noise and seasonality, representing the metrics of various Yahoo services (yah). Besides, the dataset is collected from 100 sensors in a traffic network and hourly marked by UNIX timestamp, which contains 1,680 hourly data samples from 2014-11-23 to 2015-02-01. There are various anomaly types including outliers and change-points. Here, we only test the outlier anomaly.

5.2.2. Data Pre-processing

We collect data by extracting the values and the corresponding labels from each sensor's records and concatenate them into a feature table and a label table. In the feature table, each column represents a certain sensor's time series while each row represents all the sensors' records at a certain timestamp. Each row has a corresponding label in the label table. We normalize the original data in each column by using the centering and standard deviation techniques.

5.2.3. Quantitative Comparison with State-of-the-art

With the ground truth label for this dataset, we select all the anomalous time sequences and the same amount of normal time sequences as the testing data, while use rest normal time sequence as the training data. The time window is set to 3. We compare the proposed method GGM-VAE with the same existing methods as in Case Study I. Table 2 demonstrates the performance comparison in accuracy, precision, recall, F1, and AUC (area under ROC curve). We find that the proposed GGU-VAE reaches the best performance.

5.2.4. Sensitivity analysis of hyperparameters

Two factors in sensitivity analysis deserve our significant attention: the number of components in the Gaussian Mixture and the dimension of latent vectors. For the first factor, we

Methods	Performance Evaluation				
	Accuracy	Precision	Recall	F1	AUC
EM-GMM	0.4603	0.0036	0.3333	0.0072	0.3972
GRU	0.5976	0.0293	0.6667	0.0561	0.6315
GRU-AE	0.6752	0.6062	**1.0000**	0.7548	0.6752
GRU-VAE	0.8077	0.7432	0.9402	0.8302	0.8077
GGU-VAE	**0.8396**	**0.8125**	0.8845	**0.8470**	**0.8396**

Table 2: Performance evaluation on different models for Case Study II.

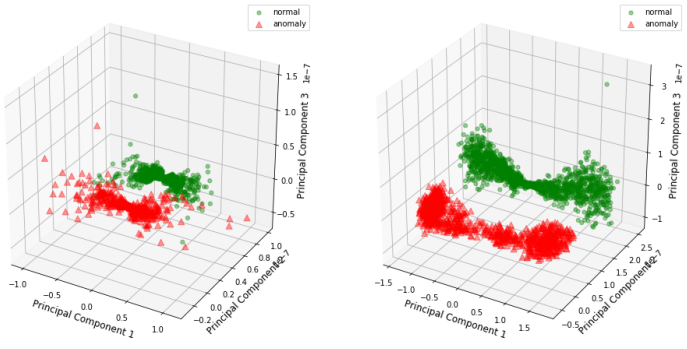
use the number of nature clusters as the metric to determine the number of components in Gaussian Mixture. Particularly, in anomaly detection, samples are either from normal and anomalous, thus we set the number of component to two. When it comes to the second factor, Table 3 shows the performance of the proposed GGU-VAE with different dimension of latent vectors. We can find that the F1 score remains stable with different dimension of latent vectors, and achieves the peak when the latent variables' dimension is 8.

Dimensionality	Precision	Recall	F1
2	0.7650	**0.9287**	0.8389
4	0.7903	0.9011	0.8421
8	**0.8125**	0.8845	**0.8470**
16	0.7916	0.9073	0.8421
32	0.7765	0.9129	0.8392

Table 3: Sensitivity analysis with different dimension of latent vectors.

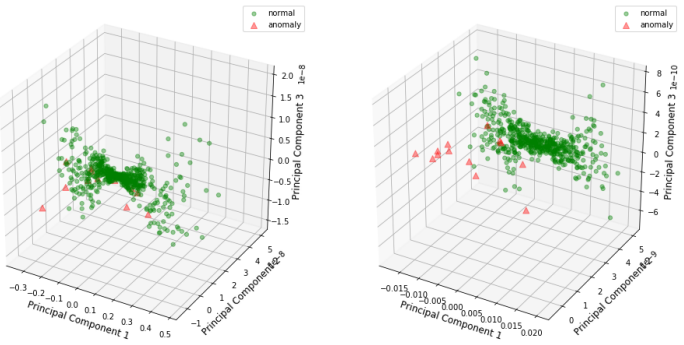
5.3. Visualization of the learned latent representation

In order to further verify the effectiveness of our model, we would demonstrate the advantage of leveraging Gaussian Mixture prior over a single Gaussian prior on separating the anomalies from normal data through the learned latent low-dimensional representation. Figure 3 shows the learned low-dimensional representation on two datasets with unit Gaussian prior and Gaussian Mixture prior, respectively. Particularly, we use the Principal Component Analysis (PCA) to reduce the original high dimension of latent vectors to three dimension by taking the top three components' eigenvectors and visualize the latent space in Figure 3. For the Intel dataset, by comparing Figure 3(a) and Figure 3(b), we can easily find that the model with Gaussian Mixture prior can better separate anomalous samples from normal samples in the 3D latent space. For the Yahoo dataset, it is obvious that a large portion of anomalous samples cannot be clearly separated from normal samples with unit Gaussian prior in Figure 3(c). In contrast, by leveraging Gaussian Mixture prior, the majority of anomalous samples can be separated from the normal samples as shown in Figure 3(d).



(a) Unit Gaussian prior on Intel dataset

(b) Gaussian Mixture prior on Intel dataset



(c) Unit Gaussian prior on Yahoo dataset

(d) Gaussian Mixture prior on Yahoo dataset

Figure 3: Visualization of the learned latent representation

6. Conclusions

In this paper, we have proposed an unsupervised GRU-based Gaussian Mixture VAE called GGU-VAE for anomaly detection on multidimensional time-series data. Specifically, we have trained a GRU based deep latent embedding to capture the correlations among time sequences. Instead of assuming single Gaussian distribution as prior in the data generative procedure, we employ the Gaussian Mixture model to better describe the latent space with a series of Gaussian distributions. Experiment results show that the proposed scheme GGM-VAE achieves obvious improvements compared with existing methods.

Acknowledgments

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

References

URL http://db.csail.mit.edu/labdata/labdata.html.

URL https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70.

Naoki Abe, Bianca Zadrozny, and John Langford. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 504–509. ACM, 2006.

Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2:1–18, 2015.

Rana Aamir Raza Ashfaq, Xi-Zhao Wang, Joshua Zhexue Huang, Haider Abbas, and Yu-Lin He. Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, 378:484–497, 2017.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Outlier detection: A survey. *ACM Computing Surveys*, 2007.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

Xuhui Chen, Jinlong Ji, Kenneth Loparo, and Pan Li. Real-time personalized cardiac arrhythmia detection and diagnosis: A cloud computing architecture. In *Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on*, pages 201–204. IEEE, 2017.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Shekhar R Gaddam, Vir V Phoha, and Kiran 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*, 19(3):345–354, 2007.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in NIPS*, pages 2946–2954, 2016.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yufeng Kou, Chang-Tien Lu, Sirirat Sirwongwattana, and Yo-Ping Huang. Survey of fraud detection techniques. In *Networking, sensing and control, 2004 IEEE international conference on*, volume 2, pages 749–754. IEEE, 2004.

Yuancheng Li, Rong Ma, and Runhai Jiao. A hybrid malicious code detection method based on deep learning. *methods*, 9(5), 2015.

Weixian Liao, Sergio Salinas, Ming Li, Pan Li, and Kenneth A Loparo. Cascading failure attacks in the power system: a stochastic game perspective. *IEEE Internet of Things Journal*, 4(6):2247–2259, 2017.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

Eric Nalisnick, Lars Hertel, and Padhraic Smyth. Approximate inference for deep latent gaussian mixtures. In *NIPS Workshop on Bayesian Deep Learning*, volume 2, 2016.

Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

Leonid Portnoy, Eleazar Eskin, and Sal Stolfo. Intrusion detection with unlabeled data using clustering. In *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)*, pages 5–8, 2001.

Rui Shu, James Brofos, Frank Zhang, Hung Hai Bui, Mohammad Ghavamzadeh, and Mykel Kochenderfer. Stochastic video prediction with conditional density estimation. In *ECCV Workshop on Action and Anticipation for Visual Learning*, volume 2, 2016.

Liang Xiong, Barnabás Póczos, and Jeff G Schneider. Group anomaly detection using flexible genre models. In *Advances in NIPS*, pages 1071–1079, 2011.

Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, 2004.

Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674. ACM, 2017.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. 2018.