Uncertainty Quantification Of Deep Anomaly Detection Algorithms - Initial Research Update

Yuchen Fama, Xingyu Cai

The Hartford Steam Boiler Inspection and Insurance Company

April 5, 2019

Overview

1. Deep Anomaly Detection (DAD) - A Brief Overview
   - Unsupervised DAD
   - Semi-Supervised DAD
   - Supervised DAD
   - The Importance of DL Model Uncertainty Quantification
2. Literature Review of DAD Model Predictive Uncertainty
   - Dropout as Bayesian Approximation
   - Dropout Distillation
   - Var Output, Adv Training and Ensembles
   - Bayesian Dark Knowledge
3. Review of Conditional VAE
   - Original CVAE
   - Relayr's Solution
   - CVAE for Anomaly Detection and Feature Recon
   - VAE with Anomaly Prior for Anomaly Detection
4. Empirical Study of Anomaly detection
   - VAE based Model
   - VAE with Lower Dimensional Latent Space
   - LSTM based Model
   - LSTM Comparison
   - VAE and LSTM based Models Comparison
5. Adversarial Attack on Sequence Data Model
   - Adversarial Attack on LSTM based Model
   - Adversarial Modification Distribution

Section 2: Deep Anomaly Detection (DAD) - A Brief Overview

Unsupervised DAD

Key Assumption

The normal regions in the original or latent feature space can be distinguished from anomalous regions

Popular Methods

The autoencoders are the most common architecture despite much higher computational cost compared to matrix decomposition based methods such as PCA.

Key Challenges

Sensitive to noise and data corruptions espeically from certain IoT devices.

Semi-Supervised DAD

Key Idea

Semi-supervised (one-class) DAD techniques assume that all training data have only one normal label, and learn a discrimative boundary around normal cases. label.

Key Assumption

Features learned within hidden layers have discriminative attributes for anomalous data points.

Key Advantages

Generally better performance than unsupervised DAD given the usage of labels such as GAN trained in an semi-supervised learning model.

Supervised DAD

Idea

Training a deep supervised binary or multi-class classifier using labels of both normal and anomalous cases.

Key Disadvantages

Not very popular due to the lack of labels.

The Importance of DL Model Uncertainty Quantification

Motivation

DL model uncertainty is indispensable for practitioners espeically in industrial applications. If the model returns a result with high uncertainty, it needs to pass the input to the human (autonomous driving, flight control, etc.)

Business Use Case

HSB IoT Insurance Pricing Model Research - how much "confidence" do we have with our IoT solutions?

Key Question

Bayesian models are popular with mathematically grounded framework for model uncertainty but can we have more practical low cost alternatives for our business use cases?

Section 2: Literature Review of DAD Model Predictive Uncertainty

Dropout as Bayesian Approximation

Paper

Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." international conference on machine learning. 2016.

Key

Link dropout to Deep Gaussian Process, providing predictive uncertainty.

Methodology

$$\mu = \text{sample } \mu, \quad \sigma^2 = \text{sample } \sigma^2 + \pi^{-1} I$$

Here sample means one forward pass of the network. π is the precision hyper-param defined as

$$p(y|x, w) = \mathcal{N}(\hat{y}(x, w), \pi^{-1} I)$$

Dropout as Bayesian Approximation

Estimate π

$$\pi = \frac{pl^2}{2N\lambda}$$

- l: length scale of GP prior
- λ: weight decay param
- p: dropout rate
- N: number of data points

Approximation

Use multiple forward pass and obtain sample mean and sample variance, ignore the π term.

Dropout Distillation

Paper

Gurau, Corina, Alex Bewley, and Ingmar Posner. "Dropout distillation for efficiently estimating model confidence." arXiv (2018).

Knowledge Distillation

Building Two models: Teacher: maximize the accuracy. Student: learn from the Monte-Carlo forward pass from the teacher, to obtain probabilistic result.

Techniques

The teacher's logit z is used as a soft label, not the softmax output. The loss function: $\mathcal{L} = \mathcal{L}_{DD} + \lambda\mathcal{L}_{CE} + \gamma\mathcal{L}_{R}$

- $\mathcal{L}_{DD}$: Distillation dropout loss, the MSE with teacher's logit z
- $\mathcal{L}_{CE}$: Cross Entropy w.r.t ground truth. $\mathcal{L}_{R}$: Regularizer

z is obtained via the mean value by several forward pass.

Var Output, Adv Training and Ensembles

Paper

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles." NIPS. 2017.

Key

- Output both mean and variance from a deep network.
- Loss function as

$$\min -\log p_\theta(y_n|x_n) = \min \frac{\log \sigma_\theta^2(x)}{2} + \frac{(y-\mu_\theta(x))^2}{2\sigma_\theta^2(x)} + \text{const}$$

  The first and the second items are trade-off
- Use adversarial training (generate adversarial samples, retrain the network using those samples) to smooth the predictive distribution
- Train several models to do ensemble

Bayesian Dark Knowledge

Paper

Balan, Anoop Korattikara, et al. "Bayesian dark knowledge." NIPS. 2015.

Key

This is not a Dropout based approach. The key idea is to use distillation.

- Teacher: train via Stochastic Gradient Langevin Dynamics (SGLD), using the training data
- Student: train via regular SGD, using generated data
- Objective:

$$\min KL(S(y|x,w)||q(y|x))$$

where $q(y|x)$ is the posterior predictive distribution obtained via MCMC of the teacher (through SGLD)

Bayesian Dark Knowledge

$$\begin{aligned}\mathcal{L}(w|x) &= KL(p(y|x, D_N)||S(y|x, w)) = -\mathbb{E}_{p(y|x,D_N)} \log S(y|x, w) + \text{const} \\ &= -\int [\int p(y|x, \theta)p(\theta|D_N)d\theta] \log S(y|x, w)dy \\ &= -\int p(\theta|D_N) \int p(y|x, \theta) \log S(y|x, w)dyd\theta \\ &= -\int p(\theta|D_N)[\mathbb{E}_{p(y|x,\theta)} \log S(y|x, w)]d\theta\end{aligned}$$

Using the Monte-Carlo approximation:

$$\mathcal{L}(w|x) = -\frac{1}{|\theta|} \sum_{\theta^S \in \theta} \mathbb{E}_{p(y|x,\theta^S)} \log S(y|x, w)$$

where θ is set of samples from $p(\theta|D_N)$ obtained via SGLD when training the teacher. To integrate out x, we geenrate D' close to D and obtained

$$\hat{\mathcal{L}}(w) = \frac{1}{|D'|} \sum_{x' \in D'} \mathcal{L}(w|x') = -\frac{1}{D}\frac{1}{D'} \sum_{\theta^S \in \theta} \sum_{x' \in D'} \mathbb{E}_{p(y|x',\theta^S)} \log S(y|x', w)$$

Bayesian Dark Knowledge

The detailed algorithm:

for t = 1 to T do:
  //train teacher with SGLD
  sample minibatch S of size M in D
  sample $z_t \sim N(0, \eta_t I)$
  update $\theta_{t+1} = \theta_t + \frac{\eta_t}{2}(\nabla_\theta \log p(\theta|\lambda) + \frac{N}{M}\sum_{(x,y)\in S} \log p(y|x,\theta)) + z_t$
  //train student using SGD
  samle D' of size M close to D
  update $w_{t+1} = w_t - \rho_t(\frac{1}{M}\sum_{x\in D'} \nabla_w \hat{\mathcal{L}}(w, \theta_{t+1}|x') + \gamma w_t)$

Bayesian Dark Knowledge

- For classification, use Cross Entropy as

$$\mathcal{L}(w|\hat{\theta}^S, x) = -\sum_{k=1} Kp(y = k|x, \theta^S) \log S(y = k|x, w)$$

- For regression: the student output both μ and σ, use $e^{\alpha(x,w)}$ instead of $\sigma^2(x|w)$ to avoid PSD constraint.

$$\begin{aligned}\hat{\mathcal{L}}(w|\theta^S, x) &= -\mathbb{E}_{p(y|x,\theta^S)} \log N(y|\mu(x, w), e^{\alpha(x,w)}) \\ &= \frac{1}{2}[\alpha(x, w) + e^{-\alpha(x,w)}((fx|\theta^S) - \mu(x, w))^2 + \frac{1}{\lambda_N}]\end{aligned}$$

where λ_N is associated with the observation model

$$p(y_i|x_i, \theta) = N(y_i|f(x_i|\theta), \lambda_N^{-1})$$

Section 3: Review of Conditional VAE

Original CVAE

Paper

Sohn, Kihyuk, Honglak Lee, and Xinchen Yan. "Learning structured output representation using deep conditional generative models." Advances in neural information processing systems. 2015.

Design

In the encoding phase, label $\rightarrow$ one-hot encoding $\rightarrow$ concatenated to x
In the decoding phase, label $\rightarrow$ one-hot encoding $\rightarrow$ concatenated to z (merge with μ, σ).

Key

Match the conditional posterior (condition on label) with prior.

$$\mathcal{L}_{CVAE} = -KL(q_\phi(z|x,y)||p_\theta(z|x)) + \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(y|x,z)]$$

Original CVAE

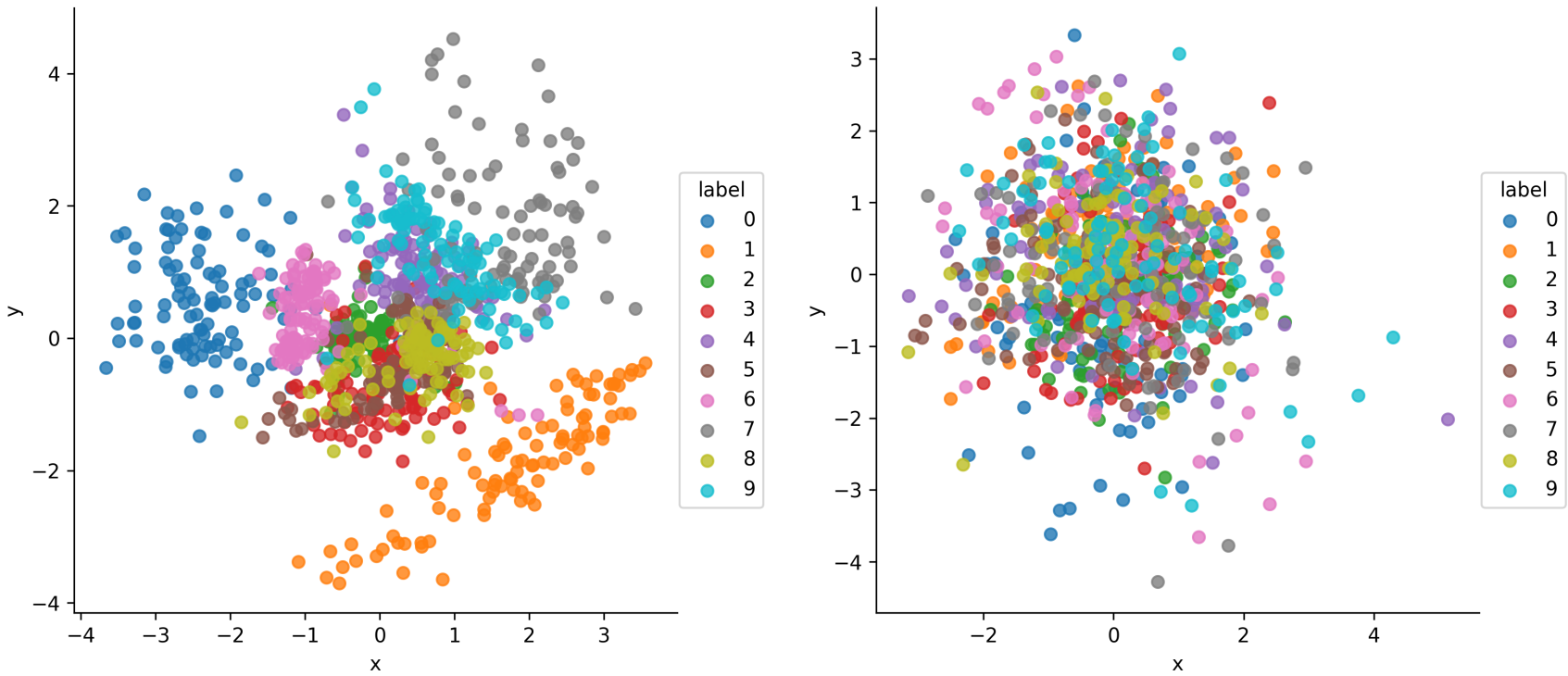


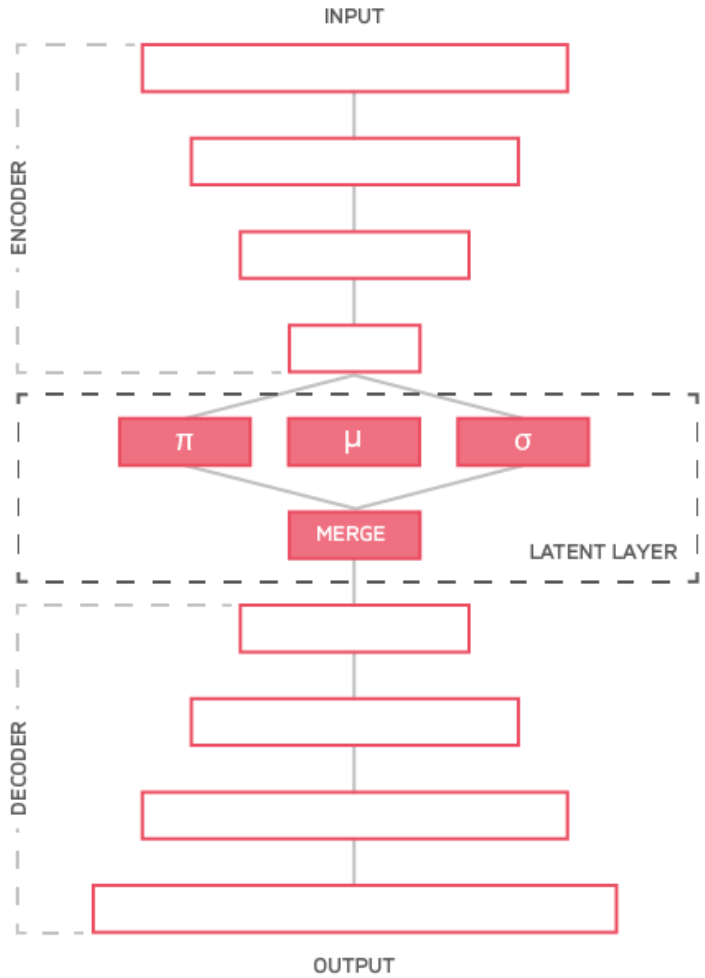
Figure: Latent space for MNIST data: left is VAE, right is CVAE

Regular VAE has to split points belonging to different categories in the latent space, but CVAE latent space is much simpler (Gaussian) for each category, because it is conditioned on the label.

Relayr's Solution

Blog

https://relayr.io/blog/one-model-to-rule-them-all/3/



Relayr's Solution

Loss Function

$$\mathcal{L} = \mathcal{L}_{\text{BCE}}(x, \hat{x}) + \mathcal{L}_{CE}(\pi, y) + KL(p(z|x)||p(z))$$

where x is original input; $\hat{x}$ is reconstructed input; BCE is binary cross entropy; CE is cross entropy; π is the predicted label; y is the original label; $p(z|x)$ is the posterior distribution of latent code z; $p(z)$ is prior of z.
As a comparison, original VAE loss is:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}}(x, \hat{x}) + KL(p(z|x)||p(z))$$

which does not have the label term.

Relayr's Solution

In supervised case (classification task)

In the training, use the entire network; in the testing, cut off the decoder part.

In anomaly detection case

In the training, use the entire network; in the testing, cut off the π part.

Key

The encoder is both a encoder and a classifier. The loss can be viewed as $\mathcal{L} = \mathcal{L}_{CE}(\pi, y) + \mathcal{R}$, which is a regularized classifier. The entire decoder and KL is the regularizer.

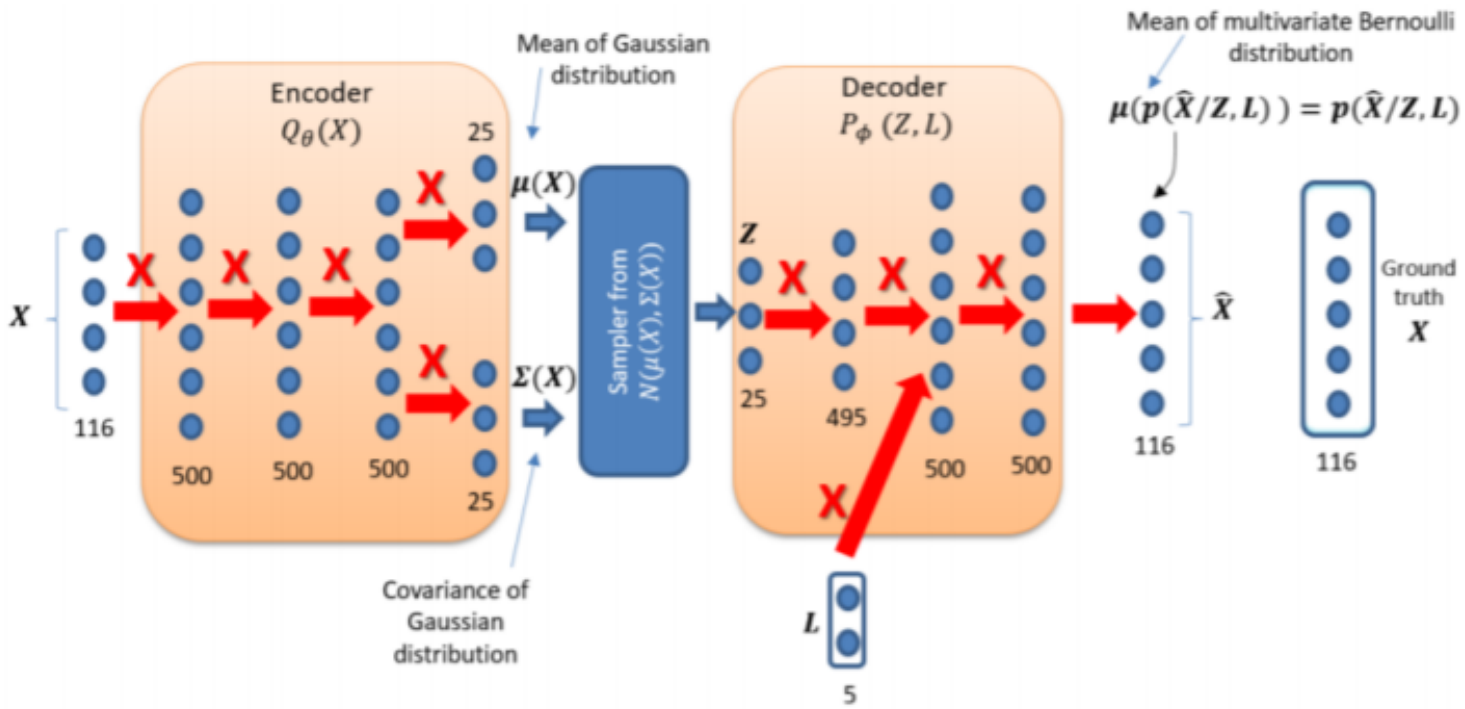
Concern

How to merge π, μ and σ?

CVAE for Anomaly Detection and Feature Recon

Paper

Lopez-Martin, Manuel, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret. "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot." Sensors 17, no. 9 (2017): 1967.



CVAE for Anomaly Detection and Feature Recon

Classification task

Find the label with the minimum reconstruction error. This step requires looping all the labels, and could be time consuming.

Feature reconstruction task

Step 1, train the model with the data that has missing features. Step 2, given the test data with missing features, first perform the classification task and obtain the most possible label, then use the predicted label to do reconstruction (output of the decoder)

Key

The label is concatenated to the 3rd layer of the decoder.

Concern

Why concatenate the label at the 3rd layer?

VAE with Anomaly Prior for Anomaly Detection

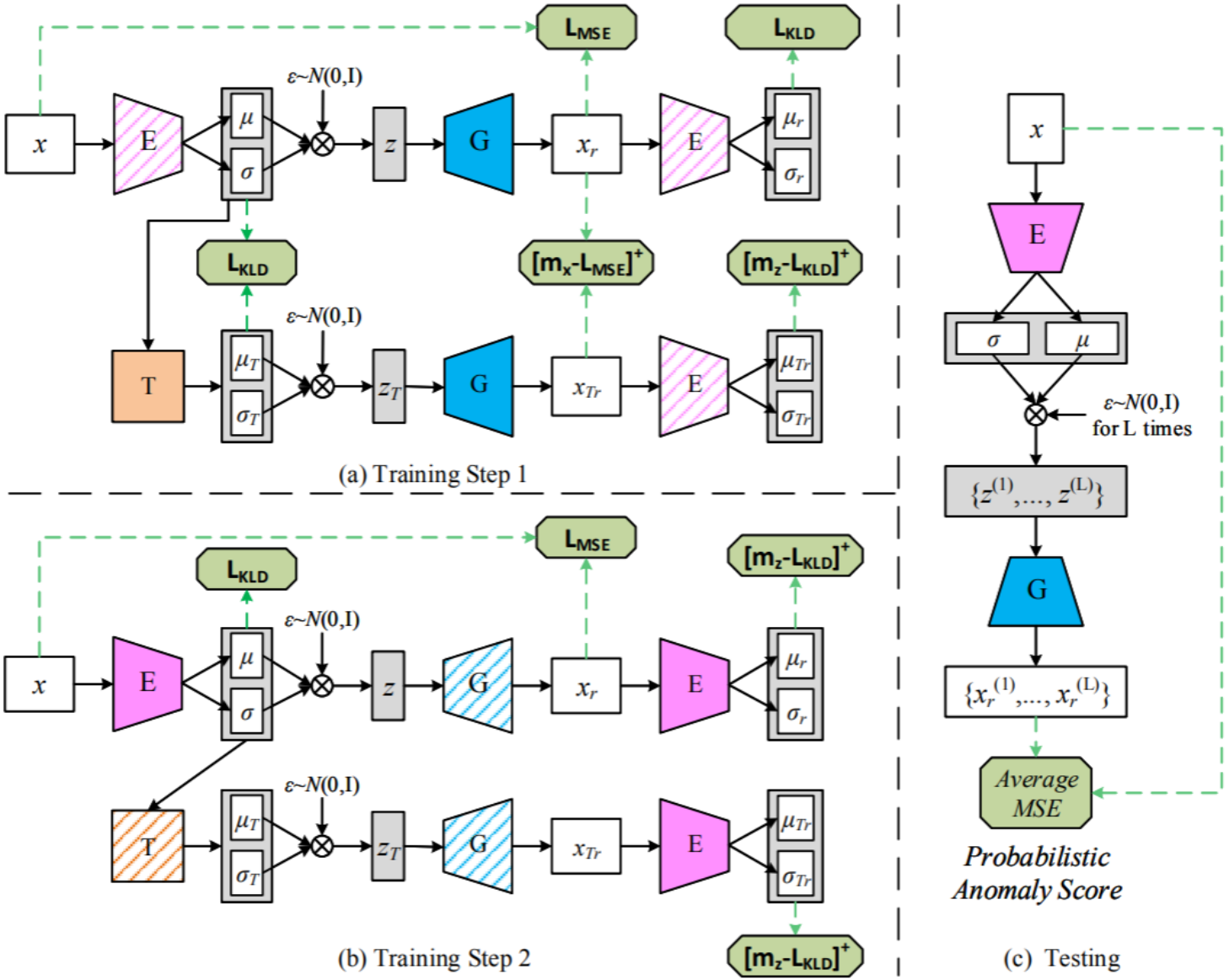
Paper

Wang, Xuhong, et al. "Self-adversarial Variational Autoencoder with Gaussian Anomaly Prior Distribution for Anomaly Detection." arXiv (2019).

Key

- Propose an anomaly prior distribution
- Propose a Transformer T that maps regular latent variable to anomaly latent variable
- Perform alternative training: train T and generator G in one step, train encoder E in another step
- Adv: T will generate anomalous latent variables that close to the normal latent variables, and G will distinguish them by different reconstruction errors.
- Testing for anomaly is the same as other VAE based approach

VAE with Anomaly Prior for Anomaly Detection



VAE with Anomaly Prior for Anomaly Detection

Train T and G

For T, minimize the difference of latent variable z and $T(z)$.

$$\mathcal{L}_T = KL(N(\mu, \sigma^2)||(N(\mu_T, \sigma_T^2))) = \log \frac{\sigma_T}{\sigma} + \frac{\sigma^2 + (\mu - \mu_T)^2}{2\sigma_T^2} - \frac{1}{2}$$

For G, two parts $\mathcal{L}_G = \mathcal{L}_{Gz} + \mathcal{L}_{GzT}$, where

$$\mathcal{L}_{Gz} = \mathcal{L}_{MSE}(x, G(z)) + \mathcal{L}_{KLD}(E(G(z))||p(z))$$

which is a regular VAE loss, with $p(z)$ regular prior. Also, the second part

$$\mathcal{L}_{GzT} = [m_x - \mathcal{L}_{MSE}(G(z), G(z_T))]^+ + [m_z - \mathcal{L}_{KLD}(E(G(z_T))||p(z))]^+$$

where m_x, m_z are threshold params. This tries to maximize diff between $G(z)$ and $G(z_T)$, and maximize the KL of $E(G(z_T))$ and normal prior. The threshold ensure the diff is bounded.

VAE with Anomaly Prior for Anomaly Detection

Train E

$$\mathcal{L}_E = \mathcal{L}_{KLD}(E(x)||p(z)) + \mathcal{L}_{MSE}(x, G(z)) + [m_z - \mathcal{L}_{KLD}(E(G(z))||p(z))]^+ + [m_z - \mathcal{L}_{KLD}(E(G(T(z)))||p(z))]^+$$

The first two terms are regular VAE loss. The last two terms prevent E from mapping the reconstructions to the prior distribution. This trains the E to discriminate the training samples (normal) and their reconstruction (anomalous)

Testing

Testing is identical to the traditional VAE testing. Perform MC samples of encoded latent code, and set threshold on reconstruction error.

Section 4: Empirical Study of Anomaly detection

VAE based Model

Methodology

Use partial data (the data sequence exclude the last data point) to train VAE, reconstruct the full sequence. Based on the prediction error (the last data point), we set 5% threshold on the empirical distribution. The variance in the latent space is represented by the encoder.

Design

- Data: Config Lab Closet Window, temperature data. Training data: 20000 points (starting from 2019-02-01 00:00:00). Testing data: 2880 points (30 days)
- Sample rate: 15 min
- Input: 27 points (28 exclude the last one); Output: 28 points
- Loss: $\mathcal{L} = \alpha \mathcal{L}_{\text{recon}}(27 \text{ points}) + (1-\alpha)\mathcal{L}_{\text{pred}}(1 \text{ point}) + KL$
- Encoder: 4 layers Conv + ReLU + BN, with dimension 64 to 1024
- Decoder: 4 layers, opposite to encoder.
- Latent space dimension: 20

VAE based Model

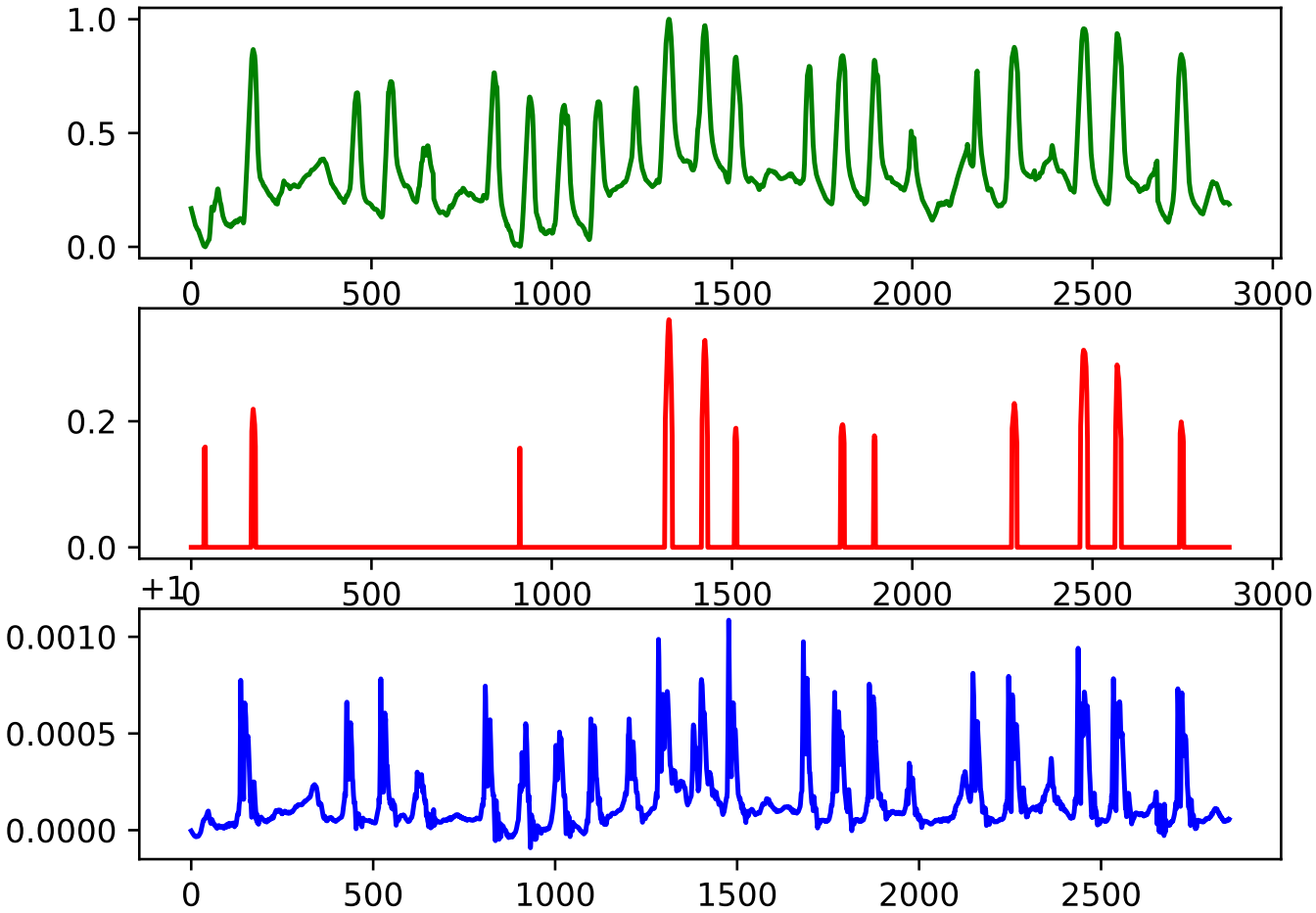


Figure: The data and prediction: from top to bottom, orignal data, detected anomaly, standard deviation of prediction

VAE based Model

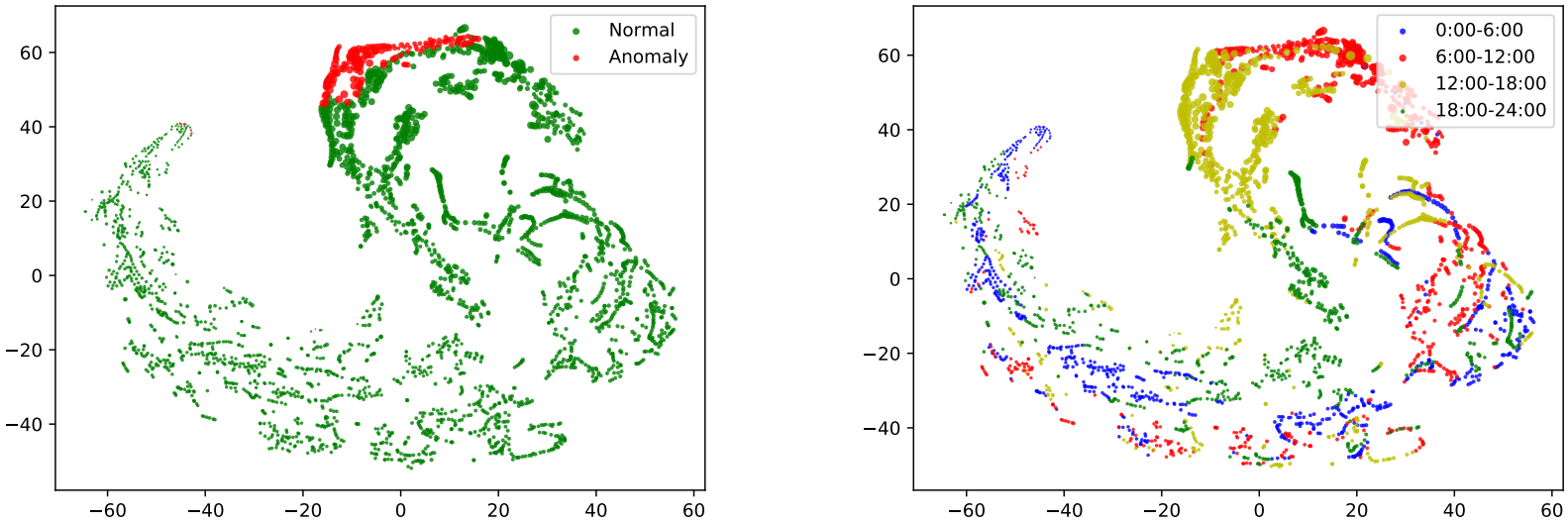


Figure: Latent space: left shows the anomaly data (red), right is the data for different time slots.

- The data lies in manifold, different time slots are different manifolds.
- High prediction error corresponds to outliers in latent space.
- The larger variance of $p(z|x)$, the larger radius of the circle in the figure.

VAE with 5 Dimensional Latent Space

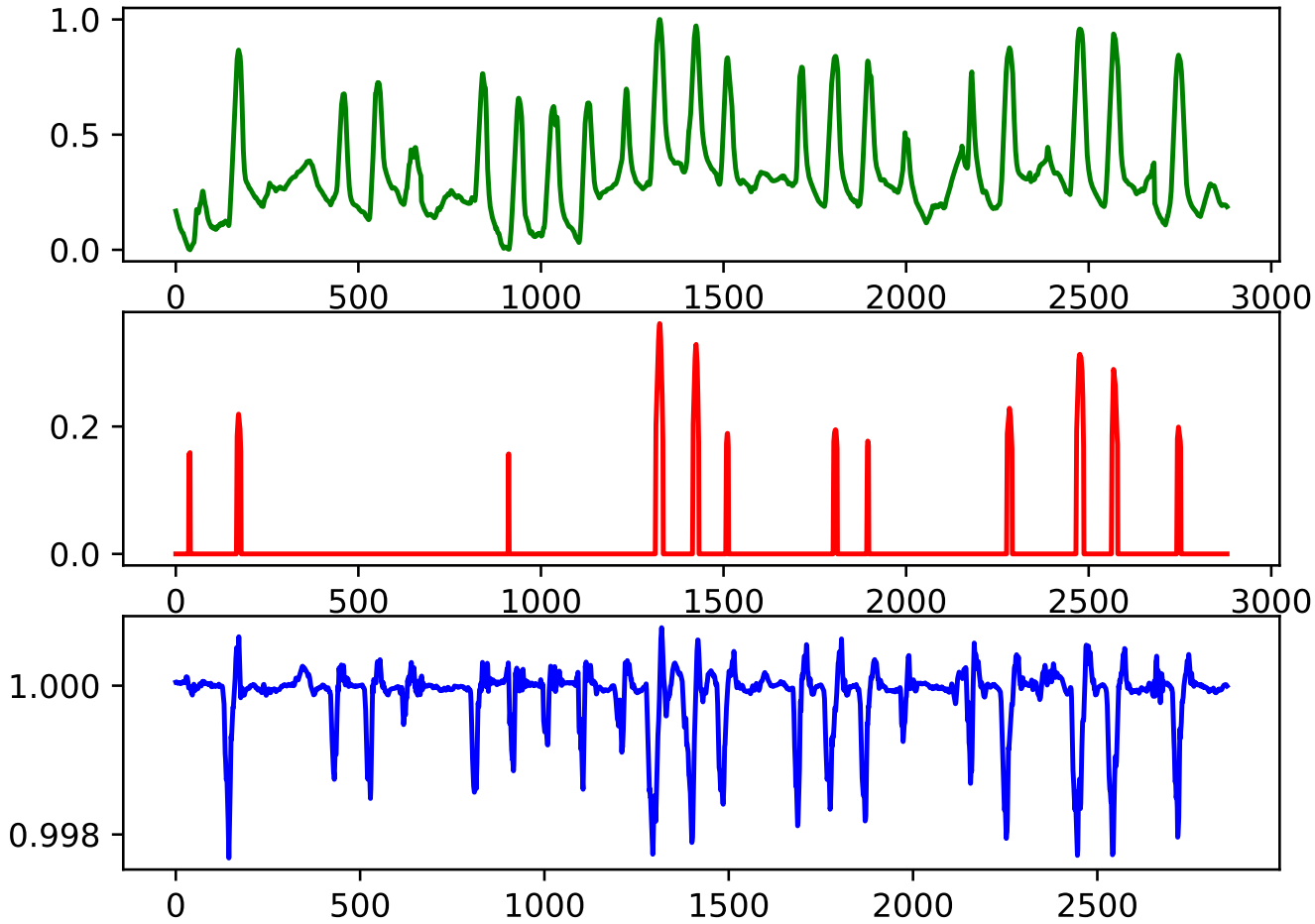


Figure: The data and prediction: from top to bottom, orignal data, detected anomaly, standard deviation of prediction

VAE with 2 Dimensional Latent Space

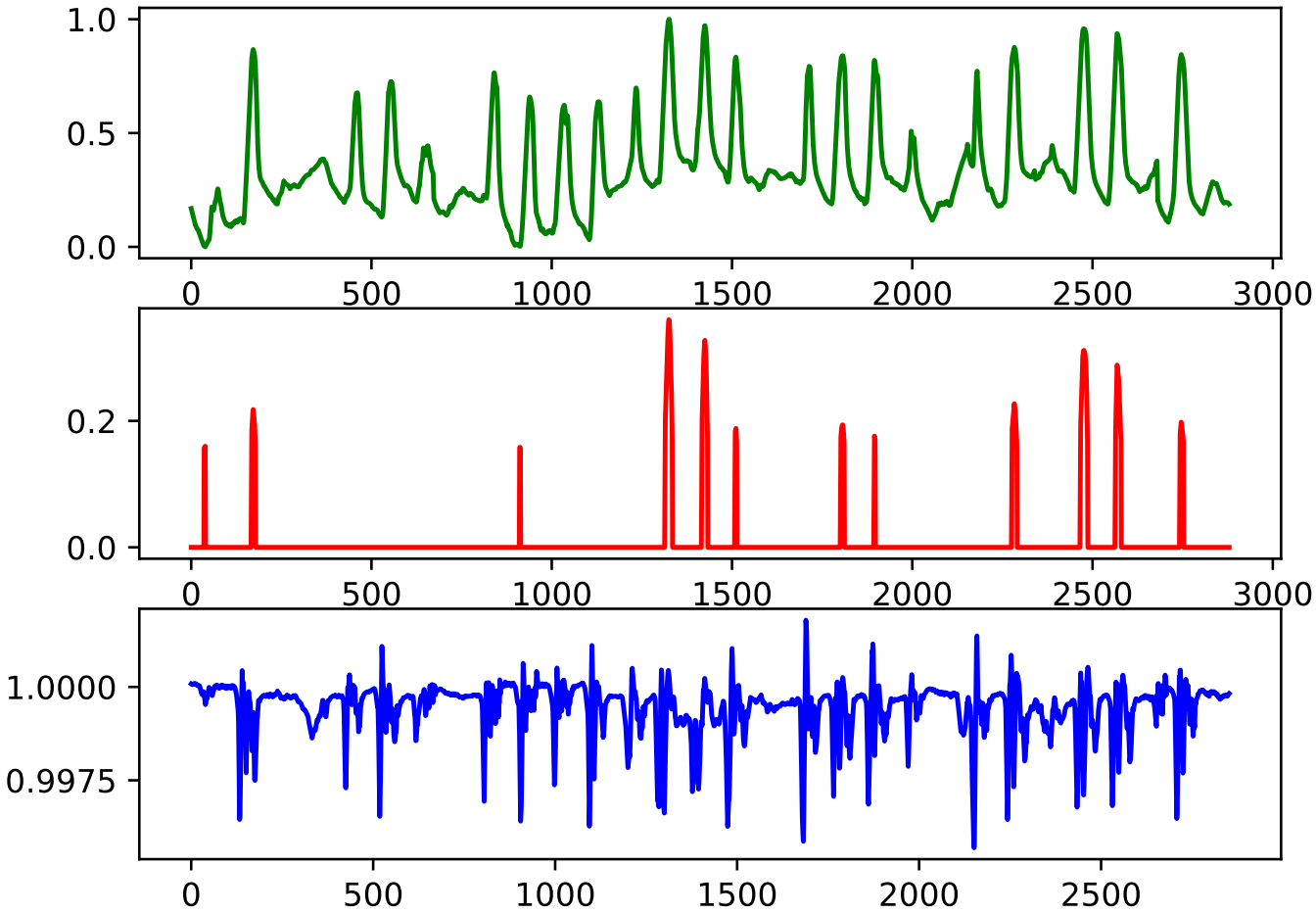


Figure: The data and prediction: from top to bottom, orignal data, detected anomaly, standard deviation of prediction

VAE with Lower Dimensional Latent Space

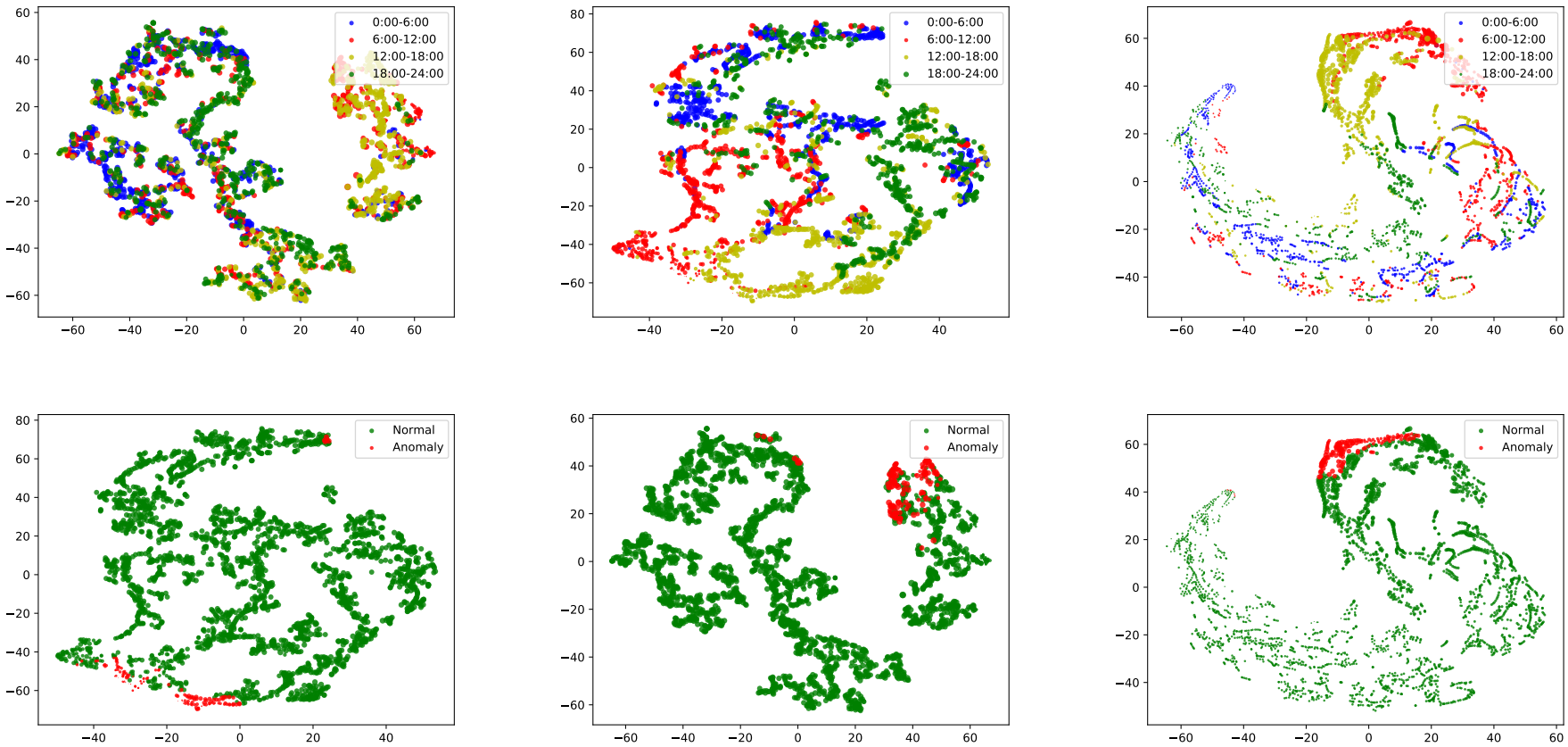


Figure: Left to Right: 2-D, 5-D, 20-D. For lower latent space dimension, the manifold is less clear.

LSTM based Model

Methodology

Use sequential data train LSTM and predict the next data point. Based on the prediction error, we set 5% threshold on the empirical distribution.

Design

- Sample rate: 15 min
- Input: 27 points
- Output: 1 point
- Loss: MSE of the prediction
- State space dimension: 5
- Model architecture 1: 2 layers LSTM + 1 layer linear (in 20, out 1)
- Model architecture 2: 4 layers LSTM + 1 layer linear (in 20, out 1)
- # of forward pass: 20

LSTM based Model - 2 Layers LSTM

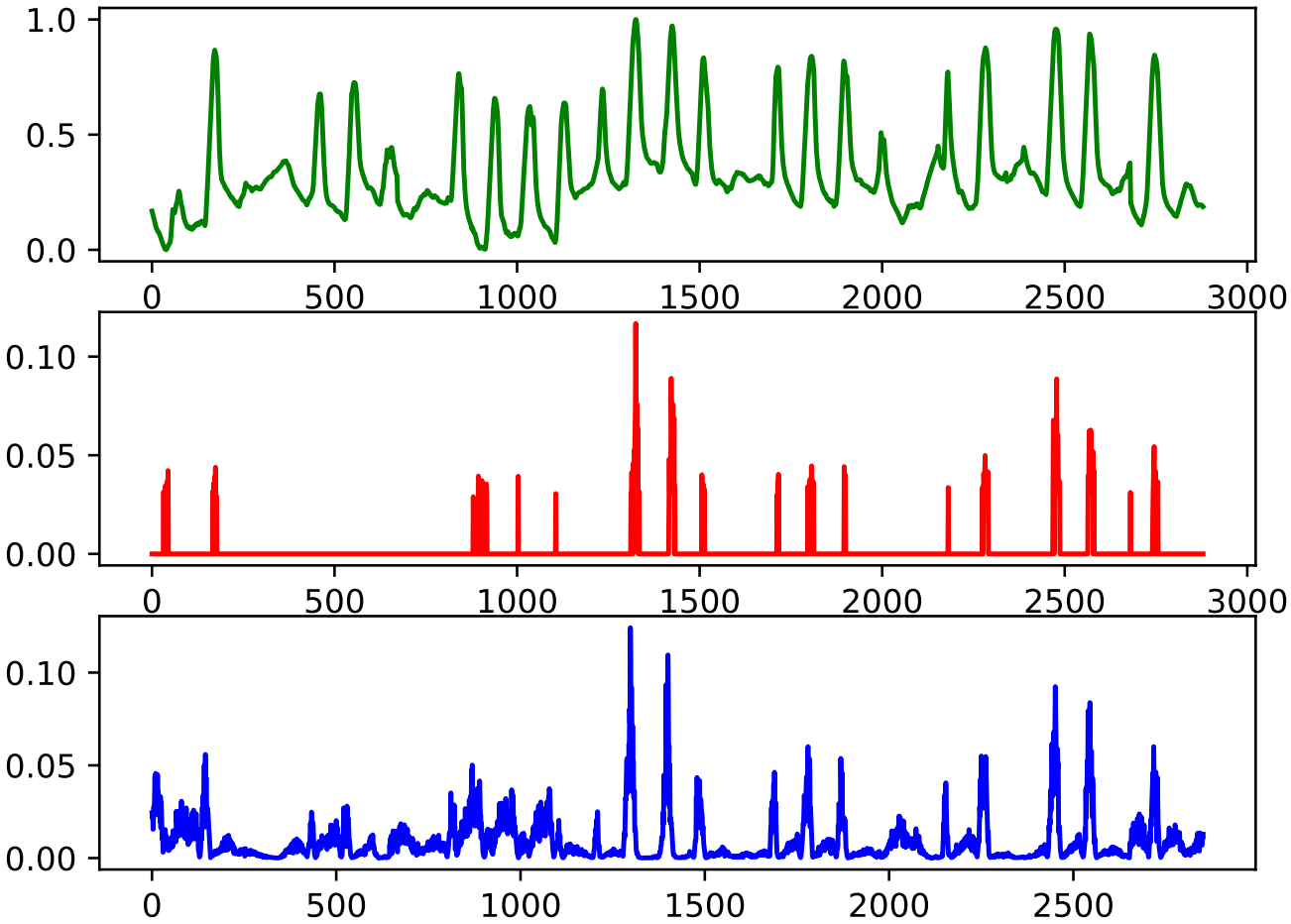


Figure: The data and prediction: top is the orignal data, middle is the detected anomaly, bottom is standard deviation of prediction

LSTM based Model - 4 Layers LSTM

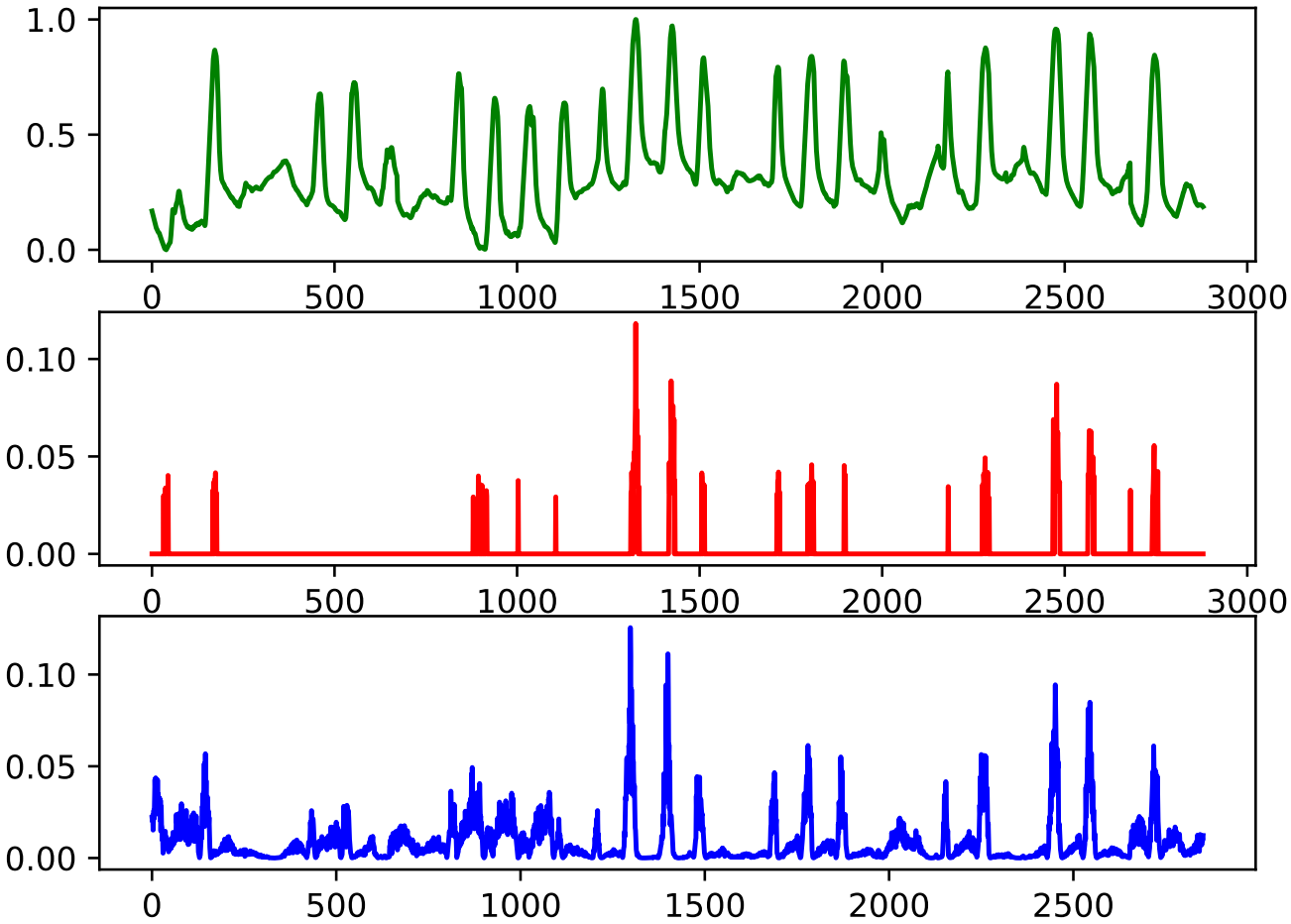


Figure: The data and prediction: top is the orignal data, middle is the detected anomaly, bottom is standard deviation of prediction

LSTM with Higher Hidden State Dimension

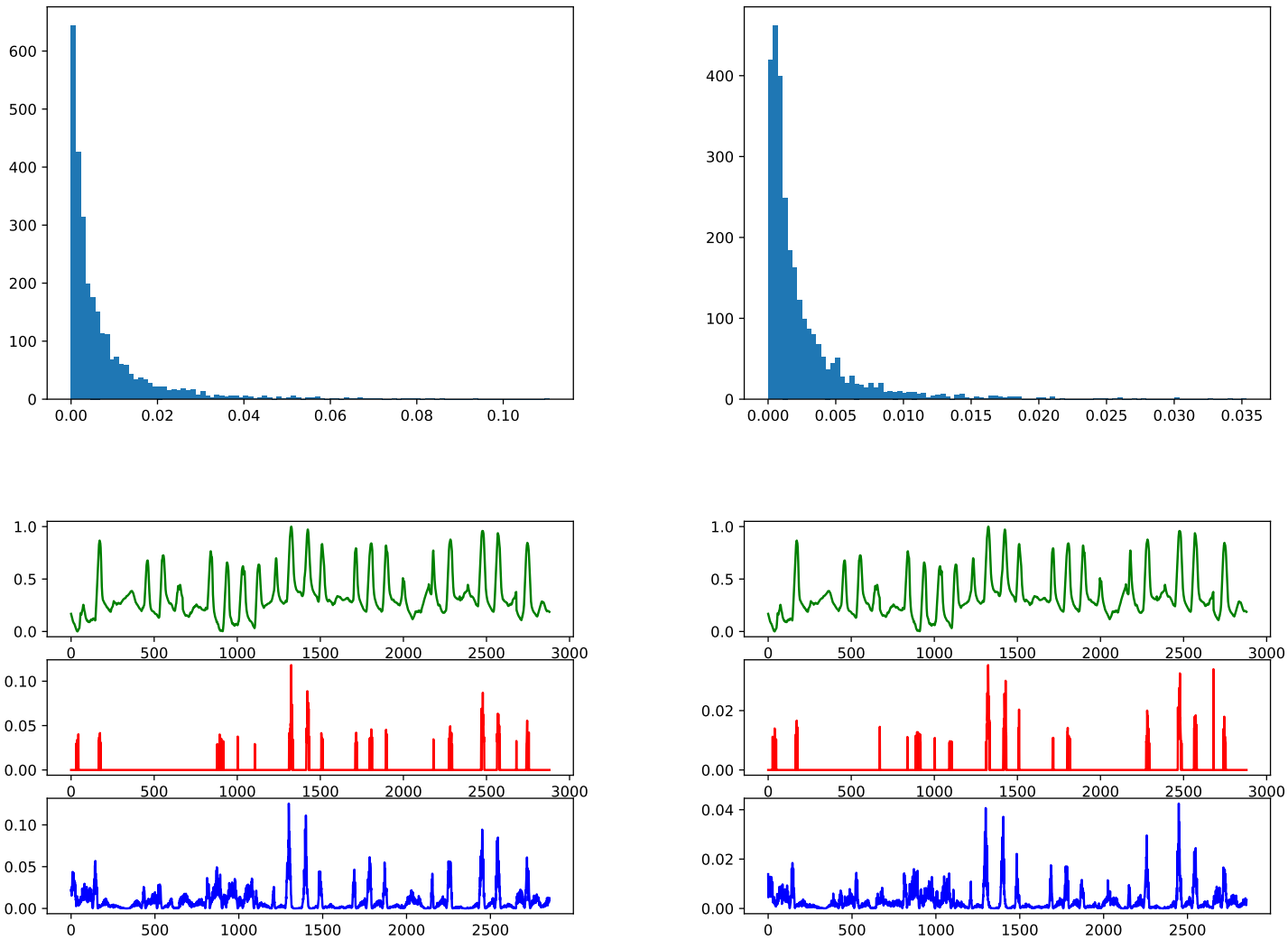


Figure: Left: 5-D hidden space; Right: 20-D hidden space; Up: pred loss; Bottom: detect.

Comparison

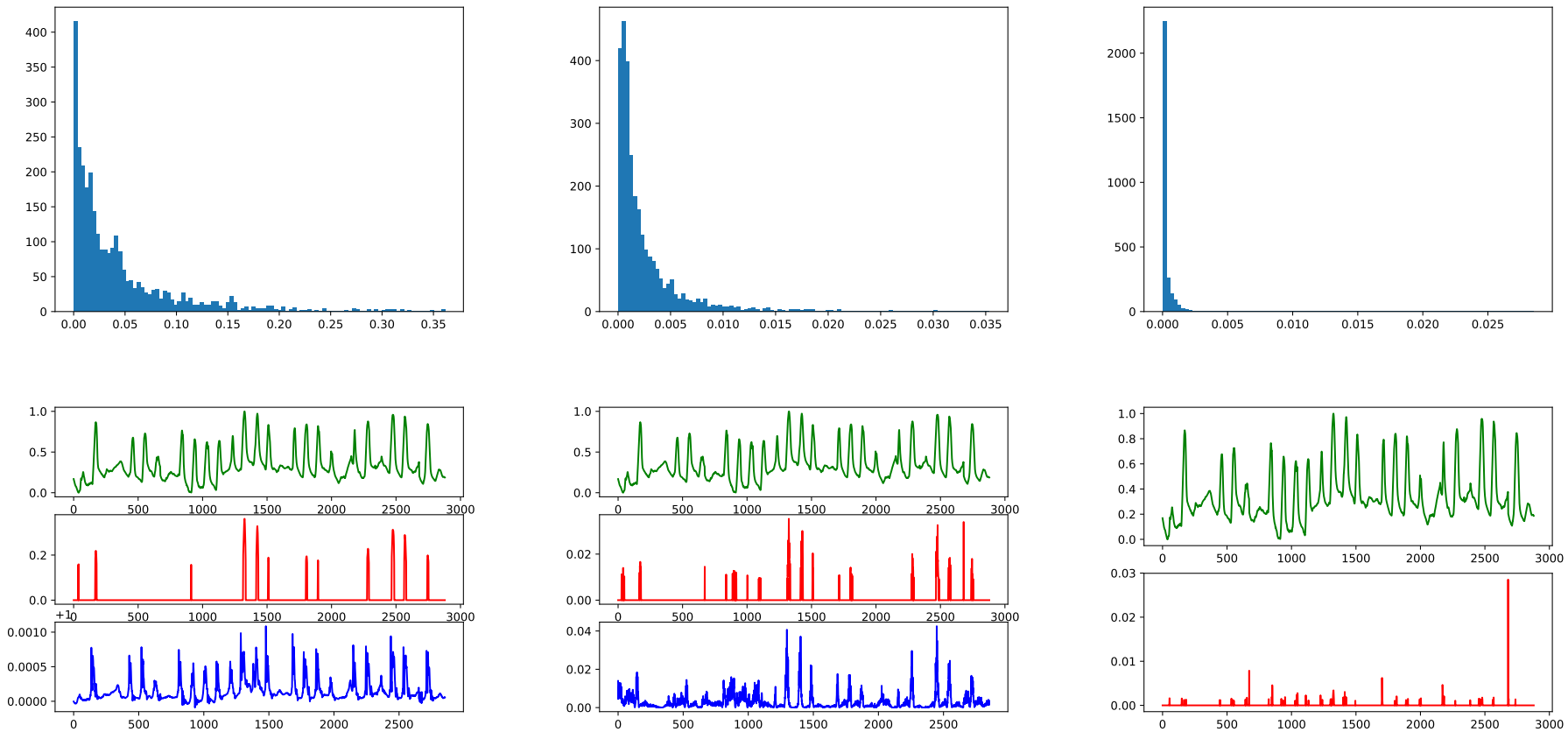


Figure: Row 1: prediction loss; Row 2: data and detection. Left to right: VAE, 4-layer LSTM with Dropout, 4-layer LSTM without Dropout

Section 5: Adversarial Attack on Sequence Data Model

Adversarial Attack on LSTM based Model

Attack Method

The adversarial goal is to maximize the accumulated MSE of the prediction, i.e.

$$\min_{\delta} \mathcal{L} = \min_{\delta} -\sum^{N}[(\mathcal{F}(x+\delta|\theta) - y)^2]$$
$$\text{s.t.} \quad x+\delta \in \mathcal{C} = \{x+\delta : ||x+\delta||_p \leq (1+\epsilon)||x||_p\}$$

where $\mathcal{L}$ is the adversarial loss, $\mathcal{F}$ is the model, x is the data, y is the label, N is the number of data points, δ is the modification, ϵ is the bound parameter, $\mathcal{C}$ represents the feasible set within the bound.
Using Projected Gradient Descent, we can perform the attack

$$\hat{x} \leftarrow \text{Proj}_{\mathcal{C}}(x - \eta \nabla_x \mathbb{E}[\mathcal{L}])$$

where $\text{Proj}_{\mathcal{C}}$ performs projection to constraint set $\mathcal{C}$.

Adversarial Attack on LSTM based Model

Settings

- ϵ: we set to 0.01 and 0.05
- p-norm: we use ∞-norm to simplify the projection
- $\text{Proj}_{\mathcal{C}_\infty}(x)$: clamp function under ∞-norm

The error amplifying ratio ρ

ρ is defined as the averaged prediction error after performing the attack, versus the error before attack.

ρ	2-layer	4-layer	8-layer
$\epsilon = 0.01$	2.66	4.05	2.10
$\epsilon = 0.05$	21.69	45.86	12.99

As expected, the larger ϵ the larger error. Even with 1% modification, the error could be increased to 4 times. For 5% modification, the error increases to 45 times. However, for 8-layer case, it is significantly better, contrary to the common thinking.

Adversarial Modification Distribution

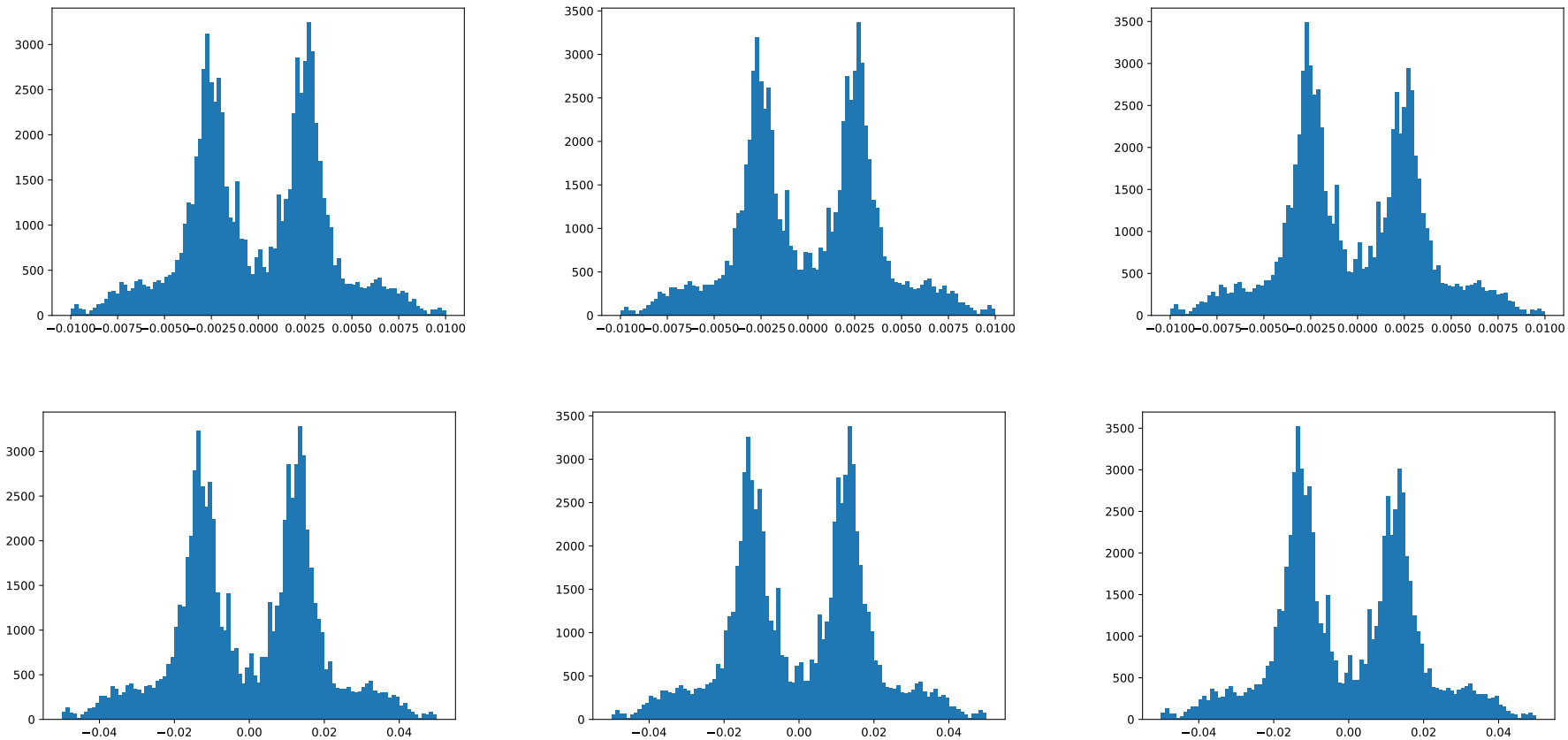


Figure: Row-1: $\epsilon = 0.01$; Row-2: $\epsilon = 0.05$. Left: 2-layer; Mid: 4-layer; Right: 8-layer. $|\delta|$ shows a skewed distribution. For 8-layer case, it is not symmetric.

Thanks!