Supplementary Material for Perturbation Learning Based Anomaly Detection

Jinyu Cai
Fuzhou University
Shenzhen Research Institute of Big Data
The Chinese University of Hong Kong (Shenzhen)
jinyucai1995@gmail.com

Jicong Fan*
Shenzhen Research Institute of Big Data
The Chinese University of Hong Kong (Shenzhen)
fanjicong@cuhk.edu.cn

Abstract

This supplementary material includes the following content:

- Detailed settings of the network architecture, hyperparameter, and optimization for each experiment.
- Visualization of the features learned by PLAD.
- Abnormal data produced from the multi-class normal data mentioned in Section 4.5 of the main paper.
- Comparison between AE-based [Hinton and Salakhutdinov, 2006] perturbator and VAE-based [Kingma and Welling, 2013] perturbator in PLAD.
- Comparison between the fully connected network based VAE perturbator and convolutional neural network based VAE perturbator in PLAD.
- Ablation study and influence of different λ.
- Extension of PLAD to time-series anomaly detection.

A Detailed settings of the network architecture, hyperparameter, and optimization

CIFAR-10[2]**:** We use LeNet-based classifier with 4 convolutional layers of kernel size 5 and 3 linear layers for CIFAR-10 in this paper. For the perturbator, we use fully-connected network based VAE. Note that we do not perform dimension reduction in the perturbator because our aim is to learn the perturbation from the data. We use LeaklyReLU as the activation function for both the classifier and perturbator. The detailed network structure is shown in Table 1.

Fashion-MNIST[3]**:** We use LeNet-based classifier with 2 convolutional layers of kernel size 5 and 3 linear layers for Fashion-MNIST in this paper. The activation function and perturbator are similar to the settings for CIFAR-10. The detailed network structure is shown in Table 2.

*Jicong Fan is the corresponding author.

[2]https://www.cs.toronto.edu/ kriz/cifar.html

[3]https://www.kaggle.com/datasets/zalando-research/fashionmnist

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Thyroid[4] **and Arrhythmia**[5]**:** For Thyroid and Arrhythmia, we use the same fully-connected network based classifier constructed with single hidden layer. Denoting the dimensionality of input data as d, the size of each layer in the perturbator is then d. The detailed network structure is shown in Table 3.

The one-class classification of each class can be regarded as an independent task, therefore the desirable settings of parameters for each class would be different. To improve the reproducibility of the paper, we provide some recommended settings for each parameter in Tables 4 and 5, including the hyper-parameter λ, the choice of the optimizer (from Adam [Kingma and Ba, 2015] and SGD), and the learning rate.

Table 1: Architecture of the LeNet-based classifier and VAE-based perturbator for CIFAR-10.

LeNet-based Classifier
Conv2d(in_channel=3, out_channel=16, kernel_size=5, bias=False, padding=2)
BatchNorm2d(16, eps=1e-4, affine=False), Leaky_ReLU(), MaxPool2d(2,2)
Conv2d(in_channel=16, out_channel=32, kernel_size=5, bias=False, padding=2)
BatchNorm2d(32, eps=1e-4, affine=False), Leaky_ReLU(), MaxPool2d(2,2)
Conv2d(in_channel=32, out_channel=64, leaky_relu_size=5, bias=False, padding=2)
BatchNorm2d(64, eps=1e-4, affine=False), Leaky_ReLU(), MaxPool2d(2,2)
Conv2d(in_channel=64, out_channel=128, kernel_size=5, bias=False, padding=2)
BatchNorm2d(128, eps=1e-4, affine=False), Leaky_ReLU(), MaxPool2d(2,2)
Flatten()
Linear(128$\times$2$\times$2, 128, bias=False)
Leaky_ReLU()
Linear(128, 64, bias=False)
Leaky_ReLU()
Linear(64, 1, bias=False)
VAE-based Perturbator
Linear(3072, 3072)
Leaky_ReLU()
$\boldsymbol{\mu}$: Linear(3072, 3072); $\boldsymbol{\sigma}$: Linear(3072, 3072)
Reparameterzie($\boldsymbol{\mu}$, $\boldsymbol{\sigma}$)
Linear(3072, 3072)
Leaky_ReLU()
Linear(3072, 3072$\times$2)

B Visualization of the learned embedding space by PLAD

We further provide the visualization of the learned embedding space by PLAD method on Fashion-MNIST in Figures 1 and 2. Note that we use t-SNE [Van der Maaten and Hinton, 2008] method to process the training samples of each class together with the test set, i.e., to reduce their dimensionality to 3, and they are marked in different colors. We can observe that although the learned decision boundary of PLAD does not based on any assumption, it still adaptively distinguishes the normal samples and anomalies. Moreover, the training samples marked in blue and normal test samples marked in green are projected to the same space, which is consistent with our expectation of learning a space that accommodates only normal samples through the training data. Of course, the 3-D visualization shown is only for an intuitive understanding of PLAD and may not be the optimal decision boundary, so the visualization on some classes (such as Pullover and Shirt) is not desirable. PLAD may obtain a better decision boundary in a higher dimension.

C Illustration of the anomalies produced from the multi-class normal data

To better understand the experiment conducted in Section 4.5, we show the anomalies produced from pair-wise samples in Figure 3. We select part of pair-wise normal samples from CIFAR-10

[4]http://odds.cs.stonybrook.edu/thyroid-disease-dataset/

[5]http://odds.cs.stonybrook.edu/arrhythmia-dataset/

Table 2: Architecture of the LeNet-based classifier and VAE-based perturbator for Fashion-MNIST.

LeNet-based Classifier
Conv2d(in_channel=1, out_channel=16, kernel_size=5, bias=False, padding=2)
BatchNorm2d(16, eps=1e-4, affine=False), Leaky_ReLU(), MaxPool2d(2,2)
Conv2d(in_channel=16, out_channel=32, kernel_size=5, bias=False, padding=2)
BatchNorm2d(32, eps=1e-4, affine=False), Leaky_ReLU(), MaxPool2d(2,2)
Flatten()
Linear($32\times7\times7$, 128, bias=False)
Leaky_ReLU()
Linear(128, 64, bias=False)
Leaky_ReLU()
Linear(64, 1, bias=False)
VAE-based Perturbator
Linear(784, 784)
Leaky_ReLU()
$\boldsymbol{\mu}$: Linear(784, 784); $\boldsymbol{\sigma}$: Linear(784, 784)
Reparameterzie($\boldsymbol{\mu}$, $\boldsymbol{\sigma}$)
Linear(784, 784)
Leaky_ReLU()
Linear(784, 784×2)

Table 3: Architecture of the MLP-based classifier and VAE-based perturbator for non-image data (Thyroid and Arrhythmia).

MLP-based Classifier	VAE-based Perturbator
Input_dim = d	Linear(d, d), ReLU()
Linear(d, 20), ReLU()	$\boldsymbol{\mu}$: Linear(d, d); $\boldsymbol{\sigma}$: Linear(d, d)
Linear(20, 1)	Reparameterzie($\boldsymbol{\mu}$, $\boldsymbol{\sigma}$)
	Linear(d, d), ReLU()
	Linear(d, $d\times2$)

Table 4: Detailed settings of the hyper-parameter λ, optimizer and learning rate for the image datasets (CIFAR-10 and Fashion-MNIST) in the one-class anomaly detection task.

CIFAR-10				Fashion-MNIST			
Class	λ	Optimizer	Learning rate	Class	λ	Optimizer	Learning rate
Airplane	10	SGD	0.005	T-shirt	5	SGD	0.001
Automobile	5	SGD	0.005	Trouser	5	SGD	0.005
Bird	50	SGD	0.005	Pullover	3	SGD	0.005
Cat	5	SGD	0.005	Dress	3	SGD	0.005
Deer	5	SGD	0.005	Coat	5	SGD	0.005
Dog	10	SGD	0.005	Sandal	5	SGD	0.005
Frog	10	Adam	0.0001	Shirt	5	SGD	0.005
Horse	5	SGD	0.005	Sneaker	15	SGD	0.002
Ship	20	Adam	0.0001	Bag	5	SGD	0.001
Truck	5	SGD	0.001	Ankle boot	5	SGD	0.005

Table 5: Detailed settings of the optimizer, learning rate, and hyper-parameter λ for the non-image datasets (Thyroid and Arrhythmia) in the one-class anomaly detection task.

Thyroid			Arrhythmia		
λ	Optimizer	Learning rate	λ	Optimizer	Learning rate
3	Adam	0.001	2	Adam	0.001

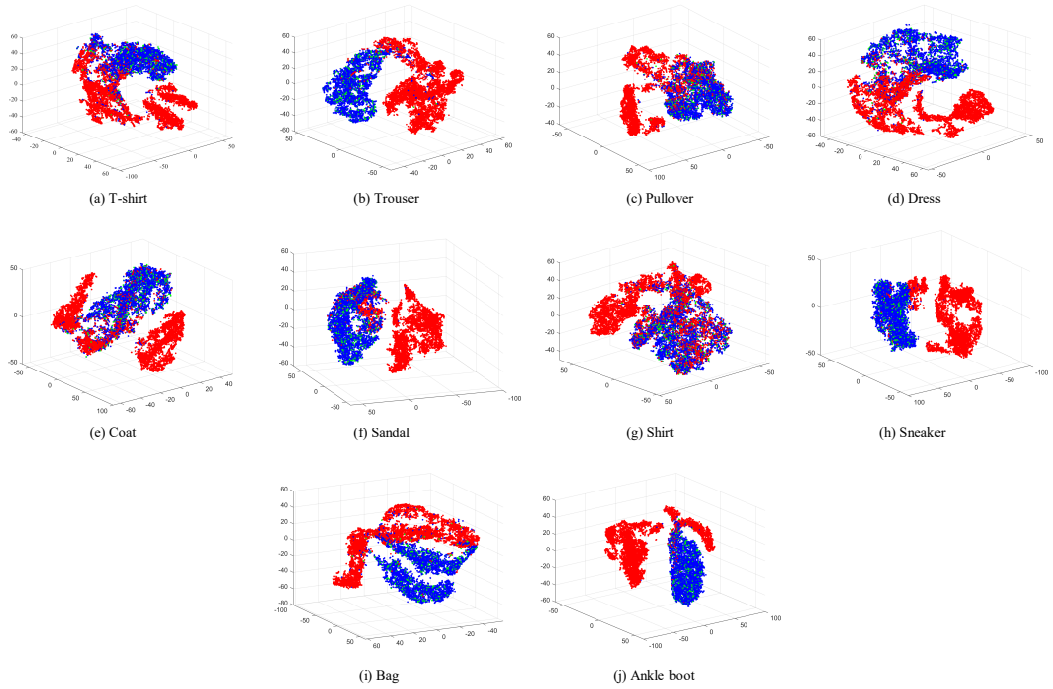


Figure 1: Visualization of the learned embedding space of PLAD on Fashion-MNIST by t-SNE. Note the points marked in blue, green, and red are training samples, normal test samples, and anomalous test samples, respectively.

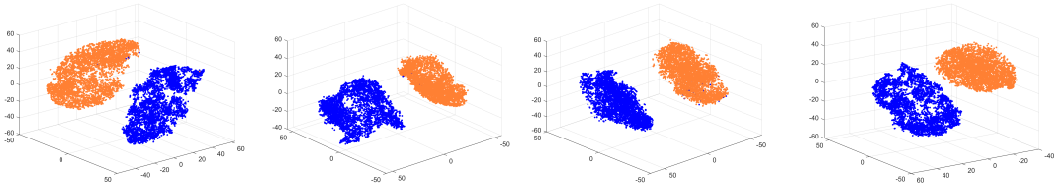
and Fashion-MNIST, then use their means in pixel level to produce anomalies. We can also observe from this figure that the anomalies produced in this way contain characteristics from multiple classes, which is a much more difficult anomaly detection task to solve because the decision boundary between the anomalous samples and the normal samples are very complicated.

D Comparison between the AE-based and VAE-based perturbators in PLAD

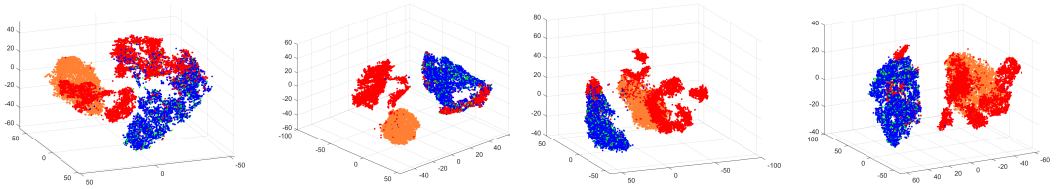
We further evaluate the performance of AE-based and VAE-based perturbators. Specifically, we build an AE-based perturbator, whose architecture is the same as the VAE-based one. Then we run the experiment on CIFAR-10 and Fashion-MNIST following the same experimental setup as mentioned in Appendix A. The experimental results are shown in Table 6. We can observe that the VAE-based perturbator generally performs better than the AE-based one in most classes, especially in the "Bird" and "Deer" classes on CIFAR-10, and in the "Shirt" and "Bag" classes on Fashion-MNIST. Nevertheless, the AE-based perturbator still achieves remarkable performance on some classes. For example, it outperforms VAE-based perturbator on "Truck", "Sandal" and "Ankle boot". Overall, the average performance of AE-based and VAE-based methods both outperform most competing methods in Section 4.3

E Comparison between the CNN-based perturbator and FCN-based perturbator in PLAD

As the above experiment can be seen, VAE-based perturbator generally performs better than AE-based one. Therefore, we further evaluate the performance of CNN-based and FCN-based VAE perturbators to provide a more comprehensive analysis. Similarly, we experiment on CIFAR-10 and Fashion-MNIST. The encoder of CNN-based perturbator is similar to the classifier, with two differences that it allows bias and contains two hidden layers to produce μ and σ for VAE. The dimension of hidden layer is set to 128, and the decoder is symmetric to the encoder. We show the



(a) T-shirt (b) Sandal (c) Sneaker (d) Ankle boot



(a) T-shirt (b) Sandal (c) Sneaker (d) Ankle boot

Figure 2: Visualization of the learned embedding space in two cases (with training and perturbed samples, and with training, perturbed and test samples, respectively) on Fashion-MNIST. Note we show four classes including "T-shirt", "Sandal", "Sneaker", and "Ankle boot", and the points marked in blue, orange, green, and red are training samples, perturbed samples, normal test samples, and anomalous test samples, respectively.

(a) Multi-class normal samples from CIFAR-10 (b) Multi-class normal samples from Fashion-MNIST (c) Anomalies produced from pair-wise samples

Figure 3: Illustration of the produced anomalies from the multi-class normal data in Section 4.5. (a) and (b) denotes the randomly selected pair-wise normal samples from CIFAR-10 and Fashion-MNIST, respectively. (c) denotes the anomalies produced by using pixel-level means of pair-wise normal samples from (a) and (b).

experimental results in Table 7. We can observe that the performance of CNN-based and FCN-based perturbators are comparable in most cases, except for the more remarkable performance achieved by the FCN-based perturbator on the "Airplane" and "Dog" classes of CIFAR-10. Nevertheless, the average performance of them are quite close, and it should be noted that the CNN-based perturbator may have an encouraging performance in more complex scenarios.

F Ablation study for PLAD

We conduct an ablation study on CIFAR-10 to validate the effectiveness of the proposed method, and also discuss the influence on the hyper-parameter λ to the anomaly detection performance. Specifically, we vary the value of λ in the range of $[0, 0.1, \ldots, 50, 100]$, and show the average AUCs in Figure 4. Note that the performance shown in the dotted line indicates the degradation model of PLAD, which drops the perturber, i.e., contains only a simple classification network without any perturbation learned. We have the following observation from this figure:

- The most significant AUC gap observed from Figure 4 is whether the perturbator is considered or not. Even with a very small value of λ (e.g., 0.1), we can see remarkable performance improvements on some classes such as "Airplane", "Bird", and ''Deer". This indicates that

Table 6: Comparison of the AE-based and VAE-based perturbators for the one-class classification tasks on CIFAR-10 and Fashion-MNIST.

CIFAR-10			Fashion-MNIST		
Class	AE-based	VAE-based	Class	AE-based	VAE-based
Airplane	79.7 ± 1.2	$\mathbf{82.5 \pm 0.4}$	T-shirt	92.3 ± 0.8	$\mathbf{93.1 \pm 0.5}$
Automobile	80.5 ± 0.8	$\mathbf{80.8 \pm 0.9}$	Trouser	98.1 ± 0.5	$\mathbf{98.6 \pm 0.2}$
Bird	62.3 ± 1.7	$\mathbf{68.8 \pm 1.2}$	Pullover	88.2 ± 0.8	$\mathbf{90.2 \pm 0.7}$
Cat	62.7 ± 1.5	$\mathbf{65.2 \pm 1.2}$	Dress	92.0 ± 0.5	$\mathbf{93.7 \pm 0.6}$
Deer	65.1 ± 2.8	$\mathbf{71.6 \pm 1.1}$	Coat	91.3 ± 0.9	$\mathbf{92.8 \pm 0.8}$
Dog	67.3 ± 1.5	$\mathbf{71.2 \pm 1.6}$	Sandal	$\mathbf{96.4 \pm 0.3}$	96.0 ± 0.4
Frog	72.1 ± 1.8	$\mathbf{76.4 \pm 1.9}$	Shirt	79.1 ± 0.8	$\mathbf{82.0 \pm 0.6}$
Horse	73.4 ± 1.7	$\mathbf{73.5 \pm 1.0}$	Sneaker	98.1 ± 0.2	$\mathbf{98.6 \pm 0.3}$
Ship	79.0 ± 1.4	$\mathbf{80.6 \pm 1.8}$	Bag	86.9 ± 1.6	$\mathbf{90.9 \pm 1.0}$
Truck	$\mathbf{81.0 \pm 0.7}$	80.5 ± 1.3	Ankle boot	$\mathbf{99.1 \pm 0.2}$	99.1 ± 0.1
Average	72.3	**75.1**	Average	92.1	**93.5**

Table 7: Performance of the CNN-based and FCN-based VAE perturbators for the one-class classification tasks on CIFAR-10 and Fashion-MNIST.

CIFAR-10			Fashion-MNIST		
Class	CNN-based	FCN-based	Class	CNN-based	FCN-based
Airplane	77.6 ± 1.1	$\mathbf{82.5 \pm 0.4}$	T-shirt	93.0 ± 0.4	$\mathbf{93.1 \pm 0.5}$
Automobile	80.2 ± 0.9	$\mathbf{80.8 \pm 0.9}$	Trouser	98.4 ± 0.2	$\mathbf{98.6 \pm 0.2}$
Bird	65.4 ± 1.5	$\mathbf{68.8 \pm 1.2}$	Pullover	88.9 ± 0.3	$\mathbf{90.2 \pm 0.7}$
Cat	64.6 ± 1.8	$\mathbf{65.2 \pm 1.2}$	Dress	$\mathbf{94.0 \pm 0.5}$	93.7 ± 0.6
Deer	$\mathbf{71.8 \pm 2.3}$	71.6 ± 1.1	Coat	91.9 ± 0.5	$\mathbf{92.8 \pm 0.8}$
Dog	67.4 ± 1.2	$\mathbf{71.2 \pm 1.6}$	Sandal	95.7 ± 0.4	96.0 ± 0.4
Frog	$\mathbf{76.6 \pm 0.8}$	76.4 ± 1.9	Shirt	$\mathbf{82.5 \pm 1.1}$	82.0 ± 0.6
Horse	70.6 ± 1.3	$\mathbf{73.5 \pm 1.0}$	Sneaker	98.5 ± 0.1	$\mathbf{98.6 \pm 0.3}$
Ship	80.1 ± 1.7	$\mathbf{80.6 \pm 1.8}$	Bag	$\mathbf{91.0 \pm 1.3}$	90.9 ± 1.0
Truck	$\mathbf{80.7 \pm 1.6}$	80.5 ± 1.3	Ankle boot	$\mathbf{99.2 \pm 0.1}$	99.1 ± 0.1
Average	73.5	**75.1**	Average	93.3	**93.5**

the proposed perturbator forces the classifier to learn more discriminative decision boundary to distinguish the normal samples and anomalies.

- Generally, we can observe the performance improves as the value of λ increases. Yet, as mentioned before, anomaly detection for each class is an independent task, so the desirable range of λ is different. For example, $\lambda \in [1, 5]$ performs well on classes "Automobile", "Deer", "Horse", and "Truck", while $\lambda \in [5, 20]$ performs well on classes "Bird", "Dog", "Frog", and "Ship".
- We can also see that the performance commonly degrades to some extent when λ is too large, except on the class 'Frog", where $\lambda = 100$ even performs better than $\lambda = 50$. Nevertheless, the proposed PLAD method shows robustness to the variation of λ and still achieves a relatively stable performance even against very large values.

G Extension of PLAD to time-series anomaly detection

In this section, we further conduct an experiment to demonstrate the effectiveness of PLAD on the time-series anomaly detection task. Specifically, we compare with five baseline methods on Epileptic Seizure, which is a multivariate time-series dataset [Andrzejak *et al.*, 2001].

Dataset description: Epileptic Seizure[6] consists of the EEG values recorded in 500 patients over 23.5 seconds, with 178 data points contained per second. Therefore, the dataset contains

[6]https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition

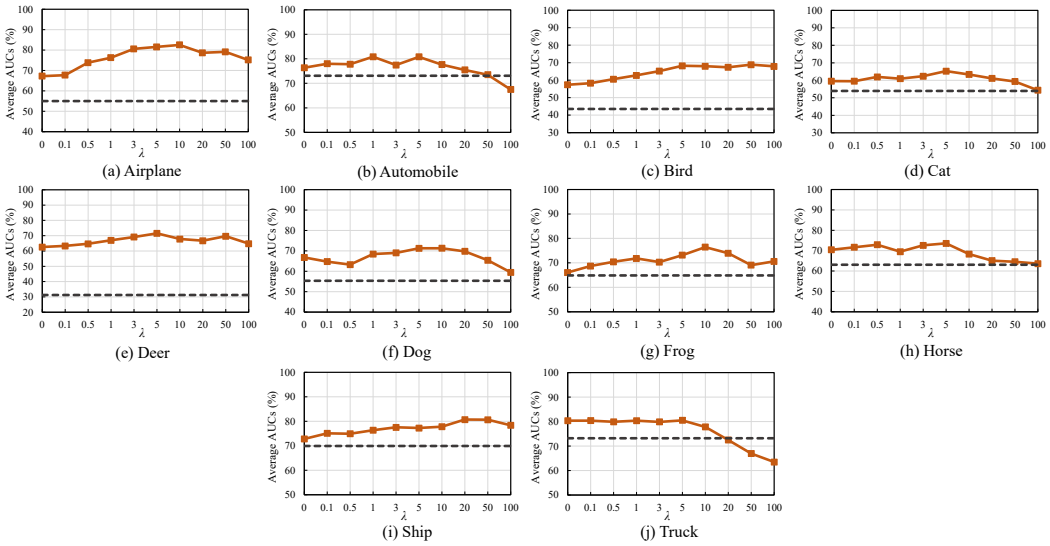


Figure 4: Average AUCs for the one-class classification task on each class of CIFAR-10 with λ varies in the range of $[0, 0.1, \ldots, 50, 100]$. Note that the dotted line denotes the performance of the degradation model that drops the perturbator.

$500 \times 23 = 11,500$ sequential data in total, with 5 categories of EEG values (each one has 2,300 samples). In this paper, we divide the dataset into two categories (epileptic seizure or not). The detection task is to recognize whether the EEG values are abnormal (i.e., is epileptic seizure activity or not).

Experimental Settings: We split the Epileptic Seizure dataset into a training set with 6,900 normal samples and a testing set with 4,600 samples (half normal and half abnormal). For fair comparison, we use a single layer LSTM for all experiments following the settings in [Goyal *et al.*, 2020] to handle time-series data. For the proposed PLAD method, we simply construct a single layer AE-based perturbator to learn perturbations, and connect a fully-connected (FC) layer after the hidden state of the last time step of the LSTM layer in the classifier to obtain anomaly detection results. We choose Adam as the optimizer with a learning rate of 1×10^{-5} and $\lambda = 0.001$. The comparative methods include k-NN, AE [Sakurada and Yairi, 2014], DAGMM [Zong *et al.*, 2018], DSVDD [Ruff *et al.*, 2018], and DROCC [Goyal *et al.*, 2020]. Each method is run 10 times to obtain the average AUC scores and their standard deviations.

Experimental results Table 8 summarizes the experimental results of each method on the Epileptic Seizure dataset. We can observe that PLAD significantly outperforms most baseline methods, e.g., k-NN, AE, DAGMM and DSVDD, and is competitive against with DROCC. The p-value v.s. DROCC also shows the effectiveness of our PLAD. Overall, the study demonstrates the potential and feasibility of PLAD to be extended to the time-series anomaly detection tasks.

Table 8: Average AUCs (%) $\pm$ std of multivariate time-series anomaly detection task on the Epileptic Seizure dataset.

Method	k-NN	AE	DAGMM	DSVDD	DROCC	PLAD	p-value (t-test) v.s DROCC
AUC $\pm$ std	91.7 $\pm$ 0.0	91.5 $\pm$ 1.9	87.0 $\pm$ 1.07	94.3 $\pm$ 2.1	98.1 $\pm$ 0.5	**98.6** $\pm$ **0.3**	0.03

References

Ralph G Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of

brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.

Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. Drocc: Deep robust one-class classification. In *Proceedings of the International Conference on Machine Learning*, pages 3711–3721. PMLR, 2020.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

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

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the International Conference on Machine Learning*, pages 4393–4402. PMLR, 2018.

Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *Proceedings of the International Conference on Learning Representations*, 2018.