33rd ACM International Conference on
Information and Knowledge Management

Transformer for Point Anomaly Detection

Harim Kim[1], Chang Ha Lee[2], Charmgil Hong[1]

{**hrkim**, charmgil}@handong.ac.kr, {yielding}@gmdsoft.com

[1]Handong Global University
[2]GMDSOFT

GitHub

Introduction

Unsupervised Anomaly Detection

Introduction

Unsupervised Anomaly Detection

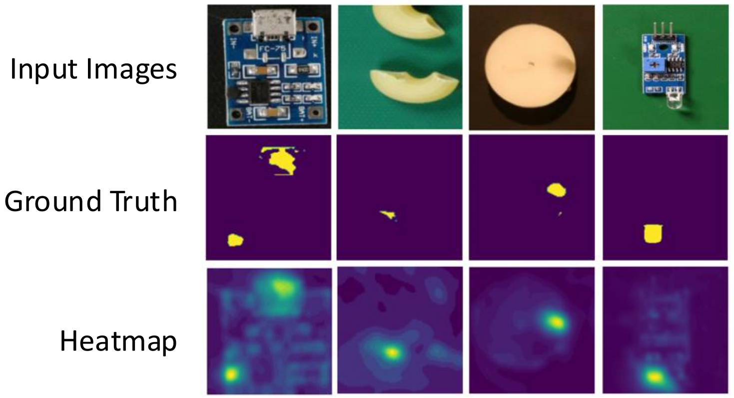
- Unsupervised anomaly detection aims to identify anomalies in data without the need for prior labeling information [Aggarwal et al., 2017]
- It typically operates under the assumption that statistical outliers are indicative of anomalies
- It has gained constant interest in various areas, including industrial manufacturing [Liu et al., 2018], cybersecurity [Alom et al., 2017], and healthcare [Pereira et al., 2019]

Introduction

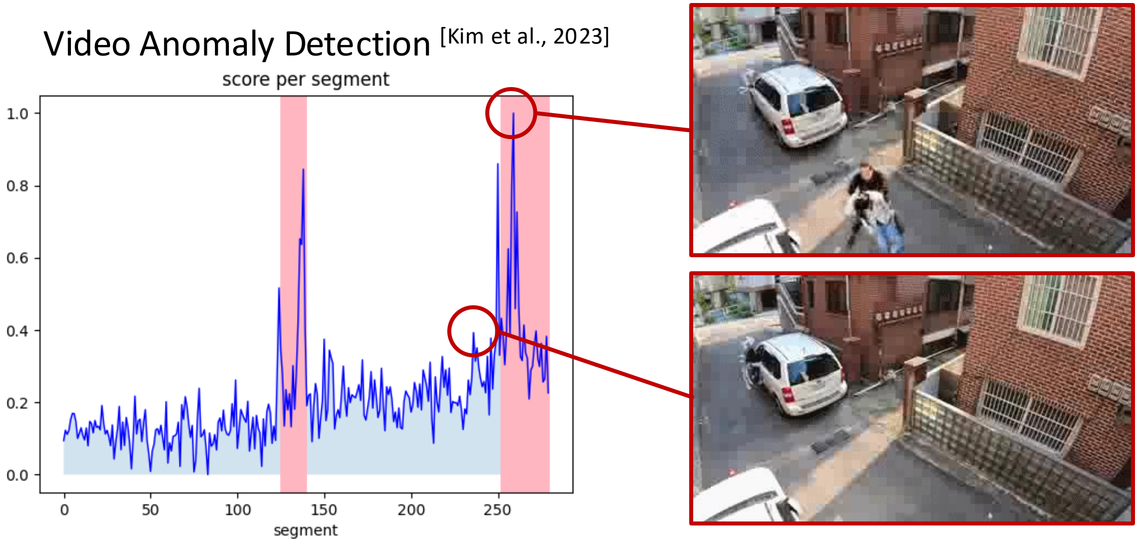
Unsupervised Anomaly Detection

- Unsupervised anomaly detection aims to identify anomalies in data without the need for prior labeling information [Aggarwal et al., 2017]
- It typically operates under the assumption that statistical outliers are indicative of anomalies
- It has gained constant interest in various areas, including industrial manufacturing [Liu et al., 2018], cybersecurity [Alom et al., 2017], and healthcare [Pereira et al., 2019]

Anomaly Localization [Mousakhan et al., 2023]



Video Anomaly Detection [Kim et al., 2023]



Introduction

Unsupervised Anomaly Detection

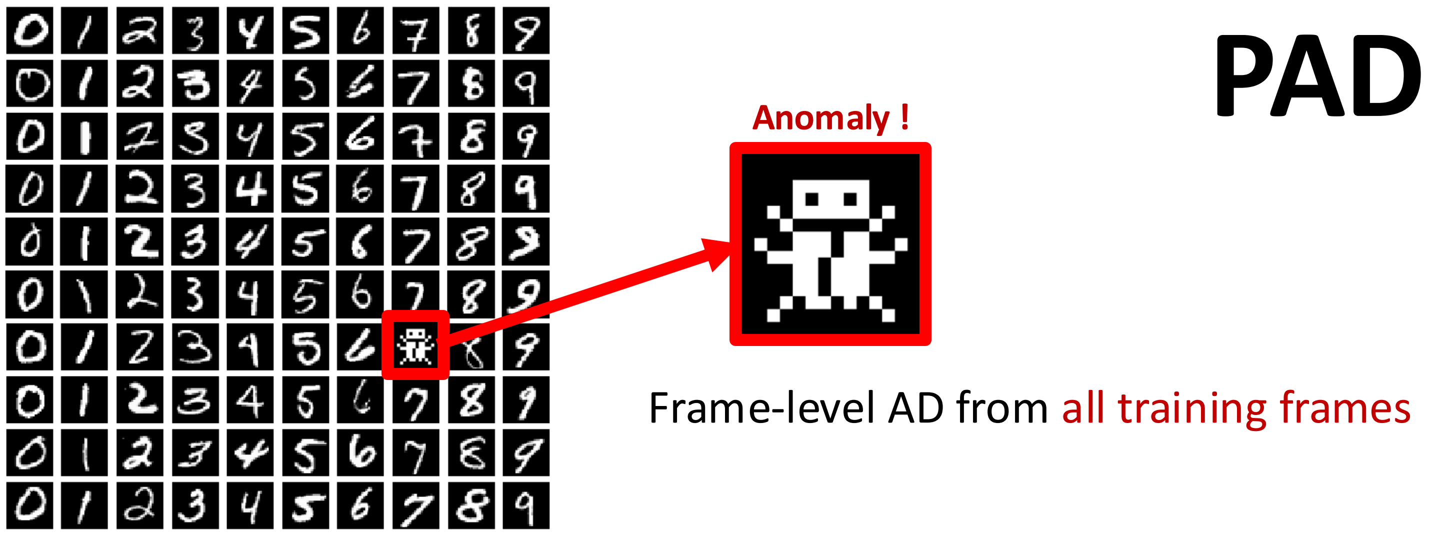
- **Point Anomaly Detection (PAD)**
  - Point anomalies are individual instances considered unusual compared to the majority of other individual instances [Pang et al., 2021]

HAIL

Introduction

Unsupervised Anomaly Detection

- **Point Anomaly Detection (PAD)**
  - Point anomalies are individual instances considered unusual compared to the majority of other individual instances [Pang et al., 2021]
  - It is like frame-level anomaly detection across all training frames



All training frames in MNIST

Two Key Components for Deep PAD Method

- Which network architecture should we use?
- How should the objective function and anomaly score be defined?

HAIL

Introduction

First Key Component : Which Network Architecture Should We Use?

- We want to use the Transformer-based architecture for anomaly detection
  - To train the Transformer-based methods, the input data needs to be an inter-dependent sequence

HAIL

Introduction

First Key Component : Which Network Architecture Should We Use?

- We want to use the Transformer-based architecture for anomaly detection
  - To train the Transformer-based methods, the input data needs to be an inter-dependent sequence
    - In the case of PAD, we cannot define the inter-dependent sequences, because the data points are assumed to be i.i.d.

HAIL

Introduction

First Key Component : Which Network Architecture Should We Use?

- We want to use the Transformer-based architecture for anomaly detection
  - To train the Transformer-based methods, the input data needs to be an inter-dependent sequence
    - In the case of PAD, we cannot define the inter-dependent sequences, because the data points are assumed to be i.i.d.

First Challenge: How should we define the input sequence for training Transformer-based PAD method

HAIL

Introduction

Second Key Component:
How Should the Objective Function and Anomaly Score Be Defined?

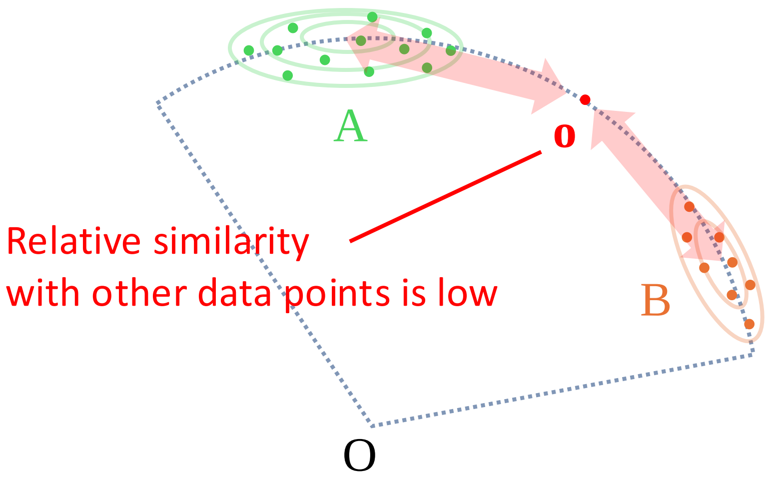
- Transformer generalizes the input data points according to their inter-similarity using attention weight

HAIL

Introduction

Second Key Component:
How Should the Objective Function and Anomaly Score Be Defined?

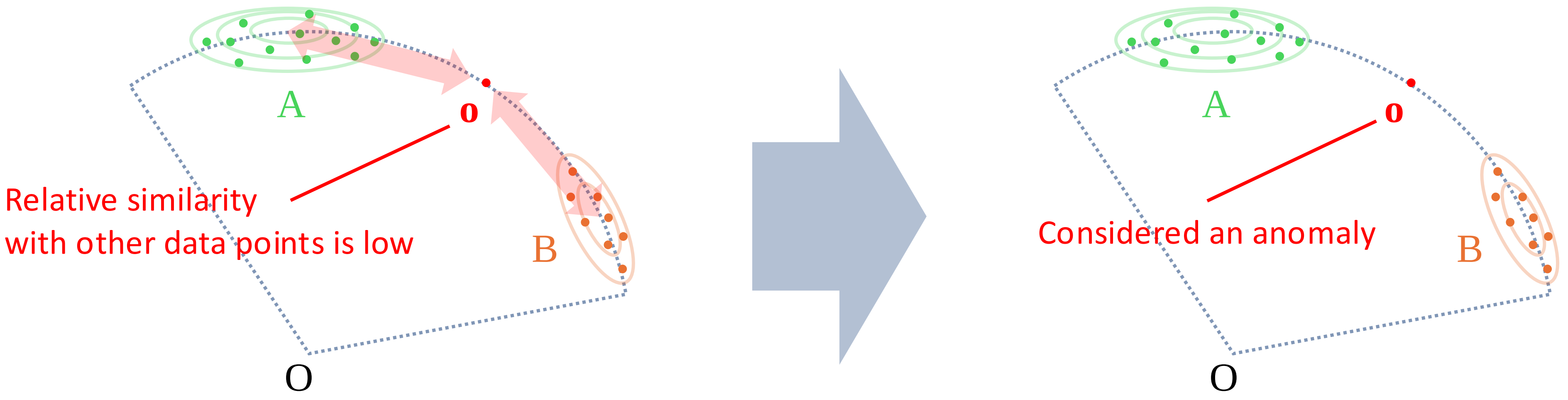
- Transformer generalizes the input data points according to their inter-similarity using attention weight
  - Assuming that the anomaly data points are relatively dissimilar with the other data points, we can utilize the attention weight to calculate the anomaly score



Introduction

Second Key Component: How Should the Objective Function and Anomaly Score Be Defined?

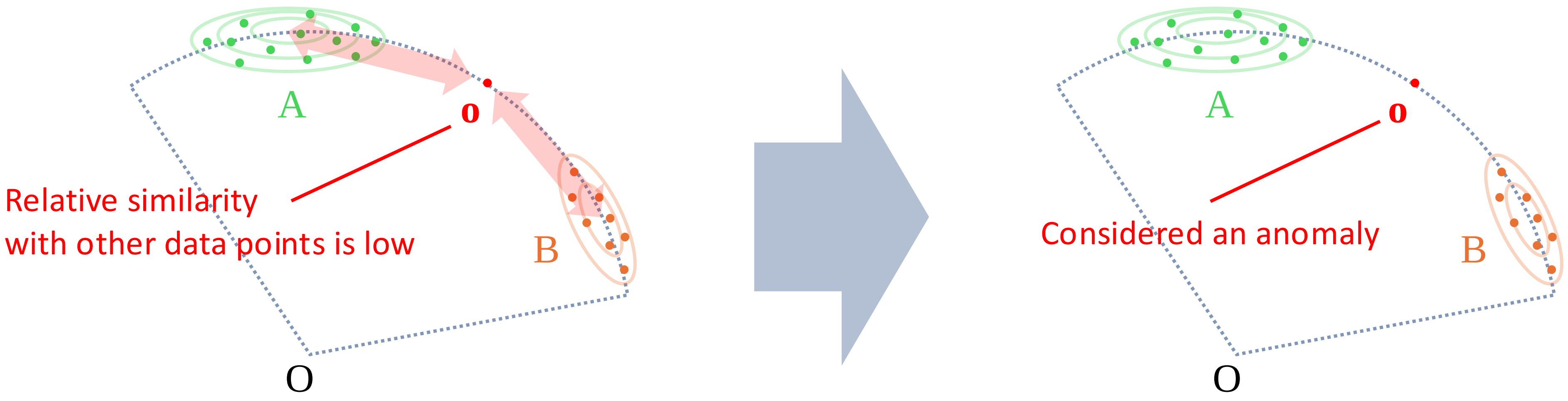
- Transformer generalizes the input data points according to their inter-similarity using attention weight
  - Assuming that the anomaly data points are relatively dissimilar with the other data points, we can utilize the attention weight to calculate the anomaly score



Introduction

Second Key Component:
How Should the Objective Function and Anomaly Score Be Defined?

- Transformer generalizes the input data points according to their inter-similarity using attention weight
  - Assuming that the anomaly data points are relatively dissimilar with the other data points, we can utilize the attention weight to calculate the anomaly score
  - However, the attention weight are calculated only among the data points within the input sequence

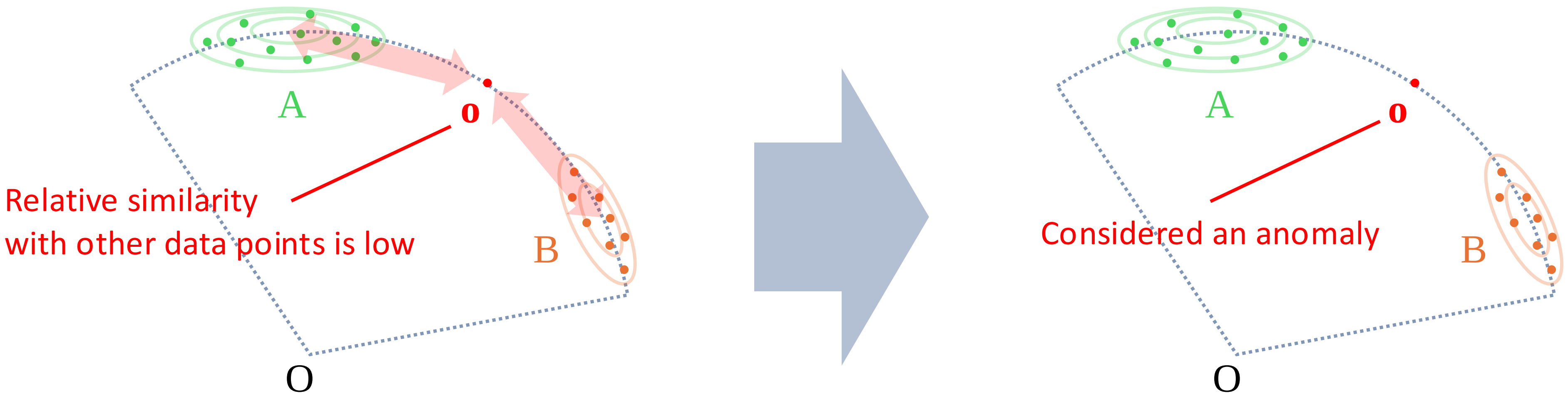


Introduction

Second Key Component:
How Should the Objective Function and Anomaly Score Be Defined?

- Transformer generalizes the input data points according to their inter-similarity using attention weight
  - Assuming that the anomaly data points are relatively dissimilar with the other data points, we can utilize the attention weight to calculate the anomaly score
  - However, the attention weight are calculated only among the data points within the input sequence

2nd Challenge: What algorithm can we utilize to obtain an anomaly score considering entire data points for PAD



HAIL

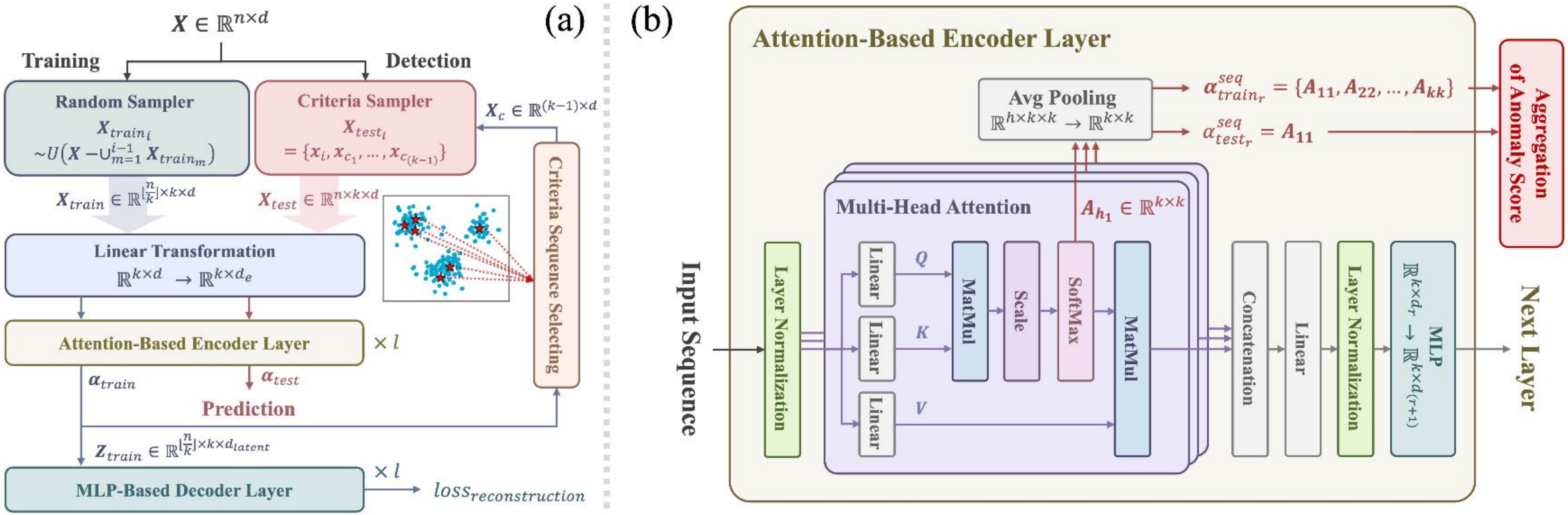
Contributions

- We propose a novel Transformer-based approach for PAD, called **TransPAD**
- This includes suitable sampling strategies for obtaining input sequence during training and detection phase
- Our approach consistently outperforms existing methods on a range of benchmark tabular datasets

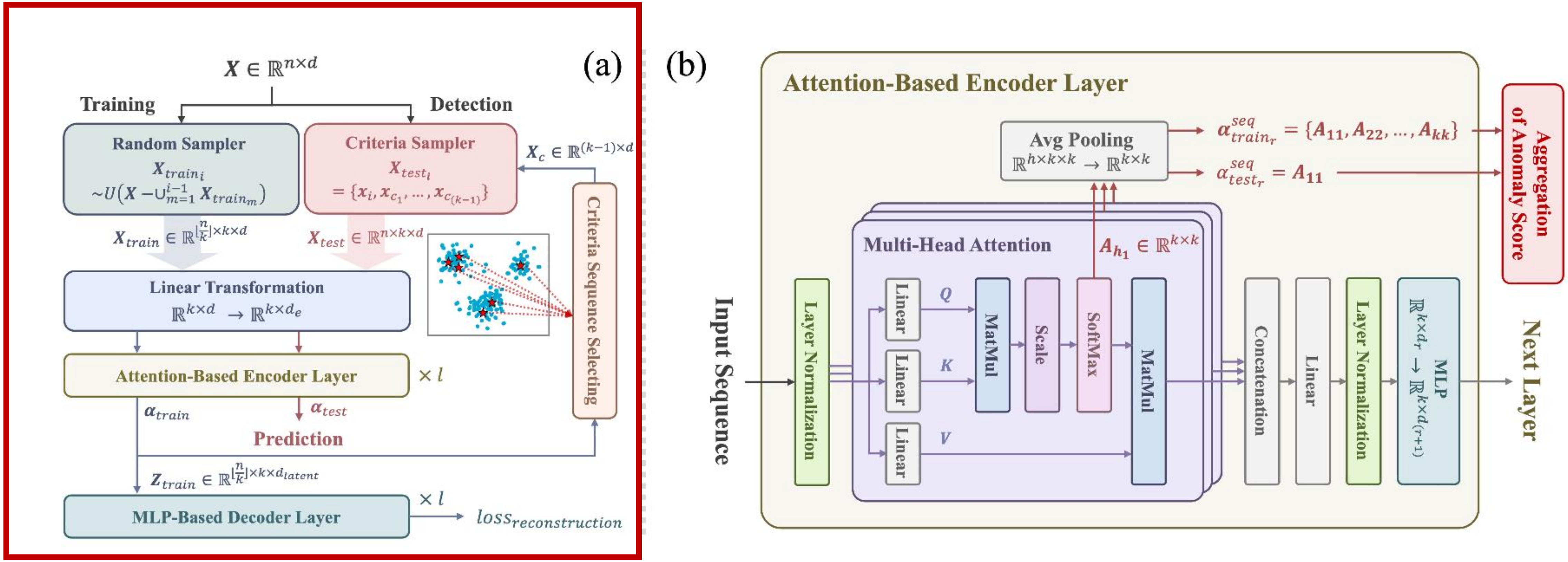
HAIL

Proposed Method

TransPAD: Transformer for Point Anomaly Detection

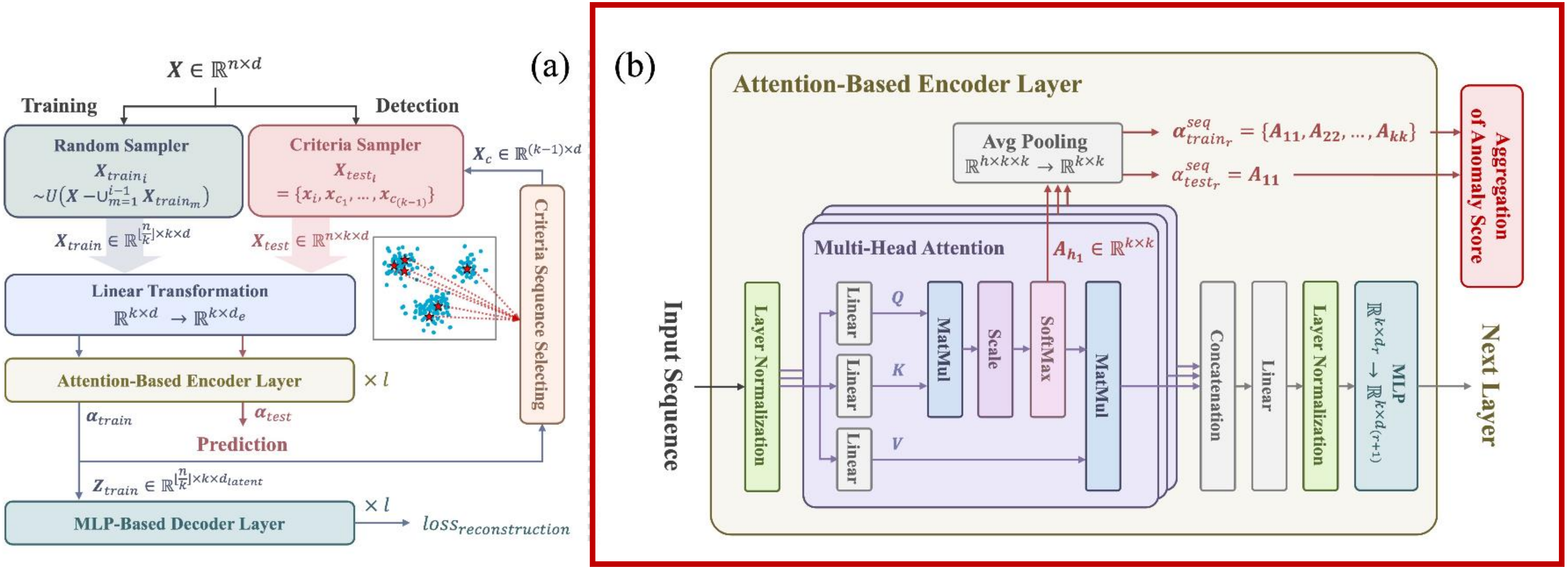


TransPAD: Transformer for Point Anomaly Detection



Proposed Method

TransPAD: Transformer for Point Anomaly Detection



Proposed Method

TransPAD: Transformer for Point Anomaly Detection

- Training
  - We utilize random sampling without replacement to obtain the input sequence
  - It makes the model consider the inter-dependencies for entire training data
  - We use the reconstruction loss for training

Random Sampler

$$\mathbf{X}_{train_i} \sim U(\mathbf{X} - \cup_{m=1}^{i-1}\mathbf{X}_{train_m})$$

$$\mathbf{X}_{train} = \{\mathbf{X}_{train_1}, \mathbf{X}_{train_2}, \ldots, \mathbf{X}_{train_{\lfloor \frac{n}{k} \rfloor}}\}$$

$$\mathbf{X}_{train} \in \mathbb{R}^{\lfloor \frac{n}{k} \rfloor \times k \times d}$$

$$loss_{recons} = \frac{1}{k}\sum_{i=1}^{k} \| \mathbf{X}_{train_i} - \mathbf{X}'_{train_i} \|_2^2$$

Algorithm 1 The Training Process of *TransPAD*

Define *TransPAD* network $\phi(\cdot)$
for number of training epochs **do**
  for $i \leftarrow 1$ to $\lfloor \frac{n}{k} \rfloor$ **do**
    Sample sequence $\mathbf{X}_{train_i} \sim U(\mathbf{X} - \cup_{m=1}^{i-1}\mathbf{X}_{train_m})$
    Add $\mathbf{X}_{train_i}$ to $\mathbf{X}_{train}$
  end for
  $\mathbf{X}_{train} \in \mathbb{R}^{\lfloor \frac{n}{k} \rfloor \times k \times d}$
  for each mini-batch $\mathbf{X}_{batch}$ from $\mathbf{X}_{train}$ **do**
    Initialize total loss: $loss_{total} \leftarrow 0$
    for each sequence $\mathbf{X}_{seq}$ in $\mathbf{X}_{batch}$ **do**
      get reconstructed $\mathbf{X}'_{seq} \leftarrow \phi(\mathbf{X}_{seq})$
      $loss_{recons} \leftarrow \frac{1}{k}\sum_{i=1}^{k} \| \mathbf{X}_{seq_i} - \mathbf{X}'_{seq_i} \|_2^2$
      $loss_{total} \leftarrow loss_{total} + loss_{recons}$
    end for
    Batch loss: $loss_{batch} \leftarrow loss_{total}/(\text{batch size})$
    Update ϕ to minimize $loss_{batch}$
  end for
end for

HAIL

First Challenge:

- How should we define the input sequence for training Transformer-based PAD method

➡ Utilize the **Random Sampler !!!**

Proposed Method

TransPAD: Transformer for Point Anomaly Detection

- Detection
  1. The decoder aims to effectively reconstruct data. To achieve this, the encoder must consistently map input data to the output latent space
  2. The encoder, executing self-attention operations, updates the input sequence by executing a weighted summation through attention weights considering interactions within the sequence
  3. With the Random Sampler generating sequences of random data points, we assume that the self-attention layer of the encoder is trained to assign higher weights to data in areas of lower variance (normal data). This approach ensures that the weighted summation operations in the self-attention process remain consistent, facilitating accurate mapping of input data to the output latent space
  4. This assumption implies that the dot-product similarity among normal data points will increase. Conversely, the similarity between rare anomalous data and frequent normal data points will decrease
  5. Applying softmax to the dot-product similarity reveals that anomalous data points tend to have higher attention weights, as illustrated below

HAIL

Proposed Method

TransPAD: Transformer for Point Anomaly Detection

- Detection
  1. The decoder aims to effectively reconstruct data. To achieve this, the encoder must consistently map input data to the output latent space

  3. With the Random Sampler generating sequences of random data points, we assume that the self-attention layer of the encoder is trained to assign higher weights to data in areas of lower variance (normal data). This approach ensures that the weighted summation operations in the self-attention process remain consistent

HAIL

Proposed Method

TransPAD: Transformer for Point Anomaly Detection

- Detection
  1. The decoder aims to effectively reconstruct data. To achieve this, the encoder must consistently map input data to the output latent space

  3. With the Random Sampler generating sequences of random data points, we assume that the self-attention layer of the encoder is trained to assign higher weights to data in areas of lower variance (normal data). This approach ensures that the weighted summation operations in the self-attention process remain consistent

	a	b	c
Normal - a	9	8	2
Normal - b	7	9	1
Anomaly - c	1	2	8

Dot-product similarity matrix

HAIL

Proposed Method

TransPAD: Transformer for Point Anomaly Detection

- Detection

1. The decoder aims to effectively reconstruct data. To achieve this, the encoder must consistently map input data to the output latent space

3. With the Random Sampler generating sequences of random data points, we assume that the self-attention layer of the encoder is trained to assign higher weights to data in areas of lower variance (normal data). This approach ensures that the weighted summation operations in the self-attention process remain consistent

	a	b	c
Normal - a	9	8	2
Normal - b	7	9	1
Anomaly - c	1	2	8

Dot-product similarity matrix

Proposed Method

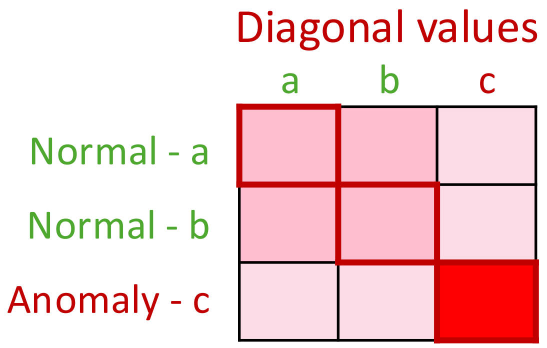
TransPAD: Transformer for Point Anomaly Detection

- Detection
  1. The decoder aims to effectively reconstruct data. To achieve this, the encoder must consistently map input data to the output latent space
  3. With the Random Sampler generating sequences of random data points, we assume that the self-attention layer of the encoder is trained to assign higher weights to data in areas of lower variance (normal data). This approach ensures that the weighted summation operations in the self-attention process remain consistent

	a	b	c
Normal - a	9	8	2
Normal - b	7	9	1
Anomaly - c	1	2	8

Dot-product similarity matrix

Softmax



Attention weight matrix

Anomaly Score !

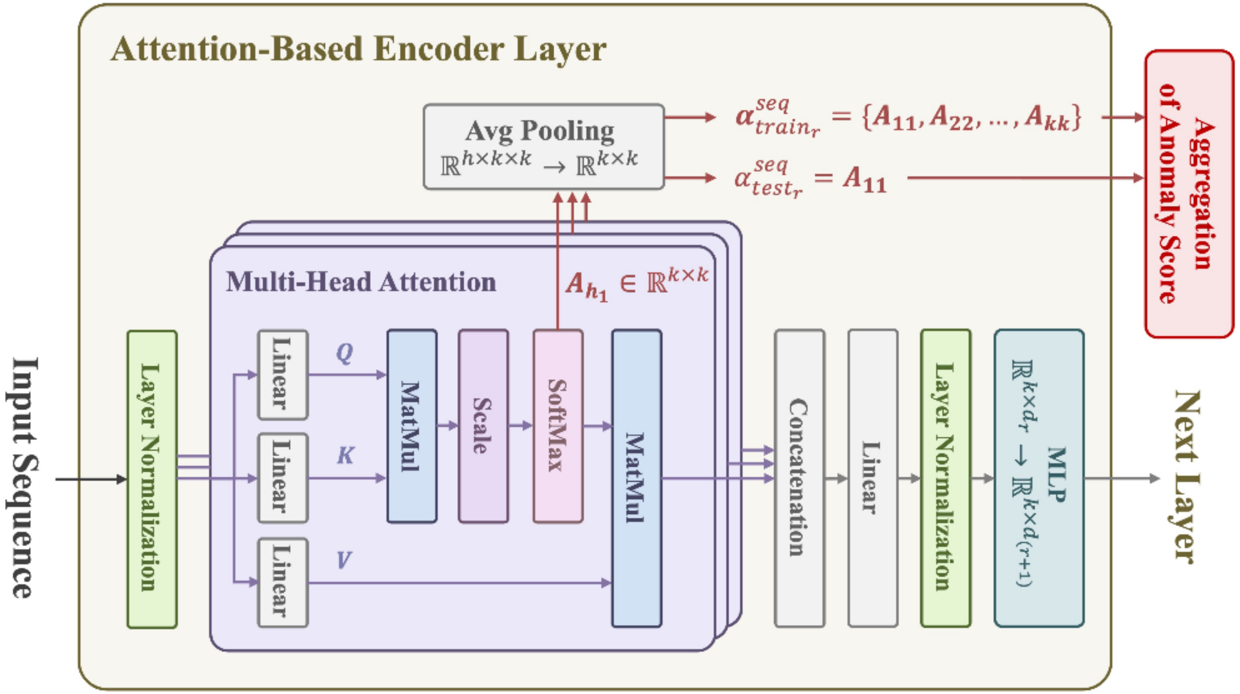
Proposed Method

TransPAD: Transformer for Point Anomaly Detection

- Detection
  - Therefore, we aggregate the diagonal values of the attention weight matrices from each head and layer to calculate the anomaly score

$$\alpha_t = \frac{1}{l} \cdot \frac{1}{h} \sum_{r=1}^{l} \sum_{j=1}^{h} \mathrm{A}_{tt}^{layer_r, head_j}$$

(b)



2nd Challenge:

- What algorithm can we utilize to obtain an anomaly score considering entire data points for PAD

Proposed Method

TransPAD: Transformer for Point Anomaly Detection

- Detection
  - To overcome this challenge, we introduce the **Criteria Sampler**
  - It literally creates a criteria sequence that represent the normal data points from the entire dataset
  - It utilizes the anomaly scores produced during the training phase

Criteria Sampler

$$\mathbf{c} = Index(Lowest_{(k-1)}(\alpha_{train}))$$

$$\mathbf{X}_{test_i} = \{\mathbf{x}_i, \mathbf{x}_{c_1}, ..., \mathbf{x}_{c_{(k-1)}}\}, c_i \in \mathbf{c}$$

$$\psi(\mathbf{X}_{test_i}) = \{\alpha_1, \alpha_2, ..., \alpha_k\}, \; pred_{\mathbf{x}_i} = \alpha_1$$

Algorithm 2 The Detection Process Using the *Criteria Sampler*

Step 1: Get initial anomaly scores
for $i \leftarrow 1$ to $\lfloor \frac{n}{k} \rfloor$ **do**
    Sample sequence $\mathbf{X}_{train_i} \sim U(\mathbf{X} - \cup_{m=1}^{i-1} \mathbf{X}_{train_m})$
    Add $\mathbf{X}_{train_i}$ to $\mathbf{X}_{train}$
end for
$\mathbf{X}_{train} \in \mathbb{R}^{\lfloor \frac{n}{k} \rfloor \times k \times d}$
Load trained *TransPAD* network $\psi(\cdot)$
for each sequence $\mathbf{X}_{seq}$ in $\mathbf{X}_{train}$ **do**
    Get initial scores $\{\alpha_1, \alpha_2, ..., \alpha_k\} \leftarrow \psi(\mathbf{X}_{seq})$ (see Eq. 5)
    Add $\{\alpha_1, \alpha_2, ..., \alpha_k\}$ to α_{train}
end for
$\alpha_{train} \in \mathbb{R}^{(\lfloor \frac{n}{k} \rfloor \times k)}$
Step 2: Get anomaly score of each data point ($\mathbf{x}_i \in X$)
$\mathbf{c} \leftarrow Index(Lowest_{(k-1)}(\alpha_{train}))$
$\mathbf{X}_{test_i} \leftarrow \{\mathbf{x}_i, \mathbf{x}_{c_1}, ..., \mathbf{x}_{c_{(k-1)}}\}, c_i \in \mathbf{c}$
$\{\alpha_1, \alpha_2, ..., \alpha_k\} \leftarrow \psi(\mathbf{X}_{test_i})$
$pred_{\mathbf{x}_i} \leftarrow \alpha_1$

HAIL

2nd Challenge:

- What algorithm can we utilize to obtain an anomaly score considering entire data points for PAD

Utilize the **Criteria Sampler !!!**

Experiments

Experimental Setting

- **Performance Comparisons**
  - To evaluate the effectiveness of our proposed method for PAD, we conduct extensive experiments on diverse set of tabular datasets
  - **First benchmark test**
    - Extended experiment based on the RDP paper [Wang et al., 2019] which is SOTA for PAD task

Datasets	aPascal [Farhadi et al., 2009], Lung [Hong et al., 1991], Probe, Secom [Stolfo et al., 1999], U2R [McCann et al., 2008]
Baseline models	iForest [Liu et al., 2008], AE [Hinton et al., 2006], REPEN [Pang et al., 2018], DAGMM [Zong et al., 2018] RND [Burda et al., 2018], RDP [Wang et al., 2019], Deep SVDD [Ruff et al., 2018], VAE-SVDD [Zhou et al., 2021]

  - **Second benchmark test**
    - Head-to-head comparison between TransPAD and RDP, which have exhibited superior performance in the first benchmark test

Datasets	Optdigits [Alpaydin et al., 1998], Pendigits [Keller et al., 2012], WBC [Wolberg et al., 1993], Lympho [Zwitter et al., 1998], Speech [Micenková et al., 2014]
Baseline models	RDP [Wang et al., 2019]

HAIL

Experiments

Experimental Setting

- **Evaluation Metrics**
  - To evaluate the performance of anomaly detection methods, we employ two key metrics
    - AUROC (the Area Under the Receiver Operating Characteristic Curve)
    - AUPRC (the Area Under the Precision-Recall Curve)
  - For a qualitative analysis, we utilize the UMAP (Uniform Manifold Approximation and Projection)

- **TransPAD Configurations**
  - We optimize the hyperparameters of TransPAD based on AUROC, with adjustments within specified ranges for each dataset

Batch size	64, 128	Learning rate	10^{-3}, 10^{-4}, 10^{-5}
Sequence length	32, 64, 128	Number of heads	16, 32
Input dimension	64, 128, 256	Number of layers	4, 5, 6
Layer configurations	same, smaller, hybrid		

HAIL

Experiments

Quantitative Analysis (for First Benchmark Test)

Dataset	*aPascal*	*Lung*	*Probe*	*Secom*	*U2R*
n	12,695	145	64,759	1,567	60,821
d	64	3312	34	590	34
anomaly ratio	1.38%	4.13%	6.43%	6.63%	0.37%
iForest	0.514±0.051	0.893±0.057	0.995±0.001	0.548±0.019	**0.988±0.001**
AE	0.623±0.005	0.953±0.004	**0.997±0.000**	0.526±0.000	0.987±0.000
REPEN	0.813±0.004	0.949±0.002	**0.997±0.000**	0.510±0.004	0.978±0.000
DAGMM	0.710±0.020	0.830±0.087	0.953±0.008	0.513±0.010	0.945±0.028
RND	0.685±0.019	0.867±0.031	0.975±0.000	0.541±0.006	0.981±0.001
RDP	0.823±0.007	0.982±0.006	**0.997±0.000**	**0.570±0.004**	0.986±0.001
DeepSVDD	0.845±0.031	0.985±0.022	0.988±0.023	0.567±0.016	0.969±0.024
VAE-SVDD	0.555±0.018	0.779±0.139	0.900±0.117	0.563±0.011	0.799±0.086
TransPAD-R	0.893±0.041	0.847±0.159	0.990±0.010	0.551±0.026	0.978±0.009
TransPAD-C	**0.928±0.036**	**0.995±0.004**	0.995±0.001	0.557±0.018	0.985±0.001

Table 1: Comparison of AUROC performance (mean±std).

Dataset	*aPascal*	*Lung*	*Probe*	*Secom*	*U2R*
iForest	0.015±0.002	0.379±0.092	0.923±0.011	**0.106±0.007**	0.180±0.018
AE	0.023±0.001	0.565±0.022	**0.964±0.002**	0.093±0.000	0.230±0.004
REPEN	0.041±0.001	0.429±0.005	**0.964±0.000**	0.091±0.001	0.116±0.007
DAGMM	0.023±0.009	0.042±0.003	0.409±0.153	0.066±0.002	0.025±0.019
RND	0.021±0.005	0.381±0.104	0.609±0.014	0.086±0.002	0.217±0.011
RDP	0.042±0.003	0.705±0.028	0.955±0.002	0.096±0.001	**0.261±0.005**
DeepSVDD	0.047±0.012	0.817±0.219	0.885±0.145	0.095±0.007	0.168±0.123
VAE-SVDD	0.017±0.003	0.139±0.054	0.674±0.265	0.081±0.006	0.063±0.066
TransPAD-R	0.092±0.046	0.474±0.255	0.918±0.042	0.087±0.010	0.171±0.126
TransPAD-C	**0.164±0.101**	**0.878±0.097**	0.943±0.013	0.085±0.008	0.259±0.104

Table 2: Comparison of AUPRC performance (mean±std).

HAIL

Experiments

Quantitative Analysis (for First Benchmark Test)

Dataset	*aPascal*	*Lung*	*Probe*	*Secom*	*U2R*
n	12,695	145	64,759	1,567	60,821
d	64	3312	34	590	34
anomaly ratio	1.38%	4.13%	6.43%	6.63%	0.37%
iForest	0.514±0.051	0.893±0.057	0.995±0.001	0.548±0.019	**0.988±0.001**
AE	0.623±0.005	0.953±0.004	**0.997±0.000**	0.526±0.000	0.987±0.000
REPEN	0.813±0.004	0.949±0.002	**0.997±0.000**	0.510±0.004	0.978±0.000
DAGMM	0.710±0.020	0.830±0.087	0.953±0.008	0.513±0.010	0.945±0.028
RND	0.685±0.019	0.867±0.031	0.975±0.000	0.541±0.006	0.981±0.001
RDP	0.823±0.007	0.982±0.006	**0.997±0.000**	**0.570±0.004**	0.986±0.001
DeepSVDD	0.845±0.031	0.985±0.022	0.988±0.023	0.567±0.016	0.969±0.024
VAE-SVDD	0.555±0.018	0.779±0.139	0.900±0.117	0.563±0.011	0.799±0.086
TransPAD-R	0.893±0.041	0.847±0.159	0.990±0.010	0.551±0.026	0.978±0.009
TransPAD-C	**0.928±0.036**	**0.995±0.004**	0.995±0.001	0.557±0.018	0.985±0.001

Table 1: Comparison of AUROC performance (mean±std).

Dataset	*aPascal*	*Lung*	*Probe*	*Secom*	*U2R*
iForest	0.015±0.002	0.379±0.092	0.923±0.011	**0.106±0.007**	0.180±0.018
AE	0.023±0.001	0.565±0.022	**0.964±0.002**	0.093±0.000	0.230±0.004
REPEN	0.041±0.001	0.429±0.005	**0.964±0.000**	0.091±0.001	0.116±0.007
DAGMM	0.023±0.009	0.042±0.003	0.409±0.153	0.066±0.002	0.025±0.019
RND	0.021±0.005	0.381±0.104	0.609±0.014	0.086±0.002	0.217±0.011
RDP	0.042±0.003	0.705±0.028	0.955±0.002	0.096±0.001	**0.261±0.005**
DeepSVDD	0.047±0.012	0.817±0.219	0.885±0.145	0.095±0.007	0.168±0.123
VAE-SVDD	0.017±0.003	0.139±0.054	0.674±0.265	0.081±0.006	0.063±0.066
TransPAD-R	0.092±0.046	0.474±0.255	0.918±0.042	0.087±0.010	0.171±0.126
TransPAD-C	**0.164±0.101**	**0.878±0.097**	0.943±0.013	0.085±0.008	0.259±0.104

Table 2: Comparison of AUPRC performance (mean±std).

- Random distance-based methods does not consider the relationship between the input and training data, which may result in a different interpretation of anomalies compared to our method

HAIL

Experiments

Quantitative Analysis (for First Benchmark Test)

Dataset	*aPascal*	*Lung*	*Probe*	*Secom*	*U2R*
n	12,695	145	64,759	1,567	60,821
d	64	3312	34	590	34
anomaly ratio	1.38%	4.13%	6.43%	6.63%	0.37%
iForest	0.514±0.051	0.893±0.057	0.995±0.001	0.548±0.019	**0.988±0.001**
AE	0.623±0.005	0.953±0.004	**0.997±0.000**	0.526±0.000	0.987±0.000
REPEN	0.813±0.004	0.949±0.002	**0.997±0.000**	0.510±0.004	0.978±0.000
DAGMM	0.710±0.020	0.830±0.087	0.953±0.008	0.513±0.010	0.945±0.028
RND	0.685±0.019	0.867±0.031	0.975±0.000	0.541±0.006	0.981±0.001
RDP	0.823±0.007	0.982±0.006	**0.997±0.000**	**0.570±0.004**	0.986±0.001
DeepSVDD	0.845±0.031	<u>0.985±0.022</u>	0.988±0.023	<u>0.567±0.016</u>	0.969±0.024
VAE-SVDD	0.555±0.018	0.779±0.139	0.900±0.117	<u>0.563±0.011</u>	0.799±0.086
TransPAD-R	<u>0.893±0.041</u>	0.847±0.159	0.990±0.010	0.551±0.026	0.978±0.009
TransPAD-C	**0.928±0.036**	**0.995±0.004**	0.995±0.001	0.557±0.018	0.985±0.001

Table 1: Comparison of AUROC performance (mean±std).

Dataset	*aPascal*	*Lung*	*Probe*	*Secom*	*U2R*
iForest	0.015±0.002	0.379±0.092	0.923±0.011	**0.106±0.007**	0.180±0.018
AE	0.023±0.001	0.565±0.022	**0.964±0.002**	0.093±0.000	0.230±0.004
REPEN	0.041±0.001	0.429±0.005	**0.964±0.000**	0.091±0.001	0.116±0.007
DAGMM	0.023±0.009	0.042±0.003	0.409±0.153	0.066±0.002	0.025±0.019
RND	0.021±0.005	0.381±0.104	0.609±0.014	0.086±0.002	0.217±0.011
RDP	0.042±0.003	0.705±0.028	0.955±0.002	0.096±0.001	**0.261±0.005**
DeepSVDD	0.047±0.012	<u>0.817±0.219</u>	0.885±0.145	0.095±0.007	0.168±0.123
VAE-SVDD	0.017±0.003	0.139±0.054	0.674±0.265	0.081±0.006	0.063±0.066
TransPAD-R	<u>0.092±0.046</u>	0.474±0.255	0.918±0.042	0.087±0.010	0.171±0.126
TransPAD-C	**0.164±0.101**	**0.878±0.097**	0.943±0.013	0.085±0.008	<u>0.259±0.104</u>

Table 2: Comparison of AUPRC performance (mean±std).

- DAGMM consistently has underperformed on all datasets that indicates potential limitations in approximating data distributions using mixtures of Gaussians

HAIL

Experiments

Quantitative Analysis (for First Benchmark Test)

Dataset	*aPascal*	*Lung*	*Probe*	*Secom*	*U2R*
n	12,695	145	64,759	1,567	60,821
d	64	3312	34	590	34
anomaly ratio	1.38%	4.13%	6.43%	6.63%	0.37%
iForest	0.514±0.051	0.893±0.057	0.995±0.001	0.548±0.019	**0.988±0.001**
AE	0.623±0.005	0.953±0.004	**0.997±0.000**	0.526±0.000	0.987±0.000
REPEN	0.813±0.004	0.949±0.002	**0.997±0.000**	0.510±0.004	0.978±0.000
DAGMM	0.710±0.020	0.830±0.087	0.953±0.008	0.513±0.010	0.945±0.028
RND	0.685±0.019	0.867±0.031	0.975±0.000	0.541±0.006	0.981±0.001
RDP	0.823±0.007	0.982±0.006	**0.997±0.000**	**0.570±0.004**	0.986±0.001
DeepSVDD	0.845±0.031	0.985±0.022	0.988±0.023	0.567±0.016	0.969±0.024
VAE-SVDD	0.555±0.018	0.779±0.139	0.900±0.117	0.563±0.011	0.799±0.086
TransPAD-R	0.893±0.041	0.847±0.159	0.990±0.010	0.551±0.026	0.978±0.009
TransPAD-C	**0.928±0.036**	**0.995±0.004**	0.995±0.001	0.557±0.018	0.985±0.001

Table 1: Comparison of AUROC performance (mean±std).

Dataset	*aPascal*	*Lung*	*Probe*	*Secom*	*U2R*
iForest	0.015±0.002	0.379±0.092	0.923±0.011	**0.106±0.007**	0.180±0.018
AE	0.023±0.001	0.565±0.022	**0.964±0.002**	0.093±0.000	0.230±0.004
REPEN	0.041±0.001	0.429±0.005	**0.964±0.000**	0.091±0.001	0.116±0.007
DAGMM	0.023±0.009	0.042±0.003	0.409±0.153	0.066±0.002	0.025±0.019
RND	0.021±0.005	0.381±0.104	0.609±0.014	0.086±0.002	0.217±0.011
RDP	0.042±0.003	0.705±0.028	0.955±0.002	0.096±0.001	**0.261±0.005**
DeepSVDD	0.047±0.012	0.817±0.219	0.885±0.145	0.095±0.007	0.168±0.123
VAE-SVDD	0.017±0.003	0.139±0.054	0.674±0.265	0.081±0.006	0.063±0.066
TransPAD-R	0.092±0.046	0.474±0.255	0.918±0.042	0.087±0.010	0.171±0.126
TransPAD-C	**0.164±0.101**	**0.878±0.097**	0.943±0.013	0.085±0.008	0.259±0.104

Table 2: Comparison of AUPRC performance (mean±std).

- The SVDD-based methods have shown lower results on most datasets compared to TransPAD
- This indicates that generalizing embeddings to the central point of all training data in latent space could be ineffective for datasets with complex distributions

HAIL

Experiments

Quantitative Analysis (for Second Benchmark Test)

Dataset	*Optdigits*	*Pendigits*	*WBC*	*Lympho*	*Speech*
n	5,216	6870	278	148	3686
d	64	16	30	18	400
anomaly ratio	2.88%	2.27%	7.55%	4.05%	1.65%
RDP	0.604±0.121	0.903±0.040	**0.933±0.014**	0.740±0.044	0.477±0.026
TransPAD-C	**0.843±0.139**	**0.916±0.111**	0.910±0.041	**0.879±0.096**	**0.519±0.050**

Table 3: Comparison AUROC performance (mean±std)

Dataset	*Optdigits*	*Pendigits*	*WBC*	*Lympho*	*Speech*
RDP	0.039±0.017	0.143±0.077	0.419±0.043	**0.581±0.116**	0.017±0.004
TransPAD-C	**0.187±0.139**	**0.356±0.230**	**0.541±0.150**	0.366±0.239	**0.019±0.003**

Table 4: Comparison AUPRC performance (mean±std)

- While both methods have exhibited strong performances, TransPAD has shown greater consistency and superiority

HAIL

Experiments

Qualitative Analysis (Encoder Architecture and Detection Performance)

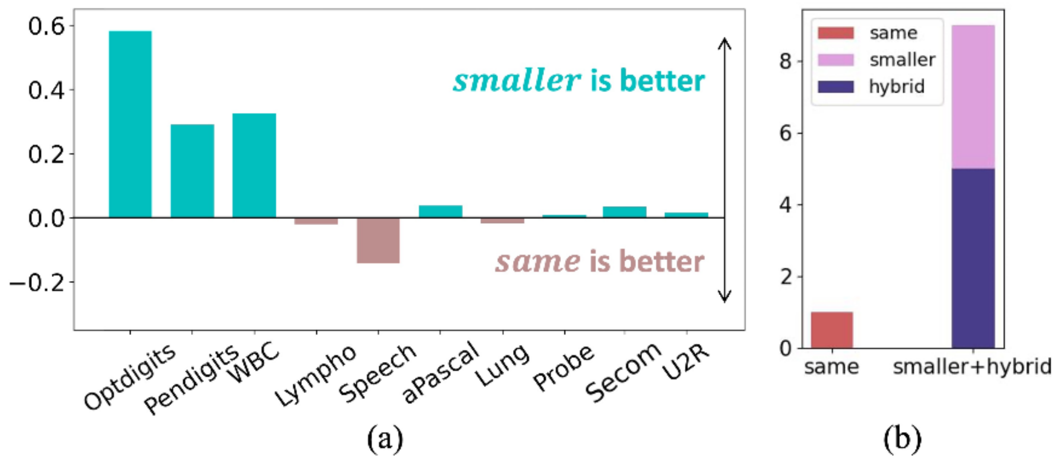


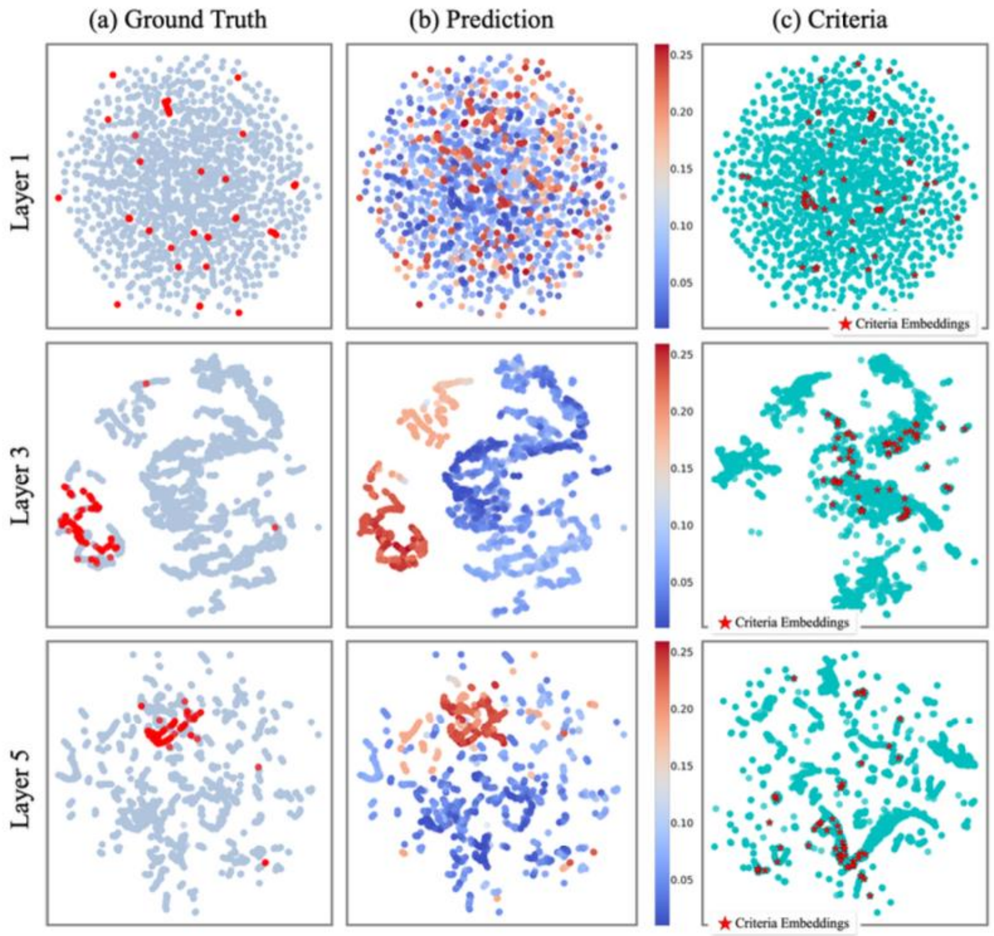
Figure 4: (a): The difference in the AUROC between cases where the layer option is *smaller*, and cases where it is *same*. (b): The number of dataset selected for each layer option during hyperparameter tuning.

➡ We interpret this because reducing the dimensionality enables careful comparisons of anomaly levels as data progresses through the multi-scaled layers

HAIL

Experiments

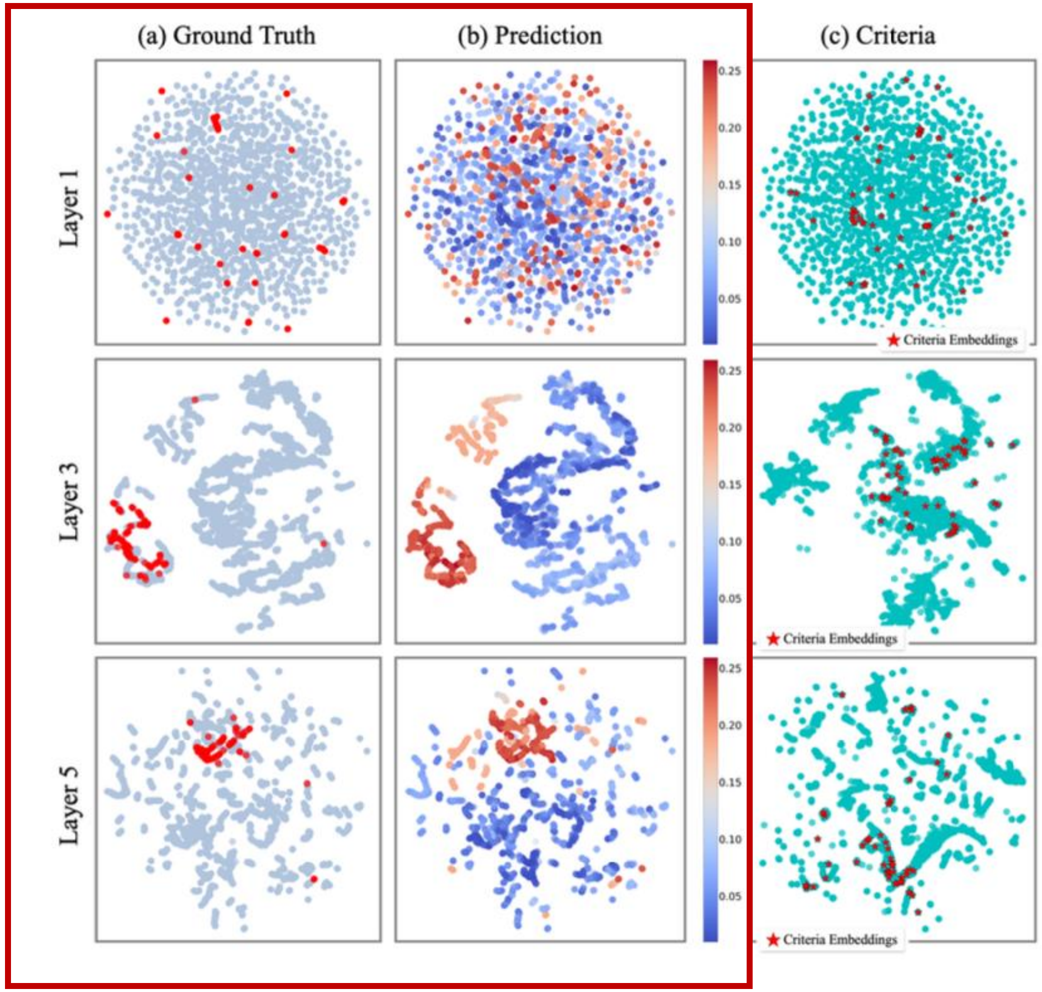
Qualitative Analysis (Visualization of Embeddings)



HAIL

Experiments

Qualitative Analysis (Visualization of Embeddings)

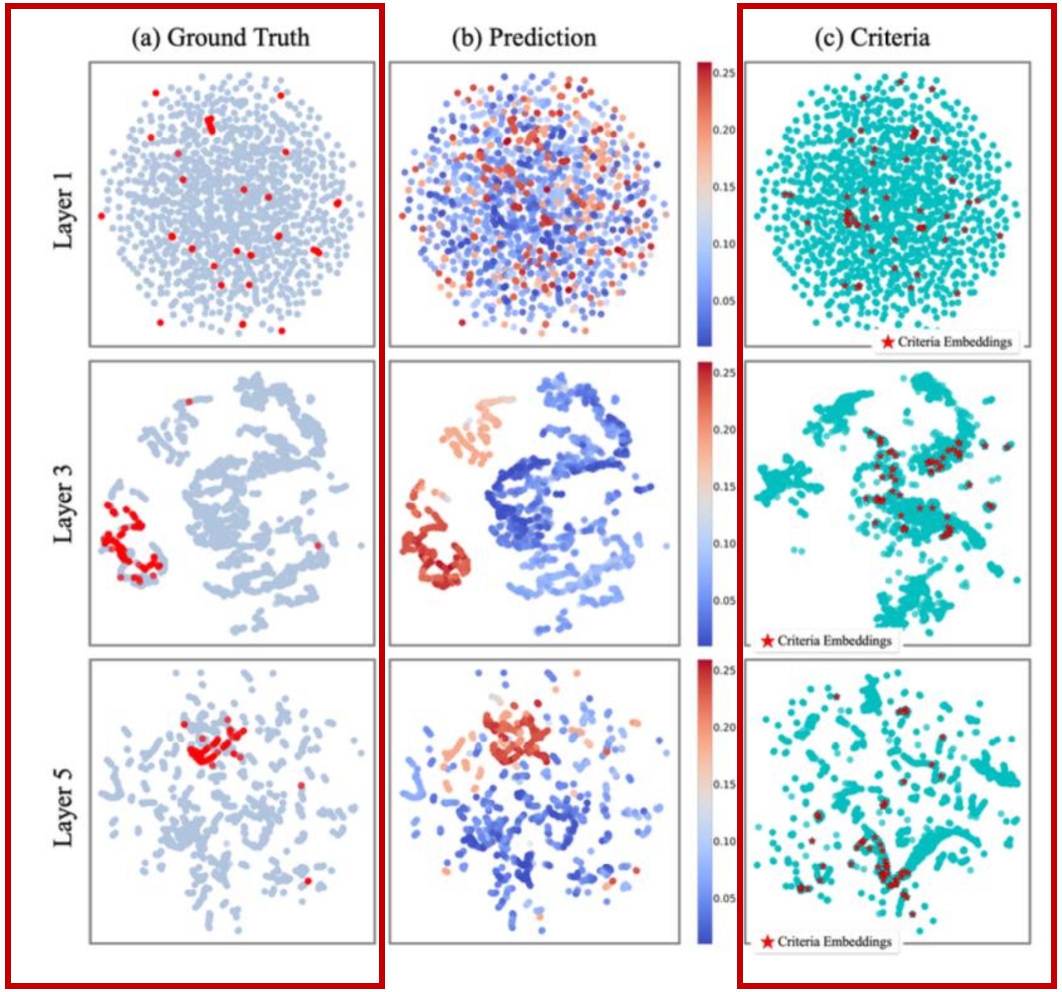


- Training with the Random Sampler effectively separates statistical anomalies from the overall dataset
- The attention weight-based anomaly scores are effective in recognizing anomalies based on similarity

HAIL

Experiments

Qualitative Analysis (Visualization of Embeddings)



- The capability of the criteria sequence to accurately reflect the characteristics of the training data

References

Aggarwal, C. C., & Aggarwal, C. C. (2017). *Outlier analysis* (pp. 185-218). Springer International Publishing.

Liu, J., Guo, J., Orlik, P., Shibata, M., Nakahara, D., Mii, S., & Takáč, M. (2018, July). Anomaly detection in manufacturing systems using structured neural networks. In *2018 13th world congress on intelligent control and automation (wcica)* (pp. 175-180). IEEE.

Alom, M. Z., & Taha, T. M. (2017, June). Network intrusion detection for cyber security using unsupervised deep learning approaches. In *2017 IEEE national aerospace and electronics conference (NAECON)* (pp. 63-69). IEEE.

Pereira, J., & Silveira, M. (2019, February). Learning representations from healthcare time series data for unsupervised anomaly detection. In *2019 IEEE international conference on big data and smart computing (BigComp)* (pp. 1-7). IEEE.

Kim, H., Lee, C. H., & Hong, C. (2023, July). Crime Scene Detection in Surveillance Videos Using Variational AutoEncoder-Based Support Vector Data Description. In *Asian Conference on Intelligent Information and Database Systems* (pp. 451-464). Cham: Springer Nature Switzerland.

Mousakhan, A., Brox, T., & Tayyub, J. (2023). Anomaly detection with conditioned denoising diffusion models. *arXiv preprint arXiv:2305.15956*.

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, *54*(2), 1-38.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Brown, T. B. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.
Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y. Z., & Hospedales, T. M. (2019). Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1446-1455).

Farhadi, A., Endres, I., Hoiem, D., & Forsyth, D. (2009, June). Describing objects by their attributes. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 1778-1785). IEEE.

Hong, Z. & Yang, J. (1991). Lung Cancer [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C57596.
Stolfo, S., Fan, W., Lee, W., Prodromidis, A., & Chan, P. (1999). KDD Cup 1999 Data [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C51C7N.
McCann, M. & Johnston, A. (2008). SECOM [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C54305.
Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413-422). IEEE.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504-507.

Pang, G., Cao, L., Chen, L., & Liu, H. (2018, July). Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2041-2050).

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., & Chen, H. (2018, February). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*.

Burda, Y., Edwards, H., Storkey, A., & Klimov, O. (2018). Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*.

Wang, H., Pang, G., Shen, C., & Ma, C. (2019). Unsupervised representation learning by predicting random distances. *arXiv preprint arXiv:1912.12186*.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., ... & Kloft, M. (2018, July). Deep one-class classification. In *International conference on machine learning* (pp. 4393-4402). PMLR.

Zhou, Y., Liang, X., Zhang, W., Zhang, L., & Song, X. (2021). VAE-based deep SVDD for anomaly detection. *Neurocomputing*, *453*, 131-140.

Alpaydin, E. & Kaynak, C. (1998). Optical Recognition of Handwritten Digits [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C50P49.

Keller, F., Muller, E., & Bohm, K. (2012, April). HiCS: High contrast subspaces for density-based outlier ranking. In *2012 IEEE 28th international conference on data engineering* (pp. 1037-1048). IEEE.

Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1993). Breast Cancer Wisconsin (Diagnostic) [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.

Zwitter, M. & Soklic, M. (1988). Lymphography [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C54598.

Micenková, B., McWilliams, B., & Assent, I. (2014, August). Learning outlier ensembles: The best of both worlds–supervised and unsupervised. In *Proceedings of the ACM SIGKDD 2014 Workshop on Outlier Detection and Description under Data Diversity (ODD2). New York, NY, USA* (pp. 51-54).

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

HAIL

Thank you for your attention

GitHub

Title: Transformer for Point Anomaly Detection

Presenter: Harim Kim (hrkim@handong.ac.kr)
Collaborator: Chang Ha Lee
Advisor: Charmgil Hong (charmgil@handong.ac.kr)

TransPAD
Transformer for Point Anomaly Detection