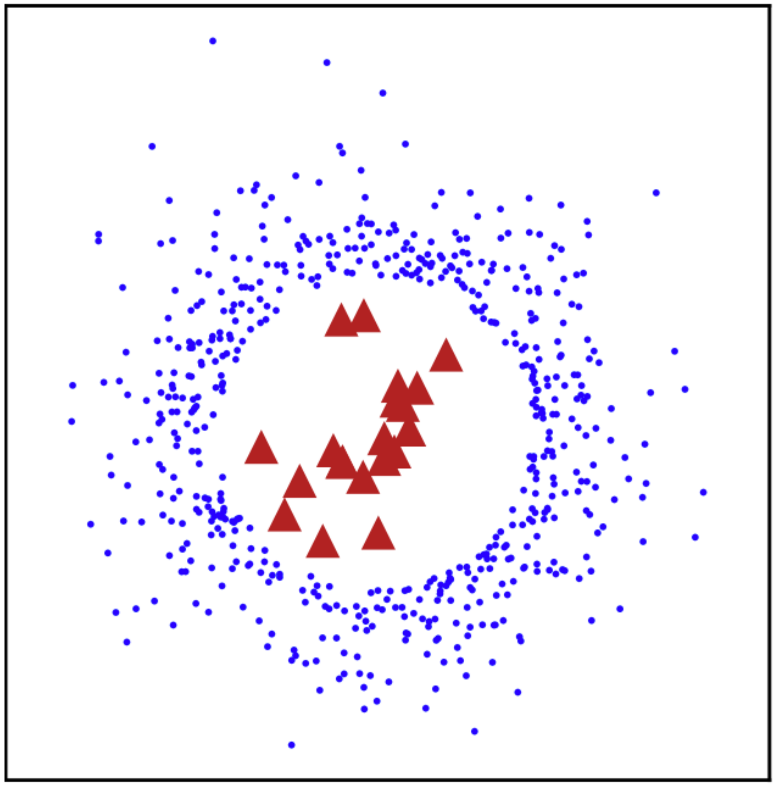
IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 35, NO. 12, DECEMBER 2023

Deep Isolation Forest for Anomaly Detection

Hongzuo Xu, Guansong Pang, *Member, IEEE*, Yijie Wang, and Yongjun Wang

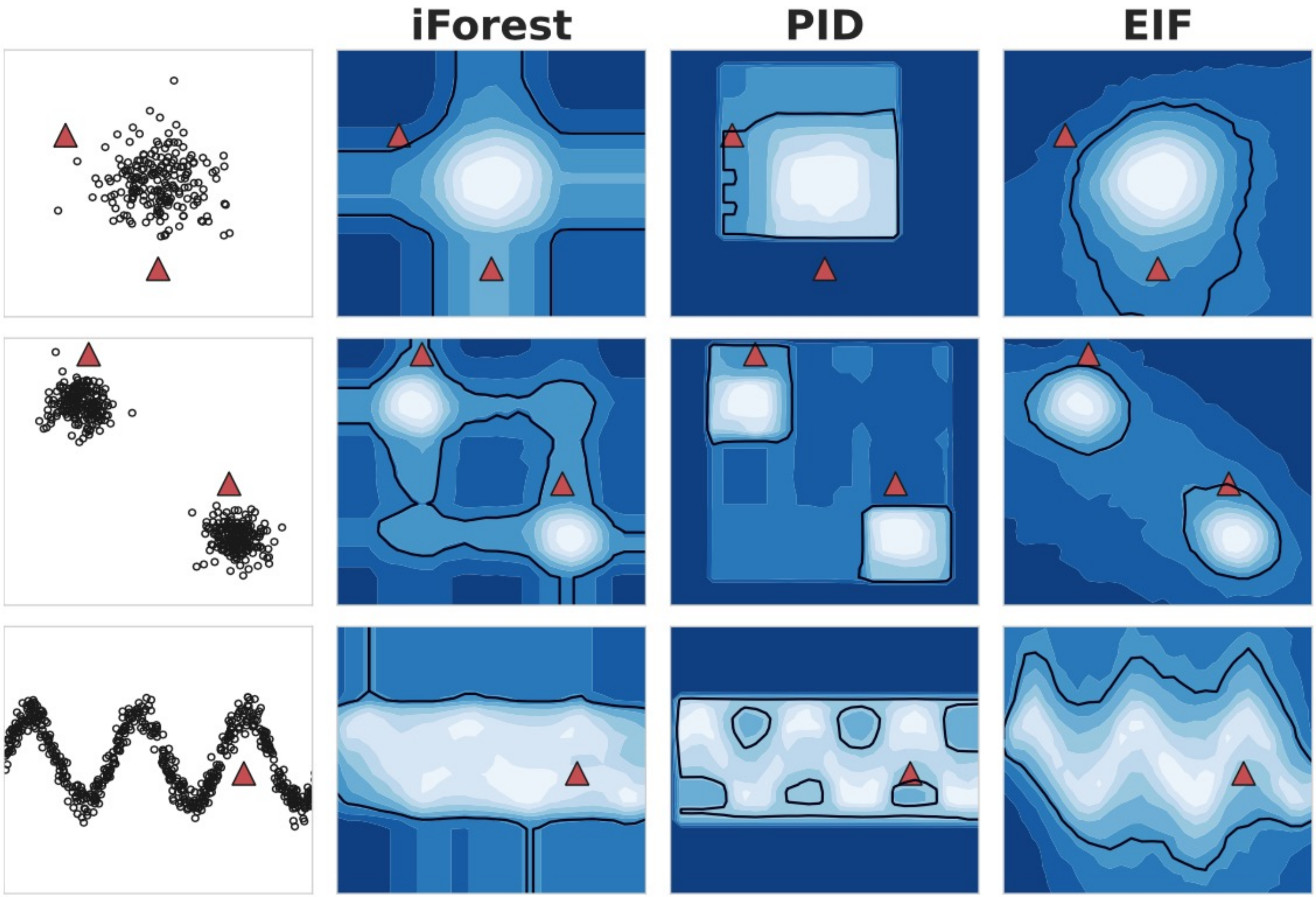
Limitation of iForest

- It cannot handle hard anomalies that are difficult to isolate in high-dimensional/non-linear-separable data space.



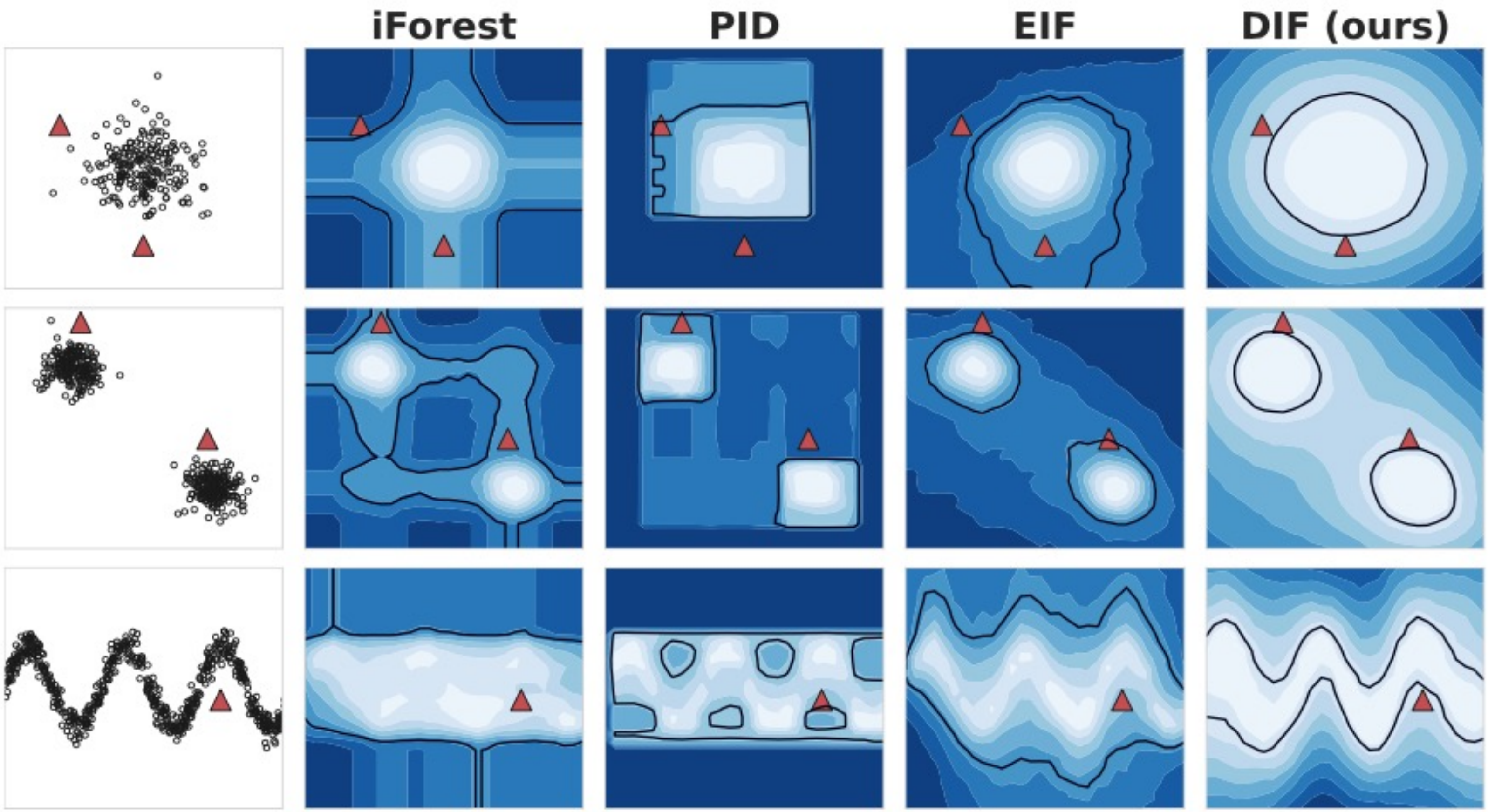
Limitation of iForest

- It assigns unexpectedly lower anomaly scores to artefact regions.



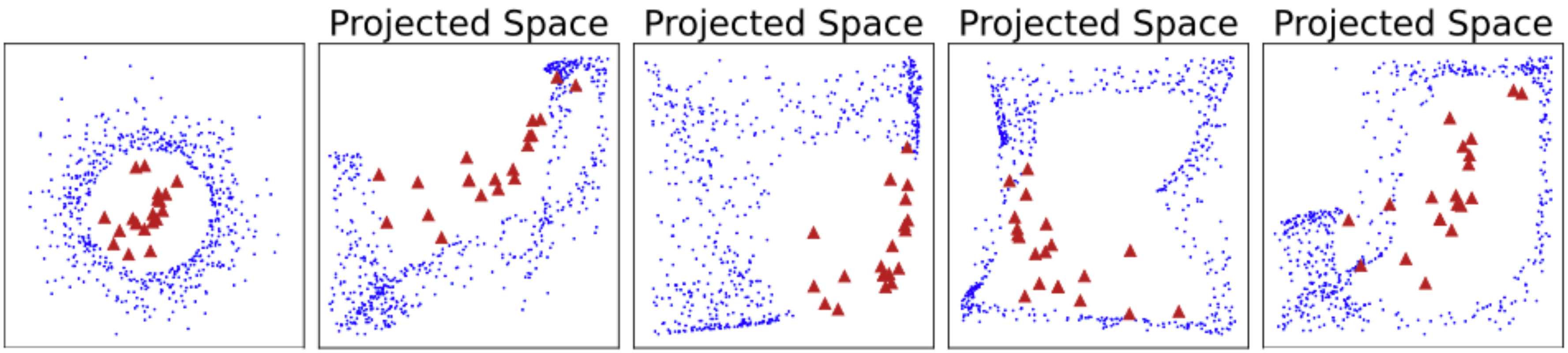
Limitation of iForest

- It assigns unexpectedly lower anomaly scores to artefact regions.



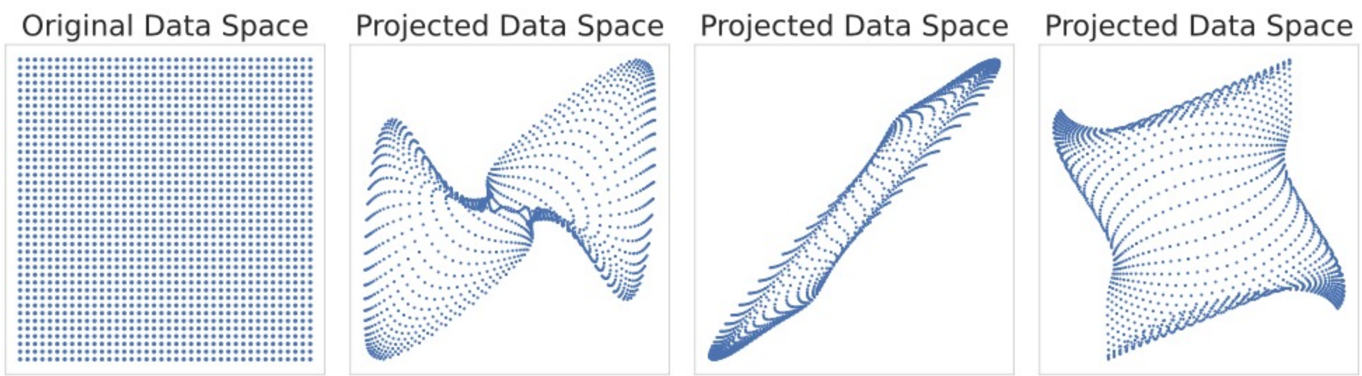
Deep IForest

- Idea of Deep iForest:
  - map the original data into a group of new data spaces, and non-linear isolation can be easily achieved by performing simple axis-parallel partitions upon these newly created data spaces



Deep IForest

- novel representation scheme – random representation ensemble – produced by optimization-free deep neural networks
  - These networks are only casually initialized and do not involve any optimization or training process.



- simply utilizes random axis-parallel cuts upon these representations

Deep IForest

DIF first produces the random representation ensemble via optimisation-free neural networks, which is defined as

$$\mathscr{G}(\mathcal{D}) = \left\{ \mathcal{X}_u \subset \mathbb{R}^d \middle| \mathcal{X}_u = \phi_u(\mathcal{D}; \theta_u) \right\}_{u=1}^{r}, \tag{1}$$

where r is the ensemble size, $\phi_u : \mathcal{D} \mapsto \mathbb{R}^d$ is the network that maps original data into new d-dimensional spaces, and the network weights in θ_u are randomly initialised. Each representation is assigned with t iTrees, and a forest $\mathcal{T} = \{\tau_i\}_{i=1}^{T}$ containing $T = r \times t$ iTrees is constructed.

Deep IForest

Algorithm 1: Construction of Deep Isolation Trees.

Input: $\mathcal{D}$ - input dataset
Output: $\mathcal{T}$ - forest of deep isolation trees

Initialise $\mathcal{T} \leftarrow \varnothing$
Generate representations $\{\mathcal{X}_u\}_{u=1}^{r}$ via $\mathscr{G}_{\text{CERE}}$
for $u = 1$ to r **do**
  for $i = 1$ to t **do**
    Initialise an isolation tree τ_i by setting the root node using $\mathcal{P}_1 \subseteq \mathcal{X}_u$, $|\mathcal{P}_1| = n$
    while $\mathcal{P}_k$ is a leaf node of tree τ_i **do**
      if $|\mathcal{P}_k| > 1$ and the depth is smaller than J **then**
        Randomly select a dimension $j_k \in \{1, \dots, d\}$
        Randomly select a split point η_k between the *max* and *min* values of dimension j_k in $\mathcal{P}_k$
        $\mathcal{P}_{2k} \leftarrow \{\boldsymbol{x} | \boldsymbol{x}^{(j_k)} \leq \eta_k, \boldsymbol{x} \in \mathcal{P}_k\}$
        $\mathcal{P}_{2k+1} \leftarrow \{\boldsymbol{x} | \boldsymbol{x}^{(j_k)} > \eta_k, \boldsymbol{x} \in \mathcal{P}_k\}$
      end if
    end while
    $\mathcal{T} \leftarrow \mathcal{T} \cup \tau_i$
  end for
end for
return $\mathcal{T}$

Deep IForest

- Deviation-Enhanced Anomaly Scoring function
  - utilise the deviation degree of the feature value to the branching threshold as additional weighting information to further improve the measurement of isolation difficulty.

- The averaged deviation degree of x_u in τ_i

$$g(\boldsymbol{x}_u|\tau_i) = \frac{1}{|p(\boldsymbol{x}_u|\tau_i)|} \sum_{k \in p(\boldsymbol{x}_u|\tau_i)} |\boldsymbol{x}_u^{(j_k)} - \eta_k|.$$

- Deviation-enhanced isolation anomaly scoring function

$$\mathscr{F}_{\mathrm{DEAS}}(\boldsymbol{o}|\mathcal{T}) = 2^{-\mathbb{E}_{\tau_i \in \mathcal{T}} \frac{|p(\boldsymbol{x}_u|\tau_i)|}{C(\mathcal{T})}} \times \mathbb{E}_{\tau_i \in \mathcal{T}}\left(g(\boldsymbol{x}_u|\tau_i)\right)$$

Deep IForest

Algorithm 2: Deviation-Enhanced Anomaly Scoring.

Input: $\boldsymbol{o}$ - data object, $\mathcal{T}$ - set of deep isolation trees

Output: anomaly score $\mathscr{F}_{\text{DEAS}}(\boldsymbol{o}|\mathcal{T})$

Generate representations $\{\boldsymbol{x}_u\}_{u=i}^{r}$ via $\mathscr{G}_{\text{CERE}}$

for $u = 1$ to r **do**

  for $i = 1$ to t **do**

    Initialise $k \leftarrow 1$, $\beta \leftarrow 0$, $p(\boldsymbol{x}_u|\tau_i) \leftarrow \varnothing$

    while $|\mathcal{P}_k| > 1$ and not reaching J **do**

      if $\boldsymbol{x}_u^{(j_k)} \leq \eta_k$ **then**

        $k \leftarrow 2k$

      else

        $k \leftarrow 2k + 1$

      end if

      $p(\boldsymbol{x}_u|\tau_i) \leftarrow p(\boldsymbol{x}_u|\tau_i) \cup k$, $\beta \leftarrow \beta + |\boldsymbol{x}_u^{(j_k)} - \eta_k|$

    end while

    $g(\boldsymbol{x}_u|\tau_i) \leftarrow \beta / |p(\boldsymbol{x}_u|\tau_i)|$

  end for

end for

return

$$\mathscr{F}_{\text{DEAS}}(\boldsymbol{o}|\mathcal{T}) \leftarrow 2^{-\mathbb{E}_{\tau_i \in \mathcal{T}} \frac{|p(\boldsymbol{x}_u|\tau_i)|}{C(T)}} \times \mathbb{E}_{\tau \in \mathcal{T}}(g(\boldsymbol{x}_u|\tau_i))$$

Experiments

AUC-ROC AND AUC-PR PERFORMANCE (MEAN ± STANDARD DEVIATION) OF DIF AND IF-BASED COMPETING METHODS ON TABULAR DATASETS

Data	AUC-ROC					AUC-PR				
	DIF (ours)	EIF	PID	LeSiNN	IF	DIF (ours)	EIF	PID	LeSiNN	IF
Analysis	**0.931** ± 0.006	0.910 ± 0.005	0.820 ± 0.019	0.903 ± 0.008	0.782 ± 0.017	**0.404** ± 0.051	0.198 ± 0.022	0.075 ± 0.007	0.183 ± 0.028	0.063 ± 0.006
Backdoor	**0.918** ± 0.002	0.902 ± 0.005	0.808 ± 0.016	0.894 ± 0.006	0.731 ± 0.021	**0.453** ± 0.051	0.218 ± 0.028	0.066 ± 0.005	0.205 ± 0.031	0.046 ± 0.004
DoS	**0.932** ± 0.003	0.918 ± 0.004	0.802 ± 0.013	0.896 ± 0.009	0.747 ± 0.020	**0.440** ± 0.023	0.269 ± 0.027	0.075 ± 0.004	0.185 ± 0.028	0.060 ± 0.005
Exploits	**0.858** ± 0.010	0.840 ± 0.008	0.797 ± 0.011	0.816 ± 0.005	0.745 ± 0.010	**0.273** ± 0.020	0.167 ± 0.011	0.077 ± 0.003	0.120 ± 0.013	0.062 ± 0.003
R8	**0.930** ± 0.008	0.854 ± 0.006	0.881 ± 0.018	0.859 ± 0.001	0.853 ± 0.016	**0.145** ± 0.031	0.101 ± 0.009	0.078 ± 0.011	0.094 ± 0.000	0.075 ± 0.008
Cover	**0.972** ± 0.010	0.872 ± 0.017	0.939 ± 0.007	0.885 ± 0.008	0.888 ± 0.017	**0.246** ± 0.069	0.040 ± 0.006	0.069 ± 0.006	0.051 ± 0.004	0.055 ± 0.008
Fraud	**0.953** ± 0.002	0.950 ± 0.001	0.950 ± 0.002	0.952 ± 0.000	0.950 ± 0.001	0.387 ± 0.031	0.378 ± 0.027	0.186 ± 0.033	**0.401** ± 0.001	0.155 ± 0.015
Pageblocks	**0.903** ± 0.006	0.902 ± 0.001	0.851 ± 0.003	0.887 ± 0.002	0.900 ± 0.005	**0.547** ± 0.012	0.537 ± 0.006	0.421 ± 0.011	0.511 ± 0.007	0.476 ± 0.013
Shuttle	**0.941** ± 0.006	0.843 ± 0.009	0.864 ± 0.017	0.805 ± 0.005	0.862 ± 0.019	**0.150** ± 0.017	0.061 ± 0.003	0.059 ± 0.008	0.048 ± 0.001	0.075 ± 0.014
Thrombin	**0.913** ± 0.003	OOM	OOM	0.912 ± 0.000	0.905 ± 0.002	**0.468** ± 0.020	OOM	OOM	0.458 ± 0.001	0.372 ± 0.008
Average	**0.925** ± 0.006	0.888 ± 0.006	0.857 ± 0.011	0.881 ± 0.004	0.836 ± 0.013	**0.351** ± 0.033	0.219 ± 0.015	0.123 ± 0.010	0.226 ± 0.011	0.144 ± 0.008
p-value	-	0.004	0.004	0.002	0.002	-	0.004	0.004	0.006	0.002

PID and EIF run out of memory (OOM) on the ultrahigh-dimensional dataset *Thrombin*.
The best performer is boldfaced.

Experiments

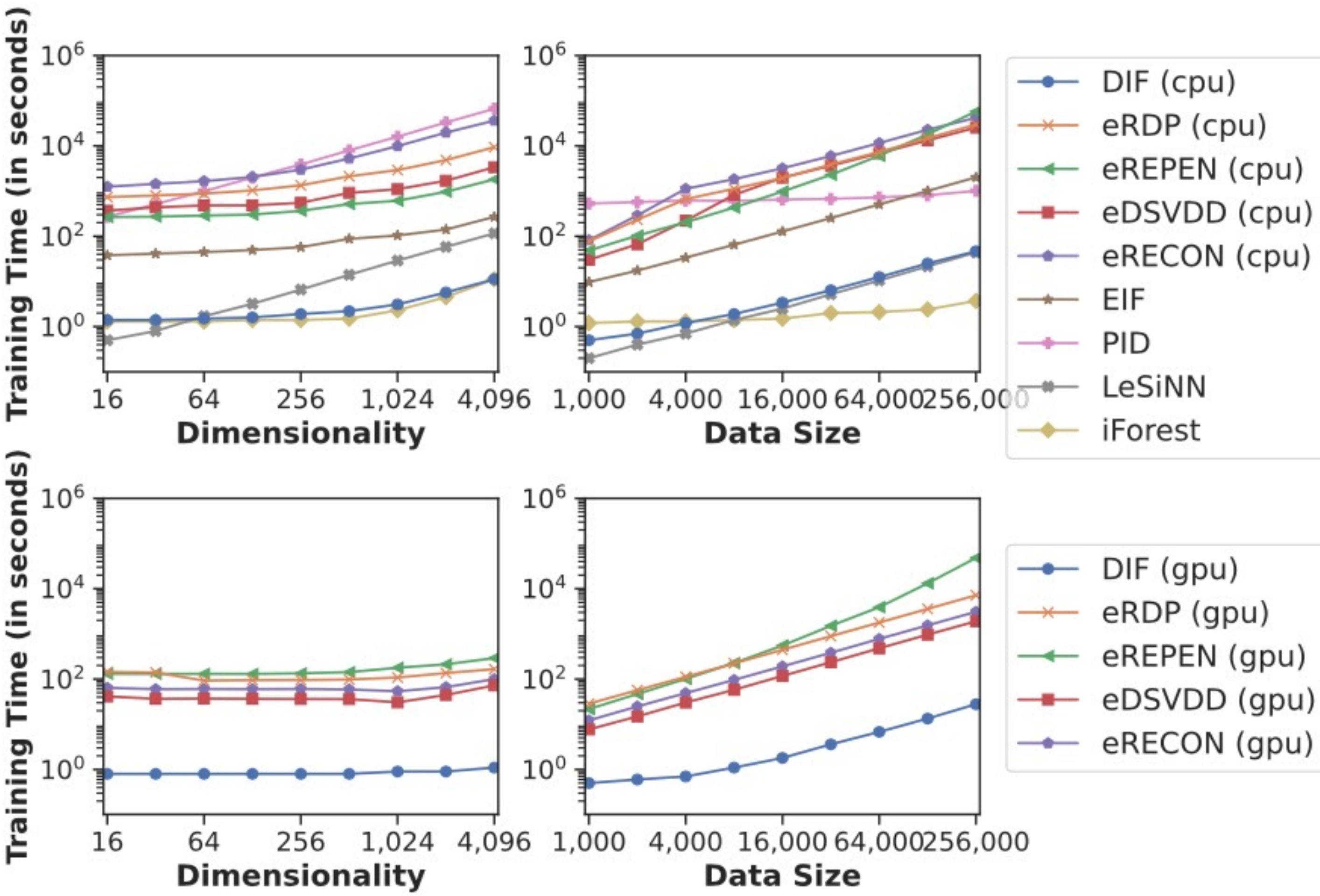
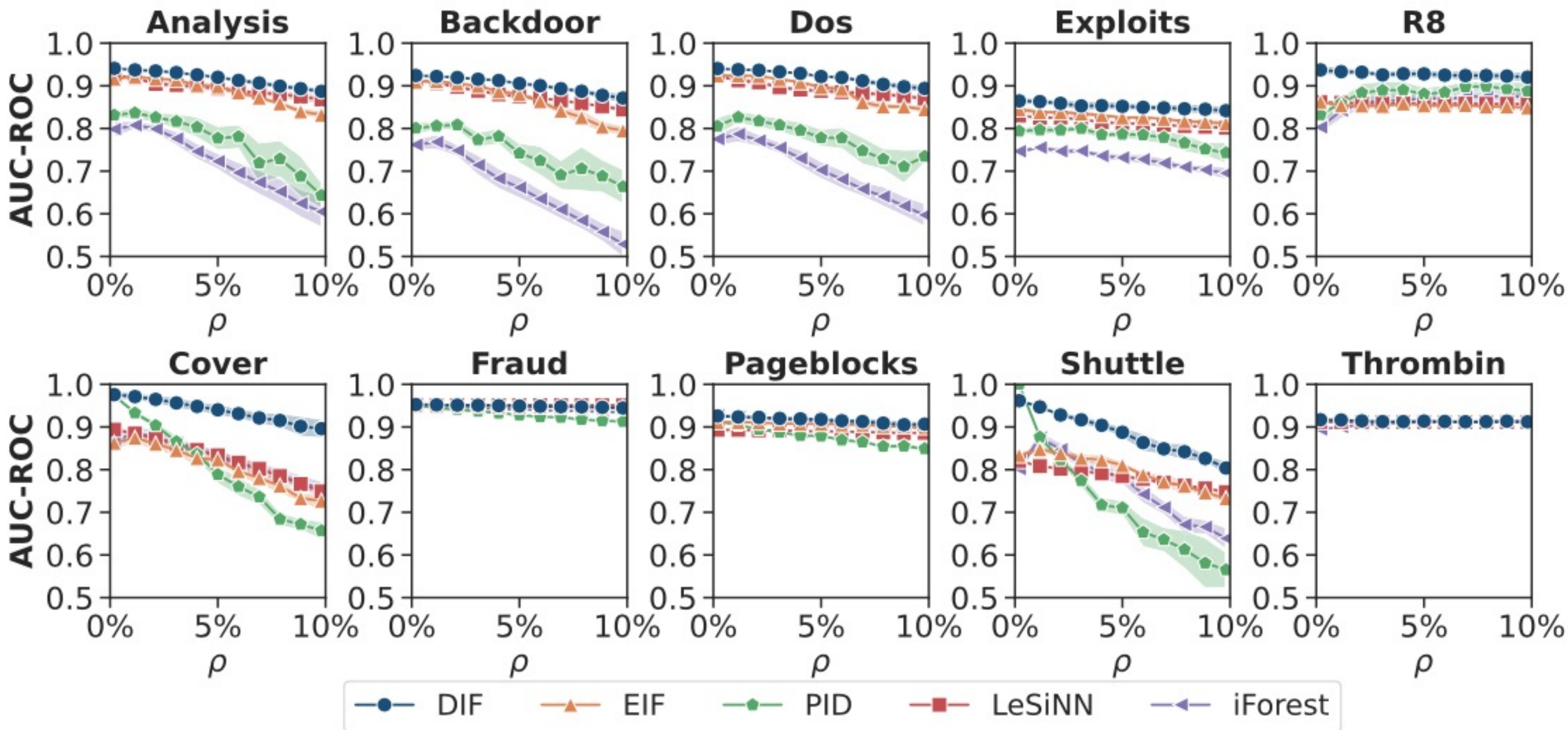


Fig. 5. Scalability test results. (**Top**) The training time of all the anomaly detectors on a CPU device; and (**Bottom**) The results of deep ensemble-based methods (including DIF) on a GPU device.

Experiments

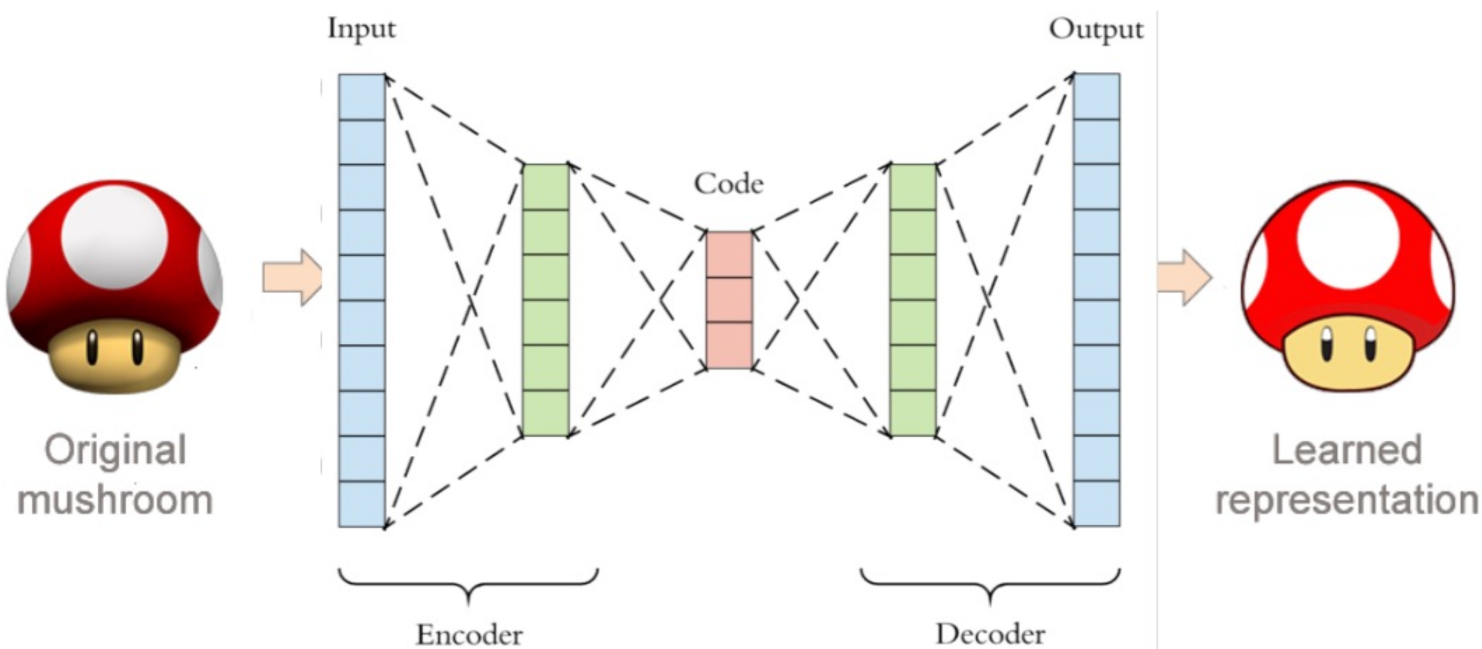


(a) DIF vs. IF-based Competing Methods

Fig. 6. AUC-ROC w.r.t. different contamination ratios ρ

Auto Encoder (AE)

- An encoder network maps inputs into a lower-dimensional representation (code), and a decoder network reconstructs the original input data



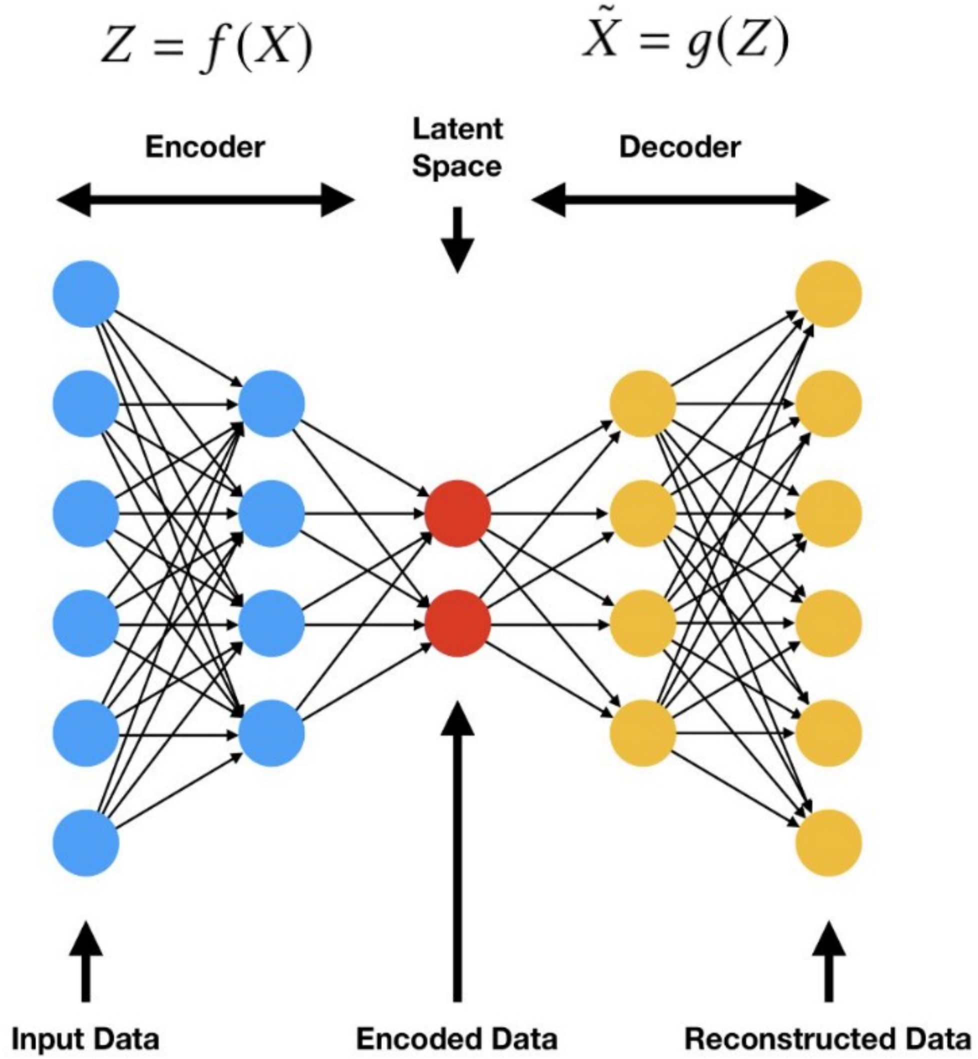
Auto Encoder (AE)

- A special case of MLP
- induce a latent representation Z to enable dimension reduction (dim(Z) < dim(X))
- output the reconstruction of input data

$$\tilde{X} = g(Z)) = g(f(X))$$

- f and g are learnt jointlv by minimizing the reconstruction loss $L(X, \tilde{X})$, the differences between the original input and the reconstruction

Auto Encoder (AE)



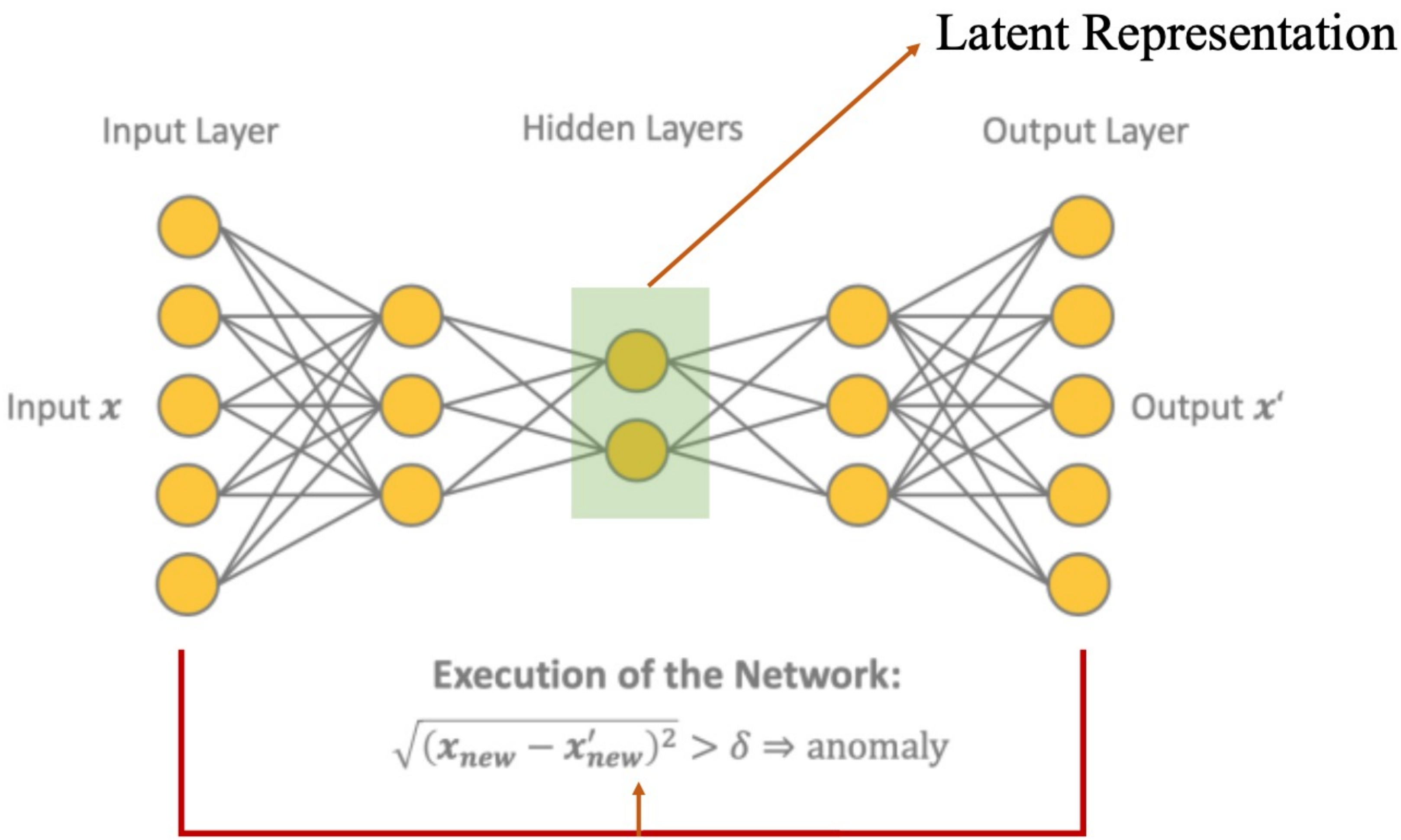
Auto Encoder (AE)

Training of AE requires X are normal data. Once the model is learnt, any input data that cannot be reconstructed back are taken as anomalies

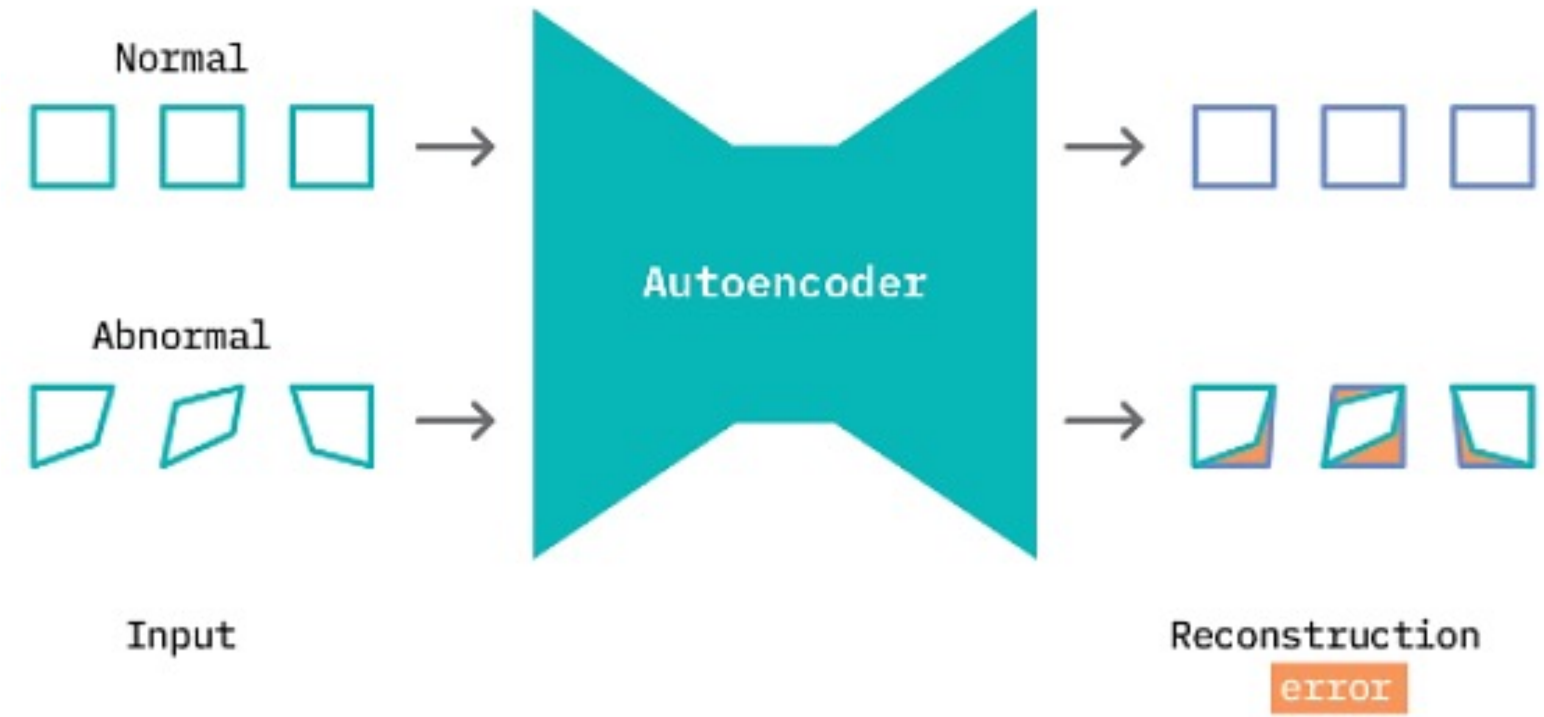
Formulation of AutoEncoder

Mapping Function Map input data to a unified space	**Latent Space** **Encoder $f(\cdot;W_f): X \rightarrow Z$, Decoder $g(\cdot;W_g): Z \rightarrow X$**
Anomaly Score Define the anomaly measure in the unified space	**$d(x) = \|\| x - f(g(x)) \|\|$**
Decision Threshold Determine whether the data is anomalous	**x is anomaly if $d(x) > \tau$, normal o.w.** **τ fine-tuned by precision/recall**

Auto Encoder (AE)



Auto Encoder (AE)



Auto Encoder (AE)

- Approach:
  1. Train the autoencoder on normal data (to model normal behavior)
  2. At inference, calculate the reconstruction error: e.g., RMSE deviation between the input instance and the corresponding reconstructed output
  3. If the reconstruction error is less than a threshold then label the instance as normal data, if it is greater than the threshold then label it as abnormal data (anomaly)
     - The manually-selected threshold value allows the user to tune the “sensitivity” to anomalies