Feature extraction via autoencoder-based network for anomaly detection

(異常検出のためのオートエンコーダベースのネットワークを介した特徴抽出)

Qien Yu

Supervisor: Prof. Takio Kurita

Department of Information Engineering
Hiroshima University

This dissertation is submitted for the degree of
Doctor of Philosophy in Information Engineering
Graduate School of Engineering September 2021

After continuous efforts and exploration, we have reached to this together. To my loving parents and brothers, your support is the driving force behind my hard-working.

Declaration

I declare that this thesis is my original work, except where stated. The thesis has been composed by myself and represents my own work. Any data, idea or opinion taken from an external source is appropriately acknowledged in the text. The work presented has not been submitted for any other degree or professional qualification. Technical and editorial guidance were provided by Professor Takio Kurita. I designed all the experiments presented in this thesis with advice from colleagues and I analysed all the experimental data.

Qien Yu
September 2021

Acknowledgements

To **Professor. Takio Kurita** I am very grateful that he accepted me and taught me a lot of knowledge. More importantly, I learned the attitude of scientific research from him. His guidance allowed me to experience the joy of scientific research and the thirst for exploring the unknown.

To **Fellow research, master and PhD students** In our team, we always help each other, we always discuss ideas and experiments together. I must also thank all my Hiroshima University educators, all the teachers who imparted knowledge and encouraged us to think boldly.

To **my family, friends and kindred spirits** – Thank you for your great support over the past few years.

They have given me great support in the past few years. Without your support, the thesis could not be completed.

Abstract

Optimizing the extraction of feature sets for anomaly detection tasks (AD) is still a fundamental and challenging problem in the field of deep leaning. Anomaly detection is an identification of instances, events or observations, which do not conform to an expected pattern or other instances in dataset.

To perform AD tasks, we extract the discriminative features and efficient coding in the latent space. Information from detection structure is encoded using convolutional-based schemes and later is mapped in the low-dimensional space. We propose several architectures to extract features of normal data for AD that encode the information according to multiple learning strategy. Our models are based on autoencoder, which can learn disrciminative information in encoder-decoder pipeline and a tight boundaries is set for normal data.

For the one-dimensional data, we proposed vector-based convolutional autoencoder (V-CAE) for one dimensional anomaly detection. Given the good performance of convolution network on matrix data, it is promising to transform one-dimensional data into a matrix form and learn important relationship features by convolutional network. The core of this model is a linear autoencoder, which is used to construct a low-dimensional manifold of feature vectors for normal data. At the same time, we used vector-based convolutional neural network (V-CNN) to extract the features from vector data before and after the linear autoencoder (fully connected autoencoder with F-norm reconstruction error and linear activation function) that makes the model learn deep features for efficient anomaly detection. The V-CNN is used to extract non-linear feature vector from the input vector by 2-D convolutional neural network. In this study, we used input data as a vector form and only the features extracted from the normal input data are used to train our proposed model.

However, the reconstruction error and abnormal score introduced in the aforementioned studies used to tune the threshold only for latent sampled variables, and as a result, such methods reported poor reconstruction performance in the abnormal data. Unlike in the previous studies on applying CVAE to anomaly detection in which the intention of variational autoencoder (VAE) deviated from learning an acceptable

pattern for anomalous data identification. As for AD classification through the normal data, examining the latent representation is promising and effective. Thus, we introduce maximization of mutual information (MMI) regularization that help in low-dimensional representation of learned features to emerge. The proposed convulutional variational autoencoder (CVAE) is optimized by combining the representations learned across the three different objectives targeted at MMI on both local and global variables with the original training objective function of Kullback-Leibler divergence distributions. This feature leads to the losses of information about the input data distribution and mapping to the prior probability distribution, thereby generating the output with the high false positives. Therefore, the application of VAEs to anomaly detection tasks needs to be facilitated by adding suitable regularization techniques . In the present study, we investigate the possibility to address this issue by regularizing multiple discriminator spaces aiming to estimate how precisely the output matches its input, rather than relying only on the encoder latent space.

Though, the above latent-based methods for AD achieved better results for vector datasets, it is not ideal for matrix datasets. Furthermore, those methods considered detection using only one latent space and did not consider the possibility of a mixture of low-dimensional nonlinear manifolds of multiple latent spaces. Linearly combining different manifolds in latent spaces can generate best latent representation. However, most of the existing AD methods solely based on the reconstruction errors or latent representation using a single low-dimensional manifold are often not ideal for the image objects with complex background. In this study to realize the promise of multi-manifold latent information for AD, we propose a mixture of experts ensemble with two convolutional variational autoencoders (CVAEs) and convolution network (MEx-CVAEC) which explicitly learns manifold relationships of data that make use of multiple encoded detections. In addition, in order to enhance the model detection performance, we re-encode the output of the CVAE by generating a new data manifold for AD. Thereby each expert is developed to comprise an encoder-decoder-encoder pipeline (EDE) based on CVAE. Additionally, we use a tower structure in the mixture-of-expert model to assign a latent score to each latent representation.

Inspired by multi-space detection in autoencoder, orthogonal projection is introduced to capture the null subspace that consists of noisy information for AD, which is explicitly ignored in the existing approaches. The exploration of double subspaces, called normal space (NS) and abnormal space (AS) can improve the

discriminative manifold information. All these insights have a direct application to the low-dimensonal representation of latent space in autoencoder-based methods for the field of anomaly detection. The range subspace and null subspace are two subspaces of the original space decomposed by their direct sum. To comprehensively exploit the manifolds in two subspaces for robust AD, in this study, we propose an autoencoder framework based on an orthogonal projection constraints (OPC) learning method. The primary objective involves the calculation of projected norms in the range and null subspace. By constraining the projection operator to approximate the orthogonal projections, the model can be trained in an end-to-end manner via BP. In the proposed autoencoder framework model, the features are firstly extracted from the raw input and projected into the subspaces by projection operator.

Compared with the state-of-the-art methods, the proposed methods achieve the best performances, which demonstrates the effectiveness and robustness of anomaly detection using the autoencoder-based method. In the future, we will try to redesign the discrimination objective of the generator to further enhance the generator's ability to recognize anomalies.

Table of contents

List of figures **xvii**

List of tables **xxi**

1 Introduction **1**
1.1 Motivation 1
1.2 Structure of thesis 2

2 Detection of One Dimensional Anomalies using a Vector-based Convolutional Autoencoder **5**
2.1 Introduction 5
2.2 Related Works 7
2.3 Proposed Method 9
2.3.1 Overview 9
2.3.2 Vector-based Convolutional Autoencoder 10
2.3.3 Anomaly Scores 11
2.4 Experimental Setup 12
2.4.1 Data Set 12
2.4.2 Parameter Settings and Evaluation 13
2.5 Results 13
2.5.1 Comparison with the State-of-the-art Methods 13
2.5.2 Ablation Study on Proposed Framework 14

3 Extensive framework based on novel convolutional and variational autoencoder based on maximization of mutual information for anomaly detection **21**
3.1 Introduction 21
3.2 Related Work 25
3.2.1 Decision boundary and density estimation models 25
3.2.2 Deep autoencoder and generative models 26

3.3 Background . . . 27
3.3.1 Variational autoencoder . . . 27
3.3.2 Mutual Information . . . 28
3.4 Method . . . 29
3.4.1 Convolutional and VAE-MMI Ensemble Framework for Anomaly Detection . . . 31
3.4.2 MMI based Training Objective . . . 32
3.4.3 Anomaly Score . . . 37
3.5 Experimental Setup . . . 38
3.5.1 Dataset . . . 38
3.5.2 Experimental Evaluation and Performance Measure . . . 38
3.5.3 Parameter Settings . . . 39
3.6 Experimental Results . . . 39
3.6.1 Performance Comparison considering the CAE-based networks 39
3.6.2 Performance Comparison against State-of-the-art methods . . 41
3.6.3 Convergence of the proposed architecture . . . 42
3.6.4 Ablation study . . . 44
3.6.5 Performance Visualization on Latent Space . . . 44

4 Mixture of experts with convolutional and variational autoencoders for anomaly detection 53
4.1 Introduction . . . 53
4.2 Related Works . . . 55
4.2.1 Reconstruction-based methods . . . 55
4.2.2 Latent space detection-based models . . . 55
4.2.3 Mixture-of-experts model for AD . . . 56
4.3 Proposed Mixture of Experts network . . . 56
4.3.1 Constructing an Experts Network Structure . . . 57
4.3.2 Gating network based on Convolutional Autoencoder . . . 59
4.3.3 Training Gated Mixture of Experts structure . . . 60
4.4 Testing Anomaly Score . . . 61
4.5 Experimental Setup . . . 62
4.5.1 Dataset . . . 62
4.5.2 Experimental Evaluation and Performance Measure . . . 62
4.5.3 Parameter Settings . . . 63
4.6 Experimental Results . . . 64

4.6.1 Performance Comparison based on Encoder-Decoder Mixture Models . . . 64
4.6.2 Performance Comparison based on State-of-the-arts . . . 64
4.6.3 Performance Visualization on Latent Space . . . 67

5 Autoencoder framework based on orthogonal projection constraints improves anomalies detection 75
5.1 Introduction . . . 75
5.2 Related work . . . 79
5.2.1 Reconstruction-based methods . . . 79
5.2.2 Latent space detection . . . 80
5.2.3 Orthogonal projection mechanism in deep learning . . . 81
5.3 Methods . . . 81
5.3.1 Constructing an autoencoder network structure based on OPC 81
5.3.2 Structure implementation . . . 84
5.3.3 Proposed anomaly score using orthogonal subspaces score with reconstruction error . . . 86
5.4 Experimental setup . . . 89
5.4.1 Dataset . . . 89
5.4.2 Experimental evaluation and performance measure . . . 89
5.4.3 Parameter settings . . . 91
5.5 Experimental results . . . 91
5.5.1 Performance comparison of OSS with RES . . . 91
5.5.2 Performance comparison against state-of-the-art methods . . 92
5.5.3 Individual class performance analysis . . . 97
5.5.4 Parameter sensitivity analysis . . . 100
5.5.5 Evaluation of rubustness to the additional noises . . . 101
5.5.6 Convergence of the proposed model . . . 102
5.5.7 Ablation study . . . 104
5.5.8 Performance visualization based on normal and abnormal subspace . . . 105

6 Conclusion 107

References 111

Appendix A Caculation of maximazation of mutual information **121**
A.1 121

Appendix B Results **125**
B.1 126

List of figures

2.1 Demonstration of intrusion detection on KDD99 dataset. The two features of network connections are represented as 'dst host count' and' dst host srv count'. The green and red color points indicated the normal and abnormal network connections 6

2.2 Pipeline of the proposed approach for anomaly detection. The parameters of the model shown in this figure are selected according to the feartures of KDD dataset. In fc1,conv1,conv2 the activation functions are leaky relu; In deconv1,deconv2 the activation functions are relu; In fc2 the activation function is tanh.The parameters in this figure are the size of the output data of each layer. The input data is 43-dimensional vector data and first passes through fc1 layer. Then the data dimension becomes 64-dimensional. And the data is converted into matrix with $8 \times 8 \times 1$ format as the input of conv1 after fc1; After output from conv2 layer, the data is changed into vector form, as input of linear encoder. The dimension of output data of linear encoder is 64; The process of data output from decoder to the whole model is similar to the previous process. 7

2.3 Detection of distributions of abnormal scores using our proposed method on KDD99. (a) distribution of S_1 (b) distribution of S_2 and (c) without linear Autoencoder. 15

2.4 Interpreting of correlations between categories of data set and its features 16

2.5 Comparison of ROC curves for abnormal scores on different data sets. (a) KDD, (b) Optdigits and (c) Default of credit card clients. 17

3.1 Performance visualization of the proposed approach over conventional methods for anomaly detection based on the toy sampled data. (a) The distribution of the raw input, (b) latent representation of the proposed method, (c) latent presentation of CVAE, and (d) latent representation of OCGAN. 24
3.2 Proposed convolutional and variational autoencoder framework with maximization of mutual information for image anomaly detection. . 31
3.3 Proposed fully connected and variational autoencoder framework with maximization of mutual information for vector anomaly detection. 32
3.4 Performance comparison of our proposed over the state-of-the-artmethods on image datasets in terms of mean AUC values. 42
3.5 Performance comparison of CVAE-MMI and the state-of-the-arts on overall classes of four data sets in terms of AUC 43
3.6 Performance of CVAE-MMI on each class in four datasets in terms of AUC. 44
3.7 Performance comparison of FVAE-MMI and the state-of-the-arts on overall classes of two data sets in terms of AUC 46
3.8 Convergence curve of the proposed model on the 'bag' class of in IMAGENET dataset. The horizontal axis and the vertical axis represent the number of epochs and loss values respectively. It can be clearly seen that the model tends to a fixed point at the 700^{th} epoch. 49
3.9 Anomaly detection performance of CVAE-MMI increases with the increasing number of iterations interms of mean AUC on image dataset. 49
3.10 Anomaly detection performance of FVAE-MMI. It improves with the increasing number of iterations in terms of AUC based on the vector dataset. 50
3.11 Latent space visualization comparison of the proposed model and the state-of-the-art models(SCG and OCGAN) for anomaly detection on CIFAR100. (a), (b) and (c) shows the performance of the proposed, SCG, and OCGAN, respectively, based on the normal class 'Bicycle', and (d), (e), and (f) the performance of the proposed, SKG and OCGAN, respectively based on the normal class 'Pine tree' dataset. . 51

4.1 Proposed mixture of convolutional variational autoencoder structure for anomaly detection 57

4.2 Proposed gating network and experts structure.(a) Convolutional autoencoder gating network. (b) Convoltional variational autoencoder and convolution showing encode-decoder-encoder pipeline in experts structure. $\mathbf{z^g}$ represents the latent representation of $gate_{AE}$. $\mathbf{z_{l1}}$ and $\mathbf{z_{l2}}$ are latent representation corresponding to $expert_1$ and $expert_2$, and $\mathbf{z_{e1}}$ and $\mathbf{z_{e2}}$ is latent representation corresponding to $expert_1$ and $expert_2$. 58
4.3 Mixture of encoder-decoder models on ED pipline (a) convolutional variational autoencoder based on convolutional autoencoder gating network (b) convolutional variational autoencoder based on logistic regression gating network 63
4.4 Performance of the proposed approach and the state-of-the-art methods on overall classes in three datasets in terms of AUC 65
4.5 Performance of the proposed approach on each class in three datasets in terms of AUC 66
4.6 Latent space visualization comparison of the proposed model over the proposed model with only one expert structure. (a) and (b) shows proposed model with single expert and two expert structures, respectively using CIFAR10 (class DEER), (c) and (d) shows proposed model with single expert and two expert structures, respectively using STL10 (class DOG) dataset 73

5.1 Illustration of orthogonal projection from the full signal space of dimension N. Subspace H_1 is built by a set of normal data $\widehat{\mathbf{x}}_1$, $\widehat{\mathbf{x}}_2$,...,$\widehat{\mathbf{x}}_n$. The $\widehat{\mathbf{x}}_t$ and $\widehat{\mathbf{x}}_a$ denote a new normal observation and abnormal observation, respectively. Therefore, we get the following inequality: $\|\widehat{\mathbf{y}}_{H_1}\| > \|\widehat{\mathbf{y}}_1\|$, $\|\widehat{\mathbf{y}}_{H_2}\| < \|\widehat{\mathbf{y}}_2\|$, $\theta_1 < \theta_2$. 78
5.2 Proposed orthogonal projection constraint-based convolutional autoencoder for anomaly detection. 82
5.3 Proposed orthogonal projection constraint based fully connected autoencoder for anomaly detection 82
5.4 Illustration of orthogonal projection from the full signal space of dimension N. The subspace created by the m vector base (assumed horizontal) is used to find the best approximation (orthogonal projection) of $\widehat{\mathbf{x}}_t$ in this space. 84
5.5 Performance comparison of orthoganal subsepaces score (OSS) and reconstruction error score (RES) on DOG, HORSE, SHIP, and TRUCK categories in terms of ROC curve based on CIFAR10 dataset. 88

5.6 Performance comparison of OPC-CAE and state-of-the-art methods on each individual class in terms of AUC based on all data set 94
5.7 Performance of OPC-CAE on each individual class in terms of AUC based on all datasets 95
5.8 Performance of OPC-FAE on each individual class in terms of AUC based on all vector datasets 96
5.9 Performance of OPC-FAE on each individual class in terms of AUC based on all image datasets according to different hyper-parameter sets. 101
5.10 Convergence curve of the proposed model on the 'CAR' class in CIFAR10 dataset. The horizontal and the vertical axis represent the number of epochs and loss values, respectively. (a). Total loss. (b), (c), and (d) denote individual losses corresponding to reconrtruction loss, middle loss (the MSE beteen the input feature vector and the approximation of the embedded linear autoencoder), and OPC loss, respectively. 103
5.11 Anomaly detection performance of the proposed method with the increasing number of iterations in terms of AUC. 104
5.12 Space visualization comparison of the proposed OPC-CAE over the baseline CAE without using OPC based on CIFAR100. (a), (b), and (c) shows the latent space representation of the class 'rocket' using baseline autoencoder without OPC, proposed OPC-CAE with normal space, and proposed OPC-CAE with abnormal space, respectively, (d) (e), and (f) shows latent space representation of the class 'bicycle' using baseline autoencoder without OPC, normal space in the proposed OPC-CAE , and abnormal space in the proposed OPC-CAE, respectively. 106

List of tables

2.1 Details of the benchmark datasets used for performance evaluation. . 12
2.2 Performance comparison of ours and the state-of-the-art methods in terms of AUC 14
2.3 Performance statistics of our proposed model in terms of precision, recall and F1-score 14
2.4 Performance of our proposed framework with and without linear autoencoder interms of AUC 16
2.5 Performance of our proposed framework using with and without linear autoencoder on KDD99 18
2.6 Performance of our proposed framework using with and without linear autoencoder on Optdigits 18
2.7 Performance of our proposed framework using with and without linear autoencoder on default of credit card clients 18
2.8 Performance of our proposed framework with and without V-CNN interms of AUC 19

3.1 CVAE-MII structure for image anomaly detection 30
3.2 FVAE-MMI structure for vector anomaly detection 30
3.3 Performance comparison of CAE, LCAE, and CVAE in terms of average AUC. 40
3.4 Performance comparison of CVAE-MMI and the state-of-the-art methods on overall classes in image datasets in terms of mean AUC 41
3.5 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR10 . 45
3.6 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR100 . 46
3.7 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning STL-10 . . . 47

3.8 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning IMAGENET 48
3.9 Performance comparison of FVAE-MMI and the state-of-the-art methods on overall classes of vector data sets in terms of AUC 48
3.10 Performance comparison based on ablation validation in terms of average AUC 50

4.1 Proposed Gated Mixture of Experts for Anomaly Detection 68
4.2 Performance comparison of the proposed over encoder-decoder mixture models in terms of average AUC 69
4.3 Performance comparison of the proposed network and the state-of-the-art methods on overall class in terms of mean AUC based on three different datasets 69
4.4 Performance comparison of the proposed network and the state-of-the-art methods on each individual class in terms of AUC based concerning CIFAR10 70
4.5 Performance comparison of the proposed network and the state-of-the-art methods on each individual class in terms of AUC based concerning CIFAR100 71
4.6 Performance comparison of the proposed network and the state-of-the-art methods on each individual class in terms of AUC concerning STL-10 72

5.1 Model structure for anomaly detection 86
5.2 Performance comparison of OSS and RES on each individual class in terms of AUC based on CIFAR10 92
5.3 Performance comparison of OPC-CAE and state-of-the-art methods on overall class in terms of mean AUC based on all four image datasets 93
5.8 Performance comparison of OPC-FAE and the state-of-the-art methods on overall classes of vector datasets in terms of AUC 95
5.4 Performance comparison of OPC-CAE and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR10 . 96
5.5 Performance comparison of OPC-CAE and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR100 . 97
5.6 Performance comparison of OPC-CAE and the state-of-the-art methods on each individual class in terms of AUC concerning STL-10 . . . 98

5.7 Performance comparison of OPC-CAE and the state-of-the-art methods on each individual class in terms of AUC concerning IMAGENET 99
5.9 Performance comparison of CAE, OCGAN and the proposed on each individual class in terms of AUC based on CIFAR10 with noise of Gaussion distribution $\mathcal{N}(0, 0.1)$. 102
5.10 Performance comparison of CAE, OCGAN and the proposed on each individual class in terms of AUC based on CIFAR10 with noise of Uniform distribution $\mathcal{U}(0, 0.3)$. 102
5.11 Performance comparison based on ablation validation in terms of AUC104

B.1 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR10 . 126
B.2 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR100 . 127
B.3 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning STL-10 . . . 128
B.4 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning IMAGENET 129

Chapter 1

Introduction

1.1 Motivation

With the improvement in the field of big data and the great improvement of computer hardware, artificial intelligence (AI) technology is advancing by leaps and bounds, and more and more attention is being paid to it in various fields. Therefore, as an important branch of artificial intelligence, machine learning technology is particularly important. Humans obtain information from the real world and use brain to process the data. Data in the real world mainly includes signals, images and videos. It is very limited for humans to manually process high-dimensional complicated data, and difficult to pay a attention to all the information, so the use of machine learning will effectively solve this problem. Machine learning has demonstrated its superior strength in many difficult and high- dimensional data analysis tasks.

With the development of technology, artificial intelligence has been applied to all walks of life. As an important branch of artificial intelligence, detection plays an important role. In the industrial field, inspection is the most effective means to ensure safety and quantity. Departments engaged in hazardous industries, such as high-voltage power transmission and nuclear industries, mainly use manual detection, which consumes a lot of manpower and material resources. At the same time, such industries have high requirements for response. Manual detection will reduce accuracy, detection efficiency, and prolong response time. Also, in order to ensure the quantity of products in production lines, a lot of manpower is often required to inspect products, which will undoubtedly bring a heavy burden to the finances. In addition, artificial intelligence is also in great demand in the hospital system, which have a greater demand for monitoring the patient's physical condition. In the field of security, whether it is a private place or a public place, intelligent

monitoring and analysis play an important role in the safety of public life and property. Compared with traditional manual monitoring, intelligent monitoring analysis has huge advantages in monitoring. With the increase in international anti-terrorism demand, the flow of people at airports and railway stations has increased, and manual security detection cannot always pay attention to all important places. In addition, the scanning of a large number of luggage items will bring visual fatigue to the inspectors, which will greatly increase the missed detection rate and the false detection rate. And people's attention can only be concentrated for a short time, which requires a large number of personnel to conduct inspections. Recruiting a large number of video inspectors will increases the cost of enterprises. At the same time, camera technology and monitoring and detection algorithms continue to improve, and intelligent video analysis systems are applied to more scenarios and tasks.

Anomaly detection (AD) technology is a cutting-edge technology in computer vision and intelligent monitoring, and has a wide range of application values in security and convenience services. First of all, AD is generally considered a binary classification, and abnormal data is rarely or even unavailable. In most cases, the system is running under normal condition; only a few cases will be abnormal, and the abnormal state is unknown and unpredictable. Secondly, abnormal condition may be fatal, so extremely fast response time is required. For example, in network intrusion detection, once abnormal access is encountered, it needs to be blocked immediately to ensure information security. However, this also requires accurate identification of abnormal states supported by effectiveness of monitoring system. Because of the uncertainty of abnormal conditions, the boundary division of normal data is very important. A compact boundary based on normal features can effectively prevent abnormalities. AD is a complex and widely used technology in computer vision. The basic assumption of anomaly detection is that only normal data is available, and abnormal data can only be used during testing. This is also in line with the reality, that is, the normal state is the majority, and the abnormal state rarely occurs. Therefore, the core task of anomaly detection is to learn the discriminative features of normal data to distinguish between normal and abnormal data.

1.2 Structure of thesis

The human brain is simulated by a computer and stored in the computer in the form of parameters, which is a way of memory. The parameters represent the running state

of the system, and this idea is applicable in many disciplines. Designing algorithms for feature learning and anomaly detection requires an in-depth understanding of related knowledge and the direction that may improve performance. Map high-dimensional data to low-dimensional data through unsupervised learning, which provides valuable information on how to understand and evaluate. Therefore, when performing anomaly detection tasks, the effectiveness of feature extraction needs to be considered. Useless features will reduce the accuracy of detection. This work must first understand the features of normal data, and extract and analyze all the steps of this information for model design. The reconstruction task will be an important step in anomaly detection. The low-dimensional popularity of low-dimensional space represents the most fundamental feature of the data, which is essential for reconstruction. In addition, the distribution of low-dimensional space will become more and more concentrated during the training process. The more concentrated the normal clustering, the more accurate the detection of abnormal points.

Chapter 2 provides a brief overview of anomaly detection (AD) and the way knowledge is extracted and used for AD tasks. We analyze one-dimensional data in the learning process by analyzing the influence of vector-based convolutional networks in this chapter. During deep network training, a low-dimensional representation of the learned features will appear in the linear autoencoder. This is expected to improve the generalization ability of the network and set a tight boundary on normal data. These findings are confirmed by experiments on classification tasks. Chapter 3 follows the neural network model in Chapter 2 to study the feature location ability after mutual information learning is introduced into the convolutional neural network. In this chapter, we propose a model based on a linear autoencoder to classify image AD tasks, and it performs better than the most state-of-the-arts methods. We have studied the applicability of events learned in three different targets of MMI. The original training target is KLD for accurate anomaly detection. In addition, we believe that the adaptability of the model is important for different domains of the input data. Therefore, we evaluated the proposed objective function on image and vector data.

Chapter 4 propose a mixture of experts ensemble with two convolutional variational autoencoders and convolution (MEx-CVAEC) model. The manifolds information are extracted using the convolutional variational autoencoder (CVAE) which can be trained end-to-end via back-propagation (BP). Moreover, to obtain more manifold information in the latent space, we map the reconstructions into the new low-dimensional latent space in each CVAE using an encoder-decoder-encoder (EDE)

pipeline . Furthermore, in order to obtain a quantitative index from the latent space which is used to measure whether the manifold of the latent space can represent the discriminative features of anomaly-free data, a tower structure is used in the mixture-of-expert model to assign a latent score to each latent representation.

In Chapter 5, we propose a convolutional autoencoder based on orthogonal projection constraint (OPC-CAE). The space after the CNN is called as the full signal space. The data in the full signal space are projected into the range and null subspace by the projection operator. The range subspace and null subspace are named as normal space (NS) and abnormal space (AS), respectively. The NS contains the main information related to normal data; the information not related to the normal data is projected to AS. To ensure disjoint that it is disjoint between the two subspaces, OPC are adopted for the projection operator. Using OPC, we can obtain two mutually orthogonal subspaces. Orthogonality is responsible for the disjoint between two subspaces, implying that there is no common non-zero element between them. To the best of our knowledge, this is the first study that introduces an AEs-based model with two orthogonal subspaces for AD.

Appendix A gives full mathematical definition of the maximization of mutual information learning of the proposed deep models in chapter 4. The update rules with maximization of mutual information are proposed in this appendix highlighting the benefits of MMI training for the generalization and robustness. Appendix B gives detailed results on each dataset, which can demonstrate the stability and effectiveness of our proposed model.

Chapter 2

Detection of One Dimensional Anomalies using a Vector-based Convolutional Autoencoder

2.1 Introduction

Deep learning has achieved encouraging performance in many visual task applications, which were included with labels. The cost of labeling increases, as the amount of data increases. Generally unusual data appeared in real life entities cannot be effectively trained by a classification model because of less number of data. Hence anomaly detection algorithms is used to identify unusual/abnormal samples by training the model using normal samples [23]. For example, the practical application of anomaly intrusion detection demonstrates the anomaly detection task as shown in Fig. 2.1. The red color points in the plot represents the abnormal data and it took various positions and values due to different external factors.

In general, anomaly detection tasks used large number of normal samples to train the model parameters Θ to generate the feature distribution $p(x)$ for normal samples. However, in training phase the number of abnormal samples are very small or sometimes not available to identify the abnormal samples in the test phase. In this case, only normal samples can be used to optimize the parameters of the model and hence the abnormal score $S(x)$ can be calculated using the test data for identifying the abnormal samples.

Varying number of neurons and layers has been observed to largely affect the performance of the anomaly detection models [76, 43]. Several intrusion classification

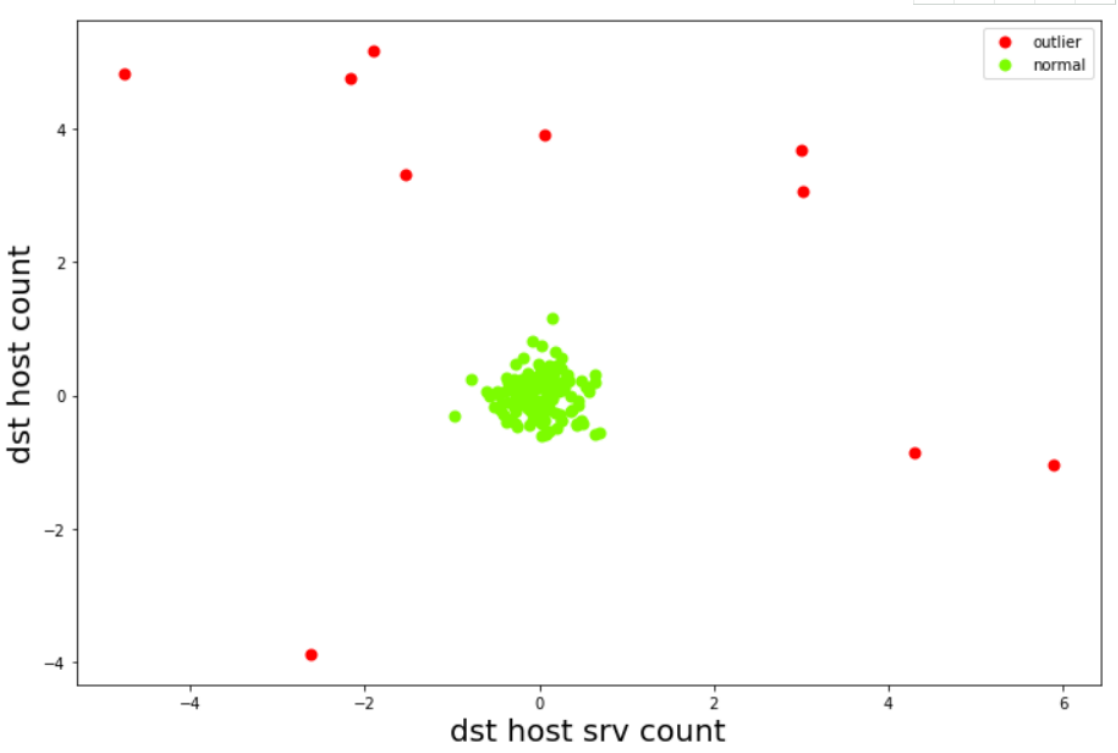


Fig. 2.1 Demonstration of intrusion detection on KDD99 dataset. The two features of network connections are represented as 'dst host count' and' dst host srv count'. The green and red color points indicated the normal and abnormal network connections

models have focused deep belief networks with stacked Restricted Boltzmann Machine and showed superior performance in identifying anomalies [7**?**]. Inspired from the aforementioned studies, we proposed to develop a V-CAE model for anomaly detection. The core of the proposed architecture is a linear autoencoder, which is used to find the sub space of the normal data by using the feature vectors extracted by the vector-based convolutional neural network (V-CNN) [38]. The V-CNN is used to extract non-linear feature vector from the input vector by 2-D convolutional neural network. The proposed V-CAE framework for identifying anomalies is shown in Fig 2.2. In this study, we used input data as a vector form and only the features extracted from the normal input data are used to train our proposed model.

The main contributions of this paper are as follows: 1) We used a autoencoder based on mutual information to enable the encoder and decoder to learn the most significant features of the input data. 2) We added a linear autoencoder to construct a low-dimensional manifold of the normal samples. 3)We used combined abnormal score computed from two different reconstruction errors: first one is calculated

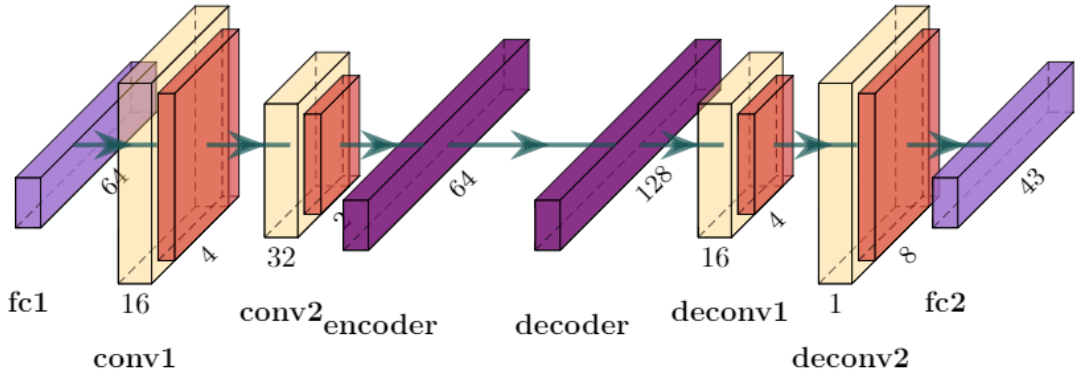


Fig. 2.2 Pipeline of the proposed approach for anomaly detection. The parameters of the model shown in this figure are selected according to the feartures of KDD dataset. In fc1,conv1,conv2 the activation functions are leaky relu; In deconv1,deconv2 the activation functions are relu; In fc2 the activation function is tanh.The parameters in this figure are the size of the output data of each layer. The input data is 43-dimensional vector data and first passes through fc1 layer. Then the data dimension becomes 64-dimensional. And the data is converted into matrix with $8 \times 8 \times 1$ format as the input of conv1 after fc1; After output from conv2 layer, the data is changed into vector form, as input of linear encoder. The dimension of output data of linear encoder is 64; The process of data output from decoder to the whole model is similar to the previous process.

between the input and output of the model, and the second one is calculated between the input and output of the linear autoencoder. 4)The effectiveness of the proposed method is experimentally evaluated by comparing with the state-of-the-art methods. 5)We conducted ablation study on our proposed framework by removing linear autoencoder in detecting anomalies.

2.2 Related Works

Anomaly detection has always been the focus of researchers, especially in the fields of finance, information security, video surveillance and medical imaging. The traditional methods are used to measure the similarity between data based on distance [78], density [13], angle [47], isolation and [56], clustering [30], etc. These algorithms are actually similar in lower dimension, because the core assumption is that "the representation of abnormal points is different from normal points and also it is a minority group". However, most similarity based algorithms will face

the curse of dimensionality, that is, common similarity measures (such as Euclidean distance) will often fail on high-dimensional data [113][83].

In order to solve this problem, many methods have been proposed, including:

1. Dimension reduction or feature selection [68]
2. Subspace methods, such as detection and merging on multiple low-dimensional spaces, random projection (randomly generating multiple subspaces and modeling separately on each subspace, feature bagging) and random forest.
3. Graph based methods are used to represent the relationships and extracted features of data [6].
4. Intrinsic dimensionality based reverse nearest neighbors methods[77]

Furthermore, based on the availability of data labels, anomaly detection technology can be divided into the following two types:

Supervised anomaly detection: The supervised anomaly detection mode assumes that we have labeled normal data and abnormal data. The most typical method is to transform the problem into a special two-class problem and establish a predictive classification model. Many general machine learning classification algorithms can be applied to model training [26]. The predicted data can be used to determine whether it is normal or abnormal. The supervised anomaly detection mode mainly has two application difficulties. Firstly, in the training data, the amount of abnormal data is far less than the amount of normal data, which brings a common data imbalance problem in the field of machine learning and data mining. Secondly, it is very challenging to obtain accurate and representative anomaly class label data. Researchers have been proposed sampling, price sensitivity, active learning and other methods to solve the above two problems. However, in practical application, the supervised anomaly detection model is still very limited.

Unsupervised anomaly detection: Unsupervised anomaly detection does not need to label data sets, and only normal data in the training set, so it has the widest applicability. This technique contains an implicit assumption that normal samples occur more frequently and are easier to obtain than abnormal samples. This assumption is also based on the fact that the number of abnormal samples in the data set is far lesser than the number of normal samples. Khreich et

al.[40] used one-class support vector machine (SVM) to map the data to high-dimensional space by kernel function, looking for hyperplanes to maximize the interval between the data and the origin of coordinates. Tax et al. [113] used support vector domain description (SVDD) method to map the data to high-dimensional space by using kernel function to find the hypersphere as small as possible to wrap the normal data. Yang et al. [105] modeled the normal data with Gaussian mixture model and estimated the parameters with maximum likelihood. When anomaly detection is carried out, the probability that it belongs to normal data can be obtained by bringing its features into the model. Liu et al. [57] used isolation forest method for anomaly detection.This method is suitable for the case where there are few abnormal points, and adopts the method of constructing multiple decision trees for anomaly detection. It is entirely based on the concept of isolation to detect anomalies without any distance or density measurement. He et al. [30] heuristically divided the data set into large and small clusters. If an example belongs to a large cluster, the abnormal score is calculated by using the example and the large cluster to which it belongs; if an example belongs to a small cluster, the abnormal score is calculated by using the example and the nearest large cluster.

2.3 Proposed Method

2.3.1 Overview

This paper proposes an anomaly detection method based on vector-based convolutional autoencoder. The flow chart of our proposed method is shown in Fig. 2.2. In this study, we consider the anomaly detection in one dimensional feature vectors. After the one dimensional feature vector is fed into the first fully connected layer, the vector data is converted to two dimensional matrix form. Then the deep features are extracted by the standard convolutional layers. The core of our model is a linear autoencoder, whose function is to reduce the dimension of data and finds the linear subspace of the normal samples. It is expected that this linear autoencoder in the middle of the convolutional autoencoder can help to find the tight boundary of the normal samples. The reconstructed vector by the linear autoencoder are used to reconstruct the output vector by using the deconvolutional layers. In the test phase, an abnormal sample is detected by using the scores defined by using two reconstruction errors of the convolutional autoencoder and the linear autoencoder.

2.3.2 Vector-based Convolutional Autoencoder

In order to extract the non linear manifold of the normal data, we adopt the vector-based convolution autoencoder. As shown in Fig. 2.2, the vector-based autoencoder includes an input layer, fully connected (FC) layers, a linear autoencoder, convolution layers before and after the linear autoencoder and output layer.

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ be the set of one dimensional feature vectors in the normal data and $\mathbf{x}_i \in R^m$ where m is the dimension of each sample.

The input vector first passes through FC layer and is converted into the two dimensional array. Then convolution neural network is used to extract the features of the input vector. The extracted features are converted into the vector form and then it is used as a input into the linear autoencoder. This procedure is defined by a function $C(.)$ and the flattened feature vector $\widehat{\mathbf{x_i}} \in R^d$ is given as

$$\widehat{\mathbf{x_i}} = C(\mathbf{x_i}) \tag{2.1}$$

where $d = l \times h \times ch$ and h, l, and ch are the width, the height and the number of channels of the output of the conv2 layer, respectively.

The linear autoencoder extracts the dimension reduced feature vector $\mathbf{z}_i$ from the flattened feature vector $\widehat{\mathbf{x_i}}$ as

$$\mathbf{z_i} = \mathbf{W}\widehat{\mathbf{x_i}} + \mathbf{b} \tag{2.2}$$

where $\mathbf{W} \in R^{d \times k}$ and $\mathbf{b} \in R^k$ are the weights and the bias of the linear encoder. The dimension of the extracted feature vector $\mathbf{z}_i$ is shown as k. The approximation $\widehat{\mathbf{y_i}}$ of $\widehat{x_i}$ is calculated by

$$\widehat{\mathbf{y_i}} = \mathbf{W'}\mathbf{z_i} + \mathbf{b'} \tag{2.3}$$

where $\mathbf{W'} \in R^{k \times d}$ and $\mathbf{b'} \in R^d$ are the weights and the bias of the linear decoder, respectively.

The approximation $\widehat{\mathbf{y_i}}$ of $\widehat{\mathbf{x_i}}$ by the linear autoencoder is reshaped into the original tensor format. It is used as the input of the next deconvolution layers. Finally, the output vector $\mathbf{y_i}$ of the vector-based convolutional autoencoder is obtained through another FC layer. This procedure is defined by function $C'(.)$ as

$$\mathbf{y_i} = C'(\widehat{\mathbf{y_i}}). \tag{2.4}$$

The loss function is defined based on the mean squared errors (MSEs) of the convolutional autoencoder and the embedded linear autoencoder as

$$\ell = \alpha(\frac{1}{n}\sum_{0}^{n}(\mathbf{x}_i - \mathbf{y}_i)^2) + (1-\alpha)(\frac{1}{n}\sum_{0}^{n}(\widehat{\mathbf{x}}_i - \widehat{\mathbf{y}}_i)^2), \alpha \in [0,1] \tag{2.5}$$

where the first term is the mean squred errors (MSE) between the input vector $\mathbf{x_i}$ and its approximation $\mathbf{y_i}$ by the convolutional autoencoder and the second term is the mean squared errors (MSE) between the feature vector $\widehat{\mathbf{x}}_i$ and its approximation $\widehat{\mathbf{y}}_i$ by the linear autoencoder. The parameter α is used to adjust the degree of contribution of these two MSEs to the objective function ℓ.

2.3.3 Anomaly Scores

In the test phase, the model calculates the anomaly score of each test sample $\mathbf{x}$. Again the anomaly score is defined based on the reconstruction error $S_1(\mathbf{x})$ of the convolutional autoencoder and the reconstruction error $S_2(\mathbf{x})$ of the linear autoencoder as

$$S(\mathbf{x}) = \lambda S_1(\mathbf{x}) + (1-\lambda)S_2(\mathbf{x}) \tag{2.6}$$

where λ is the tuning parameter that can be adjusted according to the tasks. The reconstruction error $S_1(\mathbf{x})$ between the input vector $\mathbf{x}$ and it approximation $\mathbf{y}$ by the convolutional autoencoder is defined as

$$S_1(\mathbf{x}) = ||\mathbf{x} - \mathbf{y}||^2 \tag{2.7}$$

Similary, the reconstruction error $S_2(\mathbf{x})$ between the feature vector $\widehat{\mathbf{x}}$ and its approximation by the linear autoencoder is defined as

$$S_2(\mathbf{x}) = ||\widehat{\mathbf{x}} - \widehat{\mathbf{y}}||^2 \tag{2.8}$$

In order to evaluate the impact of the overall anomaly detection performance, the anomaly score are normalized. At first, the anomaly scores $S = \{S(\mathbf{x}_i)|\mathbf{x}_i \in X\}$ for all training samples X are calculated and the maximum $\max(S)$ and the minimum $\min(S)$ of the anomaly scores are obtained. Then the anomaly score $S(\mathbf{x})$ for the new samples is normalized as

$$p = \frac{S(\mathbf{x}) - min(S)}{max(S) - min(S)} \tag{2.9}$$

2.4 Experimental Setup

2.4.1 Data Set

In order to confirm the effectiveness and the efficiency of the proposed method, we have performed experiments using three benchmark data sets which are KDD99, Optdigits, default of credit card clients. We first carried out experiments on KDD99 abnormal intrusion data, treating the 'normal' class data in the training phase and defining other classes as abnormal data. The test set contains normal as well as abnormal data. Optdigits data is experimented by treating one class (class '3') being an anomaly, while another class (class '1') is considered as the normal data. Default of credit card clients data set is an open source data set of a foreign organization. The content of the data includes some attributes such as gender, education, marriage, age, etc. It also includes the credit card consumption and bill situation of the user over a period of time. 'Payment next month', which only includes 0 or 1, is one of the feartures from data indicates whether the user has repaid the credit card bill, '1' indicates repayment, and we classify this sample with 1 as category 1; Similarly,and '0' indicates no repayment. We classify the samples with features of 'Payment next month' which equal to '1' into class '1'; Similarly, We classify the samples with features of 'Payment next month' which equal to '0' into class '0'

The data sets used in our experiments are converted into binary data sets, i.e. normal and abnormal data. The class which is considered as normal data is used to train the model. The labels of the data sets are converted into binary labels, which are used during testing. We calculated the abnormal scores of the test sets in each data set and selected an appropriate threshold to distinguish them. The original data set is randomly divided into training/testing with a ratio of 7/3. The details of the data set used in our experiments are shown in Table 1.

Table 2.1 Details of the benchmark datasets used for performance evaluation.

Datasets	Features	Normal class	abnormal class
KDD99	43	1	Others
Optdigits	47	1	3
Default of credit card clients	24	0	1

2.4.2 Parameter Settings and Evaluation

We used adam to optimize the network parameters. The proposed method is implemented in tensorflow. The parameter α is adjusted depending on the data sets. The training was done with 1,000, 2,000 and 400 epochs for KDD99, Optdigits and default of credit card clients, respectively. In the experiments, we compared the proposed method with nine state-of-the-art methods, including several traditional supervised methods and unsupervised methods. The proposed method is compared with the four most advanced supervised methods including active learning (AL) [96], feature packing (FB) [53], local outlier factor (LOF) [13] and pattern window (PW) [107]. The proposed method is compared with the four unsupervised methods including sparse coding (SC) [3], $L21 - SRC(L21)$ [20], reverse nearest neighbors (RNN) [77] and self-representation outlier detection (SRO) [108]. In addition to these eight methods, the proposed method is also compared with the sparse reconstruction (SR) method proposed by Hou et al. [32]. We computed area under the curve (AUC) value using receiver operating curve analysis as the main evaluation measure for performance evaluation. If the AUC score is large, then the performance of the anomaly detection algorithm is good. Furthermore, we used precision, recall and F1 score to evaluate the performance of the proposed system.

2.5 Results

2.5.1 Comparison with the State-of-the-art Methods

Table 2 presented the results of our experiments. Based on this our method showed more robust performance, which is better than those of the nine state-of-the-arts methods. Among all the compared models, our model scored the highest AUC score on all the three data sets, especially it is much higher compared over the latest method [32]. Experiments on KDD99 and Optdigits selected the best α of 0.5 for detecting the anomaly data. Experiment with default of credit card clients, the best α is chosen as 0.4. By adjusting the values of λ , the detection results of the model will also change accordingly. On the KDD99 and Optdigits, the choice of λ has little influence on the model, because the distribution of S_1 and S_2 terms is enough to separate the abnormal data, so the choice of λ is 0.5. But on default of credit card clients dataset, the choice of λ has great influence on the results. We chose λ=0.6 by assigning more weight on approximating S_1 which plays a leading role in the data set of credit card detection.

Table 2.2 Performance comparison of ours and the state-of-the-art methods in terms of AUC

Data sets	FB	AL	LOF	PW	SC
KDD99	0.140	0.2971	0.134	0.196	0.627
Optdigits	0.577	–	0.523	0.734	0.589
Default of credit card clients	0.535	0.484	0.524	0 .643	0.496
Data sets	L21	RNN	SRO	SR	Ours
KDD99	0.799	0.798	0.368	0.819	**0.996**
Optdigits	0.833	0.767	0.515	0.722	**0.996**
Default of credit card clients	0.599	0.506	0.600	0.606	**0.657**

Figure 3 shows the distribution of anormaly scores on the KDD99 dataset. Based on the distribution of anomaly scores of S_1 and S_2, it is suggested that these two terms are sufficient to distinguish normal from abnormal data.

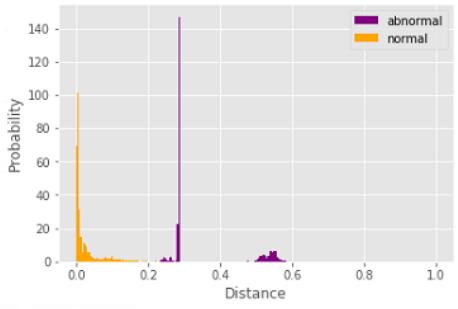
Table 2.3 Performance statistics of our proposed model in terms of precision, recall and F1-score

Data sets	Precision	Recall	F1-socre
KDD99	0.9932	0.9932	0.9932
Optdigits	0.9677	0.9928	0.9831
Default of credit card clients	0.6200	0.6201	0.6192

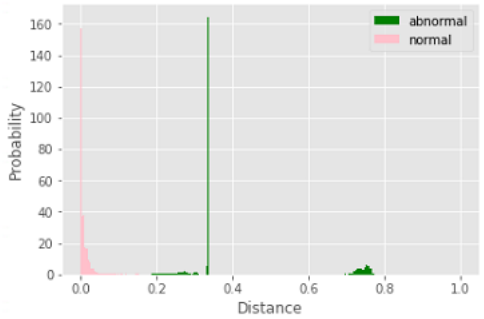
The detection results of the proposed model on default of credit card clients are not as good as those of the other two datasets. It is because it showed very poor relationship between the data and its features. Fig. 4 shows the correlation of features and the data set. It clearly explained that there is no significant correlation between the characteristics of some of its features (sex, education, marriage and age) and the categories of the data sets. Therefore, the appearance of these irrelevant features increases the difficulty of finding the anomaly data.

2.5.2 Ablation Study on Proposed Framework

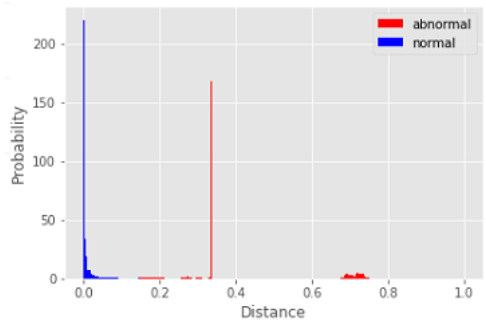
The core of our model is using linear autoencoder, whose function is to reduce the dimensions as well as compress the boundary of normal data. If the linear autoencoder is removed from the framework, the anomaly score can be calculated only from the term S_1. As shown in Figure 3(c), removing the linear autoencoder has no effect on KDD99 dataset, and the same is true for Optdigits dataset. For default



(a) First subfigure



(b) Second subfigure



(c) Second subfigure

Fig. 2.3 Detection of distributions of abnormal scores using our proposed method on KDD99. (a) distribution of S_1 (b) distribution of S_2 and (c) without linear Autoencoder.

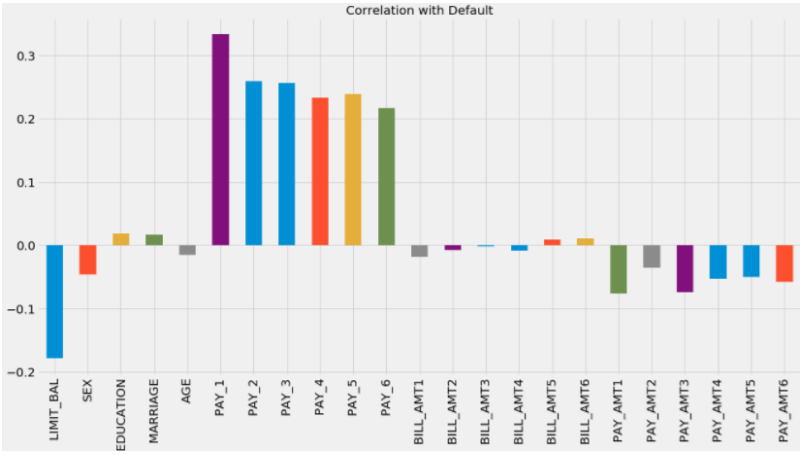


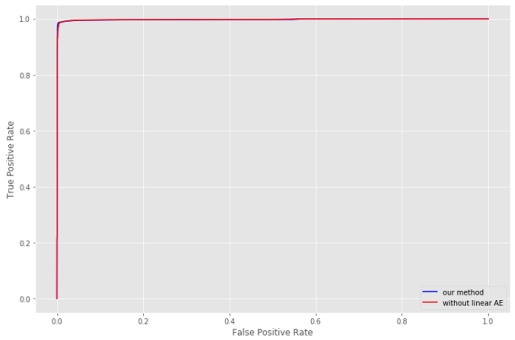
Fig. 2.4 Interpreting of correlations between categories of data set and its features

of credit card clients dataset, the detection performance will be drastically reduced. As can be seen from Table 4, the proposed framework without linear autoencoder showed less value of AUC compared over the framework with linear autoencoder. Furthermore, the precision, recall and F1-score of our method with linear autoencoder is significantly better than those values without linear autoencoder.

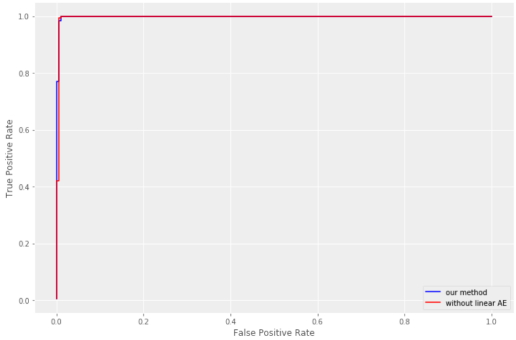
Table 2.4 Performance of our proposed framework with and without linear autoencoder interms of AUC

Datasets	with linear AE	without linear AE
KDD99	**0.9973**	0.9953
Optdigits	**0.9986**	0.9967
Default of credit card clients	**0.6570**	0.5894

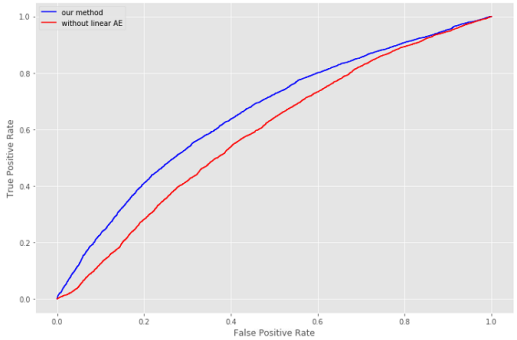
As can be seen from Figure 5, the results of our proposed method are better than without autoencoder on KDD99 data set and Optdigits dataset. Meanwhile, the results on default of credit card clients data set, our proposed framewok with linear autoencoder is better than that without linear autoencoder. It proved that our proposed V-CAE structure has potential ability to detect abnormal samples. In addition, as shown in Tables 5 and 6, the precision, recall and F1-score results of our method with or without linear autoencoder are almost similar. According to Table 7, those measures on default of credit card clients data set, our method with linear autoencoder showed better performance than those values without linear autoencoder.



(a) KDD



(b) Optdigits



(c) Default of credit card clients

Fig. 2.5 Comparison of ROC curves for abnormal scores on different data sets. (a) KDD, (b) Optdigits and (c) Default of credit card clients.

Table 2.5 Performance of our proposed framework using with and without linear autoencoder on KDD99

Measures	with linear AE	without linear AE
Precision	**0.9932**	0.9908
Recall	**0.9932**	0.9908
F1 socre	**0.9932**	0.9908

Table 2.6 Performance of our proposed framework using with and without linear autoencoder on Optdigits

Measures	with linear AE	without linear AE
Precision	**0.9946**	0.9945
Recall	**0.9946**	0.9942
F1 socre	**0.9945**	0.9940

As can be seen from Table 8, We removed the vector-based convolutional neural network(V-CNN) before and after the linear Autoencoder, and only used the linear Autoencoder for experiments. We found our results were better than the results of structure without the V-CNN.

Overall, experimental results can clearly explained that our proposed system with linear autoencoder can perfectly separate the abnormally distributed data as shown in Fig.3. The reconstruction error of the abnormal data is always larger than that of normal data and thus the results demonstrated that the anomaly data can be well detected by our proposed V-CAE approach. However, the differences of reconstruction errors between the normal and abnormal data is not very high on the credit card data set. Hence it is difficult to distinguish the anomalies from the normal data set. But still our proposed model achieved the second highest score in detecting anomalies on credit card data set.

Table 2.7 Performance of our proposed framework using with and without linear autoencoder on default of credit card clients

Measures	with linear AE	without linear AE
Precision	**0.6200**	0.5613
Recall	**0.6201**	0.5698
F1 socre	**0.6192**	0.5649

Table 2.8 Performance of our proposed framework with and without V-CNN interms of AUC

Datasets	with V-CNN	without V-CNN
KDD99	**0.9973**	0.9958
Optdigits	**0.9986**	0.9914
Default of credit card clients	**0.6570**	0.5268

Chapter 3

Extensive framework based on novel convolutional and variational autoencoder based on maximization of mutual information for anomaly detection

3.1 Introduction

With the rapid development of science and technology, greater attention has been paid to the questions of establishing security. As a security-related task, anomaly detection has been attracting the interest of the increasing number of researchers. It has been widely used in the fields of video surveillance [14, 29], defect detection [50, 101] and medical imaging [90, 38]Anomaly detection can be used to identify various patterns in a sample often including important information. In reality, considering that using abnormal samples are insufficient, only normal samples are employed to train the network aiming to learn the parameter Θ to generate the feature distribution $p(x)$ of the normal data for anomaly detection. The target of anomaly detection task is to distinguish between normal samples and abnormal samples, which is considered as a binary classification. Several research works on anomaly detection attempted to identify abnormalities in samples using machine learning and reported that with an increase in the abnormal score, the probability of finding an outlier augmented as well [77, 13]. However, significant limitations

associated with hand-engineered features deviated the goals of learning anomalous events, due to the lack of flexibility required to detect local anomalies. Moreover, estimating the amount of anomalies in advance is difficult.

It was found that deep structures could be parameterized by nonlinear functions and can learn numerous conceptual representations to detect anomalies. Hence researchers attempted to improve the effectiveness of feature extraction by incorporating deep network structures into vectors, images and videos for anomalies [77, 13, 81, 15, 41]. Recently, deep generative models are widely employed for anomaly detection. These approaches applied to learn probability distributions by training a model on an anomaly-free dataset and therefore, outliers could be identified according to their deviation from the probability model. Generative adversarial neural (GAN) networks were implemented to efficiently train a model aiming to fit a data distribution for anomaly detection [111, 63]. The convolutional autoencoder (CAE) was applied to estimate the probability of data distribution based on the criterion of large reconstruction errors [17]. Denoizing autoencoder [90] used to identify whether the anomaly data prior distribution is an interdependent set of the normal prior distribution in the latent space. However, the reconstruction error and abnormal score introduced in the aforementioned studies used to tune the threshold only for latent sampled variables, and as a result, such methods reported poor reconstruction performance in the abnormal data.

To resolve these issues, in the present paper, we propose a novel convolutional kernel based on a variational autoencoder (CVAE) to obtain robust reconstruction results for both normal and abnormal samples. Unlike in the previous studies on applying CVAE to anomaly detection in which the intention of variational autoencoder (VAE) deviated from learning an acceptable pattern for anomalous data identification, we propose to introduced the concept of the maximization of mutual information (MMI) between multiple discriminator spaces to regularize the objective of CVAE. The generalization ability of variational autoencoders (VAEs) is higher compared with that of the autoencoders (AEs), as it relies on probabilities. Therefore VAEs have been widely applied in various anomaly detection tasks [75, 8]. However, VAEs employ the regularization, which minimize or maximize the Kullback-Leibler divergence (KLD) distributions to control the space of the encoder, and no regularization is considered in the generator. This feature leads to the losses of information about the input data distribution and mapping to the prior probability distribution, thereby generating the output with the high false positives. Therefore, the application of VAEs to anomaly detection tasks needs to be facilitated by adding

suitable regularization techniques [58, 39]. In the present study, we investigate the possibility to address this issue by regularizing multiple discriminator spaces aiming to estimate how precisely the output matches its input, rather than relying only on the encoder latent space. This can be achieved by maximizing the mutual information (MI) between three different pairs of variable targets, including both local and global information in multiple discriminator spaces.

To show the effectiveness of our proposed approach in distinguishing anomalies, we utilized toy sampled data using normal samples from Gaussion distribution with mean 1, standard deviation 2, abnormal samples from Gaussion distribution with mean 2, standard deviation 2 and the size of 1000×40 (normal and abnormal samples are both 1000) as shown in Fig. 1(a). We intend to maximize the mutual information (MMI) between the raw input and the latent space, which can effectively constrain the representation of the latent space. It makes the latent space to learn the discriminative features of the raw data effectively and thus avoid the noise which is being projected into the latent space is shown in Fig. 1(b). For visualization, we used principal components analysis (PCA) to reduce the dimensionality of the raw data and the latent representation to the two dimension vectors.

As shown in Fig. 1 the performance of the proposed approach is highly acceptable in distinguishing the anomalies from anomaly-free data, compared over the methods using manifolds of the latent space by convolution variational autoencoder (CVAE) without MMI (Fig. 1(c)), and OCGAN [70] (Fig. 1(d)). Because the conventional methods were not implementing constraints on the latent space, the noise information of the raw data could also be possible to be mapped into the low-dimensional latent space and important features are ignored for accurate anomaly detection.

However, it is not a simple task to introduce the maximization of MI in the high-dimensional space as a measure of the true dependence between variables for anomaly detection. A recent thread of research works have been focused on representing MI in deep generative models concerning various domains and tasks [11, 36, 65]. These approaches have integrated MI maximization with prior matching to restrict the learning process depending on necessary statistical properties [60]. The approaches for realization of MI have diverged, including Jensen-Shannon divergence (JSD) [12], Monte-Carlo method [46], and the original objective of KLD [33] between the joint and marginal for data reconstruction. Unlike in the aforementioned studies, in the present research work, we investigate the applicability of the events learned across the three different objectives targeted at MMI with original training objective of KLD for accurate anomaly detection. Furthermore, we consider that the adaptability

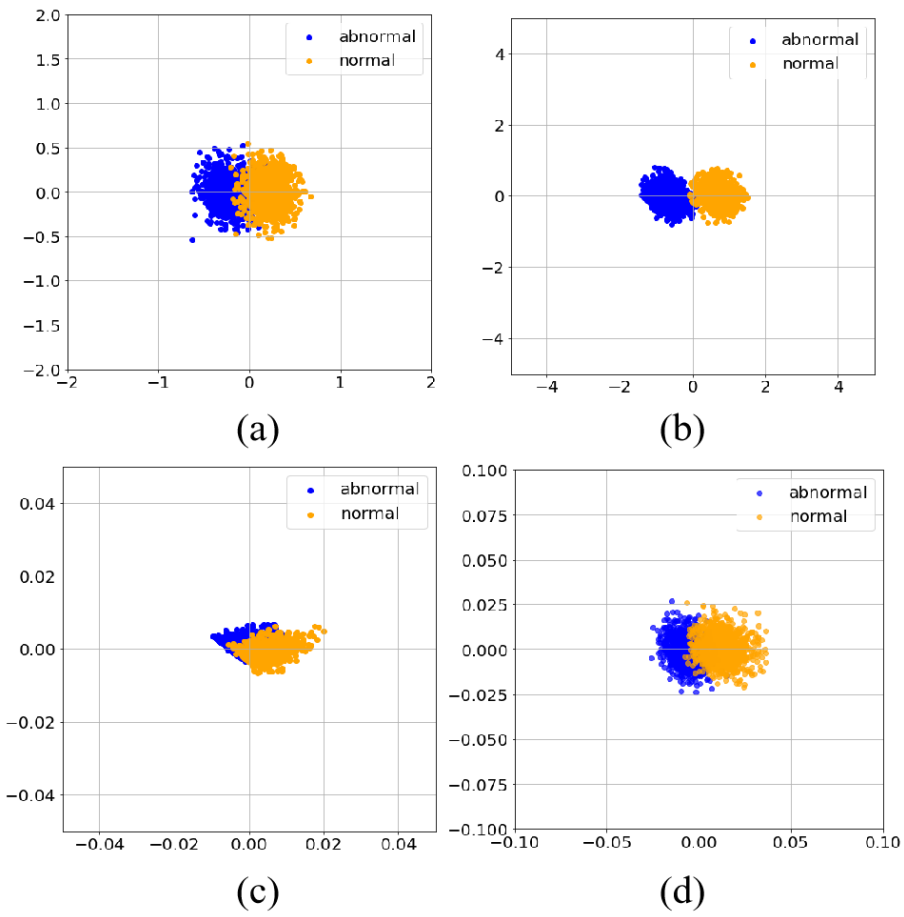


Fig. 3.1 Performance visualization of the proposed approach over conventional methods for anomaly detection based on the toy sampled data. (a) The distribution of the raw input, (b) latent representation of the proposed method, (c) latent presentation of CVAE, and (d) latent representation of OCGAN.

of a model is important for the different domain of the input data, and therefore, we evaluate the proposed objective function on both image and the vector data. To the best of our knowledge, this is the first study suggesting the CVAE-MMI model for anomaly detection.

The main contributions of this paper can be summarized as follows:

1. We proposed a novel convolutional kernel based on a variational autoencoder (CVAE) for complex image anomaly detection by the maximization of MI (MMI) through regularizing multiple discriminator spaces to control the boundary of the distribution.

2. We investigated the suitability of MMI that was learned considering the three different objectives targeted at both the local and global variables facilitate to achieve the additional supervision power for accurate detection of anomalies.

3. We intuitively verified that the proposed approach was robust and easy to adopt for image and vector anomalies, including the convolutional and the fully connected layers, respectively, for the encoder-decoder structure in the proposed MMI-based framework.

4. Extensive experiments were conducted to confirm the reliability of the proposed MMI-based loss (L_{MMI}) and anomaly score (S_{MMI}) for the detection of abnormality in the convolutional autoencoder-based architecture.

5. We compared the performance of the proposed framework with the state-of-the-art approaches for the detection of image and vector anomalies.

6. The ablation study was conducted to check the capability of the network with or without our proposed objectives for anomaly detection concerning both image and vector datasets.

3.2 Related Work

In this section, we briefly review the related works dedicated to the decision boundary and density estimation models, as well as to applying autoencoder and deep generative techniques for anomaly detection.

3.2.1 Decision boundary and density estimation models

To solve the anomaly detection problem, several studies have proposed different terms such as abnormality detection [35], one-class classification [62], and outlier detection [82]. Although the used terminology differs across the papers, the idea of solving the problem is basically the same. It is considered that estimating a decision boundary between different anomalies with maximal separability using support vector machine (SVM) allows effectively solving high-dimensional data modeling problems under finite sample conditions [100]. One class SVM (OCSVM) [40] has been introduced assuming that the hyperplane represented by the support vector can classify the target class samples against the anomaly origin within the maximum interval.

Support vector data description has been proposed [98] to identify a super sphere including target samples called optimal hypersphere. This approach allows easily identifying anomalous points located outside of the hypersphere. A density-based method for anomaly detection has utilized the Parzen window [107] to measure the locality information and distribution density of data. The nearest neighbor method [45] has been implemented to estimate the local density of data based on the distance between two points and to calculate the number of neighbors aiming to determine the outlier. The method inspired by the aforementioned studies [78] has improved the local density estimation approach by estimating the distance between each point and its neighbors. It was used to identify outliers by traversing the neighboring points.

3.2.2 Deep autoencoder and generative models

In recent years several deep generative networks have been developed to perform parameter projection and data reconstruction. Generally, it is assumed that the outliers produce larger residuals compared with the regular data. Convolutional autoencoder (CAE) [15] can be used to learn characteristics of data distribution based on the anomaly-free data, at the same time, generating larger reconstruction errors in the abnormal data. Linear-based CAE (LCAE) [109] can be utilized as the core part to better learn the manifold of data and compress the boundaries of normal data. A denoizing autoencoder based on bidirectional long short-term memory recurrent neural networks (RNN) [61] has been applied to process the auditory spectral features and to compute the reconstruction error between the input and output of an autoencoder to detect new events. A sparse representation framework [97] introduced to learn dictionaries based on the latent space of a variational autoencoder can be used to find the anomalous data that exhibit significant reconstruction errors. Similar denoizing autoencoder [70] has been introduced to explicitly specify constraints in the latent space to determine the given class for anomaly detection. However, existing autoencoder network architectures rely on mean square error loss (MSE) to comply with the features of anomaly detection, which is not optimal to establish representational learning in hidden layers. To address this deficiency, in the present study, we construct a model to learn the stabilized comprehensible features for both local and global variables that preserve the required information represented by the input to the greatest possible extent.

A simple VAE using the Monte-Carlo reconstruction probability has demonstrated the superior performance compared with AE[21]. However, a generative VAE [22]

approach constructed with intrinsic limitations without the knowledge of specific regularization is not applicable to secured anomaly detection. Therefore, in this study, we aim to addresses the problem of distribution by introducing a more generative model that can be tuned according to different problem domains by combining the ability of variational autoencoder (VAE) with MMI. The proposed network can be used to learn the features based on MMI and set the tight boundary around the normal data. The models based on GAN have introduced a discriminator considering the latent representations (either a generator input or from the encoder) for anomaly detection [111]. To exploit both generator and discriminator [54], the trained discriminator has been coupled with the residuals observed between the reconstructed and actual data concerning possible anomalies. The generative model f-anoGAN [89] has been introduced as a fast mapping technique based on a trained encoder and anomalies have been detected considering a combined anomaly score. Similarly, in the current study, we are motivated to use the combined MI-based anomaly scores to control the boundary of the distribution.

3.3 Background

3.3.1 Variational autoencoder

The goal of applying VAE is to make the posterior distribution as close as possible to the prior one. The idea underlying VAE is similar to that of AEs, with the difference that the encoder of VAE forces the representation code Z to prior probability distribution $P(Z)$, and the decoder generates new realistic data with code Z sampled from $P(Z)$. The conditional distributions of both encoder and decoder are represented as follows: $Q\phi(Z|X)$, and $P_\theta(X|Z)$. VAE employs the regularization of KLD to limit the capacity of the encoder and to measure the similarity between two distributions. To estimate the maximum likelihood, VAE maximizes the evidence variational lower bound (ELBO) $\mathcal{L}(x)$. To optimize KLD between $Q\phi(Z|X)$ and $P_\theta(X|Z)$, the encoder calculates the vectors of Gaussian distribution $Q\phi(Z|X)$, and the vectors are denoted as mean μ and standard deviation σ.

To optimize the probability distributions, VAE aims to minimize the reconstruction errors between the inputs and the outputs. Given the data point $x \in R$, the objective function can be written as follows:

$$\mathcal{L}_{\mathcal{V}} = \mathcal{L}_{\mathcal{MSE}}(x, G_\theta(Z)) + \lambda \mathcal{L}_{\mathcal{KLD}}(E_\phi(x)) \tag{3.1}$$

Parameters G and E denote the generator and encoder, respectively. The term $\mathcal{L}_{\mathcal{MSE}}(x, G_\theta(Z))$ represents MSE between the inputs and their reconstructions. The term $\mathcal{L}_{\mathcal{KLD}}(E_\phi(x))$ regularizes the encoder by enabling the approximate posterior $Q\phi(Z|X)$ to match the prior $P(Z)$. To maintain the trade-off between these two targets, each KLD target term is multiplied by a scaling hyperparameter λ and defined by the following equation:

$$\mathcal{L}_{\mathcal{KLD}}(E_\phi(x)) = \sum_{x \in X} \frac{1}{2}(-log\sigma^2(x) + \mu^2(x) + \sigma^2(x) + 1) \tag{3.2}$$

where $x \in X$, $\sigma(.)$ and $\mu(.)$ correspond to the mean and standard deviation, respectively, according to the given x [44]. AE computes the reconstruction error as an anomaly score in the test phase, whereas VAE obtains the reconstruction probability to estimate anomalies [69]. To compute the probabilistic anomaly score, VAE samples Z according to the prior $P_\theta(X|Z)$ N times and calculates the mean average reconstruction error as the reconstruction probability [95].

3.3.2 Mutual Information

MI is utilized to learn the model distribution $P_\theta(X)$ aiming to fit the true data distribution $Q(X)$ in the best possible way. In information theory, MI between two random variables is a measure of the mutual dependence between variable defined as follows:

$$I(X; Z) = H(X) - H(X|Z) \tag{3.3}$$

where H is the Shannon entropy, and $H(X|Z)$ is the conditional entropy of Z with regard to the given X. MI is minimum when two random variables are statistically independent, or maximum when two variables contain identical information. Therefore, if MI is high, the variables are highly predictive with regard to each other. Unlike the correlation coefficient, MI captures nonlinear statistical dependencies between variables, and therefore, it can be applied to measure true dependence of variables [60].

MI is equivalent to KLD between the joint distribution P_{XY} and the product of the marginals $P_X \bigotimes P_Y$, as defined by the following equation

$$H(X; Y) = D_{KL}(P_{XY}|P_X \bigotimes P_Y) \tag{3.4}$$

where P represents the probability distribution of variables X and Y, and D_{KL} is defined as follows:

$$D_{KL}(P_{XY}|P_X \bigotimes P_Y) = \int \int p(y|x)p(x)log\frac{p(y|x)p(x)}{p(y)p(x)}dxdy \tag{3.5}$$

Here, $x \in X$ and $y \in Y$, p denote the probability densities of P. Therefore, according to Eq. 3, the larger the divergence between the joint and the product of the marginals, the stronger the dependence between X and Y. If two variables X and Y are dependent, then KL divergence of the joint and the marginal probability distributions represents the closeness to independent variables. To reduce bias and variance, recent works have suggested employing MI and bave relied on approximating the Gaussian data distribution, as well as estimating entropy with varying neighborhood sizes and dual representations of the KLD [103]. However, these MI estimators have failed to be scaled effectively corresponding to a sample dimension and therefore, several studies have aimed at maximizing the mutual information under the generative joint distribution $P_\theta(X, Z)$ [16, 22]. However, in the present study, we aim to maximize MI under ensemble learning considering different variational joint distributions $Q\phi(X, Z)$, which forces VAE to learn robust features for image and vector datasets. MI could capture nonlinear statistical dependencies between variables, and therefore, we exploit the maximization criterion for MI estimation across three different objectives as key information

3.4 Method

In the present study, we proposed a novel CVAE combined with multiple MI-based anomaly discriminator space variables that maximize the relative similarity between the input and the output feature map. The features of CVAE combined with maximization of MI (CVAE-MMI) facilitate achieving the additional supervision power over the original training objective function of the KLD term of in VAE model. The adoptability of the proposed architecture is evaluated for the image and vector datasets. Concerning image anomaly detection, convolutional neural network (CNN) including fully connected (FC) layer is used as the core element in the encoder-decoder structure of the proposed CVAE-MMI framework, as shown in Fig. 2. Concerning vector-based anomaly detection, FC layers is embedded in the encoder-decoder structure of the FVAE-MMI framework, as shown in Fig.3.

Table 3.1 CVAE-MII structure for image anomaly detection

CVAE	Discriminator
conv1(channel:32, filter:5)	conv1(channel:32, filter:5)
batch normalization	batch normalization
max pooling(2*2)	max pooling(2*2)
conv2(channel:64, filter:5)	conv2(channel:64, filter:5)
batch normalization	batch normalization
conv3(channel:128, filter:5)	conv3(channel:128, filter:5)
batch normalization	batch normalization
max pooling(2*2)	max pooling(2*2)
fully-connected(500)	fully-connected(neuron:100,activation:ReLU)
fully-connected(2048)	fully-connected(neuron:1,activation:Sigmoid)
deconv1(channel:64, filter:5)	
batch normalization	
up sampling(2*2)	
deconv2(channel:32, filter:5)	
batch normalization	
up sampling(2*2)	
deconv3(channel:3, filter:5)	
batch normalization	
up sampling(2*2)	

[1] CIFAR10 dataset is considered as an example

Table 3.2 FVAE-MMI structure for vector anomaly detection

FVAE	Discriminator
fully-connected(neuron:20,activation:ReLU)	fully-connected(neuron:20,activation:ReLU)
fully-connected(neuron:10,activation:ReLU)	fully-connected(neuron:10,activation:ReLU)
fully-connected(neuron:5,activation:ReLU)	fully-connected(neuron:5,activation:ReLU)
fully-connected(neuron:10,activation:ReLU)	fully-connected(neuron:1,activation:Sigmoid)
fully-connected(neuron:20,activation:ReLU)	
fully-connected(neuron:43,activation:ReLU)	

[1] KDD99 dataset is considered as an example

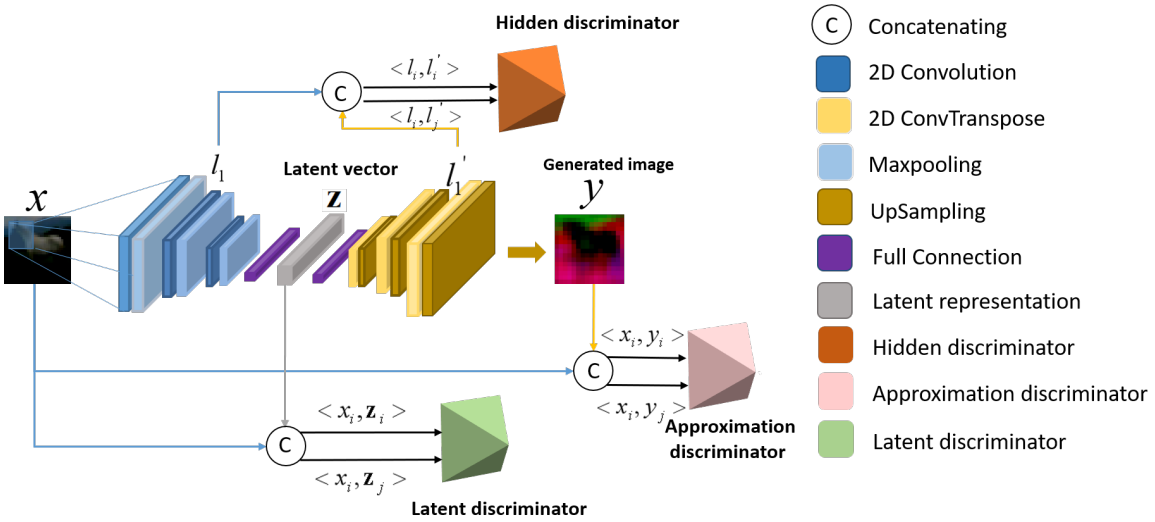


Fig. 3.2 Proposed convolutional and variational autoencoder framework with maximization of mutual information for image anomaly detection.

3.4.1 Convolutional and VAE-MMI Ensemble Framework for Anomaly Detection

We implement three convolutional layers in the encoder network with 5×5 kernels, and the stride is fixed to be 1 aiming to establish spatial downsampling. Each convolutional layer is followed by batch normalization, a leaky rectified linear unit (ReLU)activation function, and the max poolinglayer. Then, two FC output layers are added in the encoder to retrieve mean and variance of input datasets. The mean value and the variance value are used to calculate the KLD loss and latent sample variables. In the decoder, we implement three transpose convolutional and three upsampling layers with 5×5 kernels and set stride to 1. Each convolutional layer is followed by batch normalization, a ReLU activation function and the upsampling layer; however, in the third convolutional layer, we utilize tanh activation function.

We introduce three discriminators so that each of them includes two convolution layers with 5×5 kernels. We set stride equal to be 1. Each convolutional layer in the discriminators is followed by batch normalization, a ReLU activation function and the max pooling layer. Finally, a FC layer is included in the discriminator. The details of the proposed CVAE-MMI architecture is are provided in Table 1. The approximation discriminator allows estimating MI between the input and the output of the entire model. The latent discriminator retrieves MI between the input and the latent representation phase. The hidden discriminator computes MI between the output of the first convolutional layer in the encoder part (l_1, $l_1 \in L_1$) and the

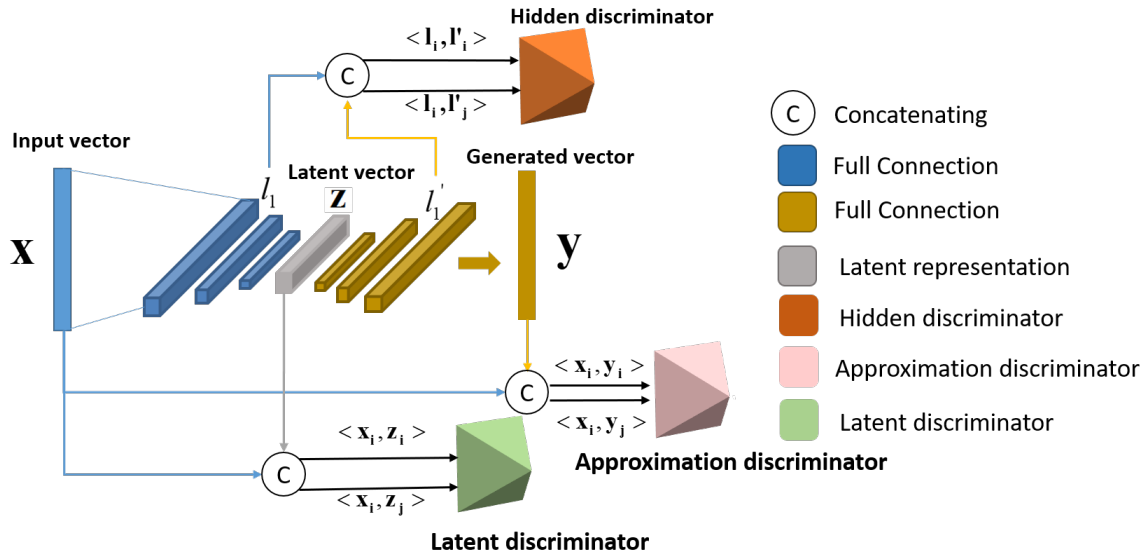


Fig. 3.3 Proposed fully connected and variational autoencoder framework with maximization of mutual information for vector anomaly detection.

input of the last convolutional layer in the decoder part (l'_1, $l'_1 \in L'_1$) in terms of forward-propagation.

Additionally, to confirm the applicability of the proposed FVAE-MMI framework to the vector data input, we replace the convolutional layers with a FC network in the encoder-decoder. We utilize two FC layers in the encoder followed by ReLU activation function. The number of neurons is set to 20 and 10 in the first and second layers, respectively. Similarly as in the decoder we use two FC layers followed by the ReLU activation function. In the discriminators of FVAE-MMI for vector datasets, we replace two convolutional layers with two FC ones. Therefore, we implement totally 3 FC layers, so that first two FC layers are followed by the ReLU activation function. The details of the proposed FVAE-MMI for vector data are provided in Table 2. The aim of each discriminator used in the proposed architecture is to maximize MI between two variables. In fact, it is rather difficult to estimate the MI from input. To solve this problem the proposed CVAE network is optimized using multiple discriminators of MMI to promote the shared information representation from input [63].

3.4.2 MMI based Training Objective

Inspired by [63, 31], we aim to develop a network incorporating MMI between two distributions. In the training process, we train CVAE-MMI using normal samples

and therefore, the encoder-decoder and encoder parts are fit for normal samples. We define the objective function as the combination of three discriminators loss functions with the original KLD term. In such way, the proposed CVAE-MMI model is optimized based on the entire context information presented in the input data.

The first loss function computes the MI in the latent discriminator between the input and its encoded representation latent space. This step is used to narrow the difference between the original input and the latent representation and is defined as follows:

$$I(X,Z) = \int\int p(\mathbf{z}|x)p(x)log\frac{p(\mathbf{z}|x)}{p(\mathbf{z})}dxdz \tag{3.6}$$

where X is a set of original input and $x \in X$. The parameter Z is defined as a set of encoded latent representations, and $\mathbf{z}$ is a vector, where $\mathbf{z} \in Z$. Here, $p(.)$ represents probability distribution.

According to the study [63], Eq. (5) can be transformed and optimized as follows

$$I(X,Z) \approx D(x,\mathbf{z}) = E_{(x,\mathbf{z})\sim p(\mathbf{z}|x)p(x)}[logH(x,\mathbf{z})] + E_{(x,y)\sim p(\mathbf{z})p(x)}[logH(x,\mathbf{z})] \tag{3.7}$$

where E denotes an expectation value. The variable $H(.)$ is equal to $\frac{1}{1+exp(-v(.))}$, where $v(.)$ is a function that can be defined according to [65] .

To solve $v(x,\mathbf{z})$ function, negative sampling estimation (NSE) is introduced based on noise-contrastive estimation [31]. Latent discriminant network $D(x,\mathbf{z})$ is used so that x and its corresponding $\mathbf{z}$ are regarded as a positive sample pair. The variable x and its non-corresponding (not encoded by x) $\mathbf{z}$ are regarded as a negative pair. Similarly, we define approximation discriminator $D(x,y)$ and hidden discriminator $D(l_1, l_1')$, as shown in Fig. 2 and 3 .

The second loss function corresponds to the feature matching error that is used to stabilize the training in the approximation discriminator. It computes the MI distance between the original image feature representation and the generated output image, which can be written as follows:

$$\begin{aligned} I(X,Y) &\approx D(x,y) \\ &= E_{(x,y)\sim p(y|x)p(x)}[logH(x,y)] + E_{(x,y)\sim p(y)p(x)}[logH(x,y)] \end{aligned} \tag{3.8}$$

The parameter Y is a set of the outputs, and $y \in Y$.

The third loss function corresponds to the active tensor space in the hidden discriminator. The latent and approximation discriminators can enable the generator

to learn the stabilized context correlated features for classification. The hidden representation is capable of obtaining the comprehensible features of spatial correlations between the input and output and therefore, it is highly significant to improve the quality of the output. The hidden discriminator calculates the MI between the first convolutional layer output of the encoder and the last convolutional layer input of the decoder. It is defined as follows:

$$\begin{aligned} I(L_1, L_1') &\approx D(l_1, l_1') \\ &= E_{(l_1, l_1') \sim p(l_1'|l_1)p(l_1)}[logH(l_1, l_1')] \\ &\quad + E_{(l_1, l_1') \sim p(l_1)p(l_1')}[logH(l_1, l_1')] \end{aligned} \tag{3.9}$$

Here, the parameters L_1, and L_1' are the output of the first convolutional layer and the input of the last convolutional layer in the encoder and decoder, respectively, in terms of the forward propagation network. The variables $l_1 \in L_1$ and $l_1' \in L_1'$ are generated from the set of input data x. According to the knowledge of data processing inequality (DPI) in [110], we consider $I(X, Y) > I(L_1, L_1')$ as a loss network to train the classification. Therefore, we compute the maximization of MI between L_1 and L_1' , which allows increasing the lower bound of MI between X and Y represented as $I(X, Y)$.

The final MMI-based reconstruction loss (L_{MMI}) combines the three aforementioned discriminator loss functions along with the KLD loss to control the boundary of the distribution. The goal of this study is to maximize the MI thereby theoretically minimizing the KLD, which is equivalent to maximizing MI between data distributions. Hence the objective function to train the proposed network is defined as follows:

$$\begin{aligned} L_{MMI} &= \lambda_{KLD} D_{KL}(P(Z)|Q(Z)) - (\lambda I(X, Z) + \lambda_A I(X, Y) + \lambda_H I(L_1, L_1')) \\ &= \lambda_{KLD} E_{x \sim p(x)}[D_{KL}(P(Z|X)||Q(Z))] \\ &- \lambda_L \left\{ E_{(x, \mathbf{z}) \sim p(\mathbf{z}|x)p(x)}[logH(x, \mathbf{z})] + E_{(x, \mathbf{z}) \sim p(\mathbf{z})p(x)}[log(1 - H(x, \mathbf{z}))] \right\} \\ &- \lambda_A \left\{ E_{(x, y) \sim p(\mathbf{z}|x)p(x)}[logH(x, y)] + E_{(x, y) \sim p(y)p(x)}[log(1 - H(x, y))] \right\} \\ &- \lambda_H \left\{ E_{(l_1, l_1') \sim p(l_1'|l_1)p(l_1)}[logH(l_1, l_1')] + E_{(l_1, l_1') \sim p(l_1')p(l_1)}[log(1 - H(l_1, l_1'))] \right\} \end{aligned} \tag{3.10}$$

where λ_{KLD} , λ_L, λ_A, and λ_H are the weighting parameters of discriminators that are used to adjust the impact of individual losses on the overall objective function $\lambda_L = \lambda_{KLD} + \lambda$. We can reformulate the proposed objective function presented in Eq.

(9) as follows:

$$L_{MMI} = \lambda_{KLD} E_{x \sim p(x)}[D_{KL}(P(Z|X)||Q(Z))] - \lambda_L D(x, \mathbf{z}) - \lambda_A D(x, y) - \lambda_H D(l_1, l_1') \quad (3.11)$$

The detailed specific inference processes related to the considered training objective functions are described in Appendix A. The proposed MMI-based training objective function is summarized in Algorithm 1.

Algorithm 1: Training objective of the proposed model

Input: Set of training data x, $x \in X$, iteration size N, weighting parameters λ_{KLD} , λ_L, λ_A, and λ_H.

Output: Y, Z, L_1, L_1'

Process from x to $\mathbf{z}$ is defined as $En(x)$, so $\mathbf{z} = En(x)$;

Similarly we can get: $y = De(\mathbf{z})$, $l_1 = En(x)$, $l_1' = En(\mathbf{z})$;

initialization;

for *iteration* $1 \rightarrow N$ **do**

Take a mini-batch of M $[x_1, ..., x_m]$ as the input;

$\mathbf{z_i} = En(x_i)$, $\mathbf{z_j} = En(x_j)$, $x_i, x_j \in M$, $i \neq j$;

$y_i = De(\mathbf{z_i})$, $y_j = De(\mathbf{z_j})$;

$l_{1i} = En(x_i)$, $l_{1j} = De(x_i)$;

$l_{1i}' = En(\mathbf{z_i})$, $l_{1j}' = De(\mathbf{z_j})$;

if *Latent discriminator update* **then**

$H_{real} \leftarrow concatenating(x_i, \mathbf{z_i})$;

$H_{fake} \leftarrow concatenating(x_i, \mathbf{z_j}), i \neq j$;

$D(x, \mathbf{z}) \leftarrow D(H_{real}, 1) + D(H_{fake}, 0)$;

Back-propagate $D(x, \mathbf{z})$ to change D;

end

if *Approximate discriminator update* **then**

$H_{real} \leftarrow concatenating(x_i, y_i)$;

$H_{fake} \leftarrow concatenating(x_i, y_j, i \neq j)$;

$D(x, y) \leftarrow D(H_{real}, 1) + D(H_{fake}, 0)$;

Back-propagate $D(x, y)$ to change D;

end

if *Hidden discriminator update* **then**

$H_{real} \leftarrow concatenating(l_{1i}, l_{1i}')$;

$H_{fake} \leftarrow concatenating(l_{1i}, l_{1j}', i \neq j)$;

$D(l_1, l_1') \leftarrow D(H_{real}, 1) + D(H_{fake}, 0)$;

Back-propagate $D(l_1, l_1')$ to change D;

end

Optimized

$L_{MMI} = \lambda_{KLD} E_{x \sim p(x)}[D_{KL}(P(Z|X)||Q(Z))] - \lambda_H D(x, \mathbf{z}) - \lambda_A D(x, y) - \lambda_L D(l_1, l_1')$

end

3.4.3 Anomaly Score

In the proposed CVAE-MMI both the trained encoder, latent representation and output of decoder part contribute to detecting anomalies. Spontaneously, the trained representation should be able to separate normal data from abnormal ones. Given a test sample t, $t \in T$, the encoder estimates the parameters of the latent variables μ and σ as the output of the encoder. The generated latent representation, and the output of the decoder received these from encoder as inputs and outputs the representation $\mathbf{z}'$, and y', respectively.

If the model takes an anomalous sample as an input, the differences between the representation and the average calculated during training is large. This difference in the scores can be calculated using the MMI-based anomaly score (S_{MMI}) as $S_{MMI} = \{s : V(t), t \in T\}$, where $V(t)$ is the abnormal score of each sample. It is defined as follows:

$$V(t) = \lambda S_a(t) + (1 - \lambda) S_l(t), \lambda \in [0, 1] \tag{3.12}$$

where λ is the tuning parameter adjusted according to the task. The outlier at the output representation for testing sample $S_a(t)$ can be defined by calculating the Euclidean distance between the generated average y_m of training data X and the output y' according to t . It is defined as follows:

$$S_a(t) = \|y' - y_m\|^2 \tag{3.13}$$

where $y_m = \frac{1}{n} \sum_{y \in Y} y$, $y \in Y$.

Similarly, the outlier at the latent space for testing sample $S_l(\mathbf{t})$ can be defined by calculating the Euclidean distance between the average latent sampled vector z_m obtained in the training phase and the generated latent output z' in VAE:

$$S_l(t) = \|\mathbf{z}' - \mathbf{z_m}\|^2 \tag{3.14}$$

where $\mathbf{z_m} = \frac{1}{n} \sum_{\mathbf{z} \in Z} \mathbf{z}$, $\mathbf{z} \in Z$. Here, the MMI-anomaly score is normalized to the interval of [0, 1]. The abnormal score close to 1 indicates the higher abnormality of the data. To address this, we set the following threshold: if $V(t) > threshold$, then it is considered as the abnormal data. The normalized score is defined as follows:

$$\tilde{s} = \frac{s - min(S_{MMI})}{max(S_{MMI}) - min((S_{MMI})} \tag{3.15}$$

3.5 Experimental Setup

3.5.1 Dataset

We intuitively whether verified the proposed CVAE-MMI and FVAE-MMI models can be robust and easy to adopt for image and vector datasets, respectively. For image datasets we used CIFAR10, CIFAR100, STL-10, and IMAGENET, the most challenging and complex datasets including various contents compared to the other properly aligned object recognition datasets such as Fashion Mnist and COIL.

The datasets CIFAR10 and STL-10 comprise the images corresponding to ten different classes, whereas CIFAR100 and IMAGENET included the images regarded to multiple classes. Therefore we select only ten classes from CIFAR100 and IMAGENET for the evaluation of the proposed anomaly detection framework. In order to simulate a anomaly detection setting for image datasets, one class is considered as the normal class, the union of other classes are considered as abnormal class, as proposed in [1, 85, 70]. The network is trained using only samples of the normal class. During testing, we use the mixture of the normal samples and the abnormal samples for test data. For vector datasets, we employ the KDD99 abnormal intrusion data including five classes: one normal class and four abnormal classes. Moreover, we consider the Default of credit card clients dataset composed of two classes: one normal class and one abnormal class.

3.5.2 Experimental Evaluation and Performance Measure

The effectiveness of both CVAE-MMI and FVAE-MMI models are evaluated to through training on normal samples and testing on the mixture of the normal and abnormal samples. The existing training and testing data proportions was adopted, including CIFAR10 [48], STL10 [19], and CIFAR100 [48] for image datasets and KDD99 [73] and default of credit card clients [106] for vector ones, while in the case of IMAGENET [21] we set the training and testing ratios of 60% and 40%, respectively.

To confirm the reliability of the proposed (L_{MMI}) and (S_{MMI}) using CVAE for anomaly detection, it is compared with MSE loss ((l_{mse}) and Euclidean distance-based anomaly score (s_{eu}) considering the CAE-based model. Furthermore, the effectiveness of CVAE-MMI in image anomaly detection is evaluated through the comparison with nine state-of-the-art methods. This included One-class Gaussian mixture model (GMM) [18], kernel density estimation (KDE) [52], CAE [17], VAE

[44], pixel convolutinal neuron network decoders (Pixel CNN) [99], GAN [90], skip connected ganomaly (SCG) [5] , anomaly detection with generative adversarial networks (AnoGAN) [90], and one-class GAN (OCGAN) [70].

The effectiveness of the proposed framework in terms of vector anomaly detection is tested through the comparison with seven state-of-the-art methods that, including active learning (AL) [2], feature packing (FB) [53], local outlier factor (LOF) [13], sparse coding (SC) [3], reverse nearest neighbors (RNN) [77], self-representation outlier detection (SRO) [3], and sparse reconstruction (SR) [32]. The individual and overall class performance of the proposed networks and state-of-the-art methods on the image and vector datasets is evaluated and compared in terms of an average area under the curve (AUC) value of the receiver operating characteristic (ROC) curve. The high AUC score indicates the good performance of a method in detecting anomalies.

3.5.3 Parameter Settings

We apply Adam optimizer to optimize the network parameters for the image and vector datasets. We set the parameters $\beta_1(0.5)$ and $\beta_2(0.99)$ for the image and vector datasets, respectively. The network is trained using 1,000 epochs for the CIFAR10, CIFAR100, STL10 and IMAGENET datasets and 2,000 epochs for the KDD99 and Default of credit card clients datasets. For image datasets, we set the learning rate of 0.0001 for CIFAR10 and CIFAR100, 0.00005 for STL10 and IMAGENET, and for each iteration, we specify the batch size of 100. For vector datasets, we set the learning rate equal to 0.00005 and the batch size of 200 for each iteration. The proposed framework is implemented in Python 3.6 using Tensorflow 1.9.

3.6 Experimental Results

3.6.1 Performance Comparison considering the CAE-based networks

We conduct several experiments to confirm the reliability of the proposed (L_{MMI}) and (S_{MMI}) using CVAE for anomaly detection . It is done through estimating the performance in terms of MSE loss ((L_{Mse}) and Euclidean distance-based anomaly score (S_{Eu}) considering the convolutional autoencoder-based model on the CIFAR10 and STL-10 datasets. To ensure fair comparison, the performance of CVAE is compared

with the CAE [15] and LCAE [109] methods in terms of the AUC value. Overall, we conduct four different experiments considering the cases with and without using the proposed loss and anomaly score to compare the prediction performance estimates of on CAE, LCAE, and CVAE, which are represented as follows: 1) structures with (l_{mse}) and (s_{eu}), 2) structures with (l_{mse}) and proposed (S_{MMI}), 3) structures with proposed (L_{MMI}), and (s_{eu}), 4) structures with both proposed (L_{MMI}) and (S_{MMI}).

Table 3.3 Performance comparison of CAE, LCAE, and CVAE in terms of average AUC.

Datasets	CAE	LCAE	CVAE
	l_{mse}+s_{eu}		
CIFAR10	0.5234	0.5942	**0.6193**
STL10	0.5698	0.5853	**0.6017**
	l_{mse}+S_{MMI}		
CIFAR10	0.5137	0.6005	**0.6071**
STL10	0.5306	0.5989	**0.6038**
	L_{MMI}+s_{eu}		
CIFAR10	0.5303	0.6026	**0.6091**
STL10	0.5262	0.5772	**0.5816**
	L_{MMI}+S_{MMI}		
CIFAR10	0.5519	0.6472	**0.6590**
STL10	0.5426	0.6418	**0.6448**

According to Table 3, the performance of all architectures incorporating the proposed L_{MMI}+S_{MMI} learning outperforms the other combinations of learning methods. Consequently, we demonstrate that the proposed CVAE-MMI model performs better compared with the other two basic models in all combinations of learning methods. This is achieved by addressing latent space irregularities with appropriate regularization. We can also observe that the performance of LCAE is acceptable compared with the simple CAE owing to effectively compressing the data boundaries.

3.6.2 Performance Comparison against State-of-the-art methods

As a result of the conducted experiment, we observe that the performance of the proposed CVAE-MMI model based on the considered ten classes corresponding to each of the four datasets achieved a better performance in terms of the average AUC value compared with those of the considered nine state-of-the-art methods for anomaly detection (Table 4 and Fig. 4). It confirms that in detecting rating image anomalies, the proposed model achieved rather better performance compared with the advanced OCGAN model [70]. Specifically, the performance of the proposed framework is much better than those of the other state-of-the-art approaches on the complex datasets (STL-10 and IMAGENET).

Table 3.4 Performance comparison of CVAE-MMI and the state-of-the-art methods on overall classes in image datasets in terms of mean AUC

Datasets	GMM	KDE	CAE	VAE	Pixel CNN
CIFAR10	0.5875	0.6097	0.5234	0.5833	0.5506
Cifar100	0.6170	0.6456	0.4912	0.5081	0.5146
STL10	0.6156	0.5842	0.5698	0.5942	0.5002
IMAGENET	0.5706	0.5310	0.5602	0.5769	0.4911
Datasets	GAN	SCG	ANOGAN	OCGAN	Proposed
CIFAR10	0.5916	0.6172	0.6179	0.6566	**0.6590**
Cifar100	0.4774	0.4886	0.4678	0.6526	**0.7128**
STL10	0.5043	0.5155	0.5212	0.6177	**0.6448**
IMAGENET	0.5537	0.5405	0.5780	0.6225	**0.6583**

The stable performance of the CVAE-MMI framework in the case of image anomalies on the four data sets is represented in Fig. 5. We observe that the proposed framework outperforms the state-of-the-art methods by achieving the relatively stable and high average AUC values on all four datasets in terms of detecting image anomalies. Then, the performance of the proposed CVAE-MMI framework in terms of estimating anomalies based on each individual class is compared over the state-of-the-art methods on CIFAR10, CIFAR100, STL-10, and IMAGENET, as is presented in Tables 7,8,9, and 10 in Appendix B. Among the considered ten classes, the classes number four, eight, five, and three corresponding to CIFAR10, CIFAR100, STL-10, and IMAGENET, respectively, demonstrate higher AUC in the case of the proposed framework compared with the other methods, as shown in Figure 6.

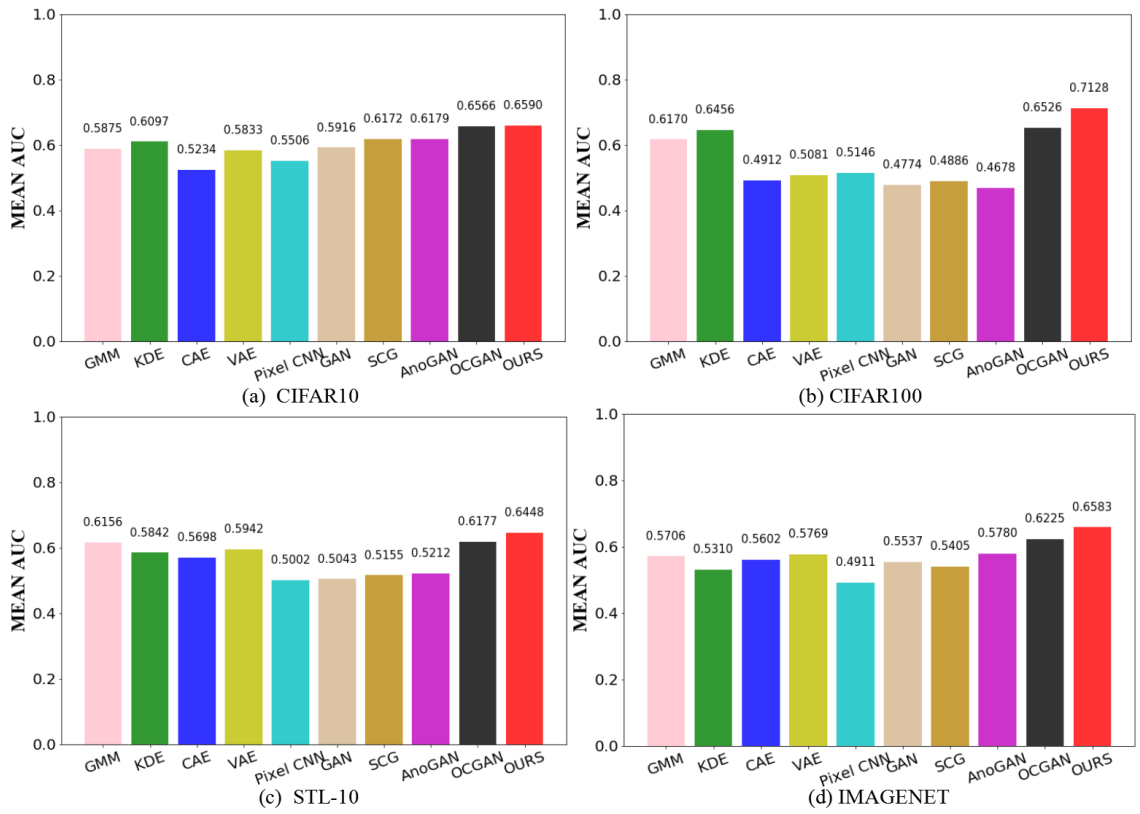
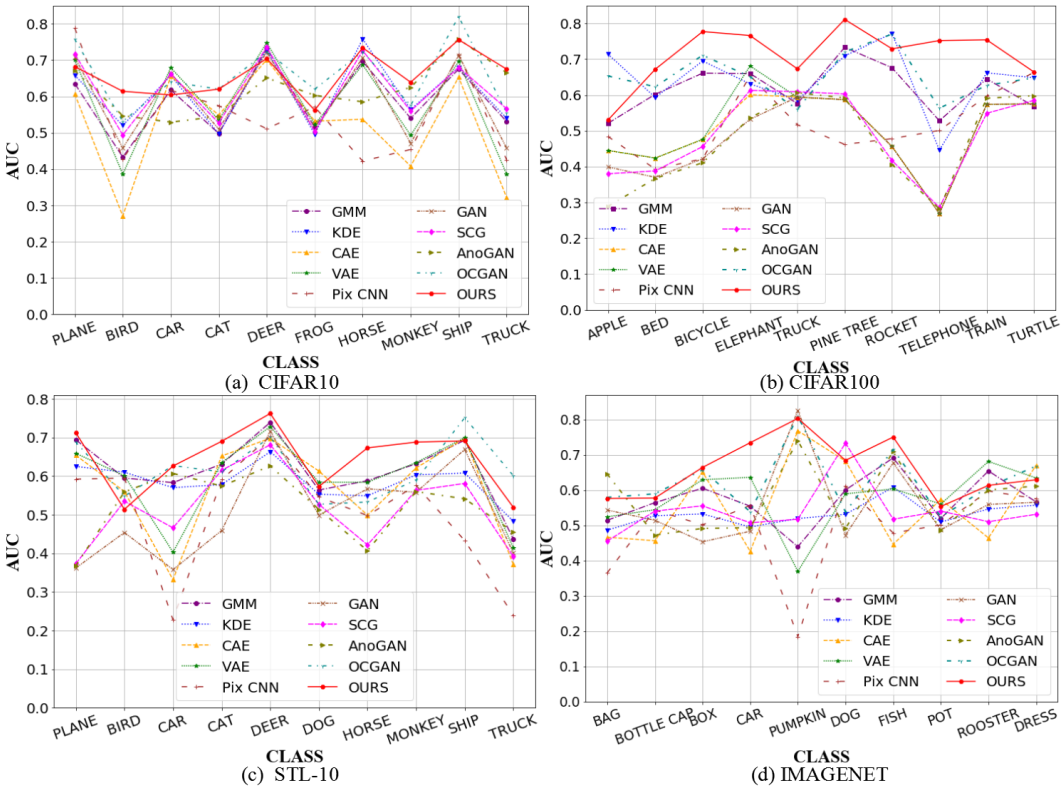


Fig. 3.4 Performance comparison of our proposed over the state-of-the-artmethods on image datasets in terms of mean AUC values.

To verify the generality and adaptability of the proposed FVAE-MII on the vector datasets, we conduct the experiments considering seven state-of-the-art methods. As represented in Table 5 and Fig. 7, the proposed model outperforms the state-of-the-art methods in the case of the KDD99 dataset. However , in the case of the Default of credit card clients dataset, the proposed framework demonstrate the performance estimate lower than those of SRO and SR methods , even though still achieving the high AUC value.

3.6.3 Convergence of the proposed architecture

Several reconstruction error detection methods [5, 4] used less (below 50 epochs) training iterations. Using a less number of iterations often results in a high AUC score; however, it may not be sufficiently stable to produce a well-reconstructed image result. Specifically, in the case of complex datasets, the autoencoder-based standard reconstruction error detection model often tends to produce an identity mapping. To overcome these problems, the proposed model learning process is



(a) CIFAR10 (b) CIFAR100

(c) STL-10 (d) IMAGENET

Fig. 3.5 Performance comparison of CVAE-MMI and the state-of-the-arts on overall classes of four data sets in terms of AUC

implemented using the large number of iterations (1,000 epochs), and therefore, produce more reliable and sound results compared with the conventional methods. To demonstrate the convergence of the proposed framework, we randomly select a class (bag) from the IMAGENET dataset and analyze the corresponding its iterative curve.

As seen in Fig 8, the learning curve of the proposed model converge and stabilize after the 700^{th} epoch. In addition, the image anomaly detection performance of the proposed network improve with an increase in the number of iterations with the high AUC value and it tends to be stable after 700 iterations according to Fig 9. Similarly, all class performance estimates on the image and vector datasets demonstrate the similar tendency in their learning convergence curves. According to Fig 10, AUC values tend to be stable after 700 and 1200 iterations corresponding to KDD99 and Default of credit card clients, respectively.

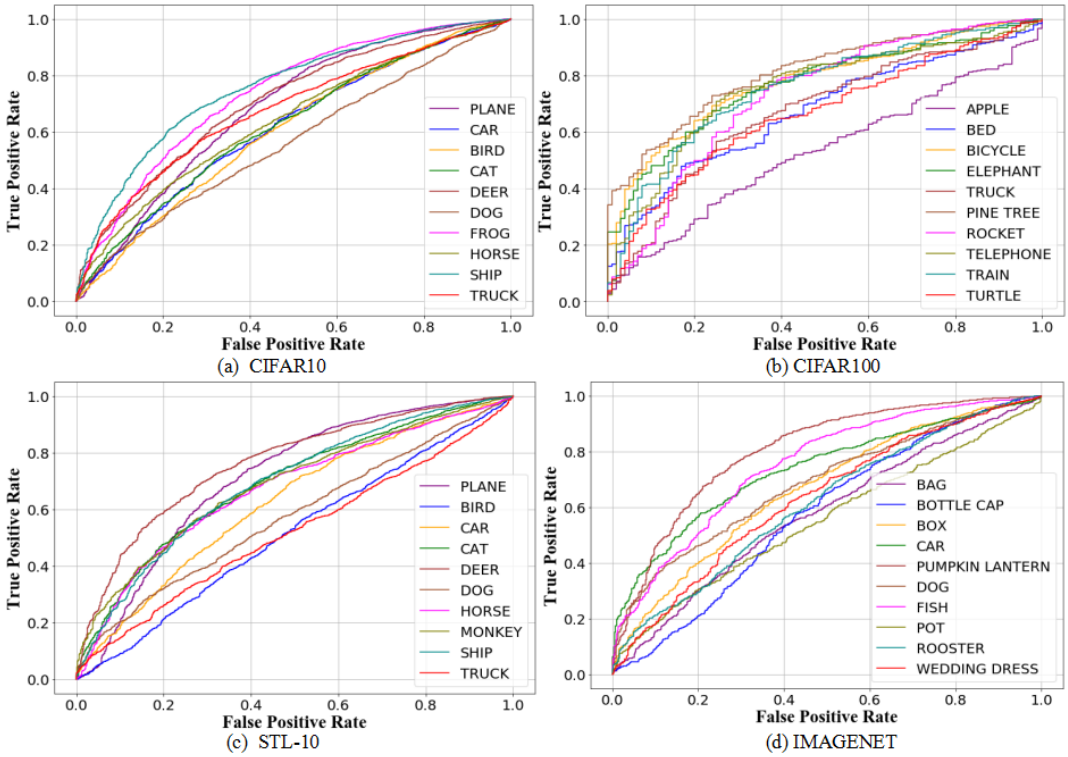


Fig. 3.6 Performance of CVAE-MMI on each class in four datasets in terms of AUC.

3.6.4 Ablation study

The effectiveness of the CVAE-MMI and FVAE-MMI frameworks components is validated by ablation experiments conducted using both image and vector datasets, respectively. We consider the following three learning settings: 1)implemented the complete three discriminator spaces model, 2) removed the latent discriminator and 3) removed both latent and hidden discriminators. The experimental results are presented in Table 6. For image datasets, it is observed that in the cases of settings 2 and 3, the anomaly detection performance deteriorate by almost 1% and more than 1%, respectively, compared to the performance of the complete model using setting 1. For vector datasets, the anomaly detection performance in settings 2 and 3 using KDD decrease by more than 4% and in the case of using Default dredit card clients datasets it decline by more than 2% compared to that of the complete model.

3.6.5 Performance Visualization on Latent Space

The latent space is visualized by using PCA that reduce the dimensionality of the latent space into two dimensions. The performance of the latent distribution of the

Table 3.5 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR10

Method	PLANE	CAR	BIRD	CAT	DEER	DOG
GMM	0.635	0.433	0.618	0.498	0.733	0.515
KDE	0.658	0.520	0.657	0.497	0.727	0.496
CAE	0.606	0.271	0.655	0.549	0.701	0.532
VAE	0.700	0.386	**0.679**	0.535	**0.748**	0.523
Pix CNN	**0.788**	0.428	0.617	0.574	0.511	0.571
GAN	0.708	0.458	0.664	0.510	0.722	0.505
SKG	0.717	0.494	0.662	0.527	0.736	0.504
AnoGAN	0.671	0.547	0.529	0.545	0.651	0.603
OCGAN	0.757	0.531	0.640	**0.620**	0.723	**0.620**
Ours	0.682	**0.614**	0.604	**0.620**	0.704	0.562

Method	FROG	HORSE	SHIP	TRUCK	MEAN
GMM	0.696	0.540	0.675	0.531	0.5874
KDE	**0.758**	0.564	0.680	0.540	0.6097
CAE	0.537	0.408	0.653	0.322	0.5234
VAE	0.687	0.493	0.696	0.386	0.5833
Pix CNN	0.422	0.454	0.715	0.426	0.5506
GAN	0.707	0.471	0.713	0.458	0.5916
SKG	0.726	0.560	0.680	0.566	0.6172
AnoGAN	0.585	0.625	0.758	0.665	0.6179
OCGAN	0.723	0.575	**0.820**	0.554	0.6566
Ours	0.734	**0.639**	0.756	**0.675**	**0.6590**

proposed model is compared with that of the SCG and OCGAN based on CIFAR100 dataset. It is found that the latent representation distribution of the proposed (Fig 11. (a), and (d)) is more efficient to distinguish between normal and abnormal samples than that of SCG model (Fig. 11(b) and (e)) and OCGAN (Fig. 11(c) and (f)). Compared with the GAN-based method, the proposed method makes the distribution more compact, which shows that the MMI can enable the model to learn more discriminative features from the raw input.

Table 3.6 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR100

Method	APPLE	BED	BICYCLE	ELEPHANT	TRUCK	PINE TREE
GMM	0.521	0.602	0.661	0.660	0.576	0.733
KDE	**0.714**	0.593	0.695	0.631	0.586	0.709
CAE	0.440	0.414	0.456	0.601	0.592	0.589
VAE	0.445	0.424	0.476	0.681	0.594	0.587
Pix CNN	0.484	0.393	0.422	0.654	0.517	0.462
GAN	0.399	0.370	0.422	0.532	0.594	0.587
SKG	0.380	0.388	0.456	0.613	0.609	0.603
AnoGAN	0.289	0.367	0.411	0.536	0.606	0.592
OCGAN	0.653	0.623	0.711	0.651	0.560	0.720
Ours	0.530	**0.672**	**0.777**	**0.766**	**0.673**	**0.811**

Method	ROCKET	TELEPHONE	TRAIN	TURTLE	MEAN
GMM	0.676	0.528	0.645	0.568	0.6170
KDE	**0.772**	0.446	0.662	0.648	0.6456
CAE	0.450	0.238	0.571	0.565	0.4912
VAE	0.456	0.269	0.574	0.575	0.5081
Pix CNN	0.419	0.688	0.444	0.663	0.5146
GAN	0.456	0.280	0.564	0.570	0.4774
SKG	0.417	0.286	0.549	0.585	0.4886
AnoGAN	0.406	0.282	0.591	0.598	0.4678
OCGAN	0.770	0.563	0.627	0.648	0.6526
Ours	0.729	**0.752**	**0.754**	**0.664**	**0.7128**

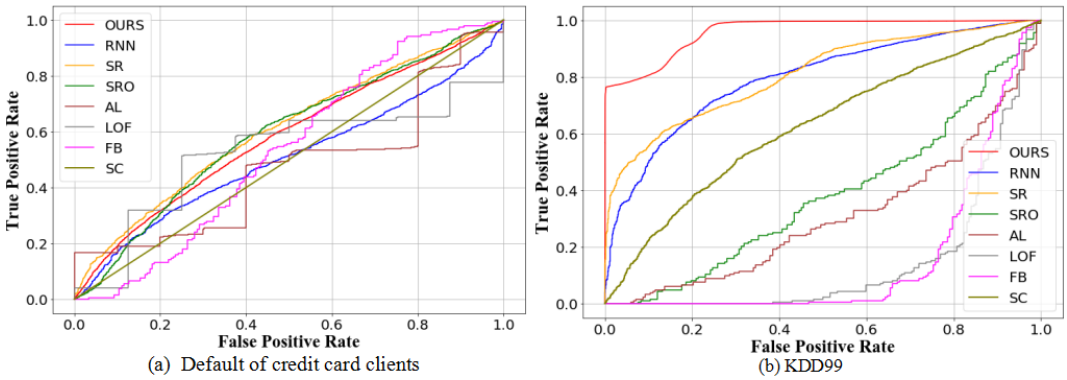


Fig. 3.7 Performance comparison of FVAE-MMI and the state-of-the-arts on overall classes of two data sets in terms of AUC

Table 3.7 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning STL-10

Method	PLANE	BIRD	CAR	CAT	DEER	DOG
GMM	0.694	0.595	0.583	0.631	0.739	0.564
KDE	0.625	**0.610**	0.570	0.578	0.663	0.553
CAE	0.654	0.560	0.332	0.652	0.698	**0.613**
VAE	0.659	0.601	0.403	0.635	0.728	0.584
Pix CNN	0.592	0.595	0.228	0.591	0.703	0.546
GAN	0.362	0.454	0.358	0.459	0.716	0.499
SKG	0.373	0.535	0.466	0.615	0.681	0.527
AnoGAN	0.368	0.559	0.607	0.574	0.626	0.514
OCGAN	0.688	0.548	**0.627**	0.611	0.701	0.527
Ours	**0.712**	0.514	0.626	**0.690**	**0.762**	0.573

Method	HORSE	MONKEY	SHIP	TRUCK	MEAN
GMM	0.588	0.632	0.693	0.437	0.6156
KDE	0.549	0.603	0.608	0.483	0.5842
CAE	0.499	0.621	0.698	0.371	0.5698
VAE	0.584	0.635	0.699	0.414	0.5942
Pix CNN	0.498	0.576	0.433	0.240	0.5002
GAN	0.567	0.558	0.669	0.401	0.5043
SKG	0.422	0.563	0.581	0.392	0.5155
AnoGAN	0.407	0.560	0.541	0.456	0.5212
OCGAN	0.533	0.590	**0.751**	**0.601**	0.6177
Ours	**0.673**	**0.688**	0.691	0.519	**0.6448**

Table 3.8 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning IMAGENET

Method	BAG	BOTTLE CAP	BOX	CAR	PUMPKIN	DOG
GMM	0.514	0.564	0.605	0.554	0.439	0.600
KDE	0.486	0.527	0.532	0.495	0.519	0.531
CAE	0.466	0.456	0.651	0.426	0.768	0.682
VAE	0.524	0.544	0.629	0.636	0.369	0.589
Pix CNN	0.365	0.543	0.501	0.556	0.184	0.611
GAN	0.543	0.472	0.453	0.483	**0.826**	0.472
SKG	0.455	0.540	0.555	0.507	0.517	**0.733**
AnoGAN	**0.645**	0.514	0.491	0.493	0.740	0.491
OCGAN	0.579	**0.589**	0.657	0.538	0.809	0.538
Ours	0.576	0.577	**0.664**	**0.734**	0.803	0.684

Method	FISH	POT	ROOSTER	DRESS	MEAN
GMM	0.691	0.519	**0.654**	0.566	0.5706
KDE	0.608	0.509	0.546	0.557	0.5310
CAE	0.446	**0.573**	0.464	0.670	0.5602
VAE	0.603	0.560	0.681	0.634	0.5769
Pix CNN	0.478	0.501	0.597	0.575	0.4911
GAN	0.677	0.487	0.559	0.565	0.5537
SKG	0.517	0.540	0.510	0.531	0.5405
AnoGAN	0.709	0.486	0.600	0.611	0.5780
OCGAN	0.715	0.524	0.608	**0.668**	0.6225
Ours	**0.750**	0.553	0.613	0.629	**0.6583**

Table 3.9 Performance comparison of FVAE-MMI and the state-of-the-art methods on overall classes of vector data sets in terms of AUC

Data sets	FB	AL	LOF	SC
KDD99	0.140	0.297	0.134	0.627
Default of credit card clients	0.535	0.484	0.524	0.496

Data sets	RNN	SRO	SR	Proposed
KDD99	0.798	0.368	0.819	**0.948**
Default of credit card clients	0.506	0.600	**0.606**	0.582

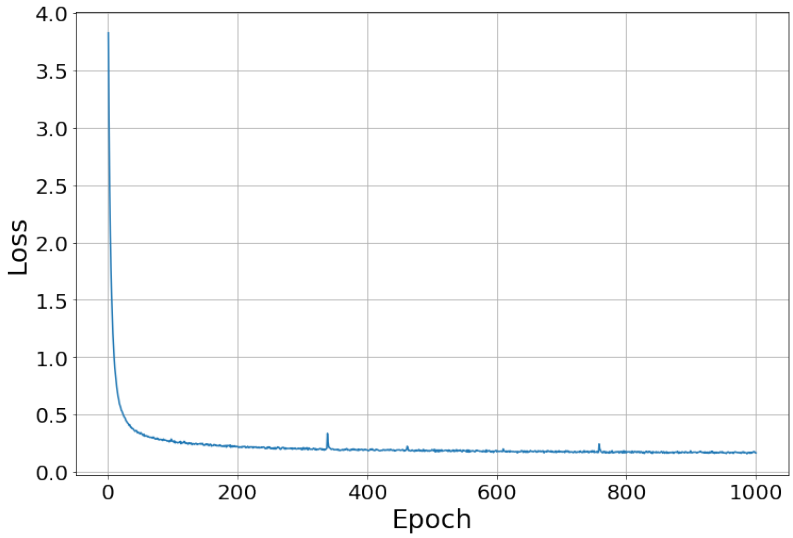


Fig. 3.8 Convergence curve of the proposed model on the 'bag' class of in IMAGENET dataset. The horizontal axis and the vertical axis represent the number of epochs and loss values respectively. It can be clearly seen that the model tends to a fixed point at the 700^{th} epoch.

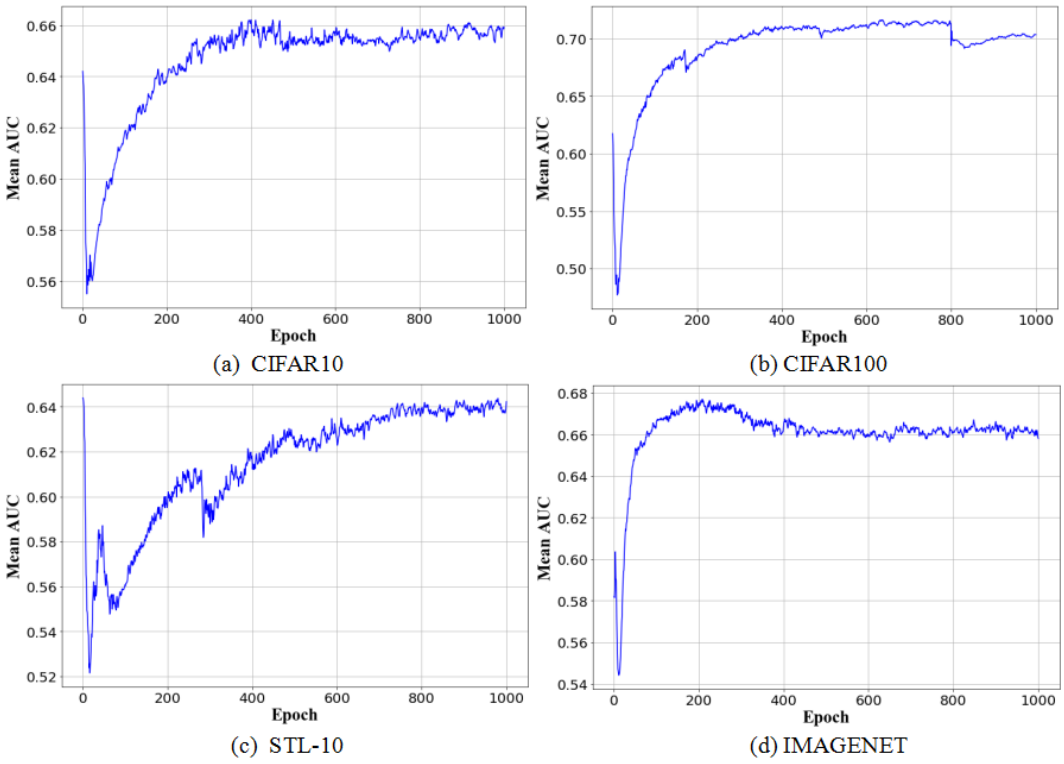
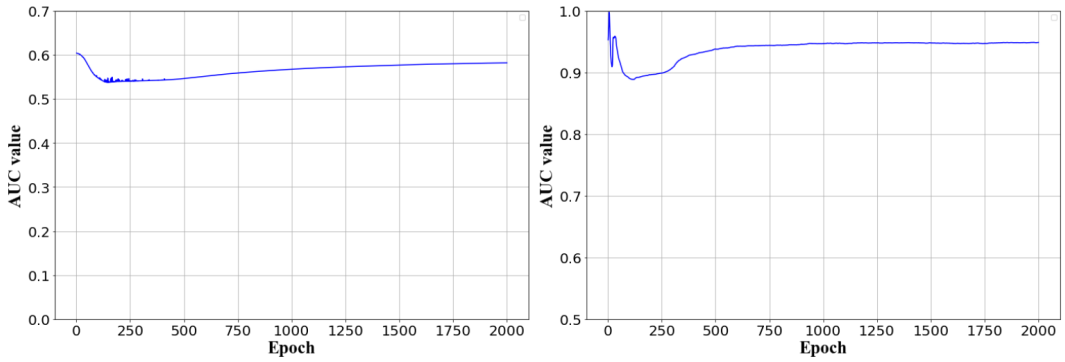


Fig. 3.9 Anomaly detection performance of CVAE-MMI increases with the increasing number of iterations interms of mean AUC on image dataset.



(a) Default of credit card clients (b) KDD99

Fig. 3.10 Anomaly detection performance of FVAE-MMI. It improves with the increasing number of iterations in terms of AUC based on the vector dataset.

Table 3.10 Performance comparison based on ablation validation in terms of average AUC

	Datasets	Setting 1	Setting 2	Setting 3
Image data	CIFAR10	**0.6590**	0.6441	0.6428
	Cifar100	**0.7128**	0.6990	0.6939
	STL10	**0.6448**	0.6271	0.6198
	IMAGENET	**0.6583**	0.6454	0.6333
vector data	KDD99	**0.948**	0.903	0.896
	Default of credit card clients	**0.582**	0.570	0.564

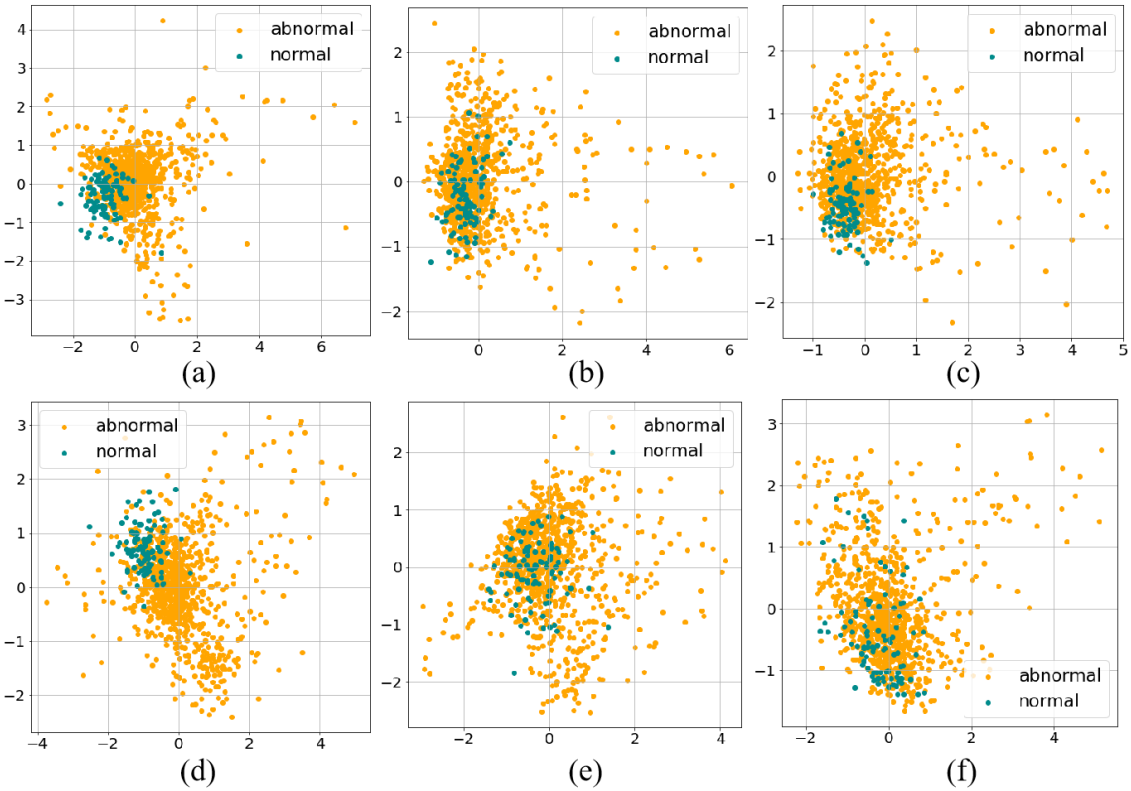


Fig. 3.11 Latent space visualization comparison of the proposed model and the state-of-the-art models(SCG and OCGAN) for anomaly detection on CIFAR100. (a), (b) and (c) shows the performance of the proposed, SCG, and OCGAN, respectively, based on the normal class 'Bicycle', and (d), (e), and (f) the performance of the proposed, SKG and OCGAN, respectively based on the normal class 'Pine tree' dataset.

Chapter 4

Mixture of experts with convolutional and variational autoencoders for anomaly detection

4.1 Introduction

In today's complex social environment, public security issues have become increasingly prominent and it is one of the hot issues in several countries. In recent years anomaly detection (AD) is gaining more and more attention in many applicative disciplines. It is widely used in video surveillance [14, 29], defect detection [50, 80], fraud detection [42], and medical imaging [38]. AD is considered as the identification of instances, events, or observations that are inconsistent with expected patterns or other instances in the dataset [23, 37, 59]. This study also follows the basic definition of the AD task by using anomaly free samples to train the model parameters θ to generate normal data distribution $p(x)$. However, the classical AD methods relied on reconstruction errors, whereas the recent studies using deep autoencoders can effectively map the data to the low-dimensional feature spaces, where data is more easily presented [90]. Hence, this study also considers the binary classification problem in the latent space that is capable of classifying the samples into normal and abnormal samples.

Convolutional autoencoder (CAE) for AD based on reconstruction error learn the features of the normal data through the convolutional neural network and calculated the Euclidean distance (reconstruction error) between the input and its generation to distinguish the normal from abnormal samples [109]. To make the latent space close

to the Gaussian distribution and achieve a better reconstruction result, convolutional variational autoencoder (CVAE) [25] is employed for anomaly detection, which results better than CAE. However, the aforementioned methods did not pay attention to the possibility of using latent space for AD. However, several recent studies have noticed the latent space importance in detecting various types of anomalies. Latent domain representation methods [49] learned a set of latent representation vectors in the source domain through examples of normal samples. Thereby, introducing the latent representation vectors from the source domain to the target domain establishes a tight boundary which can distinguish the normal from abnormal data.

Latent variable-based AD method [66] trained to encode a large amount of data into the latent space, then it detected anomalies by calculating the distance between the observation and previously defined normal cluster. Though, the above latent-based methods for AD achieved better results for vector datasets, it is not ideal for matrix datasets. Furthermore, those methods considered detection using only one latent space and did not consider the possibility of a mixture of low-dimensional nonlinear manifolds of multiple latent spaces. Linearly combining different manifolds in latent spaces can generate best latent representation. In order to solve the shortcomings of AD based on reconstruction error or latent detection, we propose a mixture of experts ensemble with two convolutional variational autoencoders and convolution (MEx-CVAEC) model. Inspired by MoE [34], we divide the dataset into two equal but non-repeating subsets as inputs of the two experts models respectively aiming to linearly combine the encoded latent spaces of the two experts. In addition, in order to enhance the model detection performance, we re-encode the output of the CVAE by generating a new data manifold for AD. Thereby each expert is developed to comprise an encoder-decoder-encoder pipeline (EDE) based on CVAE. Additionally, we use a tower structure in the mixture-of-expert model to assign a latent score to each latent representation.

The main contributions of this study are as follows:

1. Propose a novel gating mixture-of-experts based on two CVAEs and convolution (MEx-CVAEC) network which explicitly learns the underlying manifold of a group of similar objects for AD. Each expert is developed by an encoder-decoder-encoder (EDE) pipeline with VAE as a core element.

2. Introduce a convolutional autoencoder (CAE) as a gating network that learns multiple distributional information by automatically adjusting the parameter between the modelling shared information and the manifold-specific information.

3. Evaluate the efficiency of the MEx-CVAEC based on CAE gating network is compared with two other mixtures of MEx-CVAE using ED pipeline based on logistic regression (MEx-L) and based on CAE (MEx-C) gating structures for AD.

4. Compare the performance of the proposed MEx-CVAEC approach over state-of-the-art methods.

4.2 Related Works

4.2.1 Reconstruction-based methods

The reconstruction-based methods assume that outliers probably produce large reconstruction residuals. The parameters of the model for projection and reconstruction are learned from the normal samples. In traditional sparse coding [112, 20], new instances were projected into a learned subspace, and the linear combinations of the basis vectors are used to represent the normal examples. With the rise of deep learning, many researchers use autoencoders to construct a projection subspace. The CAE [109] can construct the latent space (projection space) for normal data and reconstruct the original input samples from the vectors in the latent space. Then we can use the reconstruction error to distinguish abnormal samples from the normal samples. In [90], the discriminator network of an adversarial framework was used as a novelty detector, and anomaly samples were detected by jointly using the discriminator with the reconstruction error. In OCGAN [70], taking the effectiveness and the availability of the latent space into account, the entire latent space must be used to reconstruct the normal samples. However, the studies mentioned above focused solely on the issues of reconstruction of input data. The AD methods based on reconstruction errors are not ideal for the data sets with complex backgrounds. Hence this study explored the manifold of latent space information that efficiently minimizes the false-positive rates for reliable identification of anomalies.

4.2.2 Latent space detection-based models

The latent space detection-based models mainly explore the distribution in the constructed manifold after the data are encoded. A Markov jump particle filter (MJPF) method [93] captured the low dimensional state of the video by probabilistically representing the latent space in the VAE and identified clusters with similar vectors

in the space. The latent likelihood method [10] learns the distribution of latent space by introducing an adversarial autoencoder. Given the prior distribution, adversarial learning can narrow the distance between the distribution in the latent space and the prior distribution. Compared with normal data, the likelihood of abnormal data became very small, and therefore, anomalies could be detected by comparing the estimated likelihood.

Based on [10], a dual encoder composed of a graphics encoder and a feature encoder [24] was used to encode the features and generate the correlation of the samples into a low-dimensional latent space. Then the decoder was used to reconstruct the data. Finally, a separate estimation network was used as a Gaussian mixture model to estimate the density of the latent embeddings, which led to detect the anomalies by finding the likelihood of the distribution of the observed samples from normal samples. It is motivated in this study by using the idea of a linear combination of different manifolds corresponding to the different latent space selected by the gating network helps to enable better mapping of high-dimensional data to the low-dimensional space.

4.2.3 Mixture-of-experts model for AD

Several researchers have been paying attention to AD using a mixture of expert models. The combined models [64], includes two k-nearest neighbors (k-NN), random forest (RF), joint probability approaches (RF + JP), local correlation integral (LoCI), and learned probability distribution (LPD) into a mixture-of-experts model. Though the above mentioned model used the advantages of the multiple different sub-models and performed well on vector datasets, it is not suitable on tensor datasets for AD. Different from this, our proposed approach maps the different domain data to the latent space through the EDE pipeline and thus utilizes the full use of the representation of the latent space of each model for anomalies.

4.3 Proposed Mixture of Experts network

We proposed a MEx-CVAEC netwrok model for AD composed of two convolutional VAEs and convolutional neural networks ensemble with a combination of reconstruction error and latent information distributions (Fig. 1). The variational autoencoder is used as a core element in the encoder-decoder-encoder (EDE) structure. Each CVAE and convolution framework works as an expert with exactly similar

procedural structure. The expert structures jointly improve the detection effect by utilizing the efficiency of the MoE model for highly relevant information and a convolutional autoencoder is used to train the gating network, as shown in Fig 2(a). Expert structures proposed in this study effectively combine the advantages of reconstruction and latent space learning by utilizing the multiple latent spaces (Fig. 2b).

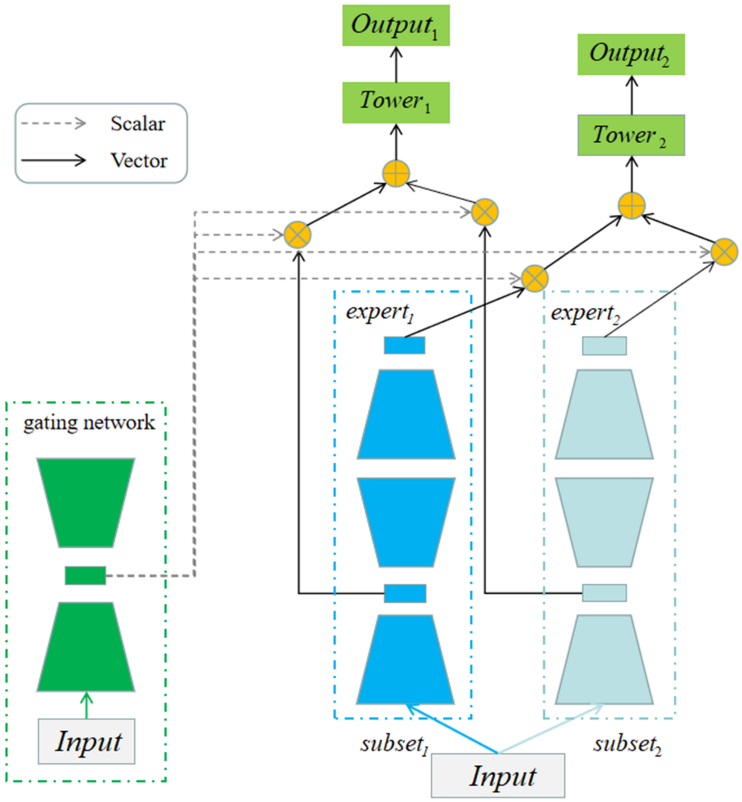


Fig. 4.1 Proposed mixture of convolutional variational autoencoder structure for anomaly detection

4.3.1 Constructing an Experts Network Structure

The proposed mixture-of-experts composes of two experts named as $expert_1$ and $expert_2$ respectively. Here we intuitively analyze the structure of experts. Each expert in our model consists of an encoder-decoder-encoder (EDE) pipeline. Each element of the expert is represented as $encoder_1$, $decoder$, $encoder_2$ as shown in Fig.2 (b). In the structure from $encoder_1$ to $decoder$ part ($encoder_1 - decoder$) of EDE, a 3-layer convolutional variational autoencoder (CVAE) is used. After CVAE (the ED part), a 3-layer convolutinal neuron networks (CNN) is connected, which consists of 3

convolutional layers, 3 pooling layers and a fully connected layer. The details of MEx-CVAEC structure are presented in $expert_1$ and $expert_2$ columns of Table 1.

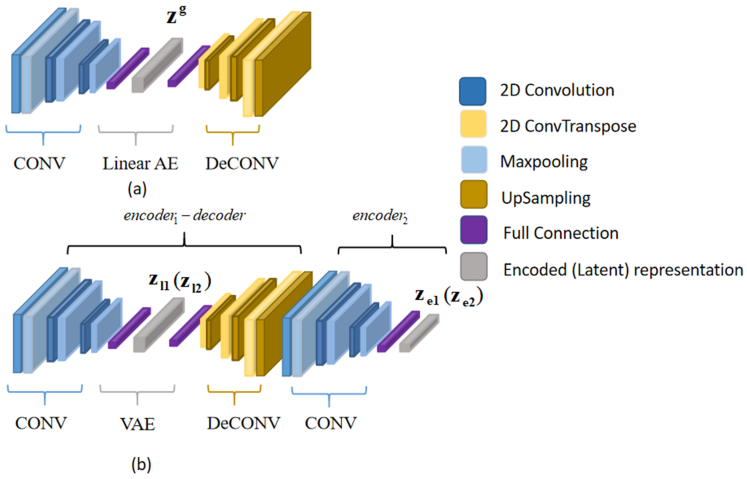


Fig. 4.2 Proposed gating network and experts structure.(a) Convolutional autoencoder gating network. (b) Convoltional variational autoencoder and convolution showing encode-decoder-encoder pipeline in experts structure. $\mathbf{z^g}$ represents the latent representation of $gate_{AE}$. $\mathbf{z_{l1}}$ and $\mathbf{z_{l2}}$ are latent representation corresponding to $expert_1$ and $expert_2$, and $\mathbf{z_{e1}}$ and $\mathbf{z_{e2}}$ is latent representation corresponding to $expert_1$ and $expert_2$.

Let us consider x is a sample from the normal dataset X as original inputs, and $x \in X$. $\mathbf{z}$ is the latent representation by encoding x, $\mathbf{z} \in Z$. The variable y is the output of CVAE and $y \in Y$. The parameters $\widehat{\mathbf{y}}$ and $\widehat{\mathbf{x}}$ are the input and the output of the VAE, respectively. Then the loss function of the $expert_1$-$decoder$ is defined as:

$$\ell_1 = \lambda_1 \left\{ \frac{1}{n} \sum_{x \in X} (x - y)^2 \right\} + \lambda_2 \left\{ \frac{1}{n} \sum_{x \in X} (\widehat{\mathbf{x}} - \widehat{\mathbf{y}})^2 \right\} + \lambda_3 D_{KL}(p(z|x)|q(z)|), \lambda_1, \lambda_2 \in [0, 1] \tag{4.1}$$

where $p(\mathbf{z}|x)$ represents the probability density function (PDF) of the latent representation distribution generated by x and D_{KL} is the Kullback-Leibler divergence distribution. The parameter $q(\mathbf{z})$ is defined as the prior Gaussion distribution.

In the expert structure, the $encoder_2$ can generate its output close to the latent space of the $encoder_1 - decoder$ structure. Let us consider the output of the $encoder_2$ is

z' and $z' \in Z'$. Then the loss function of the $encoder_2$ is defined as

$$\ell_2 = \lambda_4 \left\{ \frac{1}{n} \sum_{0}^{n} (\mathbf{z} - \mathbf{z}')^2 \right\}. \tag{4.2}$$

Therefore, the loss function of the $expert_1$ is the summation of the distribution of both $expert_1$-*decode* and $encoder_2$ in $expert_1$ which is defied as

$$\ell_{ex1} = \ell_1 + \ell_2 \tag{4.3}$$

where λ_1, λ_2, λ_3, and λ_4 are the weighting parameters that adjust the impact of individual loss on the overall objective function. Similarly, we obtained the objective function ℓ_{ex2} of $expert_2$.

4.3.2 Gating network based on Convolutional Autoencoder

We introduced a gating network $gate_{AE}$ for learning multiple distributional information as shown in Fig. 2(a). We used a convolutional autoencoder as the gating network, which is inspired by [51]. The representation of the latent space is used as the output of the gating network. The gating network utilizes input functions and output sigmoid gates to assemble experts with different weights, and thereby different learning can make full use of experts. Gating networks can learn to "select" a subset of experts to use conditioned on the input sample. This is desirable for a flexible parameter sharing in the multi-learning situation. As a special case, if only one expert with the highest gate score is selected, the gating network actually linearly separates the input space into n regions in which each region corresponds to an expert. This strategy forces the model to learn the diverse relationships in a sophisticated way by deciding how the separations resulted by different gates can overlap with each other. This method facilitates the complete model for knowledge transfer that benefits multi-information by learning as much information as possible. We randomly select the same number of samples as each expert from the original dataset which is used as input X_s into the gating network. The objective function is defined as follows:

$$\ell_{gate} = \lambda_{g1} \left\{ \frac{1}{n} \sum_{x_g \in X_g} (x_g - y_g)^2 \right\} + \lambda_{g2} \left\{ \frac{1}{n} \sum_{x_g \in X_g} (\widehat{\mathbf{x}}_g - \widehat{\mathbf{y}}_g)^2 \right\} \tag{4.4}$$
$$\lambda_{g1}, \lambda_{g2} \in [0, 1]$$

Where x_s is a sample from the normal dataset, $x_g \in X$. The variable y_g is the output of the gating network and $y_g \in Y_g$. The parameters $\widehat{\mathbf{x}}_g$ and $\widehat{\mathbf{y}}_g$ are the input and the output of the linear AE, respectively .

4.3.3 Training Gated Mixture of Experts structure

Our proposed MEx-CVAEC model fused with experts and gating neural network. It can capture the differences of latent spaces by encoding data and modeling the data as a mixture of low-dimensional nonlinear manifolds. Through the gating network, experts are assembled with different weights and thereby, different manifolds can make full use of experts. Then, the results of the gathered experts are transferred to the manifold-specific tower network (we will explain later in this section). In this way, the gating network is applied to learn different mixed modes of expert assembly, aiming to capture manifold relationships. Most importantly, we added a separate gating network g from $gate_{AE}$ according to multi-learning. Therefore, we defined the latent space output of $gate_{AE}$ as $\mathbf{z^g} = [z_1^g, z_2^g, z_3^g, z_4^g]$ and we can get

$$\begin{aligned} g(x_g)_j &= s_j(z_j^g) \\ s.t. \quad \sum_{j=1}^{n} g(x_g)_j &= 1 \end{aligned} \tag{4.5}$$

where x_g is the input of $gate_{AE}$, $x_g \in X_g$, $n = 4$ and z_j^g is a scalar, s_j represents the output of j^{th} neuron in sigmoid layer. For $expert_1$ and $expert_2$, the manifold of latent space is connected by the gating network $gate_{AE}$. The latent representations in $encoder_2 - decoder$ are named as $\mathbf{z_{l1}}$ in $expert_1$ and $\mathbf{z_{l2}}$ in $expert_2$, respectively. The encoded representation of $encoder_2$ in $expert_1$ and $expert_2$ is denoted as $\mathbf{z_{e1}}$ and $\mathbf{z_{e2}}$, respectively. In the $Tower_1$ and $Tower_2$, a one-neuron fully connected layer with sigmoid activation function is used. The function of $Tower_1$ and $Tower_2$ is represented as f_{t1} and f_{t2}, respectively. The output y_1 of $Tower_1$ and y_2 of $Tower_2$ is defined as:

$$\begin{aligned} y_1(x) &= f_{t1}(g(x)_1\mathbf{z_{l1}} + g(x)_2\mathbf{z_{l2}}) \\ y_2(x) &= f_{t2}(g(x)_3\mathbf{z_{e1}} + g(x)_4\mathbf{z_{e2}}) \end{aligned} \tag{4.6}$$

where $\mathbf{z_{l1}}$, $\mathbf{z_{l2}}$, $\mathbf{z_{e1}}$, and $\mathbf{z_{e2}}$ are represented as vectors. We label each normal data with '0' and it is used to train the model parameters as follows:

$$\ell_3 = \lambda_{t1}\frac{1}{n}\sum_{x\in X}(y_1(x)-0)^2 + \lambda_{t2}\frac{1}{n}\sum_{x\in X}(y_2(x)-0)^2, \quad \lambda_{t1},\lambda_{t2}\in[0,1] \tag{4.7}$$

Finally, the loss function of the proposed MEx-CVAEC n is defined as

$$\ell = \lambda_{ex1}\ell_{ex1} + \lambda_{ex2}\ell_{ex2} + \lambda_T\ell_3 + \lambda_g\ell_{gate}, \quad \lambda_{ex1},\lambda_{ex2},\lambda_T,\lambda_g\in[0,1] \tag{4.8}$$

4.4 Testing Anomaly Score

In the test phase, the model calculates the anomaly score of each test sample x', $x' \in X_{test}$, which is used as the input of $expert_1$, $expert_2$, and $gate_{AE}$. Let us consider calculating the anomaly score in $expert_1$. It is defined based on the reconstruction error $S_1(x')$ of the convolutional autoencoder and the reconstruction error $S_2(x')$ of the VAE as

$$S'(x') = \lambda S_1(x') + (1-\lambda)S_2(x') \tag{4.9}$$

λ is the weighting parameter controlling the relative importance of the score functions. The reconstruction error $S_1(x')$ between the input x' and it approximation y' by the $encoder_1 - decoder$ section is defined as

$$S_1(\mathbf{x}) = \|x' - y'\|^2 \tag{4.10}$$

Similary, the reconstruction error $S_2(x')$ between the feature vector $\widehat{\mathbf{x}'}$ and its approximation $\widehat{\mathbf{y}'}$ by the VAE is defined as

$$S_2(\mathbf{x}) = \|\widehat{\mathbf{x}'} - \widehat{\mathbf{y}'}\|^2 \tag{4.11}$$

Similarly, we can calculate the anomaly score S''(x) of $expert_2$. We can obtain the latent score $y_1(x')$ and $y_2(x')$ through the tower network according to Eq. (6) as follows

$$S_{g1}(x') = y_1(x')^2, S_{g2}(x') = y_2(x')^2 \tag{4.12}$$

Finally, the overall anomaly score is defined as follows

$$S(x) = \omega_1 S'(x) + \omega_2 S''(x) + \omega_3 S_{g1} + \omega_4 S_{g2} \tag{4.13}$$

where ω_1, ω_2, ω_3, and ω_4 are the weighting parameters adjusting the impact of individual score to the overall score function. In order to evaluate the impact of the overall anomaly detection performance, the anomaly scores are normalized. At first, the anomaly scores $S = \{S(x')|x' \in X_{test}\}$ for all test samples X_{test} are calculated and the maximum max(S) and the minimum min(S) of the anomaly scores are obtained. Then the anomaly score $S(x')$ for the new samples is normalized as

$$s' = \frac{S(x') - min(S)}{max(S) - min(S)} \tag{4.14}$$

The use of Eq. (14) ultimately yields an anomaly $S'(s' \in S')$ for the final evaluation of the test set X_{test}.

4.5 Experimental Setup

4.5.1 Dataset

The effectiveness of the proposed method is evaluated using three publicly available multi-class object recognition datasets (CIFAR10, CIFAR100, and STL10). The CIFAR10 [48] and STL10 [19] comprise the images corresponding to ten different classes. In the Cifar100 dataset [48], we used 10 different classes for the experiments. In order to simulate a AD setting, the network is trained using only normal class. The union of the rest of the classes are used as abnormal samples in testing the network. We divide the training set of each class into two equal and non-repeating subsets, which are fed into the input of $expert_1$ and $expert_2$. In addition, we randomly sample the entire dataset for $gate_{AE}$, to make the input number of experts the same as the input number of the gating network $gate_{AE}$.

4.5.2 Experimental Evaluation and Performance Measure

The efficiency of the proposed MEx-CVAEC incorporating EDE pipeline based on CAE gating network is compared with two other mixtures of Mo-CVAE using ED pipeline based on CAE (MEx-C) and based on logistic regression (MEx-L) gating networks (Figures 3a and 3b, respectively). Both MEx-C and MEx-L use only one

tower. The MEx-L model use convolution kernel in the logistic regression gating network. Furthermore, the performance of our proposed method in detecting

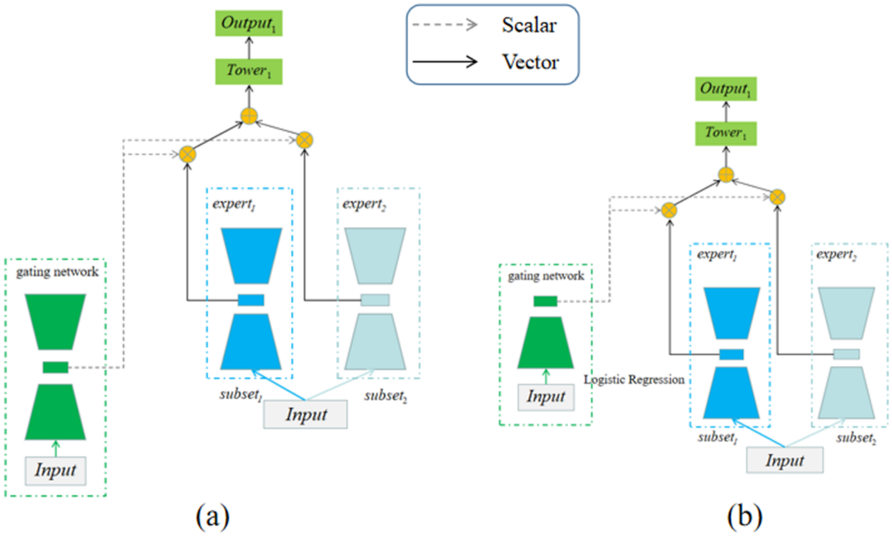


Fig. 4.3 Mixture of encoder-decoder models on ED pipline (a) convolutional variational autoencoder based on convolutional autoencoder gating network (b) convolutional variational autoencoder based on logistic regression gating network

anomalies is evaluated through the comparison with nine state-of-the-art methods, including one-class Gaussian mixture model (GMM) [105], kernel density estimation (KDE) [52], convolutional autoencoder (CAE) [17], VAE [44], pixel CNN decoders (Pixel CNN) [99], GAN [90], skip-connection Ganomaly (SCG) [5] , anomaly detection with generative adversarial networks (AnoGAN) [90] and one-class GAN (OCGAN) [70]. The comparison standard of our proposed and state-of-the-arts is measured using the area under the curve (AUC) of the receiver operating characteristic curve method.

4.5.3 Parameter Settings

All experiments carried out in this study are implemented on Python 3.6 and Tensorflow 1.9. RMSProp optimizer is used to train the network parameters. The learning rate is 0.00005 for CIFAR10 and STL10, 0.0001 for CIFAR100. The parameters *decay* is 0.9, *momentum* is 0.9. We set 300, 300 and 500 epochs corresponding to CIFAR10, STL10, and CIFAR100.

4.6 Experimental Results

4.6.1 Performance Comparison based on Encoder-Decoder Mixture Models

We conducted experiments and compared our proposed MEx-CVAEC with MEx-C and MEx-L to verify the effectiveness of the elements used in the architecture. To realize the linear association of multiple potential spaces through the expert structure it is obvious that the gating network should be capable to choose more appropriate characteristics for a specific task. It is realized that the performance of the proposed expert structures with EDE pipeline is better than the expert structures with ED pipeline models as presented in Table 2. Furthermore, different from the structure of MEx-L and MEx-C, the additional tower element ($Tower_2$) in our proposed approach pushes the network to improve the performance. Additionally, the encoded representation of $encoder_2$ plays a major role by re-encoding the generated data into a new feature space, which is different from the latent space features of the ED pipeline. Thereby the linear combination of encoded representations ($encoder_2$) gathered from different experts is highly potential in detecting anomalies. Whereas, the hybrid model of conventional experts (MEx-L) developed using only one training teamwork is not efficient to classify and thus it produced poor ability in predicting the characteristic relationship of the data. However, we can observe that the CAE gating network in MEx-C is capable of selecting better matching correlation parameters to characterize the data than the MEx-L structure.

4.6.2 Performance Comparison based on State-of-the-arts

The performance of the proposed method is compared over nine state-of-the-art methods. The proposed model using ten classes on three different datasets shows higher performance than that of state-of-the-art anomaly detection methods, with the highest average AUC value as shown in Table 3. The performance of the proposed model is better than that of the classical CAE or VAE. In addition, compared with the recently developed OCGAN model [70], MEx-CVAEC model is more efficient in rating the anomalies.

The stable performance of the MEx-CVAEC framework on image anomalies datasets is represented in Figure 4. It is observed that the stability in detecting anomalies of the proposed model on three different data sets is above average. Furthermore, we can observe that the proposed model outperforms the stare-of-

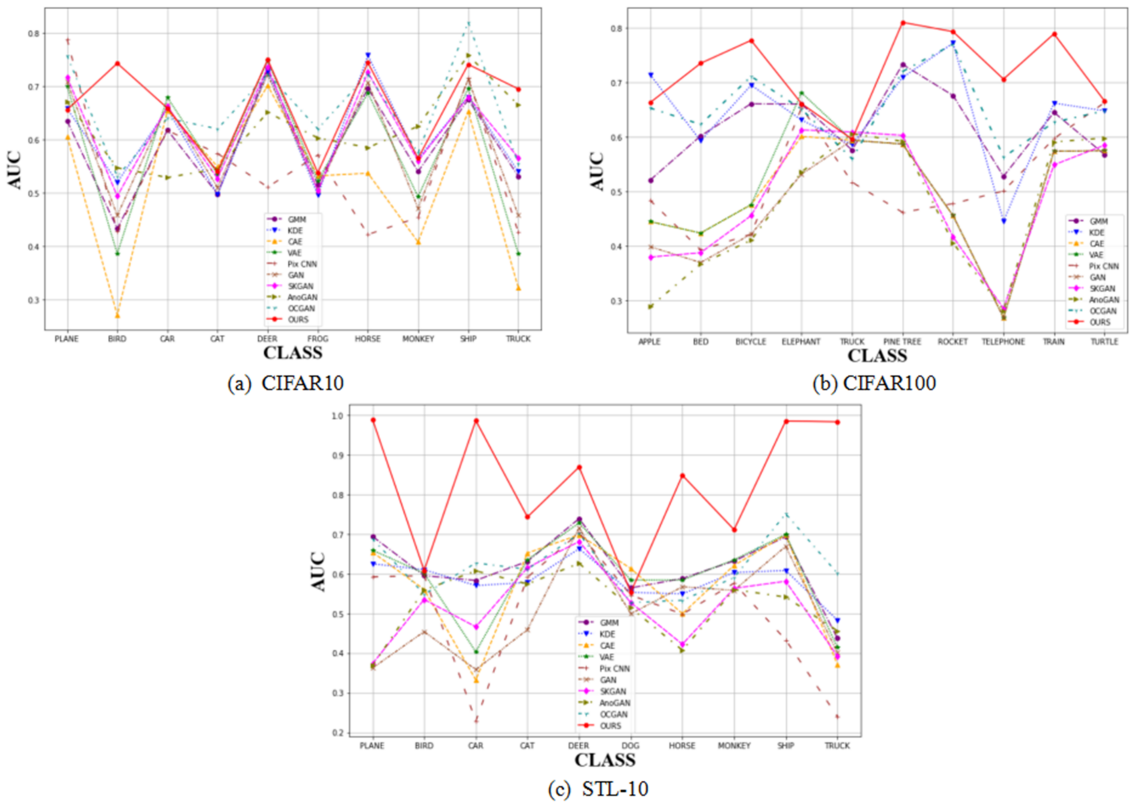


Fig. 4.4 Performance of the proposed approach and the state-of-the-art methods on overall classes in three datasets in terms of AUC

the-art methods by showing the relatively stable and high average AUC values on all data sets in detecting image anomalies. In fact, both GMM and KDE used distribution information for detection. GMM used the strategy of using the probability observations that belonged to the normal data distribution for distinguishing normal and abnormal data, and KDE used the density of distribution for AD.

The detection results of the techniques based on the data distribution information (GMM and KDE) outperform those of techniques based on the reconstruction errors (CAE, VAE, Pix CNN and GAN). But the classic AD methods, GMM and KDE Although AnoGAN and SCG utilized the discriminator in the detection part, the overall effect of AD performance is not high, because of using the encoded reconstruction data as an input into the discriminator for detection. Furthermore, the background of the complex datasets is highly ambiguous, which greatly increases the difficulty of detection. However, AnoGAN and SCG methods did not consider the impact of the latent space on reconstruction of the complex data and hence they are not effective on complex image anomalies. The OCGAN used in the comparison

experiments is based on the reconstruction errors considering all the potential latent spaces for generating only the normal data, and thus there is no space to improve the AD. Based on the different drawbacks of the above methods, the proposed method combined both the reconstruction error and latent space detection. The experimental results proved the superiority of the proposed method by using the distribution information of the latent space features and latent score to analyze the distribution of the data.

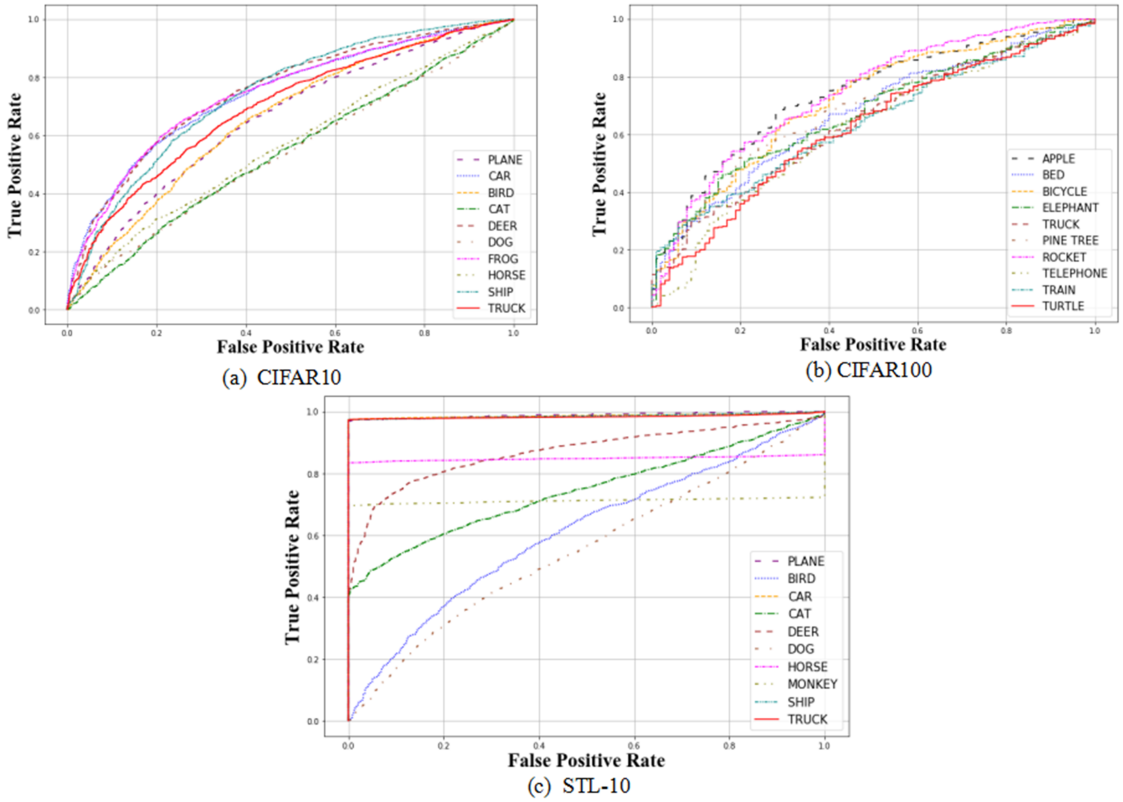


Fig. 4.5 Performance of the proposed approach on each class in three datasets in terms of AUC

Furthermore, the performance of the proposed MEx-CVAEC framework in estimating anomalies based on each individual class is compared over the state-of-the-art methods on CIFAR10, CIFAR100 and STL-10 as presented in Tables.7, 8, and 9. Among the ten class used in this study, the classes number three, five, and eight corresponding to CIFAR10, CIFAR100 and STL-10 datasets, respectively of our proposed model demonstrated higher AUC in the case of the proposed model compared with the other methods as shown in Figure 5.

4.6.3 Performance Visualization on Latent Space

This section demonstrates the conducive construction of latent space in training phase to distinguish normal from abnormal data. The latent space is visualized using PCA that reduced the dimensionality of the latent space into two dimensions for visualization. The performance of the latent representation distribution of the proposed model is compared with that of the proposed model with only one expert structure with ED pipeline. It is found that the latent representation distribution after linear combination of our model (Fig. 4b and 4d) is efficient to distinguish between normal and abnormal samples than that of the model with only one expert structure (Fig. 4). The single expert structure on ED pipline without the mixture of nonlinear manifolds in the latent space cannot be able to distinguish the anomalies from the normal data (Fig. 4a and 4c).

Table 4.1 Proposed Gated Mixture of Experts for Anomaly Detection

$Expert_1$ and $Expert_2$	Gating network
conv1(channel:32, filter:5)	conv1(channel:32, filter:5)
batch normalization	batch normalization
max pooling(2*2)	max pooling(2*2)
conv2(channel:64, filter:5)	conv2(channel:64, filter:5)
batch normalization	batch normalization
conv3(channel:128, filter:5)	conv3(channel:128, filter:5)
batch normalization	batch normalization
max pooling(2*2)	max pooling(2*2)
fully-connected(500)	fully-connected(4)
fully-connected(2048)	fully-connected(2048)
deconv1(channel:64, filter:5)	deconv1(channel:64, filter:5)
batch normalization	batch normalization
up sampling(2*2)	up sampling(2*2)
deconv2(channel:32, filter:5)	deconv2(channel:32, filter:5)
batch normalization	batch normalization
up sampling(2*2)	up sampling(2*2)
deconv3(channel:3, filter:5)	deconv3(channel:3, filter:5)
batch normalization	batch normalization
up sampling(2*2)	up sampling(2*2)
conv1(channel:32, filter:5)	
batch normalization	
max pooling(2*2)	
conv2(channel:64, filter:5)	
batch normalization	
max pooling(2*2)	
conv3(channel:128, filter:5)	
batch normalization	
max pooling(2*2)	
fully-connected(500)	
output (sigmoid)	

[1] CIFAR10 dataset is considered as an example

Table 4.2 Performance comparison of the proposed over encoder-decoder mixture models in terms of average AUC

Datasets	Proposed	MEx-C	NEx-L
CIFAR10	**0.6631**	0.6507	0.6428
CIFAR100	**0.6740**	0.6536	0.6480
STL10	**0.8275**	0.7981	0.7852

Table 4.3 Performance comparison of the proposed network and the state-of-the-art methods on overall class in terms of mean AUC based on three different datasets

Data sets	GMM	KDE	CAE	VAE	Pixel CNN
CIFAR10	0.5875	0.6097	0.5234	0.5833	0.5506
Cifar100	0.6170	0.6454	0.4912	0.5081	0.5146
STL10	0.6156	0.5842	0.5698	0.5942	0.5002
Data sets	GAN	SCG	ANOGAN	OCGAN	Proposed
CIFAR10	0.5916	0.6172	0.6179	0.6566	**0.6631**
Cifar100	0.4774	0.4886	0.4678	0.6526	**0.6740**
STL10	0.5043	0.5155	0.5212	0.6177	**0.8275**

Table 4.4 Performance comparison of the proposed network and the state-of-the-art methods on each individual class in terms of AUC based concerning CIFAR10

Method	PLANE	CAR	BIRD	CAT	DEER	DOG
GMM	0.635	0.433	0.618	0.498	0.733	0.515
KDE	0.658	0.520	0.657	0.497	0.727	0.496
CAE	0.606	0.271	0.655	0.549	0.701	0.532
VAE	0.700	0.386	**0.679**	0.535	0.748	0.523
Pix CNN	**0.788**	0.428	0.617	0.574	0.511	0.571
GAN	0.708	0.458	0.664	0.510	0.722	0.505
SKG	0.717	0.494	0.662	0.527	0.736	0.504
AnoGAN	0.671	0.547	0.529	0.545	0.651	0.603
OCGAN	0.757	0.531	0.640	**0.620**	0.723	**0.620**
proposed	0.656	**0.743**	0.659	0.540	**0.750**	0.538

Method	FROG	HORSE	SHIP	TRUCK	MEAN
GMM	0.696	0.540	0.675	0.531	0.5875
KDE	**0.758**	0.564	0.680	0.540	0.6097
CAE	0.537	0.408	0.653	0.322	0.5234
VAE	0.687	0.493	0.696	0.386	0.5833
Pix CNN	0.422	0.454	0.715	0.426	0.5506
GAN	0.707	0.471	0.713	0.458	0.5916
SKG	0.726	0.560	0.680	0.566	0.6172
AnoGAN	0.585	**0.625**	0.758	0.665	0.6179
OCGAN	0.723	0.575	**0.820**	0.554	0.6566
Proposed	0.744	0.565	0.741	**0.695**	**0.6631**

Table 4.5 Performance comparison of the proposed network and the state-of-the-art methods on each individual class in terms of AUC based concerning CIFAR100

Method	APPLE	BED	BICYCL	ELEPHAN	TRUCK	TREE
GMM	0.521	0.602	0.661	0.660	0.576	0.733
KDE	**0.714**	0.593	0.695	0.631	0.586	0.709
CAE	0.440	0.414	0.452	0.601	0.592	0.589
VAE	0.445	0.424	0.476	0.681	0.594	0.587
Pix CNN	0.484	0.393	0.422	0.654	0.517	0.462
GAN	0.399	0.370	0.422	0.532	0.594	0.587
SKG	0.380	0.388	0.456	0.613	0.609	0.603
AnoGAN	0.289	0.367	0.411	0.536	0.606	0.592
OCGAN	0.653	0.623	0.711	0.651	0.560	0.720
Proposed	**0.739**	**0.671**	**0.715**	0.675	**0.640**	0.679
Method	ROCKET	TELEPHON	TRAIN	TURTLE	MEAN	
GMM	0.676	0.528	0.645	0.568	0.6170	
KDE	**0.772**	0.446	0.662	0.648	0.6456	
CAE	0.450	0.238	0.571	0.565	0.4912	
VAE	0.456	0.269	0.574	0.575	0.5081	
Pix CNN	0.419	0.688	0.444	0.663	0.5146	
GAN	0.456	0.280	0.564	0.570	0.4774	
SKG	0.417	0.286	0.549	0.585	0.4886	
AnoGAN	0.406	0.282	0.591	0.598	0.4678	
OCGAN	0.770	0.563	0.627	0.648	0.6526	
Proposed	0.742	**0.619**	0.639	0.621	**0.6740**	

Table 4.6 Performance comparison of the proposed network and the state-of-the-art methods on each individual class in terms of AUC concerning STL-10

Method	PLANE	BIRD	CAR	CAT	DEER	DOG
GMM	0.694	0.595	0.583	0.631	0.739	0.564
KDE	0.625	**0.610**	0.570	0.578	0.663	0.553
CAE	0.654	0.560	0.332	0.652	0.698	**0.613**
VAE	0.659	0.601	0.403	0.635	0.728	0.584
Pix CNN	0.592	0.595	0.228	0.591	0.703	0.546
GAN	0.362	0.454	0.358	0.459	0.716	0.499
SKG	0.373	0.535	0.466	0.615	0.681	0.527
AnoGAN	0.368	0.559	0.607	0.574	0.626	0.514
OCGAN	0.688	0.548	0.627	0.611	0.701	0.527
Proposed	**0.989**	0.608	**0.986**	**0.743**	**0.869**	0.553

Method	HORSE	MONKEY	SHIP	TRUCK	MEAN
GMM	0.588	0.632	0.693	0.437	0.6156
KDE	0.549	0.603	0.608	0.483	0.5842
CAE	0.499	0.621	0.698	0.371	0.5698
VAE	0.584	0.635	0.699	0.414	0.5942
Pix CNN	0.498	0.576	0.433	0.240	0.5002
GAN	0.567	0.558	0.669	0.401	0.5043
SKG	0.422	0.563	0.581	0.392	0.5155
AnoGAN	0.407	0.560	0.541	0.456	0.5212
OCGAN	0.533	0.590	0.751	0.601	0.6177
Proposed	**0.848**	**0.711**	**0.985**	**0.983**	**0.8275**

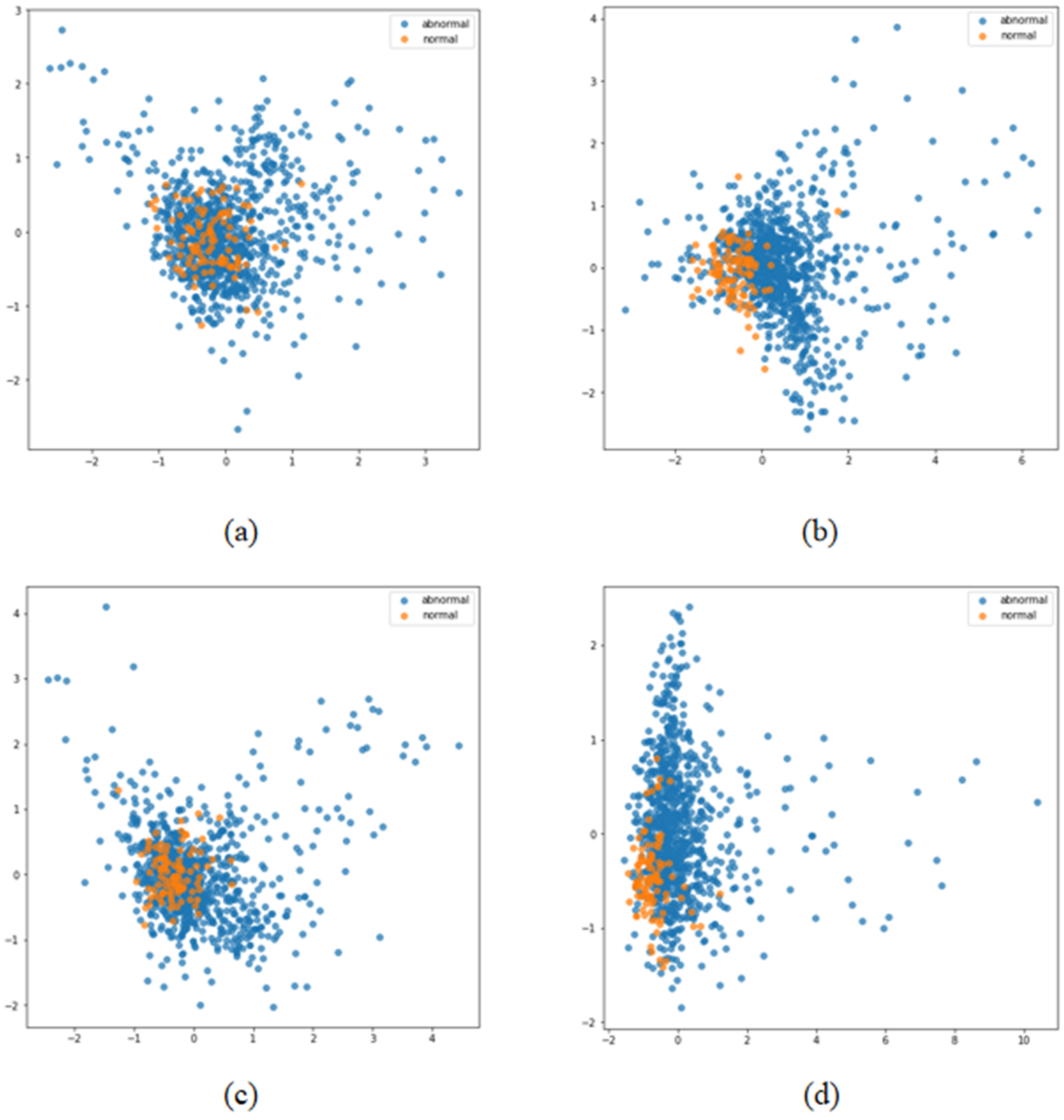


Fig. 4.6 Latent space visualization comparison of the proposed model over the proposed model with only one expert structure. (a) and (b) shows proposed model with single expert and two expert structures, respectively using CIFAR10 (class DEER), (c) and (d) shows proposed model with single expert and two expert structures, respectively using STL10 (class DOG) dataset

Chapter 5

Autoencoder framework based on orthogonal projection constraints improves anomalies detection

5.1 Introduction

With the development of information science, information security has become increasingly important. As an integral part of information security, anomaly detection (AD) has been considerably investigated by researchers. AD is also considered a classification problem [23]. AD is widely used in video surveillance [94, 14, 29, 55], security applications [27, 72, 67, 71], defect detection [50], and medical imaging [90, 84]. In this study, we follow the basic rules for AD tasks because the abnormal samples are insufficient or even missing compared to normal samples [85]. Hence, in the training phase, only normal samples can be used to train the model parameters, θ, and generate normal data distribution, $p(x)$. In the testing phase, both anomaly-free and abnormal data are included to test the performance of the model using abnormal score. The autoencoders [88] can map the original input to low-dimensional feature spaces in order to analyze the distribution of the low-dimensional feature space where the data can be better represented.

Autoencoder methods have been widely used in AD studies. Classical reconstruction error-based convolutional autoencoders (CAEs) are used to learn the features of normal data using a convolutional neural network (CNN); they are also used to find the Euclidean distance (reconstruction error) to distinguish between normal and abnormal samples [17]. However, in classical CAEs, the latent space helps in

reconstruction. As a direct extension, one-class novelty detection using generative adversarial networks (GANs) with constrained latent representations (OCGAN) outperforms several conventional AD methods on benchmark dataset classification tasks [70]. OCGAN controls the latent space and use it entirely to train normal samples. Thus, there is no latent space for abnormal samples and it is bound to cause a large residual error.

In recent years, several studies have focused on the detection of latent space to project data [49, 66, 74, 87, 10, 24]. The latent information and reconstructions are well performed in these methods. However, the orthogonal projection in the autoencoders, which is the basic projection mechanism, has not been explicitly considered. The orthogonal projection mechanism is well applied in principal component analysis (PCA) [102] for dimensionality reduction. PCA is a classic method for low-dimensional data; in this method, the first few principal components can equivalently be defined as the directions that maximize the variance of projected data. The variance in first few principal components is significantly large; therefore they require a considerable change to be detected. The last few principle components are considered as the sum of variations in the residual vector, which is very small; thus, any minor change is observable, which is a good property for AD. However, for high-dimensional data, such as image data, includes a large amount of complex contents, which deteriorates the performance of PCA detection not ideal. Inspired by PCA, we propose the introduction of orthogonal projection constrain (OPC) in the deep learning model, aiming to use the discriminative feature vectors of normal data to create the orthogonal complementary subspaces in an end-to-end manner via back-propagation (BP). A convolutional network in the proposed was utilized to extract the discriminative feature vector for normal space (NS). Additionally, previous works related to AD focused on the reconstruction error-based detection with deep learning models, especially autoencoder-based models; they did not consider examining the orthogonal complementary subspaces to detect anomalies. In our work, we attempted to explore the orthogonal complementary subspaces in the deep learning model; moreover, multi-space is more conducive to feature representation than single space, which is sensitive to normal and abnormal data.

Alternatively , to determine a null subspace (kernel subspace) and a range subspace, which are orthogonal to each other and efficiently capture the important discriminative features of subspace information and reconstructions. Most importantly, the null subspace information has been ignored. Thereby, in this study, we

focused on examining a range subspace and a null subspace for robust anomalies separation.

The range subspace and null subspace are two subspaces of the original space decomposed by their direct sum. The two subspaces are orthogonal to each other and disjoint. The range space contains the main features of data, while the null subspace contains information that is not related to the input data, such as noise. However, in previous works, the detection of latent spaces has not been ideal [91, 10, 24]. This is because some unimportant features of data, such as background or noise, greatly influences the detection performance. Thus, in this study, we explored the projection of data into the range subspace and null subspace for AD.

In comparison to a single subspace, the exploration of double subspace increases the detection effect. The manifolds in the subspaces contain multiple features with different contributions to AD. The representation of normal and abnormal data in the two subspaces is different and discriminative. To comprehensively exploit the manifolds in two subspaces for robust AD, in this study, we propose an autoencoder framework based on an OPC learning method. The primary objective involves the calculation of projected norms in the range and null subspace. By constraining the projection operator to approximate the orthogonal projections, the model can be trained in an end-to-end manner via BP.

In the proposed autoencoder framework model, the features are firstly extracted from the raw input and projected into the subspaces by projection operator. For image datasets, we propose a convolutional autoencoder based on orthogonal projection constraint (OPC-CAE). The space after the CNN is called as the full signal space. The data in the full signal space are projected into the range and null subspace by the projection operator. The range subspace and null subspace are named as normal space (NS) and abnormal space (AS), respectively. The NS contains the main information related to normal data; the information not related to the normal data is projected to AS.

To ensure disjoint that it is disjoint between the two subspaces, OPC are adopted for the projection operator. Using OPC, we can obtain two mutually orthogonal subspaces. Orthogonality is responsible for the disjoint between two subspaces, implying that there is no common non-zero element between them. To the best of our knowledge, this is the first study that introduces an AEs-based model with two orthogonal subspaces for AD.

However, only anomaly-free data can be utilized in the training period. Therefore, it is difficult to train the AS directly. To solve this problem, normal data are utilized

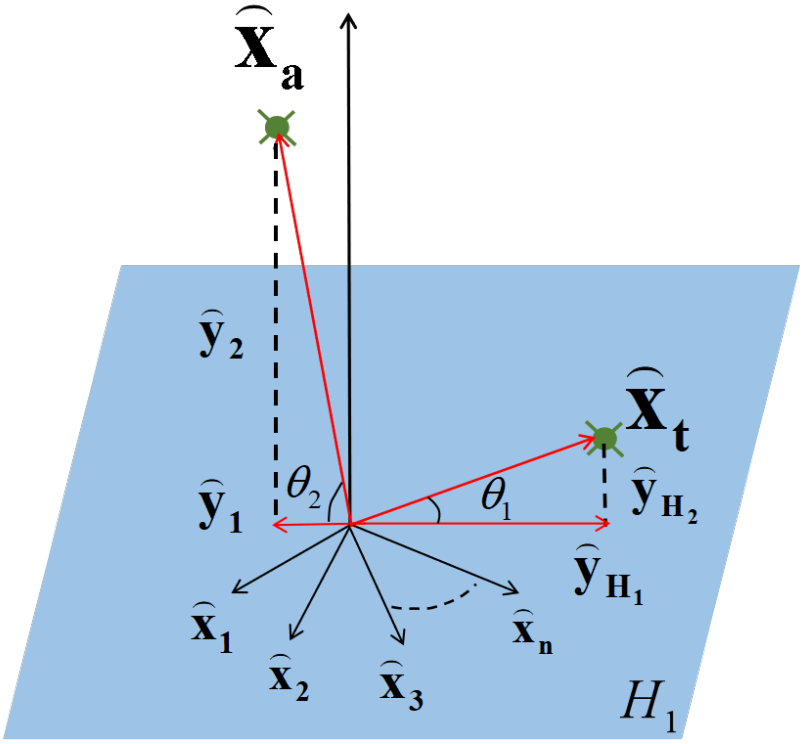


Fig. 5.1 Illustration of orthogonal projection from the full signal space of dimension N. Subspace H_1 is built by a set of normal data $\widehat{\mathbf{x}}_1$, $\widehat{\mathbf{x}}_2$,...,$\widehat{\mathbf{x}}_n$. The $\widehat{\mathbf{x}}_t$ and $\widehat{\mathbf{x}}_a$ denote a new normal observation and abnormal observation, respectively. Therefore, we get the following inequality: $\|\widehat{\mathbf{y}}_{H_1}\| > \|\widehat{\mathbf{y}}_1\|$, $\|\widehat{\mathbf{y}}_{H_2}\| < \|\widehat{\mathbf{y}}_2\|$, $\theta_1 < \theta_2$.

to train the NS and the AS is considered as orthogonal complementary space of NS. We introduce OPC to project the main information of normal data into NS, which can ensure the main information of abnormal space will be projected into the AS owing to the larger angle between abnormal data and normal space. Thereby, to maximize the projection value in the NS in the training period, we used a convolutional decoder (DeCONV in Fig. 2) to reconstruct the data from the NS for training. Anomalies were detected in the test phase. As shown in Fig. 1, $\widehat{\mathbf{x}}_t$ and $\widehat{\mathbf{x}}_a$ denote a new normal observation and abnormal observation, respectively. The set of normal data $\widehat{\mathbf{x}}_1$, $\widehat{\mathbf{x}}_2$,...,$\widehat{\mathbf{x}}_n$ is the basis for the normal subspace H_1. The angle θ_1 between $\widehat{\mathbf{x}}_t$ and H_1 is smaller and angle θ_2 between $\widehat{\mathbf{x}}_a$ and H_1 is larger; therefore, the new normal and abnormal data get a larger projection norm in NS and AS, respectively. Additionally, we propose an anomaly score based on the orthogonal subspace score (OSS) and reconstruction error score (RES). It allowed achieving the additional supervision power for generalizing AD based on the vector datasets.

To confirm the applicability of the proposed autoencoder framework in vector data input environment, we replace the convolutional layers with a fully connected (FC) network in the encoder–decoder, which is called as OPC-FAE. Consequently, we set an appropriate threshold to distinguish between normal and abnormal samples.

The main contributions of this study are as follows:

1. This study constraints the projection operator of subspaces to approximate orthogonal projections.
2. We propose a novel autoencoder framework based on the OPC model that explicitly learns the manifolds for NS and AS.
3. The generalization ability of the proposed method is proved with image and vector datasets by designing OPC-CAE and OPC-FAE, respectively.
4. A new anomaly score implemented that combines the OSS and the RES.
5. The effectiveness of the propose method using the combined OSS and RES anomaly score for anomalies is experimentally evaluated by comparing it with the state-of-the-art methods.

5.2 Related work

5.2.1 Reconstruction-based methods

Many works lean toward learning a parametric projection and reconstruction of normal data, assuming outliers will yield higher residuals. In traditional sparse coding [112, 20], new observations are encoded by sparse coding to train codes to represent normal examples, while the codes cannot be used to represent abnormal data. PCA-based AD approach [28] has been used successfully for monitoring production systems in hospitals, PCA helps in developing a reference model using the normal data collected from the normal process. In this approach, the last few principal components were found to be more sensitive to anomalies in experiments. However, for high-dimensional data, the performance of PCA is not ideal. With the rise of deep learning, several studies have focused on learning the latent space through an autoencoder-like structure. Convolutional autoencoder (CAE) [86] was introduced to learn the latent space of normal data based on the reconstructions according to the latent space and the normal, and abnormal samples were distinguished using the reconstruction error.

An adversarial framework was used in which a discriminator network was used as the novelty detector and the anomalies were jointly detected by the discriminator and reconstruction error [90]. Adversarial autoencoder neural networks (AAEs) [91] discriminate anomalies through adversarial learning to discover the differences between the distribution of data in the latent space and prior distribution. A deep autoencoder equipped with a parameter density estimator was used to learn that learns the probability distribution of its latent representation through an autoregressive process [1]. In this study, we combined the OSS and RES for AD.

5.2.2 Latent space detection

A complementary line of research investigates different strategies to explore the distributions in low-dimensional feature spaces. The low-dimensional manifold of encoded data is considered to be a useful feature for detecting normal and abnormal distributions. A deep neural network using a multivariate Gaussian fully convolution adversarial autoencoder was proposed to make the latent distribution to approximate a Gaussian distribution, while the latent representations of anomalies have no constraints. AD is performed by detecting the difference between the distributions of normal and abnormal samples [55].

B. Zong et al. [114] combined the latent space of autoencoders with a Gaussian mixture model and optimized the parameters of the deep autoencoder and mixture model in an end-to-end manner. They used a separate estimation network to facilitate parameter learning for the mixture model via density estimation of the latent space. A novel sparse representation framework was proposed to learn dictionaries based on the latent space of variational autoencoder [97]. Thus, the anomaly samples can be detected by measuring the degree of dictionary reconstruction of the latent variable.

The biggest obstacle in previous latent-based methods [55, 114, 97] was the limited performance of input data in a single latent space. It cannot be possible to very well presented the manifolds in a single latent space to each categories of the datasets. Therefore, we intend to train the NS and AS by approximating the orthogonal projection to obtain discriminative features from two subspaces for the improved representation of each complicated category of samples.

5.2.3 Orthogonal projection mechanism in deep learning

In deep learning, the orthogonal projection mechanism has garnered the significant attention from several researchers, especially in the study of clustering. To prevent all points from being grouped into the same cluster in network maps, dual autoencoders [104] introduces orthogonal projection mechanism to make the output orthonormal in expectation for deep spectral clustering. In SpectralNet [92], the loss function can be minimized by mapping all points to the same output vectors. To prevent this, the last layer in network enforces the orthogonality constraint . Orthogonal projection mechanism was also introduced into lifelong learning. In [79], the most informative features for the first task were preserved and more flexibility was provided to other features to improve the performance on the second task using the orthogonal projection mechanism.

5.3 Methods

5.3.1 Constructing an autoencoder network structure based on OPC

In this study, we propose a novel autoencoder framework that combines with the NS and the AS based on OPC. The features of AE combined with NS and AS achieve additional supervision power over the original training objective function of AE models. The adaptability of the proposed architecture was evaluated on the image and vector datasets. Concerning image AD, a CNN is utilized to extract the features; additionaly , fully connected (FC) layer (linear autoencoder) is used as the core element in the encoder-decoder structure of the proposed OPC-CAE framework, as shown in Fig. 2. Concerning vector-based AD, FC layers are embedded in the encoder-decoder structure of the OPC-FAE framework, as shown in Fig. 3. In this study, we obtain two subspaces, namely NS and AS, from the full signal space by constraining the orthogonal projection operator. After training, the two subspaces are approximately orthogonal to each other. We use only anomaly-free data in training to obtain the two subspaces; thus, the NS contains main information of normal data, while the AS is expected to contain imformation not related to normal data. In the training phase, the input samples are reconstructed from NS to ensure that the main information of the normal data is projected into the NS. Through training, the normal data can be projected by the projection operator of the NS, while

abnormal data which has different information from normal data, is expected to be projected into the AS.

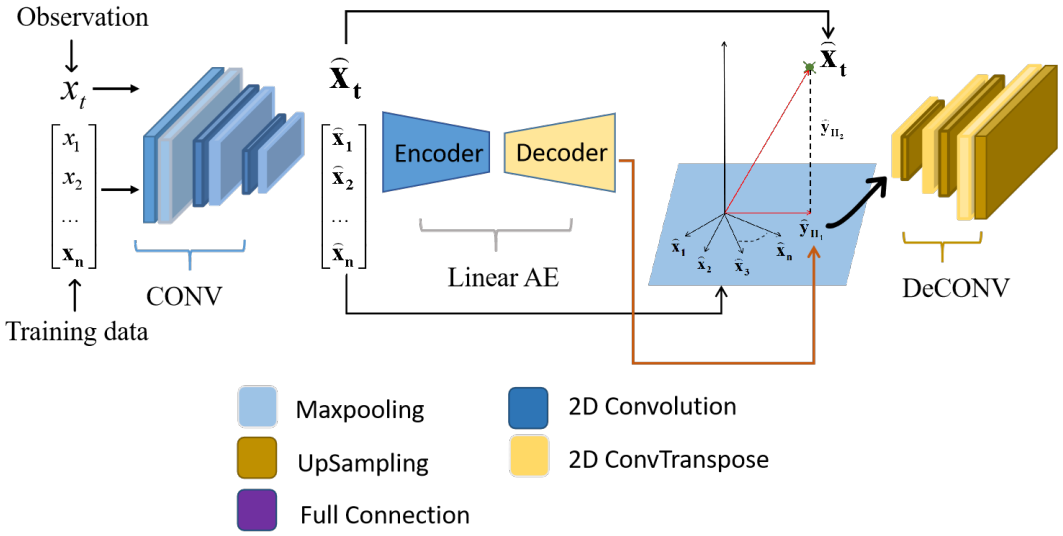


Fig. 5.2 Proposed orthogonal projection constraint-based convolutional autoencoder for anomaly detection.

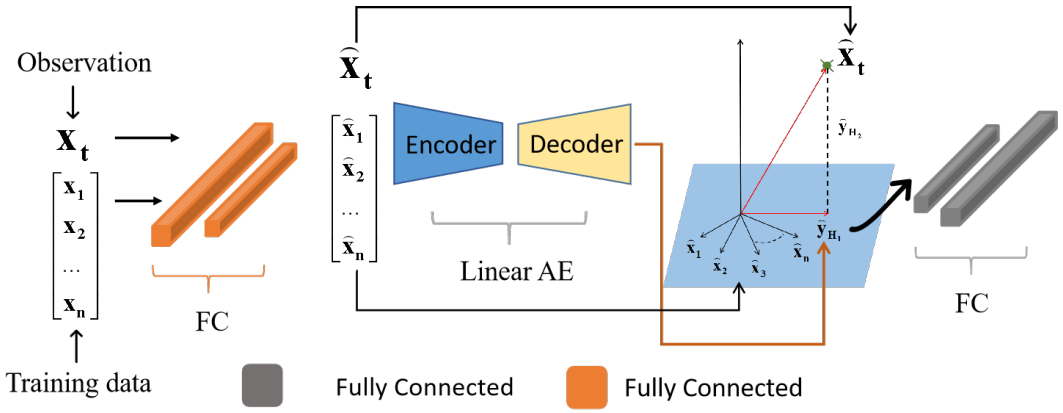


Fig. 5.3 Proposed orthogonal projection constraint based fully connected autoencoder for anomaly detection

Let us consider that x_t is a sample from the normal dataset X, which includes n samples. Assume that $\widehat{\mathbf{x}}_t$ is the feature vector of normal data after being encoded by a convolutional network or a fully connected network and $\widehat{\mathbf{x}}_t \in \widehat{\mathbf{X}}$; thus $\widehat{\mathbf{x}}_t$ is in the full signal space H. The parameter $\widehat{\mathbf{x}}_t$ is mapped into the latent space and constructed by a linear autoencoder as follows:

$$\widehat{\mathbf{y}}_{H_1} = W_e W_d \widehat{\mathbf{x}}_t \tag{5.1}$$

W_e and W_d are the weights in the linear encoder and decoder, respectively. In our target, $\widehat{\mathbf{y}}_{H_1}$ is in NS, which is with the basis vectors $\widehat{\mathbf{x}}_1$, $\widehat{\mathbf{x}}_2$,...,$\widehat{\mathbf{x}}_n$, which are feature vectors of normal data encoded by a convolutional network or a fully connected network. Let H_1 be the NS of R^n with basis vectors $\widehat{\mathbf{x}}_1$, $\widehat{\mathbf{x}}_2$,...,$\widehat{\mathbf{x}}_n$, and let $\widehat{\mathbf{X}}$ be a matrix with columns $\widehat{\mathbf{x}}_1$, $\widehat{\mathbf{x}}_2$,...,$\widehat{\mathbf{x}}_n$. P_{H_1} is defined as the projection operator on H_1 based on $\widehat{\mathbf{X}}$. Therefore, the projection operator can be defined as,

$$P_{H_1} = \widehat{\mathbf{X}}(\widehat{\mathbf{X}}^T\widehat{\mathbf{X}})^{-1}\widehat{\mathbf{X}}^T \tag{5.2}$$

However, to ensure that the entire structure can be learned an end-to-end manner, we use the W_eW_d as the projection operator to approximate P_{H_1}. Thus, NS and AS can be trained in an end-to-end manner via BP. The linear autoencoder is a fully connected autoencoder with a linear activation function. It is used to make the weight of linear autoencoder approximate to the projection operator of normal data. Therefore, the output of the linear autoencoder is the vector projected into the NS. So, we can use the following formulas to regularize weights in the linear autoencoder:

$$OPC = \|W_eW_d - \widehat{\mathbf{X}}(\widehat{\mathbf{X}}^T\widehat{\mathbf{X}})^{-1}\widehat{\mathbf{X}}^T\|_2^2 \tag{5.3}$$

This ensures that the intersection of the two subspaces in the autoencoder is zero using OPC. implying that the two subspaces are non-overlapping or disjoint , that is $H_1 \bigcap H_2 = \emptyset$. Suppose the subspaces H_1 and H_2 are the NS and AS of P_{H_1}, respectively. We have a direct sum $H = H_1 \oplus H_2$. Every vector $\widehat{\mathbf{x}}_t \in H$ may be decomposed uniquely as $\widehat{\mathbf{x}}_t = \widehat{\mathbf{y}}_{H_1} + \widehat{\mathbf{y}}_{H_2}$ with the following formulas:

$$\widehat{\mathbf{y}}_{H_1} = P\widehat{\mathbf{x}}_t \tag{5.4}$$

$$\widehat{\mathbf{y}}_{H_2} = (I - P)\widehat{\mathbf{x}}_t \tag{5.5}$$

where $P = W_eW_d$, $\widehat{\mathbf{y}}_{H_1} \in H_1$, $\widehat{\mathbf{y}}_{H_1} \in H_2$, and I is the identity operator. Hence, it can be guaranteed that H_1 and H_2 are orthogonal to each other. Therefore, $\widehat{\mathbf{y}}_{H_1}$ and $\widehat{\mathbf{y}}_{H_2}$ are also orthogonal to each other, as shown in Fig. 4.

The loss function is defined as follows:

$$\ell = \frac{1}{n}\sum_{x\in X}\|x - y\|_2^2 + \lambda_1\frac{1}{n}\sum_{0}^{n}\|\widehat{\mathbf{x}} - \widehat{\mathbf{y}}_{H_1}\|_2^2 + \lambda_2\|W_eW_d - \widehat{\mathbf{X}}(\widehat{\mathbf{X}}^T\widehat{\mathbf{X}})^{-1}\widehat{\mathbf{X}}^T\|_2^2 \tag{5.6}$$

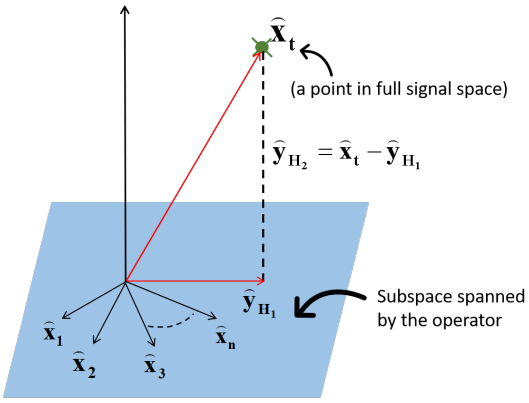


Fig. 5.4 Illustration of orthogonal projection from the full signal space of dimension N. The subspace created by the m vector base (assumed horizontal) is used to find the best approximation (orthogonal projection) of $\widehat{\mathbf{x}}_t$ in this space.

where $\widehat{\mathbf{y}}_{H_1} = W_d\mathbf{z}$, $\mathbf{z}$ is the latent representation of latent space in linear autoencoder. The first term is the mean squared error (MSE) between the input x and its approximation y. The second term is the MSE between the feature vector in the full signal space $\widehat{\mathbf{x}}$ and its approximation $\widehat{\mathbf{y}}_{H_1}$ by the linear autoencoder. The parameters λ_1 and λ_2 are the weighting parameters that adjust the impact of individual loss on the overall objective function. The proposed OPC-based training objective function is summarized in Algorithm 1.

5.3.2 Structure implementation

The core of the proposed OPC-CAE structure is a linear autoencoder. We use a CNN before and after the linear autoencoder to learn the data features as shown, in Fig. 2. The details of the proposed model are summarized in Table 1. We added a three-layer CNN before and after the linear AE to improve the learning ability of the model, respectively. To verify the generality of the model, vector datasets are used to evaluate the performance of the proposed model. The vector-based model replaces the convolution neural network with two-layered FC network as shown in Fig. 3. In the encoder, the number of neurons in the first layer is 20, and that in the second layer is 10. ReLU is used as an activation function in the encoder and decoder. The activation of the last layer introduces the sigmoid function.

Algorithm 2: Training objective of the proposed model

Input: Set of training data x, $x \in X$, iteration size N, weighting parameters λ_1 and λ_2.

Output: $Y, \widehat{\mathbf{y}}_{H_1}, \widehat{\mathbf{y}}_{H_2}$

Define $\widehat{\mathbf{x}}$ in full signal space form x;

Process from x to $\widehat{\mathbf{x}}$ is defined as $En(x)$, thus $\widehat{\mathbf{x}} = En(x)$;

Similarly, we can get $y = De(\widehat{\mathbf{y}})$, $y \in Y$;

Training:;

initialization;

for *iteration* $1 \rightarrow N$ **do**

  Take a mini-batch of M $[x_1, ..., x_m]$ as input;

  $\widehat{\mathbf{x}}_i = En(x_i)$, $x_i \in M$, $\widehat{\mathbf{x}}_i \in \widehat{\mathbf{X}}$;

  $\mathbf{z_i} = W_e\widehat{\mathbf{x}}_i$, $\widehat{\mathbf{y}}^i_{H_1} = W_d\mathbf{z_i}$, $y_i = De(\widehat{\mathbf{y}}^i_{H_1})$;

  if *Reconstruction loss update* **then**

    $L_1 \leftarrow \frac{1}{n}\sum_{x_i \in M} \|x_i - y_i\|_2^2$;

    $L_2 \leftarrow \frac{1}{n}\sum_{x_i \in M} \|\widehat{\mathbf{x}}_i - \widehat{\mathbf{y}}^i_{H_1}\|_2^2$;

    $L \leftarrow L_1 + \lambda_1 L_2$;

    Back-propagate L to change L_1 and L_2;

  end

  if *Orthogonal projection constraints update* **then**

    $OPC \leftarrow \|W_e W_d - \widehat{\mathbf{X}}(\widehat{\mathbf{X}}^T\widehat{\mathbf{X}})^{-1}\widehat{\mathbf{X}}^T\|_2^2$;

    $Cons \leftarrow \lambda_2 OPC$

    Back-propagate $Cons$ to change $Cons_{ab}$,

  end

  Optimized $\ell = L + Cons$

end

Table 5.1 Model structure for anomaly detection

OPC-CAE	OPC-FAE
conv1(channel:32, filter:5)	fc(neuron:20,activation:ReLU)
batch normalization	fc(neuron:10,activation:ReLU)
max pooling(2*2)	fc(NS)(neuron:8),fc(NS)(neuron:8)
conv2(channel:64, filter:5)	fc(neuron:10,activation:ReLU)
batch normalization	fc(neuron:20,,activation:ReLU)
conv3(channel:128, filter:5)	fc(neuron:43,activation:Sigmoid)
batch normalization	
max pooling(2*2)	
fc(NS)(neuron:500) fc(AS)(neuron:500)	
fc(neuron:2048)	
deconv1(channel:64, filter:5)	
batch normalization	
up sampling(2*2)	
deconv2(channel:32, filter:5)	
batch normalization	
up sampling(2*2)	
deconv3(channel:3, filter:5)	
batch normalization	
up sampling(2*2)	

[1] CIFAR10 dataset is considered as an example for OPC-CAE

[2] KDD99 dataset is considered as an example for OPC-FAE

5.3.3 Proposed anomaly score using orthogonal subspaces score with reconstruction error

To find the anomalies during testing and subsequent deployment, we implement the reconstruction error score (RES). RES is defined based on the reconstruction error $S_1(x)$ of the CAE and reconstruction error $S_2(x)$ of the linear autoencoder. In the test phase, the model calculates the anomaly score of each test sample x'. The reconstruction error $S_1(x')$ between the input x' and it approximation y' by the CAE is defined as

$$S_1(x') = \|x' - y'\|^2 \tag{5.7}$$

Similary, the reconstruction error $S_2(x)$ between the feature vector $\widehat{\mathbf{x}}$ and its approximation by the linear autoencoder is defined as

$$S_2(x') = \|\widehat{\mathbf{x}'} - \widehat{\mathbf{y}}'_{H_1}\|^2 \tag{5.8}$$

The RES can be defined as,

$$S_{RES}(x') = \omega_1 S_1(x') + \omega_2 S_2(x') \tag{5.9}$$

Here, ω_1 and ω_2 are the weighting parameters that control the relative importance of the score functions.

To find the anomalies more effectively, we propose a new OSS. It is defined using the score $S_3(\mathbf{x})$ and $S_4(\mathbf{x})$ of the NS and the AS, respectively. The performance of the proposed OSS in AD is better than the that of the RES. It is proved on CIFAR10 data sets in terms of the performance of the area under the curve (AUC) in detecting individual class anomaly (Figure 5).

The score of data in the NS is represented by S_3. The score S_3 is defined as follows:

$$S_3(x') = \mathbf{y}'^{T}_{\mathbf{H_1}}\mathbf{y}'_{\mathbf{H_1}} \tag{5.10}$$

The projection norm of data in the AS is S_4

$$S_4(x') = \mathbf{y}'^{T}_{\mathbf{H_2}}\mathbf{y}'_{\mathbf{H_2}} \tag{5.11}$$

To evaluate the impact of the overall AD performance, each individual anomaly score is normalized. First, the individual anomaly scores $S_i = \{S_i(x')|x'_i \in X_{test}\}$ for all test samples X_{test} are calculated and then, the maximum $\max(S_i)$ and minimum $\min(S_i)$ of individual anomaly scores are obtained. Therefore, the individual anomaly score $S_i(x')$ for new samples is normalized as

$$P_i = \frac{S_i(x') - min(S_i)}{max(S_i) - min(S_i)} \tag{5.12}$$

Thus, the OSS can be defined as follows:

$$S_{OSS}(\mathbf{x}') = \omega_3(1 - P_3(\mathbf{x}')) + \omega_4 P_4(\mathbf{x}') \tag{5.13}$$

As evident from Fig. 5, the performance of OSS on class categories of CIFAR10 is better than that of RES. Additionally, the OSS improve the performance of AD when combines with RES.

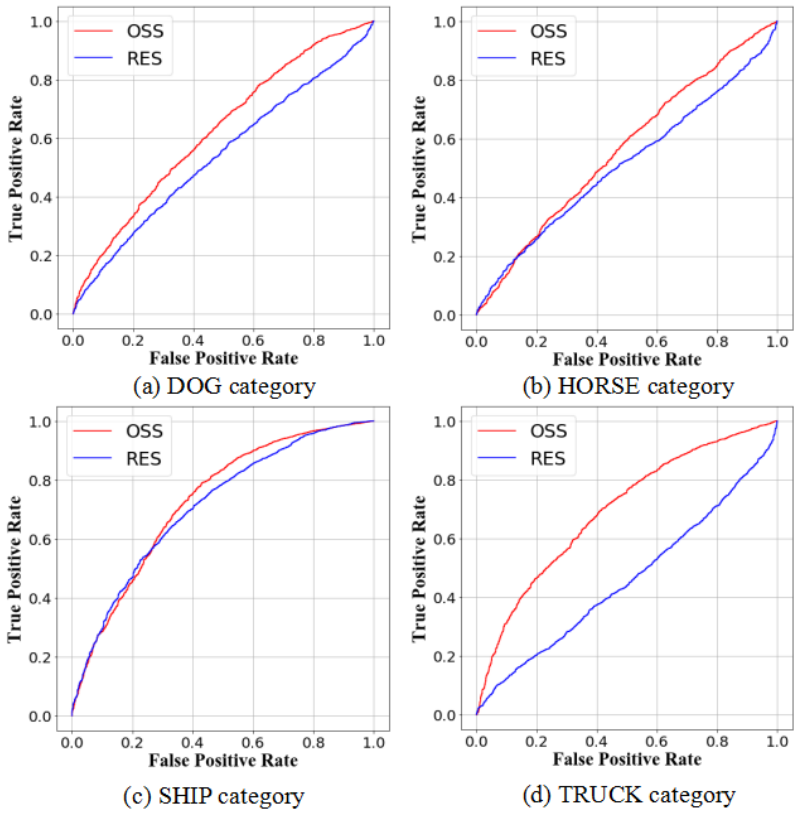


Fig. 5.5 Performance comparison of orthoganal subsepaces score (OSS) and reconstruction error score (RES) on DOG, HORSE, SHIP, and TRUCK categories in terms of ROC curve based on CIFAR10 dataset.

It is defined as follows:

$$\begin{aligned} P(x') &= S_{RES} + S_{OSS} \\ &= \omega_1 P_1(x') + \omega_2 P_2(x') + \omega_3(1 - P_3(x')) + \omega_4 P_4(x') \\ s.t. \quad & \omega_1 + \omega_2 + \omega_3 + \omega_4 = 1 \end{aligned} \tag{5.14}$$

where ω_1, ω_2, ω_3, and ω_4 are the tuning parameters used to adjust according to the tasks. It indicated a high anomaly score with the anomaly data. Hence, using Equation (14), the proposed anomaly score defined the abnormal sample, if P is larger than the threshold. However, the intuition of P_3 is contrast to P. To facilitate the calculation, we use $1 - P_3$. Finally, given a certain threshold δ, the following formula is utilized to identify whether the testing x' is anomalous:

$$Class(x') = \begin{cases} normal, & P(x') \leq \delta; \\ abnormal, & P(x') > \delta. \end{cases} \tag{5.15}$$

5.4 Experimental setup

5.4.1 Dataset

The performance of the OPC-CAE model is evaluated on pubilc datasets with CIFAR10, CIFAR100, STL-10, and IMAGENET for the image datasets. These are the most challenging datasets with various content and complexity compared to other object recognition data sets (Fashion-MNIST and COIL) with properly aligned objects and without background. Among the four datasets CIFAR10 and STL-10 contain images with 10 different classes, whereas CIFAR100 and Imagenet are composed of images with multiple classes. Thus, we selected only 10 classes from CIFAR100 and Imagenet dataset to evaluate the performance of the model. We select one class category as the normal sample for training. We consider the union of remaining classes during testing. In the experiment, each individual class in each dataset is selected as a normal category to verify the generality of our model.

In terms of vector datasets, we perform experiments on Optdigits [9] and Default of credit card clients (DCCC) [106] dataset. In Optdigits, one class (class '3') is treated as being an anomaly, while another class (class '1') is considered as the normal data. Default of credit card clients (DCCC) data set is an open-source dataset from a foreign organization; 'Payment next month', which only includes '0' or '1', is a feature of the DCCC dataset. It indicates whether the user has repaid the credit card bill, '1' indicates repayment, which is considered normal; and '0' indicates no repayment, which is considered abnormal. Moreover, we consider the DCCC dataset is composed of two classes: one normal class and one abnormal class.

5.4.2 Experimental evaluation and performance measure

To simulate an AD setting, four public image datasets is utilized to verify the performance of our model. We follow the same experimental setting as that used in [1, 85, 70], which can make the comparison between the proposed algorithm and the previous works easy. We used only normal samples to train the model and implemented two different protocols to learn the proposed model for training. For image datasets, one class is considered as the normal class, the union of other

classes are considered as abnormal class; in testing, we use the mixture of the normal samples and the abnormal samples for test data.

1. Protocol 1. We divide each class of dataset into 60%, used as training samples, and 40%, used as testing samples. We evaluate Protocol 1 on the IMAGENET dataset. [21].

2. Protocol 2. The existing training and testing data from CIFAR10 [48], STL10 [19], CIFAR100 [48], Optdigits [9], and DCCC [106] are used for experiments. Training split of normal data is used for training and testing split of all classes are used for testing.

In Subsection 5.1, we verified the effectiveness of the proposed OSS for AD in comparision to the RES on CIFAR10 dataset in terms of AUC. Furthermore, in subsection 5.2, the effectiveness of our proposed OPC-CAE based on the combined anomaly score ($RES+OSS$) is compared with nine state-of-the-art methods, including one-class Gaussian mixture odel (GMM) [18], kernel density estimation (KDE) [52], convolutional autoencoder (CAE) [17], VAE [44], pixel CNN decoders (Pixel CNN) [99], GAN [90], Skip-connection Ganomaly (SCG) [5] , anomaly detection with generative adversarial networks (AnoGAN) [90], and one-class GAN (OCGAN) [70].

To verify the generality of our model, we perform the additional experiments based on vector datasets. We use FC layers instead of CNNs, called as OPC-FAE. In the experiments, we compared the proposed method with eight state-of-the-art methods, including several traditional supervised and unsupervised methods. The proposed method is compared with the three most advanced supervised methods including active learning (AL) [2], feature packing (FB) [53], and local outlier factor (LOF) [13]. The proposed method is compared with the five unsupervised methods including sparse coding (SC) [3], L21-SRC (L21) [20], reverse nearest neighbors (RNN) [77], and self-representation outlier detection (SRO) [108]. In addition to these seven methods, the proposed method is also compared with the sparse reconstruction (SR) method proposed by Hou et al. [32].

Additionally, we conducted an ablation experiment. The proposed method was compared with a model based on the OPC without AS. It demonstrated that the individual score S_4 should be removed from the anomaly score for the OPC-based model with AS in the ablation study. Furthermore, the individual and overall class performance of the proposed model for AD based on four image datasets and two vector datasets used in this study is compared with the performance of

state-of-the-art methods in terms of AUC of the receiver operating characteristic curve method.

5.4.3 Parameter settings

We apply Adam optimizer to optimize the network parameters for the image and vector datasets. We set the parameters $\beta_1(0.5)$ and $\beta_2(0.99)$ for the image and vector datasets, respectively. The network is trained using 1,000 epochs for the CIFAR10, CIFAR100, STL10 and, IMAGENET, and Optdigits datasets and 2,000 epochs for the CIFAR100 and DCCC datasets. For image datasets, the learning rate is set to 0.0001 for CIFAR10 and CIFAR100 and 0.00005 for STL10 and IMAGENET. For each iteration, we set the batch size are set to 100. For vector datasets, the learning rate and batch size are set to 0.00005 and 100 for each iteration, respectively. We used two sets of hyper-parameters, $\lambda_1 = 1, \lambda_2 = 1$ and $\lambda_1 = 1, \lambda_2 = 0.3$, to train the model. The proposed framework is implemented in Python 3.6 using Tensorflow 1.9.

5.5 Experimental results

5.5.1 Performance comparison of OSS with RES

In this section, we discuss the experiments performed using OSS and RES on the CIFAR10 dataset. From Table 2, it can be observed that except the performance of class 'deer', the performance of all other classes using our proposed OSS is higher than that using the RES. The OSS improved by 11.82% in comparison to RES, indicating that the detection ability of OSS is higher than that of conventional RES. The conventional RES includes the reconstruction of noise from the raw input and has a significant impact on the detection performance. However, the proposed OSS with encoded data learned from the neural network (CNN or FCN), projected into different subspaces can present more efficient discriminative manifolds. Thus, it can significantly improve the detection performance.

Table 5.2 Performance comparison of OSS and RES on each individual class in terms of AUC based on CIFAR10

Method	PLANE	CAR	BIRD	CAT	DEER	DOG
RES	0.657	0.441	0.660	0.551	**0.754**	0.540
OSS	**0.747**	**0.662**	**0.661**	**0.571**	0.747	**0.623**
Method	FROG	HORSE	SHIP	TRUCK	MEAN	
RES	**0.729**	0.512	0.713	0.458	0.6015	
OSS	**0.729**	**0.563**	**0.729**	**0.694**	**0.6726**	

In most categories, the performance of OSS was better than that of RES, but in some categories (Bird, Frog, Ship), the performances were similar. This can be explained through the following. Firstly, the content in the categories of Bird, Frog and Ship, in comparison to other categories, is less diverse with low complexity. Specifically, it should be noted that these three categories are better aligned than others. Consequently, these three categories can ensure model have better memorization capabilities and accurately extract more discriminative features, which can enhance the performance of RES. Secondly, the distance of the input and output vectors of the linear AE was minimized, thereby improving the discriminative capability of reconstruction error. In detail, this measurement can be motivated by the observation that abnormal samples are suspected to have larger reconstruction errors and it is considered that the subtle changes in the feature vector encoded by the convolutional network can be more easily captured.

5.5.2 Performance comparison against state-of-the-art methods

We discuss the comparison of our method with nine and eight state-of-the-art methods corresponding to image and vector datasets, respectively. The performance of the proposed OPC-CAE model based on the overall ten classes of each of the four datasets demonstrated higher performance with the highest average AUC value than those of state-of-the-art for image AD (Table 3). In addition, the efficiency of the proposed model in rating anomalies was high compared to the recently developed OCGAN model for image datasets. [70]. Especially, more complex datasets (STL-10 and Imagenet) demonstrated significantly higher AUC value with the OPC-CAE model than other state-of-the-art methods.

Table 5.3 Performance comparison of OPC-CAE and state-of-the-art methods on overall class in terms of mean AUC based on all four image datasets

Datasets	GMM	KDE	CAE	VAE	Pixel CNN
CIFAR10	0.5875	0.6097	0.5234	0.5833	0.5506
Cifar100	0.6170	0.6454	0.4912	0.5081	0.5146
STL10	0.6156	0.5842	0.5698	0.5942	0.5002
IMAGENET	0.6326	0.5312	0.5601	0.5412	0.4911
Datasets	GAN	SCG	ANOGAN	OCGAN	Ours
CIFAR10	0.5916	0.6172	0.6179	0.6566	**0.6847**
Cifar100	0.4774	0.4886	0.4678	0.6526	**0.7435**
STL10	0.5043	0.5155	0.5212	0.6177	**0.6908**
IMAGENET	0.5550	0.5404	0.5550	0.6226	**0.6795**

The stable performance of OPC-CAE model based on AD of all classes in all four datasets is shown in Figure 6. The proposed method outperformed state-of-the-arts by showing relatively stable and the highest average AUC value on all four datasets. The performance of each individual class of the proposed OPC-CAE model was compared with the state-of-the-art methods based on CIFAR10, CIFAR100, STL-10, and ImageNet datasets, presented in Table 4, 5, 6 and 7, respectively. Among ten classes in each dataset, six, ten, nine, and seven classes corresponding to CIFAR10, CIFAR100, STL-10, and IMAGENET demonstrated higher AUC than other methods. The performance of the OPC-CAE model of some classes did not improve on all four datasets in terms of ROC curve, shown in Figure 7. However, it was observed that the mean AUC value is always higher than the other state-of-the-art methods (Table 3).

GMM and KDE are probabilistic approaches that use statistical methods to estimate the probability density function of the normal class. KDE is based on the estimation of probability density. The probability density of normal data is larger than that of the abnormal data. GMM performs AD by fitting a normal distribution that can distinguish the abnormal data. The performance of these methods is generally better than that of the methods based solely on reconstruction errors (CAE, VAE, Pix CNN, and GAN). However, reconstruction includes the complicated background score, which reduces the accuracy of reconstruction error detection. Therefore, AnoGAN and SKG utilized the discriminator trained using normal data and produced effective results for AD. However, these methods are not considered

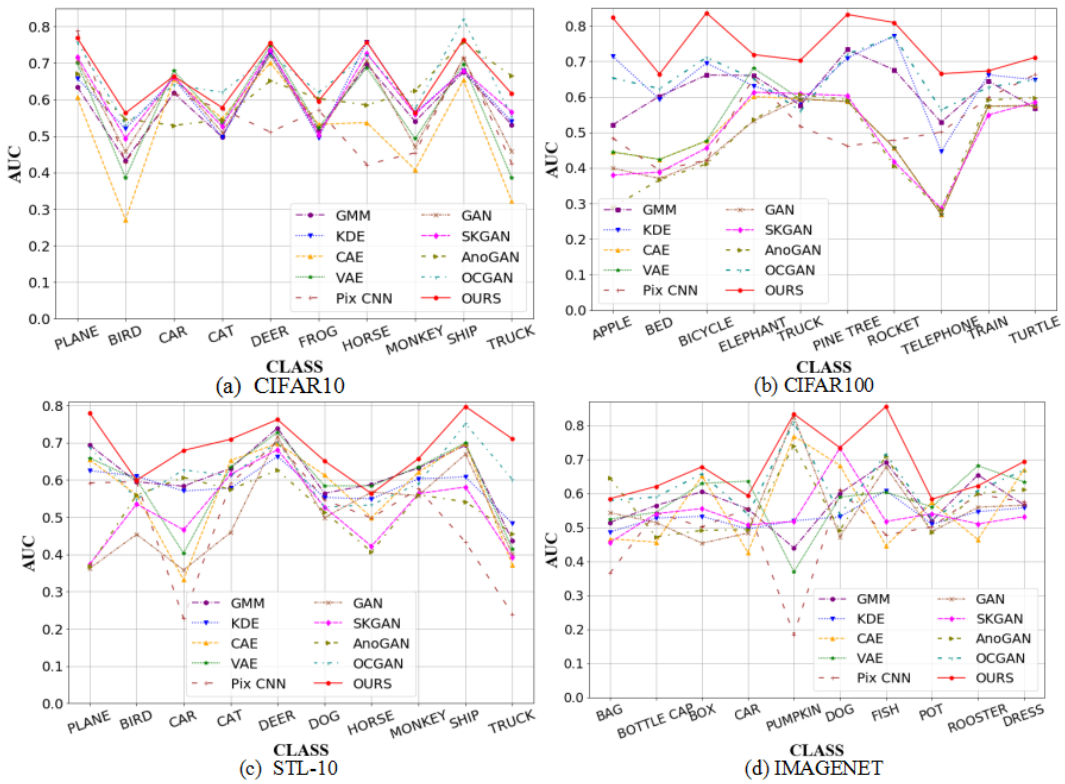


Fig. 5.6 Performance comparison of OPC-CAE and state-of-the-art methods on each individual class in terms of AUC based on all data set

for detection of the distributions in the latent space. OCGAN is a recent proposed method that only maps data into one latent space and does not explore the possibility of projecting into two subspaces. To solve the limitations from these methods, our method constrains the projection operator and obtains the NS and AS for AD. Based on the experimental results, it was found that the performance of the proposed method is better than that of state-of-the-art methods.

To verify the generality and adaptability of the proposed OPC-FAE on vector datasets, we conducted experiments considering eight state-of-the-art methods. As represented in Table 8 and Fig. 8, the proposed model outperforms state-of-the-art methods for Optdigits and DCCC datasets. In addition, the proposed framework demonstrated the higher performance with the highest AUC value than the recent SRO and SR methods.

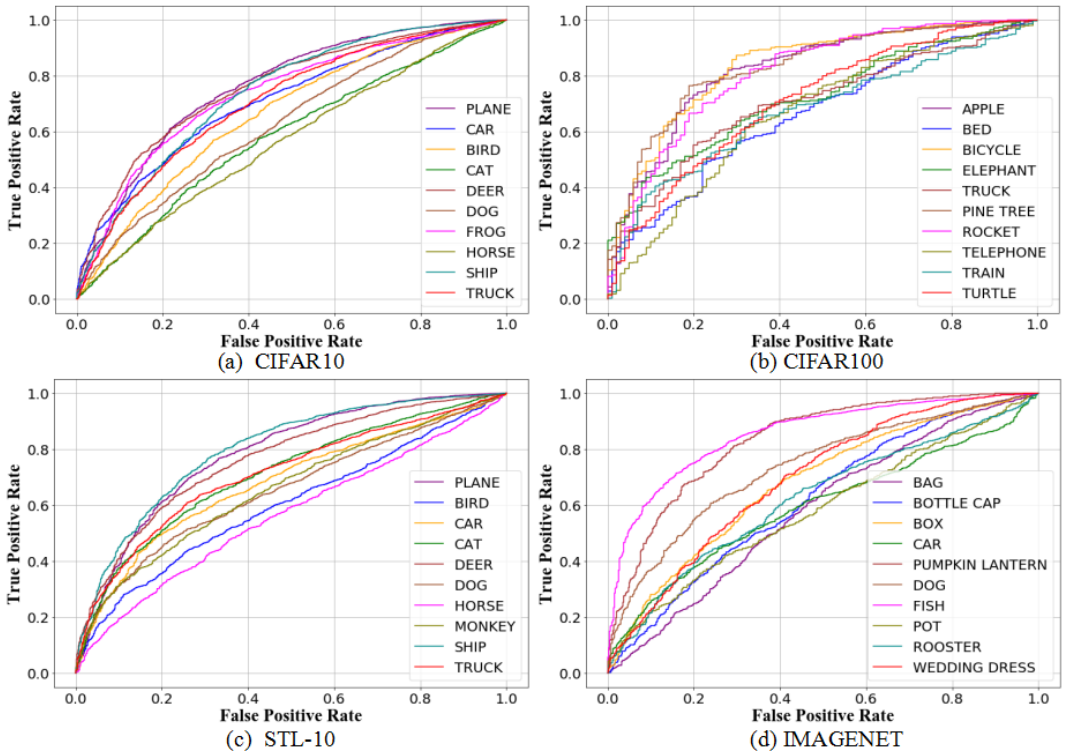


Fig. 5.7 Performance of OPC-CAE on each individual class in terms of AUC based on all datasets

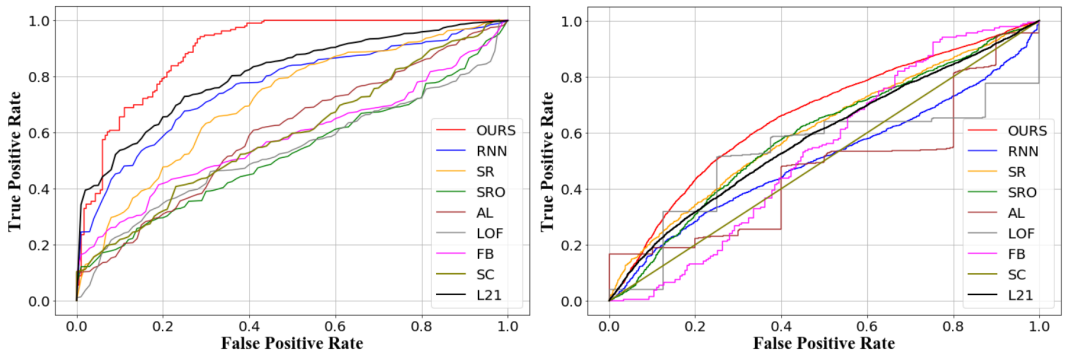
Table 5.8 Performance comparison of OPC-FAE and the state-of-the-art methods on overall classes of vector datasets in terms of AUC

Data sets	FB	AL	LOF	SC	L21
Optdigits	0.577	0.590	0.523	0.589	0.833
Default of credit card clients	0.535	0.484	0.524	0.496	0.599
Data sets	**RNN**	**SRO**	**SR**	**Ours**	
Optdigits	0.767	0.515	0.722	**0.896**	
Default of credit card clients	0.506	0.600	0.606	**0.660**	

Table 5.4 Performance comparison of OPC-CAE and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR10

Method	PLANE	CAR	BIRD	CAT	DEER	DOG
GMM	0.635	0.433	0.618	0.498	0.733	0.515
KDE	0.658	0.520	0.657	0.497	0.727	0.496
CAE	0.606	0.271	0.655	0.549	0.701	0.532
VAE	0.700	0.386	**0.679**	0.535	0.748	0.523
Pix CNN	**0.788**	0.428	0.617	0.574	0.511	0.571
GAN	0.708	0.458	0.664	0.510	0.722	0.505
SKG	0.717	0.494	0.662	0.527	0.736	0.504
AnoGAN	0.671	0.547	0.529	0.545	0.651	0.603
OCGAN	0.757	0.531	0.640	0.620	0.723	0.620
Ours	0.760	**0.708**	0.664	**0.585**	**0.757**	**0.628**

Method	FROG	HORSE	SHIP	TRUCK	MEAN
GMM	0.696	0.540	0.675	0.531	0.5874
KDE	**0.758**	0.564	0.680	0.540	0.6097
CAE	0.537	0.408	0.653	0.322	0.5234
VAE	0.687	0.493	0.696	0.386	0.5833
Pix CNN	0.422	0.454	0.715	0.426	0.5506
GAN	0.707	0.471	0.713	0.458	0.5916
SKG	0.726	0.560	0.680	0.566	0.6172
AnoGAN	0.585	0.625	0.758	0.665	0.6179
OCGAN	0.723	0.575	**0.820**	0.554	0.6566
Ours	0.734	**0.566**	0.737	**0.707**	**0.6847**



(a) Optdigits (b) Default of credit card clients

Fig. 5.8 Performance of OPC-FAE on each individual class in terms of AUC based on all vector datasets

Table 5.5 Performance comparison of OPC-CAE and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR100

Method	APPLE	BED	BICYCLE	ELEPHANT	TRUCK	PINE TREE
GMM	0.521	0.602	0.661	0.660	0.576	0.733
KDE	0.714	0.593	0.695	0.631	0.586	0.709
CAE	0.440	0.414	0.456	0.601	0.592	0.589
VAE	0.445	0.424	0.476	0.681	0.594	0.587
Pix CNN	0.484	0.393	0.422	0.654	0.517	0.462
GAN	0.399	0.370	0.422	0.532	0.594	0.587
SKG	0.380	0.388	0.456	0.613	0.609	0.603
AnoGAN	0.289	0.367	0.411	0.536	0.606	0.592
OCGAN	0.653	0.623	0.711	0.651	0.560	0.720
Ours	**0.823**	**0.664**	**0.836**	**0.719**	**0.703**	**0.832**

Method	ROCKET	TELEPHONE	TRAIN	TURTLE	MEAN
GMM	0.676	0.528	0.645	0.568	0.6170
KDE	0.772	0.446	0.662	0.648	0.6456
CAE	0.450	0.238	0.571	0.565	0.4912
VAE	0.456	0.269	0.574	0.575	0.5081
Pix CNN	0.419	0.688	0.444	0.663	0.5146
GAN	0.456	0.280	0.564	0.570	0.4774
SKG	0.417	0.286	0.549	0.585	0.4886
AnoGAN	0.406	0.282	0.591	0.598	0.4678
OCGAN	0.770	0.563	0.627	0.648	0.6526
Ours	**0.809**	**0.665**	**0.673**	**0.711**	**0.7435**

Among the comparison methods for vector AD task, only the SR method demonstrated a better effect. This is because it projects the input data into the latent space using an autoencoder and learns a sparse representation dictionary to represent the latent space distribution. Compared with other methods, the SR can map the raw input into the latent space while the proposed method can map the raw input into the NS and AS. Thus, the data manifolds can be better represented in the subspace by reducing the influence of noise on data distribution.

5.5.3 Individual class performance analysis

Overall, the proposed method outperformed other AD methods with an average AUC of 0.6847, 0.7435, 0.6908 and 0.6795 corresponding to CIFAR10, CIFAR100,

Table 5.6 Performance comparison of OPC-CAE and the state-of-the-art methods on each individual class in terms of AUC concerning STL-10

Method	PLANE	BIRD	CAR	CAT	DEER	DOG
GMM	0.694	0.595	0.583	0.631	0.739	0.564
KDE	0.625	**0.610**	0.570	0.578	0.663	0.553
CAE	0.654	0.560	0.332	0.652	0.698	**0.613**
VAE	0.659	0.601	0.403	0.635	0.728	0.584
Pix CNN	0.592	0.595	0.228	0.591	0.703	0.546
GAN	0.362	0.454	0.358	0.459	0.716	0.499
SKG	0.373	0.535	0.466	0.615	0.681	0.527
AnoGAN	0.368	0.559	0.607	0.574	0.626	0.514
OCGAN	0.688	0.548	0.627	0.611	0.701	0.527
Ours	**0.780**	0.599	**0.679**	**0.709**	**0.762**	**0.651**

Method	HORSE	MONKEY	SHIP	TRUCK	MEAN
GMM	0.588	0.632	0.693	0.437	0.6156
KDE	0.549	0.603	0.608	0.483	0.5842
CAE	0.499	0.621	0.698	0.371	0.5698
VAE	0.584	0.635	0.699	0.414	0.5942
Pix CNN	0.498	0.576	0.433	0.240	0.5002
GAN	0.567	0.558	0.669	0.401	0.5043
SKG	0.422	0.563	0.581	0.392	0.5155
AnoGAN	0.407	0.560	0.541	0.456	0.5212
OCGAN	0.533	0.590	0.751	0.601	0.6177
Ours	**0.563**	**0.657**	**0.797**	**0.710**	**0.6908**

STL10, and IMAGENET (Table 3), respectively. Furthermore, the average AUCs of 0.996 and 0.660 on two vector datasets in Table 8 showed superior performance over other AD methods on vector datasets. The performance on CIFAR-10 in Table 4 demonstrates the performance of the OCGAN method is close to our results, except the category 'SHIP', which is higher than ours. The propoesd method demonstrated an improvement of 4.28% over the OCGAN method based on the average AUCs. The performance of CIFAR100 is presented in Table 5. Our method shows high performance in all categories for CIFAR100 by indicating high AUC values. It is worth noting that the classical statistical methods, GMM and KDE, performed better than the reconstruction-based methods, (CAE, VAE, Pix CNN, GAN, SKG, and AnoGAN) and almost similar to OCGAN. This indicates that the statistical technique of data distribution is effective and promising. As evident from Table 6, the proposed

Table 5.7 Performance comparison of OPC-CAE and the state-of-the-art methods on each individual class in terms of AUC concerning IMAGENET

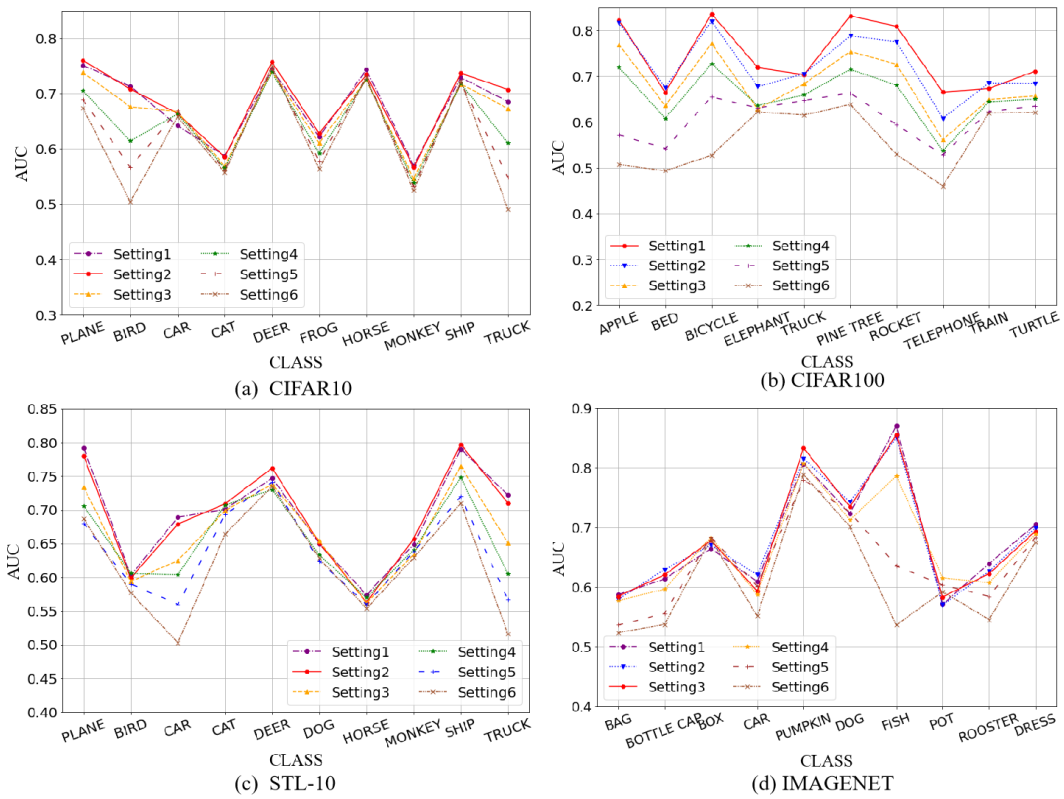
Method	BAG	BOTTLE CAP	BOX	CAR	PUMPKIN	DOG
GMM	0.514	0.564	0.605	0.554	0.439	0.600
KDE	0.486	0.527	0.532	0.495	0.519	0.531
CAE	0.466	0.456	0.651	0.426	0.768	0.682
VAE	0.524	0.544	0.629	**0.636**	0.369	0.589
Pix CNN	0.365	0.543	0.501	0.556	0.184	0.611
GAN	0.543	0.472	0.453	0.483	0.826	0.472
SKG	0.455	0.540	0.555	0.507	0.517	0.733
AnoGAN	**0.645**	0.514	0.491	0.493	0.740	0.491
OCGAN	0.579	0.589	0.657	0.538	0.809	0.538
Ours	0.584	**0.620**	**0.678**	0.593	**0.833**	**0.734**
Method	FISH	POT	ROOSTER	DRESS	MEAN	
GMM	0.691	0.519	**0.654**	0.566	0.5706	
KDE	0.608	0.509	0.546	0.557	0.5310	
CAE	0.446	0.573	0.464	0.670	0.5602	
VAE	0.603	0.560	0.681	0.634	0.5769	
Pix CNN	0.478	0.501	0.597	0.575	0.4911	
GAN	0.677	0.487	0.559	0.565	0.5537	
SKG	0.517	0.540	0.510	0.531	0.5405	
AnoGAN	0.709	0.486	0.600	0.611	0.5780	
OCGAN	0.715	0.524	0.608	0.668	0.6225	
Ours	**0.855**	**0.583**	0.622	**0.693**	**0.6795**	

method improved by 11.83% compared to OCGAN. Furthermore, the proposed method exhibited the highest AUCs in all categories except the 'BIRD' category on STL10 dataset. Additionally, the performance of GMM is almost similar to that of OCGAN on STL10 dataset. Table 7 presents the performance of the IMAGENET dataset. The performance of the proposed method improved by 9.16% compared to OCGAN. Table 8 demonstrates that the proposed OPC-FAE is superior to other classical AD methods with an AUC of 0.896 and 0.660 corresponding to Optdigits and DCCC, respectively. The performance of OPC-FAE improved by 21.61% and 8.91% compared to the SR method corresponding to Optdigits and DCCC, respectively.

5.5.4 Parameter sensitivity analysis

The detection performance of the proposed method is controlled by a set of hyper-parameters: ω_1, ω_2, ω_3, and ω_4, which present the importance degree of individual score. It is beneficial to explore the sensitivity of these parameters to understand which scores are important in testing, thereby providing effective ways to monitor and control their detection condition. We selected six sets of hyper-parameters for each dataset in the testing phase.

As evident from the Table. A12-A17 in Appendix A, better performance will be obtained when ω_1 and ω_2 is smaller and ω_3 and ω_4 is larger; this implies that the weight of RES (the combination of P_1 and P_2)is less than that of OSS (the combination of P_1 and P_2). Consequently, the performance of OSS is better than that of RES. In addition, the weight of ω_4 is larger than that of ω_3 when the best AUC value is obtained among the six sets of hyper-parameters, indicating that the performance of the AS is better than that of NS. Furthermore, the performance is better when ω_3 and ω_4 are larger among the image and vector datasets, indicating that the OSS is more sensitive to abnornal data and the robustness of the proposed model. According to Table. A12, A13, A14, A15, A16, and A17 in Appendix A, and Fig. 9, the parameters are changed to obtain the highest AUC value. It can be observed that the weight of OSS are larger than that of RES, which indicates that OSS can greatly improve the detection performance. Finally, $setting_2$(ω_1=0.10, ω_2=0.10, ω_3=0.30, ω_4=0.50) is selected for CIFAR10 , $setting_1$(ω_1=005, ω_2=0.05, ω_3=0.35, ω_4=0.55) for CIFAR100 , $setting_2$(ω_1=0.08, ω_2=0.07, ω_3=0.32, ω_4=0.53) for STL10 , $setting_3$(ω_1=0.09, ω_2=0.15, ω_3=0.41, ω_4=0.35) for IMAGENET , $setting_4$(ω_1=0.18, ω_2=0.18, ω_3=0.30, ω_4=0.34) for Optdigits , $setting_3$(ω_1=0.18, ω_2=0.13, ω_3=0.33, ω_4=0.36) for DCCC.



(a) CIFAR10 (b) CIFAR100 (c) STL-10 (d) IMAGENET

Fig. 5.9 Performance of OPC-FAE on each individual class in terms of AUC based on all image datasets according to different hyper-parameter sets.

5.5.5 Evaluation of rubustness to the additional noises

To verify the effectiveness and generalization of the proposed method, noisy dataset is utilized for experiments in comparison to CAE and OCGAN. CIFAR10 is added with two kinds of noise, sampled from a Gaussion distribution ($u = 0$, $\sigma = 0.1$) and a uniform distribution ($a = 0$, $b = 0.3$), respectively. As evident from Table. 9, the result of our method decreases from 0.6874 to 0.6452 in terms of mean AUC, compared with CAE from 0.5234 to 0.5167, and OCGAN from 0.6560 to 0.6061 when noise of Gaussion distribution is used. From Table. 10, when noise of Uniform distribution is used, the result of our method decreases from 0.6874 to 0.6539 in terms of mean AUC, compared with CAE from 0.5234 to 0.5206, and OCGAN from 0.6560 to 0.6051. Images added with noise will deteriorate the performance of image detection, implying that the proposed method and the compared method have a decline in performance of AD. However, the proposed method still ensures the highest AUC, which indicates that the proposed method is more effective and robust.

Table 5.9 Performance comparison of CAE, OCGAN and the proposed on each individual class in terms of AUC based on CIFAR10 with noise of Gaussion distribution $\mathcal{N}(0, 0.1)$.

Method	PLANE	CAR	BIRD	CAT	DEER	DOG
CAE	0.526	0.307	0.555	0.551	0.613	0.524
OCGAN	0.654	0.521	0.621	0.539	0.680	0.531
OURS	**0.730**	**0.524**	**0.659**	**0.553**	**0.756**	**0.565**
Method	FROG	HORSE	SHIP	TRUCK	MEAN	
CAE	0.629	0.433	0.660	0.369	0.5167	
OCGAN	0.670	0.589	0.737	0.519	0.6061	
OURS	**0.744**	**0.581**	**0.752**	**0.588**	**0.6452**	

Table 5.10 Performance comparison of CAE, OCGAN and the proposed on each individual class in terms of AUC based on CIFAR10 with noise of Uniform distribution $\mathcal{U}(0, 0.3)$.

Method	PLANE	CAR	BIRD	CAT	DEER	DOG
CAE	0.502	0.322	0.655	0.536	0.627	0.518
OCGAN	0.662	**0.601**	0.626	0.524	0.673	0.512
OURS	**0.770**	0.588	**0.661**	**0.567**	**0.756**	**0.548**
Method	FROG	HORSE	SHIP	TRUCK	MEAN	
CAE	0.620	0.434	0.647	0.345	0.5206	
OCGAN	0.676	0.533	0.710	0.534	0.6051	
OURS	**0.754**	**0.556**	**0.734**	**0.605**	**0.6539**	

5.5.6 Convergence of the proposed model

To demonstrate the convergence of the proposed framework, we randomly selected a class (CAR) from the CIFAR10 dataset and analyzed its corresponding iterative curve. As shown in Fig 10, the learning curve of the proposed model converges and stabilizes after the 1250^{th} epoch. In detail, the learning curve of the middle loss (MSE loss between the input vector and approximation of the linear AE) converges and

stabilizes after 200^{th} epoch, the learning curve of the reconstruction loss (MSE loss between the input and approximation of the model) and that of OPC loss converge and stabilize after the 700^{th} and 1250^{th} epoch according to Fig. 2(b), (d), respectively. In addition, as evident from Fig. 11, the anomaly detection performance of the proposed network reaches the peak at about 150^{th} epoch with the highest AUC value and it tends to be stable after 1250 epochs, finally the test AUC (0.708) can be obtained after 2000^{th} epoch. According to Fig. 10 and Fig. 11, it is indicated that the performance of the proposed method is superior and stable.

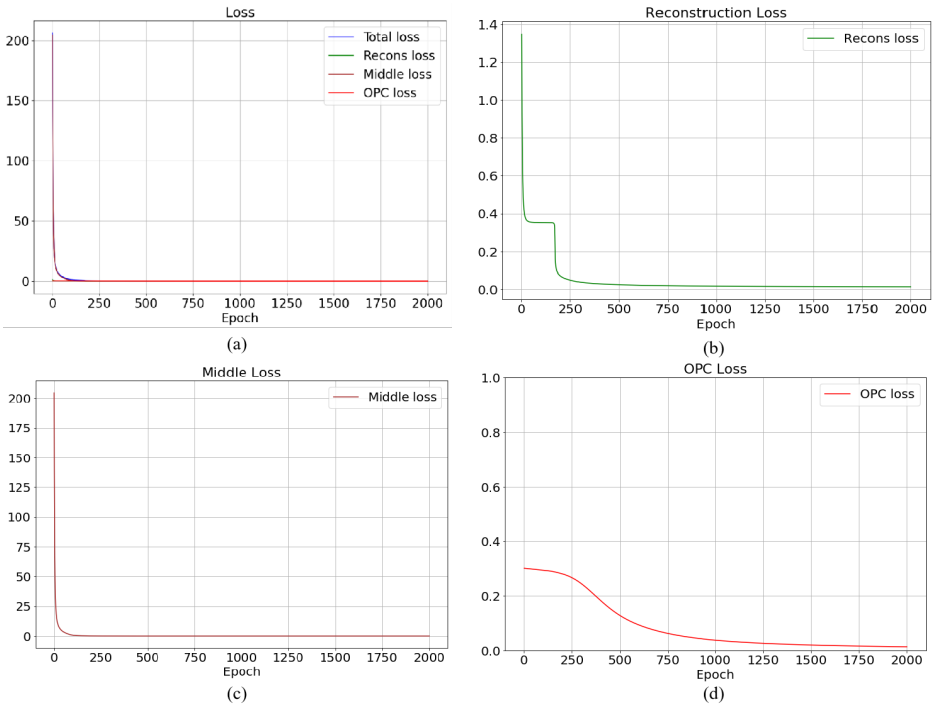


Fig. 5.10 Convergence curve of the proposed model on the 'CAR' class in CIFAR10 dataset. The horizontal and the vertical axis represent the number of epochs and loss values, respectively. (a). Total loss. (b), (c), and (d) denote individual losses corresponding to reconrtruction loss, middle loss (the MSE beteen the input feature vector and the approximation of the embedded linear autoencoder), and OPC loss, respectively.

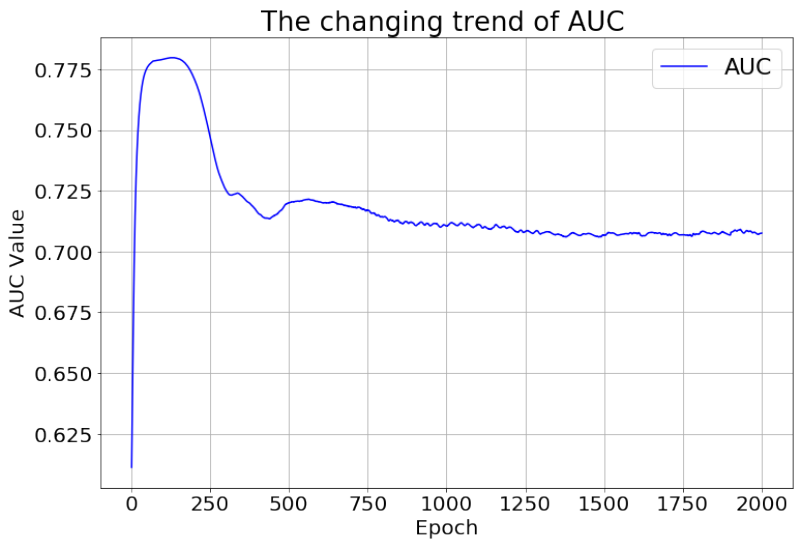


Fig. 5.11 Anomaly detection performance of the proposed method with the increasing number of iterations in terms of AUC.

5.5.7 Ablation study

The effectiveness of the OPC-CAE and OPC-FAE frameworks was validated through ablation experiments conducted using image and vector datasets, respectively. We consider the following two learning settings: 1)implementing the two subspaces model, and 2) removing the AS. The experimental results are presented in Table 11. For image datasets, it was observed that under Settings 2, the AD performance deteriorated by almost 6.1% and 7.6% on CIFAR10 and STL10 datasets, respectively, compared to the performance of the complete model using Setting 1. For vector datasets, the AD performance in Settings 2 using DCCC dataset decreases by more than 7.4% compared to that of the complete model.

Table 5.11 Performance comparison based on ablation validation in terms of AUC

Datasets	Setting 1	Setting 2
CIFAR10 (average AUC)	0.6847	0.6427
STL10 (average AUC)	0.6908	0.6385
Default of credit card clients	0.660	0.611

5.5.8 Performance visualization based on normal and abnormal subspace

The latent space was visualized using PCA that reduced the dimensionality into two dimensions. The performance of subspace representation distribution of the proposed model was compared with the latent space representation of the standard CAE model without OPC. Fig. 12(a) and (d) is the visualization of the standard CAE model without OPC, Fig. 12(b) and (e) illustrate the visualization of NS of our method, and Fig. 12(c) and (f) is the visualization of AS of our proposed method. The images in first row of Fig. 12 indicate the visualization of class 'ROCKET' in CIFAR100. The images in the second row represent the visualization of class 'BICYCLE' in CIFAR100. It was found that the subspace representation distribution in AS (Fig. 12(c), (f)) can efficiently distinguish between normal and abnormal samples in comparison to the latent presentation of the standard antoencoder model (Fig. 12(a), (d)). Fig. 10 also demonstrates that the subspace representation of NS (Fig. 12 (b), (e)) is more effective than the latent representation of standard CAE for AD tasks.

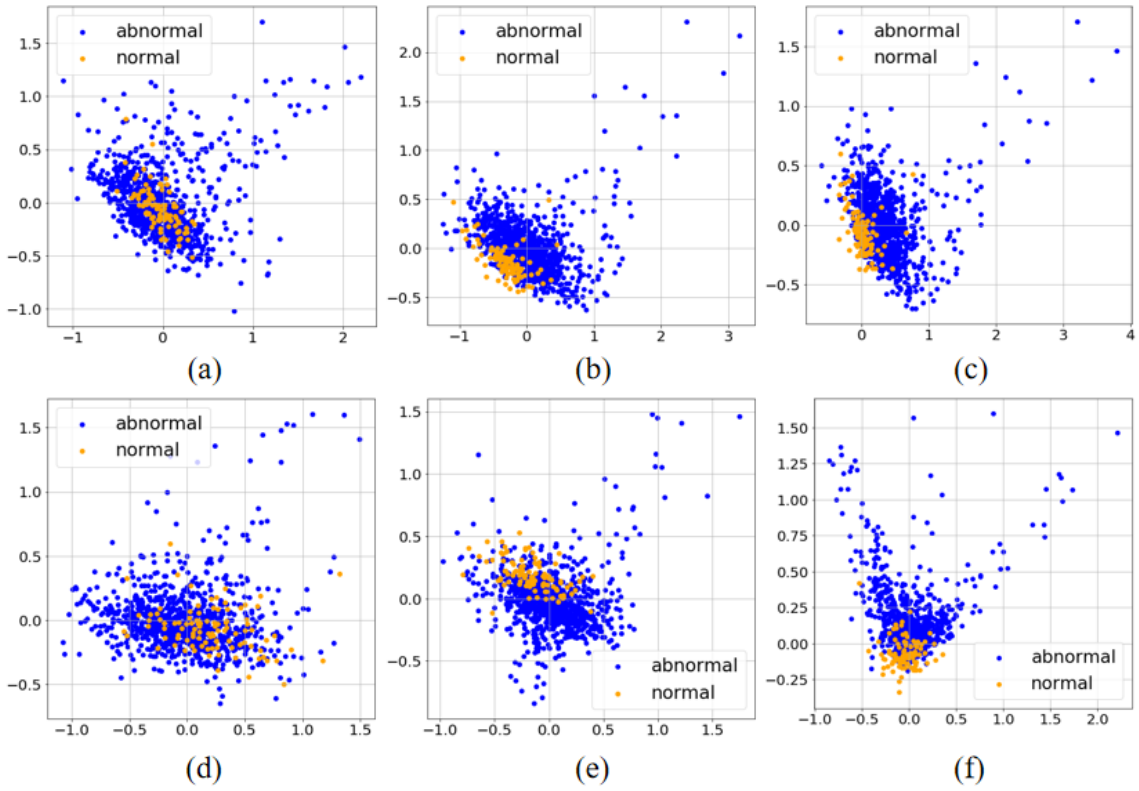


Fig. 5.12 Space visualization comparison of the proposed OPC-CAE over the baseline CAE without using OPC based on CIFAR100. (a), (b), and (c) shows the latent space representation of the class 'rocket' using baseline autoencoder without OPC, proposed OPC-CAE with normal space, and proposed OPC-CAE with abnormal space, respectively, (d) (e), and (f) shows latent space representation of the class 'bicycle' using baseline autoencoder without OPC, normal space in the proposed OPC-CAE , and abnormal space in the proposed OPC-CAE, respectively.

Chapter 6

Conclusion

In this work, we propose feature learning and data mapping scheme in neural networks for anomaly detection. It discusses the ability of generation and dimensionality reduction performance of the generative model, and analyzes the learning ability of the discriminative features of the model ； and also extends the feature representations to multiple spaces. The advantages of feature extraction have the improvements in binary classification abilities and feature representations. The relevant theoretical background of AD is discussed at the first of this work, then focus on low-dimensional representation constraints (chapter 2). The importance in Chapter 2 are discussed as to its low-dimensional feature extracting and feature understanding. The results show that these methods can be used to learn discriminative low-dimensional features and reconstruction errors. Interestingly, it has been found that embedding multiple spaces can achieve better performance for multiple deep networks. The proposed method is proved by the classification problem of public image and vector dataset. The experimental results are consistent with our hypothesis, which is supported by mathematical evidence.

Uncertain cases is unavailable to learn the feature in this tasks resulting in the model confuse on which features are useful. In our experiments, the detection of abnormal states is based on the discriminative features of normal states, which not only depends on the feature representation of normal data, but also depends on the choice of boundaries. Feature representations for anomaly detection are focused on filtering out important features from original features and are sparse. But another problem of classification description is that many similar features between different classes will result in bad performance detection of anomalies. Therefore, in this work it is suggested utilizing maximization of mutual information (MMI) as regularizer for its ability to learning the discriminative information.

In Chapter 3, we also found that the low-dimensional representation of our model has more discriminative low-dimensional features after introducing MMI as a regularizer, which will increase the learning capabilities of the model. The introduction of MMI helps to learn the local and global feature in deep neural network models. We introduce the MMI between the input and the latent representation, the approximation, and the MMI between the hidden layer representations as regularizers to enhance the latent representation. MMI introduces the learning algorithm to the most discriminative learning, and prevents latent space learning failures caused by reconstruction noise. Mutual information (MI) can reduce the noise of the generated data, so that the low-dimensional representation of the latent space will be more discriminative. This is done through an anomaly detection experiment based on an anomaly detection task, which results in a great improvements feature manifolds of the model represented by the feature. In addition, our experiment categorizes abnormal states in various data sets by injecting KLD loss into selected joints of the neural network, thereby steadily improving the accuracy of important features and clearer positioning capabilities.

A single space can only be trained to accommodate normal data, but not abnormal data, which means ignoring the basic principle of multiple spaces. The manifold in a single space is limited because some unnecessary features will be mapped into this space, which will decrease the performance of AD. Therefore, our work focus on the multi-space analysis for feature representation in Chapter 4. We also analyzed the effectiveness of the gated network, showing that different spaces are weighted to highlight the importance of the space for a specific task. A convolutional autoencoder with fully connected layers embedded in latent space is introduced to capture the discriminative features enhanced by manifolds in different spaces. Compared with other AD methods, our AD performance is superior in terms of AUC value. This can be observed through experimentation and visualization of latent representations. These findings will help to understand useful changes by looking at the evolution of the latent space that define anomalies.

Based on the theoretical considerations in Chapter 4, AD represented by multiple spaces is used to evaluate linear connection of different features. It avoids some shortcomings of single-space learning by introducing low-dimensional information. Data is represented in multiple low-dimensional spaces, and the feature representation is more discriminative through linear combination, and feature representation and boundary determination are regarded as the unique pair of description effects.

We also proposed another measure of AD through orthogonal projection mechanism. Our method can overcome the major challenges of noise and multi-space detection, which makes orthogonality more promising in detection. To verify the effectiveness of proposed model, six datasets were used, including image and vector datasets. The evaluation results show that the orthogonal complementary subspace has a robust effect on anomaly detection. By linking orthogonal complementary spaces, the model can be concentrated for feature projection.

The effective feature extraction capability of the proposed architecture proves the applicability of the extraction and understanding based on low-dimensional manifolds. In addition, the extraction of features and low-dimensional mapping can be performed in many tasks, from evaluating the features of data distribution through feature extraction, dimensionality reduction, and reconstruction, to distinguish between normal and abnormal data. Finally, it can be said that the work described here analyzes several issues that have effective feature extraction and understanding so far. Using the method suggested in this paper, we also used the new anomaly score to test the severity of the anomaly, which allows us to understand the state of the anomaly in more detail.

References

[1] Abati, D., Porrello, A., Calderara, S., and Cucchiara, R. (2019). Latent space autoregression for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 481–490.

[2] Abe, N., Zadrozny, B., and Langford, J. (2006). Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 504–509.

[3] Adler, A., Elad, M., Hel-Or, Y., and Rivlin, E. (2015). Sparse coding with anomaly detection. *Journal of Signal Processing Systems*, 79(2):179–188.

[4] Akcay, S., Atapour-Abarghouei, A., and Breckon, T. P. (2018). Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, pages 622–637. Springer.

[5] Akçay, S., Atapour-Abarghouei, A., and Breckon, T. P. (2019). Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[6] Akoglu, L., Tong, H., and Koutra, D. (2015). Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3):626–688.

[7] Alom, M. Z., Bontupalli, V., and Taha, T. M. (2015). Intrusion detection using deep belief networks. In *2015 National Aerospace and Electronics Conference (NAECON)*, pages 339–344. IEEE.

[8] An, J. and Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18.

[9] Asuncion, A. and Newman, D. (2007). Uci machine learning repository.

[10] Beggel, L., Pfeiffer, M., and Bischl, B. (2019). Robust anomaly detection in images using adversarial autoencoders. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 206–222. Springer.

[11] Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. (2018). Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*.

[12] Bereziński, P., Jasiul, B., and Szpyrka, M. (2015). An entropy-based network anomaly detection method. *Entropy*, 17(4):2367–2408.

[13] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.

[14] Calderara, S., Heinemann, U., Prati, A., Cucchiara, R., and Tishby, N. (2011). Detecting anomalies in people's trajectories using spectral graph analysis. *Computer Vision and Image Understanding*, 115(8):1099–1111.

[15] Chalapathy, R., Menon, A. K., and Chawla, S. (2017). Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 36–51. Springer.

[16] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*.

[17] Chen, Z., Yeo, C. K., Lee, B. S., and Lau, C. T. (2018). Autoencoder-based network anomaly detection. In *2018 Wireless Telecommunications Symposium (WTS)*, pages 1–5. IEEE.

[18] Clifton, L., Clifton, D. A., Watkinson, P. J., and Tarassenko, L. (2011). Identification of patient deterioration in vital-sign data using one-class support vector machines. In *2011 federated conference on computer science and information systems (FedCSIS)*, pages 125–131. IEEE.

[19] Coates, A., Ng, A., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223.

[20] Cong, Y., Yuan, J., and Liu, J. (2011). Sparse reconstruction cost for abnormal event detection. In *CVPR 2011*, pages 3449–3456. IEEE.

[21] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[22] Dieng, A. B., Kim, Y., Rush, A. M., and Blei, D. M. (2019). Avoiding latent variable collapse with generative skip models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2397–2405. PMLR.

[23] Dufrenois, F. (2014). A one-class kernel fisher criterion for outlier detection. *IEEE transactions on neural networks and learning systems*, 26(5):982–994.

[24] Fan, H., Zhang, F., Wang, R., Xi, L., and Li, Z. (2020a). Correlation-aware deep generative model for unsupervised anomaly detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 688–700. Springer.

[25] Fan, Y., Wen, G., Li, D., Qiu, S., Levine, M. D., and Xiao, F. (2020b). Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. *Computer Vision and Image Understanding*, 195:102920.

[26] Fujimaki, R., Yairi, T., and Machida, K. (2005). An anomaly detection method for spacecraft using relevance vector learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 785–790. Springer.

[27] Gandhi, T. and Trivedi, M. M. (2007). Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on intelligent Transportation systems*, 8(3):413–430.

[28] Harrou, F., Kadri, F., Chaabane, S., Tahon, C., and Sun, Y. (2015). Improved principal component analysis for anomaly detection: Application to an emergency department. *Computers & Industrial Engineering*, 88:63–77.

[29] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., and Davis, L. S. (2016). Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742.

[30] He, Z., Xu, X., and Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650.

[31] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2018). Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

[32] Hou, D., Cong, Y., Sun, G., Liu, J., and Xu, X. (2019). Anomaly detection via adaptive greedy model. *Neurocomputing*, 330:369–379.

[33] Huang, W., Zhang, J., Sun, H., Ma, H., and Cai, Z. (2017). An anomaly detection method based on normalized mutual information feature selection and quantum wavelet neural network. *Wireless Personal Communications*, 96(2):2693–2713.

[34] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

[35] Jagota, A. (1991). Novelty detection on a very large number of memories stored in a hopfield-style network. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume 2, pages 905–vol. IEEE.

[36] Ji, X., Henriques, J. F., and Vedaldi, A. (2018). Invariant information distillation for unsupervised image segmentation and clustering. *arXiv preprint arXiv:1807.06653*, 2(3):8.

[37] Kavitha, M., Lee, C.-H., Shibudas, K., Kurita, T., and Ahn, B.-C. (2020). Deep learning enables automated localization of the metastatic lymph node for thyroid cancer on 131 i post-ablation whole-body planar scans. *Scientific reports*, 10(1):1–12.

[38] Kavitha, M. S., Kurita, T., Park, S.-Y., Chien, S.-I., Bae, J.-S., and Ahn, B.-C. (2017). Deep vector-based convolutional neural network approach for automatic recognition of colonies of induced pluripotent stem cells. *PloS one*, 12(12):e0189974.

[39] Kawachi, Y., Koizumi, Y., and Harada, N. (2018). Complementary set variational autoencoder for supervised anomaly detection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2366–2370. IEEE.

[40] Khreich, W., Khosravifar, B., Hamou-Lhadj, A., and Talhi, C. (2017). An anomaly detection system based on variable n-gram features and one-class svm. *Information and Software Technology*, 91:186–197.

[41] Kim, D., Yang, H., Chung, M., Cho, S., Kim, H., Kim, M., Kim, K., and Kim, E. (2018). Squeezed convolutional variational autoencoder for unsupervised anomaly detection in edge device industrial internet of things. In *2018 international conference on information and computer technologies (icict)*, pages 67–71. IEEE.

[42] Kim, J., Kim, H.-J., and Kim, H. (2019). Fraud detection for job placement using hierarchical clusters-based deep neural networks. *Applied Intelligence*, 49(8):2842–2861.

[43] Kim, J., Shin, N., Jo, S. Y., and Kim, S. H. (2017). Method of intrusion detection using deep neural network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 313–316. IEEE.

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

[45] Knorr, E. M., Ng, R. T., and Tucakov, V. (2000). Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3):237–253.

[46] Koch-Janusz, M. and Ringel, Z. (2018). Mutual information, neural networks and the renormalization group. *Nature Physics*, 14(6):578–582.

[47] Kriegel, H.-P., Schubert, M., and Zimek, A. (2008). Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452.

[48] Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

[49] Kumagai, A., Iwata, T., and Fujiwara, Y. (2019). Transfer anomaly detection by inferring latent domain representations. In *Advances in Neural Information Processing Systems*, pages 2471–2481.

[50] Kumar, A. (2008). Computer-vision-based fabric defect detection: A survey. *IEEE transactions on industrial electronics*, 55(1):348–363.

[51] Kurita, T. and Takahashi, T. (2003). Viewpoint independent face recognition by competition of the viewpoint dependent classifiers. *Neurocomputing*, 51:181–195.

[52] Latecki, L. J., Lazarevic, A., and Pokrajac, D. (2007). Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 61–75. Springer.

[53] Lazarevic, A. and Kumar, V. (2005). Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166.

[54] Li, D., Chen, D., Goh, J., and Ng, S.-k. (2018). Anomaly detection with generative adversarial networks for multivariate time series. *arXiv preprint arXiv:1809.04758*.

[55] Li, N. and Chang, F. (2019). Video anomaly detection and localization via multivariate gaussian fully convolution adversarial autoencoder. *Neurocomputing*, 369:92–105.

[56] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE.

[57] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–39.

[58] Liu, Y., Li, Z., Zhou, C., Jiang, Y., Sun, J., Wang, M., and He, X. (2019). Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1517–1528.

[59] Loka, N. R., Kavitha, M., and Kurita, T. (2019). Hilbert vector convolutional neural network: 2d neural network on 1d data. In *International Conference on Artificial Neural Networks*, pages 458–470. Springer.

[60] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

[61] Marchi, E., Vesperini, F., Eyben, F., Squartini, S., and Schuller, B. (2015). A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks. In *Proceedings 40th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2015*, pages 5–pages.

[62] Moya, M. M., Koch, M. W., and Hostetler, L. D. (1993). One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N*, 93:24043.

[63] Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. *arXiv preprint arXiv:1606.00709*.

[64] Nun, I., Protopapas, P., Sim, B., and Chen, W. (2016). Ensemble learning method for outlier detection and its application to astronomical light curves. *The Astronomical Journal*, 152(3):71.

[65] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[66] Otomo, K., Kobayashi, S., Fukuda, K., and Esaki, H. (2019). Latent variable based anomaly detection in network system logs. *IEICE TRANSACTIONS on Information and Systems*, 102(9):1644–1652.

[67] Oza, P. and Patel, V. M. (2019). Active authentication using an autoencoder regularized cnn-based one-class classifier. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE.

[68] Pang, G., Cao, L., Chen, L., and Liu, H. (2017). Learning homophily couplings from non-iid data for joint feature selection and noise-resilient outlier detection. In *IJCAI*, pages 2585–2591.

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

[70] Perera, P., Nallapati, R., and Xiang, B. (2019). Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2898–2906.

[71] Perera, P. and Patel, V. M. (2018). Dual-minimax probability machines for one-class mobile active authentication. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE.

[72] Perera, P. and Patel, V. M. (2019). Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463.

[73] Pfahringer, B. (2000). Winning the kdd99 classification cup: bagged boosting. *ACM SIGKDD Explorations Newsletter*, 1(2):65–66.

[74] Pidhorskyi, S., Almohsen, R., and Doretto, G. (2018). Generative probabilistic novelty detection with adversarial autoencoders. In *Advances in neural information processing systems*, pages 6822–6833.

[75] Pol, A. A., Berger, V., Germain, C., Cerminara, G., and Pierini, M. (2019). Anomaly detection with conditional variational autoencoders. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1651–1657. IEEE.

[76] Potluri, S. and Diedrich, C. (2016). Accelerated deep neural networks for enhanced intrusion detection system. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE.

[77] Radovanović, M., Nanopoulos, A., and Ivanović, M. (2014). Reverse nearest neighbors in unsupervised distance-based outlier detection. *IEEE transactions on knowledge and data engineering*, 27(5):1369–1382.

[78] Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438.

[79] Rannen, A., Aljundi, R., Blaschko, M. B., and Tuytelaars, T. (2017). Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1320–1328.

[80] Rashid, M. M., Amar, M., Gondal, I., and Kamruzzaman, J. (2016). A data mining approach for machine fault diagnosis based on associated frequency patterns. *Applied Intelligence*, 45(3):638–651.

[81] Ravanbakhsh, M., Nabi, M., Sangineto, E., Marcenaro, L., Regazzoni, C., and Sebe, N. (2017). Abnormal event detection in videos using generative adversarial nets. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1577–1581. IEEE.

[82] Ritter, G. and Gallegos, M. T. (1997). Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern recognition letters*, 18(6):525–539.

[83] Ro, K., Zou, C., Wang, Z., and Yin, G. (2015). Outlier detection for high-dimensional data. *Biometrika*, 102(3):589–599.

[84] Roberts, S. J. (1999). Novelty detection using extreme value statistics. *IEE Proceedings-Vision, Image and Signal Processing*, 146(3):124–129.

[85] Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. (2018). Deep one-class classification. In *International conference on machine learning*, pages 4393–4402.

[86] Russo, S., Disch, A., Blumensaat, F., and Villez, K. (2020). Anomaly detection using deep autoencoders for in-situ wastewater systems monitoring data. *arXiv preprint arXiv:2002.03843*.

[87] Sabokrou, M., Fayyaz, M., Fathy, M., Moayed, Z., and Klette, R. (2018). Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, 172:88–97.

[88] Sakurada, M. and Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11.

[89] Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G., and Schmidt-Erfurth, U. (2019). f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44.

[90] Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer.

[91] Schreyer, M., Sattarov, T., Schulze, C., Reimer, B., and Borth, D. (2019). Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks. *arXiv preprint arXiv:1908.00734*.

[92] Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., and Kluger, Y. (2018). Spectralnet: Spectral clustering using deep neural networks. *arXiv preprint arXiv:1801.01587*.

[93] Slavic, G., Campo, D., Baydoun, M., Marin, P., Martin, D., Marcenaro, L., and Regazzoni, C. (2020). Anomaly detection in video data based on probabilistic latent space models. In *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pages 1–8. IEEE.

[94] Sodemann, A. A., Ross, M. P., and Borghetti, B. J. (2012). A review of anomaly detection in automated surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1257–1272.

[95] Suh, S., Chae, D. H., Kang, H.-G., and Choi, S. (2016). Echo-state conditional variational autoencoder for anomaly detection. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1015–1022. IEEE.

[96] Sun, G., Cong, Y., and Xu, X. (2018a). Active lifelong learning with" watchdog". In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[97] Sun, J., Wang, X., Xiong, N., and Shao, J. (2018b). Learning sparse representation with variational auto-encoder for anomaly detection. *IEEE Access*, 6:33353–33361.

[98] Tax, D. M. and Duin, R. P. (1999). Support vector domain description. *Pattern recognition letters*, 20(11-13):1191–1199.

[99] Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. (2016). Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798.

[100] Wang, G., Yang, J., and Li, R. (2017). Imbalanced svm-based anomaly detection algorithm for imbalanced training datasets. *Etri Journal*, 39(5):621–631.

[101] Wang, Y., Liu, M., Bao, Z., and Zhang, S. (2019). Stacked sparse autoencoder with pca and svm for data-based line trip fault diagnosis in power systems. *Neural Computing and Applications*, 31(10):6719–6731.

[102] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

[103] Xu, J. and Durrett, G. (2018). Spherical latent spaces for stable variational autoencoders. *arXiv preprint arXiv:1808.10805*.

[104] Yang, X., Deng, C., Zheng, F., Yan, J., and Liu, W. (2019). Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4066–4075.

[105] Yang, X., Latecki, L. J., and Pokrajac, D. (2009). Outlier detection with globally optimal exemplar-based gmm. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 145–154. SIAM.

[106] Yeh, I.-C. and Lien, C.-h. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480.

[107] Yeung, D.-Y. and Chow, C. (2002). Parzen-window network intrusion detectors. In *Object recognition supported by user interaction for service robots*, volume 4, pages 385–388. IEEE.

[108] You, C., Robinson, D. P., and Vidal, R. (2017). Provable self-representation based outlier detection in a union of subspaces. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 3395–3404.

[109] Yu, Q., Kavitha, M., and Kurita, T. (2019). Detection of one dimensional anomalies using a vector-based convolutional autoencoder. In *Asian Conference on Pattern Recognition*, pages 516–529. Springer.

[110] Yu, S. and Principe, J. C. (2019). Understanding autoencoders with information theoretic concepts. *Neural Networks*, 117:104–123.

[111] Zenati, H., Foo, C. S., Lecouat, B., Manek, G., and Chandrasekhar, V. R. (2018). Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*.

[112] Zhao, B., Fei-Fei, L., and Xing, E. P. (2011). Online detection of unusual events in videos via dynamic sparse coding. In *CVPR 2011*, pages 3313–3320. IEEE.

[113] Zimek, A., Schubert, E., and Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387.

[114] Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. (2018). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*.

Appendix A

Caculation of maximazation of mutual information

A.1

In section 3.2, we have defined the loss function (Eqn. 15) as follows

$$\begin{aligned} L_{MMI} &= \lambda_{KLD} KL(P(Z)|Q(Z)) - \lambda I(X,Z) - \lambda_O I(X,Y) - \lambda_H I(L_1, L_1') \\ &= \lambda_{KLD} \int p(\mathbf{z}) log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z} - \lambda \int \int p(\mathbf{z}|x) p(x) log \frac{p(\mathbf{z}|x)}{p(\mathbf{z})} dx d\mathbf{z} \\ &- \lambda_O \int \int p(y|x) p(x) log \frac{p(y|x)}{p(y)} dx dy - \lambda_H \int \int p(l_1'|l_1) p(l_1) log \frac{p(l_1'|l_1)}{p(l_1')} dl_1 dl_1' \end{aligned} \tag{A.1}$$

where λ_{KLD}, λ, λ_O and λ_H are the weighting parameters used to adjust the impact of individual losses on the overall objective function.

We transform Eq. B.1 to obtain the following:

$$\begin{aligned} L_{MMI} &= \lambda_{KLD} \int p(\mathbf{z}) log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z} - \lambda \int \int p(\mathbf{z}|x) p(x) log \frac{p(\mathbf{z}|x)}{p(\mathbf{z})} dx d\mathbf{z} \\ &- \lambda_O \int \int p(y|x) p(x) log \frac{p(y|x)}{p(y)} dx dy - \lambda_H \int \int p(l_1'|l_1) p(l_1) log \frac{p(l_1'|l_1)}{p(l_1')} dl_1 dl_1' \\ &= \int p(\mathbf{z}|x) p(x) \Big[\lambda_{KLD} log \frac{p(\mathbf{z}|x)}{q(\mathbf{z})} - (\lambda_{KLD} + \lambda) \frac{p(\mathbf{z}|x)}{p(\mathbf{z})} \Big] dx d\mathbf{z} \\ &- \lambda_O \int \int p(y|x) p(x) log \frac{p(y|x)}{p(y)} dx dz - \lambda_H \int \int p(l_1'|l_1) p(l_1) log \frac{p(l_1'|l_1)}{p(l_1')} dl_1 dl_1' \end{aligned} \tag{A.2}$$

We define $\lambda_L = \lambda_{KLD} + \lambda$, and thus Eqn. B.20 can be writen as follows:

$$\begin{aligned} L_{MMI} =& \lambda_{KLD} E_{x\sim p(x)}[D_{KL}(P(Z|X)||Q(Z))] - \lambda_L \int\int p(\mathbf{z}|x)p(x)log\frac{p(\mathbf{z}|x)}{p(\mathbf{z})}dxdz \\ &- \lambda_O \int\int p(y|x)p(x)log\frac{p(y|x)}{p(y)}dxdz \\ &- \lambda_H \int\int p(l_1'|l_1)p(l_1)log\frac{p(l_1'|l_1)}{p(l_1')}dl_1 dl_1' \end{aligned} \tag{A.3}$$

The first term of the loss function can be simply expressed as follows

$$E_{x\sim p(x)}[D_{KL}(P(Z|X)||Q(Z))] = \sum_{x\in X}\frac{1}{2}(-log\sigma^2(x) + \mu^2(x) + \sigma^2(x) + 1), x \in X, \tag{A.4}$$

where $\sigma(.)$ and $\mu(.)$ represent the mean and standard deviations given x, respectively [44].

Then, Eq. B.3 is converted into KL divergence as follows:

$$\begin{aligned} I(X,Z) &= \int\int p(\mathbf{z}|x)p(x)log\frac{p(\mathbf{z}|x)}{p(\mathbf{z})}dxdz \\ &= \int\int p(\mathbf{z}|x)p(x)log\frac{p(\mathbf{z}|x)p(x)}{p(\mathbf{z})p(x)}dxdz \\ &= D_{KL}(p(\mathbf{z}|x)p(x)||p(\mathbf{z})p(x)) \end{aligned} \tag{A.5}$$

Similarly, $I(X,Y)$ and $I(L_1, L_1')$ can be expressed as follows, relatively

$$I(X,Y) = D_{KL}(p(y|x)p(x)||p(y)p(x)) \tag{A.6}$$

$$I(l_1, l_1') = D_{KL}(p(l_1'|l_1)p(l_1)||p(l_1')p(l_1)) \tag{A.7}$$

It should be noted that KLD theoretically has no upper limit, but maximizing a quantity without an upper bound is likely to lead to outputting infinite results. Therefore, to perform optimization more effectively, we consider that the characteristic of maximizing MI is to widen the distance between $p(\mathbf{z}|x)p(x)$ and $p(\mathbf{z})p(x)$; accordingly, instead of KL divergence, we switch to Jensen-Shannon divergence (JSD), which is a measure with an upper bound and it is defined as follows:

$$D_{JS}(P,Q) = \frac{1}{2}D_{KL}(P|\frac{P+Q}{2}) + \frac{1}{2}D_{KL}(Q|\frac{P+Q}{2}) \tag{A.8}$$

The loss function according to Eq. 7 can be rewritten as follows:

$$\begin{aligned} L_{MMI} &= \lambda_{KLD} E_{x\sim p(x)}[D_{KL}(P(Z|X)\|Q(Z))] \\ &\quad - \lambda_L \cdot (E_{(x,\mathbf{z})\sim p(\mathbf{z}|x)p(x)}[logH(x,\mathbf{z})] + E_{(x,\mathbf{z})\sim p(\mathbf{z})p(x)}[log(1-H(x,\mathbf{z}))]) \\ &\quad - \lambda_O \cdot (E_{(x,y)\sim p(\mathbf{z}|x)p(x)}[logH(x,y)] + E_{(x,y)\sim p(y)p(x)}[log(1-H(x,y))]) \\ &\quad - \lambda_H \cdot (E_{(l_1,l_1')\sim p(l_1'|l_1)p(l_1)}[logH(l_1,l_1')] + E_{(l_1,l_1')\sim p(l_1')p(l_1)}[log(1-H(l_1,l_1'))]) \end{aligned} \tag{A.9}$$

where $H(.) = \frac{1}{1+exp(-v(.))}$, $v(.)$ is an objective function defined from the proposed MI criterion according to [31].

Appendix B

Results

B.1

Table B.1 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR10

Method	PLANE	CAR	BIRD	CAT	DEER	DOG
GMM	0.635	0.433	0.618	0.498	0.733	0.515
KDE	0.658	0.520	0.657	0.497	0.727	0.496
CAE	0.606	0.271	0.655	0.549	0.701	0.532
VAE	0.700	0.386	**0.679**	0.535	**0.748**	0.523
Pix CNN	**0.788**	0.428	0.617	0.574	0.511	0.571
GAN	0.708	0.458	0.664	0.510	0.722	0.505
SKG	0.717	0.494	0.662	0.527	0.736	0.504
AnoGAN	0.671	0.547	0.529	0.545	0.651	0.603
OCGAN	0.757	0.531	0.640	**0.620**	0.723	**0.620**
Ours	0.682	**0.614**	0.604	**0.620**	0.704	0.562

Method	FROG	HORSE	SHIP	TRUCK	MEAN
GMM	0.696	0.540	0.675	0.531	0.5874
KDE	**0.758**	0.564	0.680	0.540	0.6097
CAE	0.537	0.408	0.653	0.322	0.5234
VAE	0.687	0.493	0.696	0.386	0.5833
Pix CNN	0.422	0.454	0.715	0.426	0.5506
GAN	0.707	0.471	0.713	0.458	0.5916
SKG	0.726	0.560	0.680	0.566	0.6172
AnoGAN	0.585	0.625	0.758	0.665	0.6179
OCGAN	0.723	0.575	**0.820**	0.554	0.6566
Ours	0.734	**0.639**	0.756	**0.675**	**0.6590**

Table B.2 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning CIFAR100

Method	APPLE	BED	BICYCLE	ELEPHANT	TRUCK	PINE TREE
GMM	0.521	0.602	0.661	0.660	0.576	0.733
KDE	**0.714**	0.593	0.695	0.631	0.586	0.709
CAE	0.440	0.414	0.456	0.601	0.592	0.589
VAE	0.445	0.424	0.476	0.681	0.594	0.587
Pix CNN	0.484	0.393	0.422	0.654	0.517	0.462
GAN	0.399	0.370	0.422	0.532	0.594	0.587
SKG	0.380	0.388	0.456	0.613	0.609	0.603
AnoGAN	0.289	0.367	0.411	0.536	0.606	0.592
OCGAN	0.653	0.623	0.711	0.651	0.560	0.720
Ours	0.530	**0.672**	**0.777**	**0.766**	**0.673**	**0.811**

Method	ROCKET	TELEPHONE	TRAIN	TURTLE	MEAN
GMM	0.676	0.528	0.645	0.568	0.6170
KDE	**0.772**	0.446	0.662	0.648	0.6456
CAE	0.450	0.238	0.571	0.565	0.4912
VAE	0.456	0.269	0.574	0.575	0.5081
Pix CNN	0.419	0.688	0.444	0.663	0.5146
GAN	0.456	0.280	0.564	0.570	0.4774
SKG	0.417	0.286	0.549	0.585	0.4886
AnoGAN	0.406	0.282	0.591	0.598	0.4678
OCGAN	0.770	0.563	0.627	0.648	0.6526
Ours	0.729	**0.752**	**0.754**	**0.664**	**0.7128**

Table B.3 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning STL-10

Method	PLANE	BIRD	CAR	CAT	DEER	DOG
GMM	0.694	0.595	0.583	0.631	0.739	0.564
KDE	0.625	**0.610**	0.570	0.578	0.663	0.553
CAE	0.654	0.560	0.332	0.652	0.698	**0.613**
VAE	0.659	0.601	0.403	0.635	0.728	0.584
Pix CNN	0.592	0.595	0.228	0.591	0.703	0.546
GAN	0.362	0.454	0.358	0.459	0.716	0.499
SKG	0.373	0.535	0.466	0.615	0.681	0.527
AnoGAN	0.368	0.559	0.607	0.574	0.626	0.514
OCGAN	0.688	0.548	**0.627**	0.611	0.701	0.527
Ours	**0.712**	0.514	0.626	**0.690**	**0.762**	0.573

Method	HORSE	MONKEY	SHIP	TRUCK	MEAN
GMM	0.588	0.632	0.693	0.437	0.6156
KDE	0.549	0.603	0.608	0.483	0.5842
CAE	0.499	0.621	0.698	0.371	0.5698
VAE	0.584	0.635	0.699	0.414	0.5942
Pix CNN	0.498	0.576	0.433	0.240	0.5002
GAN	0.567	0.558	0.669	0.401	0.5043
SKG	0.422	0.563	0.581	0.392	0.5155
AnoGAN	0.407	0.560	0.541	0.456	0.5212
OCGAN	0.533	0.590	**0.751**	**0.601**	0.6177
Ours	**0.673**	**0.688**	0.691	0.519	**0.6448**

Table B.4 Performance comparison of CVAE-MMI and the state-of-the-art methods on each individual class in terms of AUC concerning IMAGENET

Method	BAG	BOTTLE CAP	BOX	CAR	PUMPKIN	DOG
GMM	0.514	0.564	0.605	0.554	0.439	0.600
KDE	0.486	0.527	0.532	0.495	0.519	0.531
CAE	0.466	0.456	0.651	0.426	0.768	0.682
VAE	0.524	0.544	0.629	0.636	0.369	0.589
Pix CNN	0.365	0.543	0.501	0.556	0.184	0.611
GAN	0.543	0.472	0.453	0.483	**0.826**	0.472
SKG	0.455	0.540	0.555	0.507	0.517	**0.733**
AnoGAN	**0.645**	0.514	0.491	0.493	0.740	0.491
OCGAN	0.579	**0.589**	0.657	0.538	0.809	0.538
Ours	0.576	0.577	**0.664**	**0.734**	0.803	0.684

Method	FISH	POT	ROOSTER	DRESS	MEAN
GMM	0.691	0.519	**0.654**	0.566	0.5706
KDE	0.608	0.509	0.546	0.557	0.5310
CAE	0.446	**0.573**	0.464	0.670	0.5602
VAE	0.603	0.560	0.681	0.634	0.5769
Pix CNN	0.478	0.501	0.597	0.575	0.4911
GAN	0.677	0.487	0.559	0.565	0.5537
SKG	0.517	0.540	0.510	0.531	0.5405
AnoGAN	0.709	0.486	0.600	0.611	0.5780
OCGAN	0.715	0.524	0.608	**0.668**	0.6225
Ours	**0.750**	0.553	0.613	0.629	**0.6583**