Correlation-aware Deep Generative Model for Unsupervised Anomaly Detection

Haoyi Fan[1], Fengbin Zhang[1], Ruidong Wang[1], Liang Xi[1], and Zuoyong Li[2]

[1] School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China.
{isfanhy, zhangfengbin}@hrbust.edu.cn,1820400010@stu.hrbust.edu.cn,
xiliang@hrbust.edu.cn
[2] Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350121, China.
fzulzytdq@126.com
Corresponding authors: Fengbin Zhang, Zuoyong Li.

Abstract. Unsupervised anomaly detection aims to identify anomalous samples from highly complex and unstructured data, which is pervasive in both fundamental research and industrial applications. However, most existing methods neglect the complex correlation among data samples, which is important for capturing normal patterns from which the abnormal ones deviate. In this paper, we propose a method of **C**orrelation aware unsupervised **A**nomaly detection via **D**eep **G**aussian **M**ixture **M**odel (**CADGMM**), which captures the complex correlation among data points for high-quality low-dimensional representation learning. More specifically, the relations among data samples are correlated firstly in forms of a graph structure, in which, the node denotes the sample and the edge denotes the correlation between two samples from the feature space. Then, a dual-encoder that consists of a graph encoder and a feature encoder, is employed to encode both the feature and correlation information of samples into the low-dimensional latent space jointly, followed by a decoder for data reconstruction. Finally, a separate estimation network as a Gaussian Mixture Model is utilized to estimate the density of the learned latent vector, and the anomalies can be detected by measuring the energy of the samples. Extensive experiments on real-world datasets demonstrate the effectiveness of the proposed method.

Keywords: Anomaly Detection · Graph Attention · Gaussian Mixture Model · Data Correlation.

1 Introduction

Anomaly detection aims at identifying abnormal patterns that deviate significantly from the normal behavior, which is ubiquitous in a multitude of application domains, such as cyber-security [15], medical care [19], and surveillance video profiling [14]. Formally, anomaly detection problem can be viewed as density estimation from the data distribution [23]: anomalies tend to reside in the

low probability density areas. Although anomaly detection has been well-studied in the machine learning community, how to conduct unsupervised anomaly detection from highly complex and unstructured data effectively, is still a challenge.

Unsupervised anomaly detection aims to detect outliers without labeled data for the scenario that only a small number of labeled anomalous data combined with plenty of unlabeled data are available, which is common in real-world applications. Existing methods for unsupervised anomaly detection can be divided into three categories: reconstruction based methods, clustering based methods, and one-class classification based methods. Reconstruction based methods, such as PCA [5] based approaches [18, 10] and autoencoder based approaches [21–23, 20], assume that outliers cannot be effectively reconstructed from the compressed low-dimensional projections. Clustering based methods [17, 6] aim at density estimation of data points and usually adopt a two-step strategy [3] that performs dimensionality reduction firstly and then clustering. Different from previously mentioned categories, one-class classification based methods [7, 11, 1] make the effort to learn a discriminative boundary between the normal and abnormal instances.
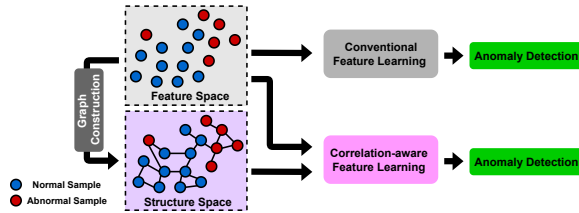

Fig. 1. Correlation-aware feature learning for anomaly detection.

Although the above-mentioned methods had their fair share of success in anomaly detection, most of these methods neglect the complex correlation among data samples. As shown in Fig. 1, the conventional methods attempt to conduct feature learning on the original observed feature space of data samples, while the correlation among similar samples is ignored, which can be exploited during feature learning by propagating more representative features from the neighbors to generate high-quality embedding for anomaly detection. However, modeling correlation among samples is far different from those conventional feature learning models, in which highly non-linear structure needs to be captured. Therefore, how to effectively incorporate both the original feature and relation structure of samples into an integrated feature learning framework for anomaly detection is still an open problem.

To alleviate the above-mentioned problems, in this paper, we propose a method of **C**orrelation aware unsupervised **A**nomaly detection via **D**eep **G**aussian **M**ixture **M**odel (**CADGMM**), which considers both the original feature and the complex correlation among data samples for feature learning. Specifically, the relations among data samples are correlated firstly in forms of a graph structure, in which, the node denotes the sample and the edge denotes the correlation between two samples from the feature space. Then, a dual-encoder that consists

of a graph encoder and a feature encoder, is employed in CADGMM to encode both the feature and correlation of samples into the low-dimensional latent space jointly, followed by a decoder for data reconstruction. Finally, a separate estimation network as a Gaussian Mixture Model is utilized to estimate the density of the learned latent embedding. To verify the effectiveness of our algorithms, we conduct experiments on multiple real-world datasets. Our experimental results demonstrate that, by considering correlation among data samples, CADGMM significantly outperforms the state-of-the-art on unsupervised anomaly detection tasks.

2   Notations and Problem Statement

In this section, we formally define the frequently-used notations and the studied problem.

Definition 1. *$\textbf{Graph}$ is denoted as $\boldsymbol{\mathcal{G}} = \{\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}}, \mathbf{X}\}$ with N nodes and E edges, in which, $\mathcal{V} = \{v_i | i = 1, 2, ..., N\}$ is a set of nodes, $\boldsymbol{\mathcal{E}} = \{e_i | i = 1, 2, ..., E\}$ is a set of edges and $e_i = (v_{i_1}, v_{i_2})$ represents an edge between node v_{i_1} and node v_{i_2}. $\mathbf{X} \in \mathbb{R}^{N \times F}$ is an feature matrix with each row corresponding to a content feature of a node, where F indicates the dimension of features. $\textbf{Adjacency Matrix}$ of a graph is denoted as $\mathbf{A} \in \mathbb{R}^{N \times N}$, which can be used to represent the topologies of a graph. The scalar element $\mathbf{A}_{i,j} = 1$ if there exists an edge between node v_i and node v_j, otherwise, $\mathbf{A}_{i,j} = 0$.*

Problem 1. **Anomaly detection**: Given a set of input samples $\boldsymbol{\mathcal{X}} = \{x_i | i = 1, ..., N\}$, each of which is associated with a F dimension feature $\mathbf{X}_i \in \mathbb{R}^F$, we aim to learn a score function $u(\mathbf{X}_i) : \mathbb{R}^F \mapsto \mathbb{R}$, to classify sample x_i based on the threshold λ:

$$y_i = \begin{cases} 1, if \ u(\mathbf{X}_i) \geq \lambda, \\ 0, \quad otherwise. \end{cases} \tag{1}$$

where y_i denotes the label of sample x_i, with 0 being the normal class and 1 the anomalous class.

3   Method

In this section, we introduce the proposed CADGMM in detail. CADGMM is an end-to-end joint representation learning framework for unsupervised anomaly detection. As shown in Fig. 2, CADGMM consists of three modules named dual-encoder, feature decoder, and estimation network, respectively. Specifically, the relations among data samples in the original feature space are correlated firstly in form of the graph structure. In the constructed graph, the node denotes the sample and the edge denotes the correlation between two samples in the feature space. Then, a dual-encoder that consists of a graph encoder and a feature
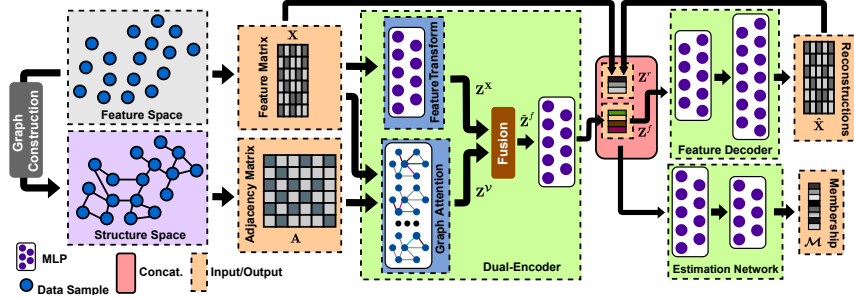
Fig. 2. The framework of the proposed method.

encoder, is employed to encode both the feature and correlation information of samples into the low-dimensional latent space jointly, followed by a feature decoder for sample reconstruction. Finally, a separate estimation network is utilized to estimate the density of the learned latent embedding in the framework of Gaussian Mixture Model, and the anomalies can be detected by measuring the energy of the samples with respect to a given threshold.

3.1   Graph Construction

To explore the correlation among non-structure data samples for feature learning, we explicitly construct a graph structure to correlate the similar samples from the feature space. More specifically, given a set of input samples $\boldsymbol{\mathcal{X}} = \{x_i | i = 1, ..., N\}$, we employ K-**NN** algorithm on sample x_i to determine its K nearest neighbors $\boldsymbol{\mathcal{N}}_i = \{x_{i_k} | k = 1, ..., K\}$ in the feature space. Then, an undirected edge is assigned between x_i and its neighbor x_{i_k}. Finally, an undirected graph $\boldsymbol{\mathcal{G}} = \{\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}}, \mathbf{X}\}$ is constructed, with $\boldsymbol{\mathcal{V}} = \{v_i = x_i | i = 1, ..., N\}$ being the node set, $\boldsymbol{\mathcal{E}} = \{e_{i_k} = (v_i, v_{i_k}) | v_{i_k} \in \boldsymbol{\mathcal{N}}_i\}$ being the edge set, and $\mathbf{X} \in \mathbb{R}^{N \times F}$ being the feature matrix of nodes. Based on the constructed graph, the feature affinities among samples are captured explicitly, which can be used during feature learning by performing message propagation mechanism on them.

3.2   Dual-Encoder

In order to obtain sufficient representative high-level sample embedding, Dual-Encoder consists of a feature encoder and a graph encoder to encode the original feature of samples and the correlation among them respectively.

To encode the original sample features $\mathbf{X}$, feature encoder employs a L_{X} layers Multi-Layer Perceptron (MLP) to conduct a non-linear feature transform, which is as follows:

$$\mathbf{Z}^{\mathbf{X}(l_{\mathrm{X}})} = \sigma(\mathbf{Z}^{\mathbf{X}(l_{\mathrm{X}}-1)}\mathbf{W}^{\mathbf{X}(l_{\mathrm{X}}-1)} + \mathbf{b}^{\mathbf{X}(l_{\mathrm{X}}-1)}) \tag{2}$$

where $\mathbf{Z}^{\mathbf{X}(l_{\mathrm{X}}-1)}$, $\mathbf{Z}^{\mathbf{X}(l_{\mathrm{X}})}$, $\mathbf{W}^{\mathbf{X}(l_{\mathrm{X}}-1)}$ and $\mathbf{b}^{\mathbf{X}(l_{\mathrm{X}}-1)}$ are the input, output, the trainable weight and bias matrix of $(l_{\mathrm{X}}$-1)-th layer respectively, $l_{\mathrm{X}} \in \{1, 2, ..., L_{\mathrm{X}}\}$,

and $\mathbf{Z}^{\mathbf{X}(0)} = \mathbf{X}$ is the initial input of the encoder. $\sigma(\bullet)$ denotes an activation function such as ReLU or Tanh. Finally, the final feature embedding $\mathbf{Z}^{\mathbf{X}} = \mathbf{Z}^{\mathbf{X}(L_{\mathrm{X}})}$ is obtained from the output of the last layer in MLP.

To encode the correlation among the samples, a graph attention layer [16] is employed to adaptively aggregate the representation from neighbor nodes, by performing a shared attentional mechanism on the nodes:

$$w_{i,j} = attn(\mathbf{X}_i, \mathbf{X}_j) = \sigma(\mathbf{a}^{\mathrm{T}} \cdot [\mathbf{W}^c \mathbf{X}_i || \mathbf{W}^c \mathbf{X}_j]) \tag{3}$$

where $w_{i,j}$ indicates the importance weight of node $\boldsymbol{v}_i$ to node $\boldsymbol{v}_j$, $attn(\bullet)$ denotes the neural network parametrized by weights $\mathbf{a} \in \mathbb{R}^{D^c}$ and $\mathbf{W}^c \in \mathbb{R}^{\frac{D^c}{2} \times F}$ that shared by all nodes and D^c is the number of hidden neurons in $attn(\bullet)$, $||$ denotes the concatenate operation. Then, the final importance weight $\alpha_{i,j}$ is normalized through the softmax function:

$$\alpha_{i,j} = \frac{\exp(w_{i,j})}{\sum_{k \in \mathcal{N}_i} \exp(w_{i,k})} \tag{4}$$

where $\mathcal{N}_i$ denotes the neighbors of node $\boldsymbol{v}_i$, which is provided by adjacency matrix $\mathbf{A}$, and the final node embedding $\mathbf{Z}^{\mathcal{V}} = \{\mathbf{Z}_i^{\mathcal{V}}\}$ can be obtained by the weighted sum based on the learned importance weights as follows:

$$\mathbf{Z}_i^{\mathcal{V}} = \sum_{k \in \mathcal{N}_i} \alpha_{i,k} \cdot \mathbf{X}_k \tag{5}$$

Given the learned embedding $\mathbf{Z}^{\mathbf{X}}$ and $\mathbf{Z}^{\mathcal{V}}$, a fusion module is designed to fuse the embeddings from heterogeneous data source into a shared latent space, followed by a fully connected layer to obtain the final sample embedding $\mathbf{Z}^f \in \mathbb{R}^{N \times D}$:

$$\tilde{\mathbf{Z}}^f = \mathrm{Fusion}(\mathbf{Z}^{\mathbf{X}}, \mathbf{Z}^{\mathcal{V}}) = \mathbf{Z}^{\mathbf{X}} \oplus \mathbf{Z}^{\mathcal{V}} \tag{6}$$

$$\mathbf{Z}^f = \tilde{\mathbf{Z}}^f \mathbf{W} + \mathbf{b} \tag{7}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the trainable weight and bias matrix, and $\oplus$ indicates the element-wise plus operator of two matrices.

3.3   Feature Decoder

Feature decoder aims at reconstructing the sample features from the latent embedding $\mathbf{Z}^f$:

$$\mathbf{Z}^{\hat{\mathbf{X}}(l_{\hat{\mathrm{X}}})} = \sigma(\mathbf{Z}^{\hat{\mathbf{X}}(l_{\hat{\mathrm{X}}}-1)} \mathbf{W}^{\hat{\mathbf{X}}(l_{\hat{\mathrm{X}}}-1)} + \mathbf{b}^{\hat{\mathbf{X}}(l_{\hat{\mathrm{X}}}-1)}) \tag{8}$$

where $\mathbf{Z}^{\hat{\mathbf{X}}(l_{\hat{\mathrm{X}}}-1)}$, $\mathbf{Z}^{\hat{\mathbf{X}}(l_{\hat{\mathrm{X}}})}$, $\mathbf{W}^{\hat{\mathbf{X}}(l_{\hat{\mathrm{X}}}-1)}$ and $\mathbf{b}^{\hat{\mathbf{X}}(l_{\hat{\mathrm{X}}}-1)}$ are the input, output, the trainable weight and bias matrix of $(l_{\hat{\mathrm{X}}}-1)$-th layer of decoder respectively, $l_{\hat{\mathrm{X}}} \in \{1, 2, ..., L_{\hat{\mathrm{X}}}\}$, and $\mathbf{Z}^{\hat{\mathbf{X}}(0)} = \mathbf{Z}^f$ is the initial input of the decoder. Finally, the reconstruction $\hat{\mathbf{X}}$ is obtained from the last layer of decoder:

$$\hat{\mathbf{X}} = \mathbf{Z}^{\hat{\mathbf{X}}(L_{\hat{\mathrm{X}}})} \tag{9}$$

3.4   Estimate Network

To estimate the density of the input samples, a Gaussian Mixture Model is leveraged in CADGMM over the learned latent embedding. Inspired by DAGMM [23], a sub-network consists of several fully connected layers is utilized, which takes the reconstruction error preserved low-dimensional embedding as input, to estimate the mixture membership for each sample. The reconstruction error preserved low-dimentional embedding $\mathbf{Z}$ is obtained as follows:

$$\mathbf{Z} = [\mathbf{Z}^f || \mathbf{Z}^r], \ \mathbf{Z}^r = \mathrm{Dist}(\mathbf{X}, \hat{\mathbf{X}}) \tag{10}$$

where $\mathbf{Z}^r$ is the reconstruction error embedding and $\mathrm{Dist}(\bullet)$ denotes the distance metric such as Euclidean distance or cosine distance. Given the final embedding $\mathbf{Z}$ as input, estimate network conducts membership prediction as follows:

$$\mathbf{Z}^{\boldsymbol{\mathcal{M}}(l_{\mathcal{M}})} = \sigma(\mathbf{Z}^{\boldsymbol{\mathcal{M}}(l_{\mathcal{M}}-1)}\mathbf{W}^{\boldsymbol{\mathcal{M}}(l_{\mathcal{M}}-1)} + \mathbf{b}^{\boldsymbol{\mathcal{M}}(l_{\mathcal{M}}-1)}) \tag{11}$$

where $\mathbf{Z}^{\boldsymbol{\mathcal{M}}(l_{\mathcal{M}}-1)}$, $\mathbf{Z}^{\boldsymbol{\mathcal{M}}(l_{\mathcal{M}})}$, $\mathbf{W}^{\boldsymbol{\mathcal{M}}(l_{\mathcal{M}}-1)}$ and $\mathbf{b}^{\boldsymbol{\mathcal{M}}(l_{\mathcal{M}}-1)}$ are the input, output, the trainable weight and bias matrix of $(l_{\mathcal{M}}-1)$-th layer of estimate network respectively, $l_{\mathcal{M}} \in \{1, 2, ..., L_{\mathcal{M}}\}$, $\mathbf{Z}^{\boldsymbol{\mathcal{M}}(0)} = \mathbf{Z}$, and the mixture-component membership $\boldsymbol{\mathcal{M}}$ is calculated by:

$$\boldsymbol{\mathcal{M}} = \mathrm{Softmax}(\mathbf{Z}^{\boldsymbol{\mathcal{M}}(L_{\mathcal{M}})}) \tag{12}$$

where $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{N \times M}$ is the predicted membership of M mixture components for N samples. With the predicted sample membership, the parameters of GMM can be calculated to facilitate the evaluation of the energy/likelihood of input samples, which is as follows:

$$\boldsymbol{\mu_m} = \frac{\sum_{i=1}^{N} \boldsymbol{\mathcal{M}}_{i,m}\mathbf{Z}_i}{\sum_{i=1}^{N} \boldsymbol{\mathcal{M}}_{i,m}}, \ \boldsymbol{\Sigma}_m = \frac{\sum_{i=1}^{N} \boldsymbol{\mathcal{M}}_{i,m}(\mathbf{Z}_i - \boldsymbol{\mu_m})(\mathbf{Z}_i - \boldsymbol{\mu_m})^{\mathrm{T}}}{\sum_{i=1}^{N} \boldsymbol{\mathcal{M}}_{i,m}} \tag{13}$$

where $\boldsymbol{\mu_m}$ and $\boldsymbol{\Sigma}_m$ are the means and covariance of the m-th component distribution respectively, and the energy of samples is as follows:

$$\mathrm{E}_{\mathbf{Z}} = -\log\left(\sum_{m=1}^{M}\sum_{i=1}^{N} \frac{\boldsymbol{\mathcal{M}}_{i,m}}{N} \frac{\exp(-\frac{1}{2}(\mathbf{Z} - \boldsymbol{\mu_m})^{\mathrm{T}}\boldsymbol{\Sigma}_m^{-1}(\mathbf{Z} - \boldsymbol{\mu_m}))}{|2\pi\boldsymbol{\Sigma}_m|_D^{\frac{1}{2}}}\right) \tag{14}$$

3.5   Loss Function and Anomaly Score

The training objective of CADGMM is defined as follows:

$$\mathcal{L} = ||\mathbf{X} - \hat{\mathbf{X}}||_2^2 + \lambda_1 E_{\mathbf{Z}} + \lambda_2 \sum_{m=1}^{M}\sum_{i=1}^{N} \frac{1}{(\boldsymbol{\Sigma}_m)_{ii}} + \lambda_3 ||\mathbf{Z}||_2^2 \tag{15}$$

where the first term is reconstruction error used for feature reconstruction, the second is sample energy, which aims to maximize the likelihood to observed

Table 1. Statistics of the public benchmark datasets.

Database	# Dimensions	# Instances	Anomaly ratio
KDD99	120	494,021	0.2
Arrhythmia	274	452	0.15
Satellite	36	6,435	0.32

samples, the third is covariance penalization, used for solving singularity problem as in GMM [23] by penalizing small values on the diagonal entries of covariance matrix, and the last is embedding penalization, which serves as a regularizer to impose the magnitude of normal samples as small as possible in the latent space, to deviate the normal samples from the abnormal ones. λ_1, λ_2, and λ_3 are three parameters which control the trade off between different terms.

The anomaly score is the sample energy $E_{\mathbf{Z}}$, and based on the measured anomaly scores, the threshold λ in Eq. 1 can be determined according to the distribution of scores, e.g. the samples of top-k scores are classified as anomalous samples.

4 Experiments

In this section, we will describe the experimental details including datasets, baseline methods, and parameter settings, respectively.

4.1 Dataset

Three benchmark datasets are used in this paper to evaluate the proposed method, including KDD99, Arrhythmia, and Satellite. The statistics of datasets are shown in Table 1.

- **KDD99** The KDD99 10 percent dataset [2] contains 494021 samples with 41 dimensional features, where 34 of them are continuous and 7 are categorical. One-hot representation is used to encode the categorical features, resulting in a 120-dimensional feature for each sample.
- **Arrhythmia** The Arrhythmia dataset [2] contains 452 samples with 274 dimensional features. We combine the smallest classes including 3, 4, 5, 7, 8, 9, 14, 15 to form the outlier class and the rest of the classes are inliers class.
- **Satellite** The Satellite dataset [2] has 6435 samples with 36 dimensional features. The smallest three classes including 2,4,5 are combined to form the outliers and the rest are inliers classes.

4.2 Baseline Methods

- **One Class Support Vector Machines (OC-SVM)** [4] is a classic kernel method for anomaly detection, which learns a decision boundary between the inliers and outliers.

– **Isolation Forests (IF)** [8] conducts anomaly detection by building trees using randomly selected split values across sample features, and defining the anomaly score as the average path length from a specific sample to the root.
– **Deep Structured Energy Based Models (DSEBM)** [21] is a deep energy-based model, which aims to accumulate the energy across the layers. DSEBM-r and DSEBM-e are utilized in [21] by taking the energy and reconstruction error as the anomaly score respectively.
– **Deep Autoencoding Gaussian Mixture Model (DAGMM)** [23] is an autoencoder based method for anomaly detection, which consists of a compression network for dimension reduction, and an estimate network to perform density estimation under the Gaussian Mixture Model.
– **AnoGAN** [13] is an anomaly detection algorithm based on GAN, which trains a DCGAN [12] to recover the representation of each data sample in the latent space during prediction.
– **ALAD** [20] is based on bi-directional GANs for anomaly detection by deriving adversarially learned features and uses reconstruction errors based on the learned features to determine if a data sample is anomalous.

4.3   Parameter Settings

The parameter settings in the experiment for different datasets are as follows:

– **KDD99** For KDD99, CADGMM is trained with 300 iterations and N=1024 for graph construction with K=15, which is the batch size for training. M=4, λ_1=0.1, λ_2=0.005, λ_3=10.
– **Arrhythmia** For Arrhythmia, CADGMM is trained with 20000 iterations and N=128 for graph construction with K=5, which is the batch size for training, M=2, λ_1=0.1, λ_2=0.005, λ_3=0.001.
– **Satellite** For Satellite, CADGMM is trained with 3000 iterations and N=512 for graph construction with K=13, M=4, λ_1=0.1, λ_2=0.005, λ_3=0.005.

The architecture details of CADGMM on different datasets are shown in Table 2, in which, $FC(D_{in}, D_{out})$ means a fully connected layer with D_{in} input neurons and D_{out} output neurons. Similarly, $GAT(D_{in}, D_{out})$ means a graph attention layer with D_{in}-dimensional input and D_{out}-dimensional output. The activation function $\sigma(\bullet)$ for all datasets is set as Tanh. For the baseline methods, we set the parameters by grid search. We independently run each experiment 10 times and the mean values are reported as the final results.

5   Results and Analysis

In this section, we will demonstrate the effectiveness of the proposed method by presenting results of our model on anomaly detection task, and provide a comparison with the state-of-the-art methods.

Table 2. Architecture details of CADGMM for different datasets.

Dataset	Dual-Enc.			Feature Dec.	Estimate Net.
	Feature Trans.	Graph Attn.	MLP		
KDD99	FC(120,64) FC(64,32)	GAT(120,32)	FC(32, 8)	FC(8,32) FC(32,64) FC(64,120)	FC(10,20) FC(20,8) FC(8,4)
Arrhythmia	FC(274,32)	GAT(274,32)	FC(32, 2)	FC(2,10) FC(10,274)	FC(4,10) FC(10,2)
Satellite	FC(36,16)	GAT(36,16)	FC(16, 2)	FC(2,16) FC(16,36)	FC(4,10) FC(10,4)

Table 3. Anomaly Detection Performance on KDD99, Arrhythmia, and Satellite datasets. Better results are marked in **bold**.

Method	KDD99			Arrhythmia			Satellite		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
OC-SVM [4]	74.57	85.23	79.54	53.97	40.82	45.81	52.42	59.99	61.07
IF [8]	92.16	93.73	92.94	51.47	54.69	53.03	60.81	**94.89**	75.40
DSEBM-r [21]	85.21	64.72	73.28	15.15	15.13	15.10	67.84	68.61	68.22
DSEBM-e [21]	86.19	64.66	73.99	46.67	45.65	46.01	67.79	68.56	68.18
DAGMM [23]	92.97	94.42	93.69	49.09	50.78	49.83	80.77	81.6	81.19
AnoGAN [13]	87.86	82.97	88.65	41.18	43.75	42.42	71.19	72.03	71.59
ALAD [20]	94.27	95.77	95.01	50	53.13	51.52	79.41	80.32	79.85
CADGMM	**96.01**	**97.53**	**96.71**	**56.41**	**57.89**	**57.14**	**81.99**	82.75	**82.37**

5.1 Anomaly Detection

As in previous literatures [21, 23, 20], in this paper, **Precision**, **Recall** and **F1** score are employed as the evaluation metrics. Generally, we expect the values of these evaluation metrics as big as possible. The sample with high energy is classified as abnormal and the threshold is determined based on the ratio of anomalies in the dataset. Following the settings in [21, 23], the training and test sets are split by 1:1 and only normal samples are used for training the model.

The experimental results shown in Table 3 demonstrate that the proposed CADGMM significantly outperforms all baselines in various datasets. The performance of CADGMM is much higher than traditional anomaly detection methods such as OC-SVM and IF, because of the limited capability of feature learning or the curse of dimensionality. Moreover, CADGMM also significantly outperforms all other deep learning based methods such as DSEBM, DAGMM, AnoGAN, and ALAD, which demonstrates that additional correlation among data samples facilitates the feature learning for anomaly detection. For small datasets such as Arrhythmia, we can find that traditional methods such as IF are competitive compared with conventional deep learning based method such as DSEBM, DAGMM, AnoGAN, and ALAD, which might because that the lack of sufficient training data could have resulted in poorer performance of the data hungry deep learning based methods, while CADGMM is capable of leveraging

more data power given the limited data source, by considering the correlation among data samples.

Table 4. Anomaly Detection Performance on KDD99 with different ratios of anomalies during training.

Radio	CADGMM			DAGMM			OC-SVM		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
1%	95.53	97.04	96.28	92.01	93.37	92.68	71.29	67.85	69.53
2%	95.32	96.82	96.06	91.86	93.40	92.62	66.68	52.07	58.47
3%	94.83	96.33	95.58	91.32	92.72	92.01	63.93	44.70	52.61
4%	94.62	96.12	95.36	88.37	89.89	89.12	59.91	37.19	45.89
5%	94.35	96.04	95.3	85.04	86.43	85.73	11.55	33.69	17.20

5.2   Impact of noise data

In this section, we study the impact of noise data for the training of CADGMM. To be specific, 50% of randomly split data samples are used for testing, while the rest 50% combined with 1% to 5% anomalies are used for training.

As shown in Table 4, with the increase of noise data, the performance of all baselines degrade significantly, especially for OC-SVM, which tends to be more sensitive to noise data because of its poor ability of feature learning on high-dimensional data. However, CADGMM performs stable with different ratios of noise and achieves state-of-the-art even 5% anomalies are injected into the training data, which demonstrates the robustness of the proposed method.
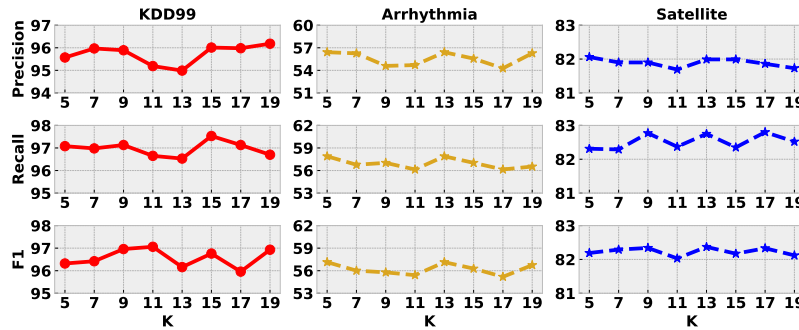

Fig. 3. Impact of different K values of K-NN algorithms in graph construction.

5.3   Impact of K values

In this section, we evaluate the impact of different K values during the graph construction on CADGMM.

More specifically, we conduct experiments on all three datasets by varying the number of K from 5 to 19, and the experimental results are illustrated in

(a) DAGMM   (b) CADGMM
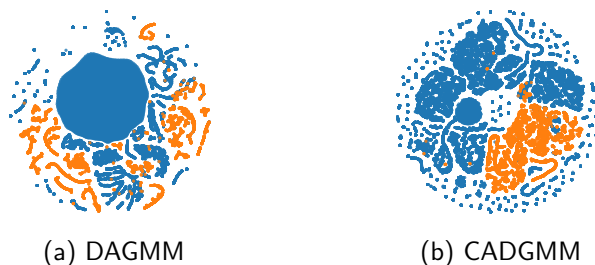
Fig. 4. Embedding visualization (Blue indicates the normal samples and orange the anomalies).

Fig. 3. During training, the batch sizes are set as 1024, 128, and 512 for KDD99, Arrhythmia, and Satellite, respectively, the experimental results show that the changing of K value causes only a little fluctuation of performance on all datasets with different settings, which demonstrates that CADGMM is less sensitive to the K value and easy to use.

5.4 Embedding Visualization

In order to explore the quality of the learned embedding, we make a comparison of the visualization of sample representation for different methods in Fig. 4. Specifically, we take the low-dimensional embeddings of samples learned by DAGMM and CADGMM, as the inputs to the t-SNE tool [9]. Here, we randomly choose 40000 data samples from the test set of KDD99 for visualization, and then we generate visualizations of the sample embedding on a two-dimensional space, in which blue colors correspond to the normal class while orange the abnormal class. We can find that CADGMM achieves more compact and separated clusters compared with DAGMM. The results can also explain why our approach achieves better performance on anomaly detection task.

6 Conclusion

In this paper, we study the problem of correlation aware unsupervised anomaly detection, which considers the correlation among data samples from the feature space. To cope with this problem, we propose a method named CADGMM to model the complex correlation among data points to generate high-quality low-dimensional embeddings for anomaly detection. Extensive experiments on real-world datasets demonstrate the effectiveness of the proposed method.

Acknowledgement

References

1. Amer, M., Goldstein, M., Abdennadher, S.: Enhancing one-class support vector machines for unsupervised anomaly detection. In: SIGKDD. pp. 8–15 (2013)
2. Bache, K., Lichman, M.: Uci machine learning repository, 2013. URL http://archive. ics. uci. edu/ml **5** (2013)
3. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys **41**(3) (7 2009). https://doi.org/10.1145/1541880.1541882
4. Chen, Y., Zhou, X.S., Huang, T.S.: One-class svm for learning in image retrieval. In: ICIP. pp. 34–37 (2001)
5. Jolliffe, I.: Principal component analysis. Technometrics **45**(3), 276 (2003)
6. Kim, J., Scott, C.D.: Robust kernel density estimation. Journal of Machine Learning Research **13**(Sep), 2529–2565 (2012)
7. Li, K.L., Huang, H.K., Tian, S.F., Xu, W.: Improving one-class svm for anomaly detection. In: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics. vol. 5, pp. 3077–3081 (2003)
8. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: ICDM. pp. 413–422 (2008)
9. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
10. Pascoal, C., De Oliveira, M.R., Valadas, R., Filzmoser, P., Salvador, P., Pacheco, A.: Robust feature selection and robust pca for internet traffic anomaly detection. In: IEEE INFOCOM. pp. 1755–1763 (2012)
11. Perdisci, R., Gu, G., Lee, W., et al.: Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In: ICDM. vol. 6, pp. 488–498 (2006)
12. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR (2016)
13. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: IPMI. pp. 146–157 (2017)
14. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: CVPR. pp. 6479–6488 (2018)
15. Tan, S.C., Ting, K.M., Liu, T.F.: Fast anomaly detection for streaming data. In: IJCAI (2011)
16. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2018)
17. Xiong, L., Póczos, B., Schneider, J.G.: Group anomaly detection using flexible genre models. In: NIPS. pp. 1071–1079 (2011)
18. Xu, H., Caramanis, C., Sanghavi, S.: Robust pca via outlier pursuit. In: NIPS. pp. 2496–2504 (2010)
19. Xu, S., Wu, H., Bie, R.: Cxnet-m1: Anomaly detection on chest x-rays with image-based deep learning. IEEE Access **7**, 4466–4477 (2018)
20. Zenati, H., Romain, M., Foo, C.S., Lecouat, B., Chandrasekhar, V.: Adversarially learned anomaly detection. In: ICDM. pp. 727–736 (2018)
21. Zhai, S., Cheng, Y., Lu, W., Zhang, Z.: Deep structured energy based models for anomaly detection. In: ICML. pp. 1100–1109 (2016)
22. Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: SIGKDD. pp. 665–674 (2017)
23. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: ICLR (2018)