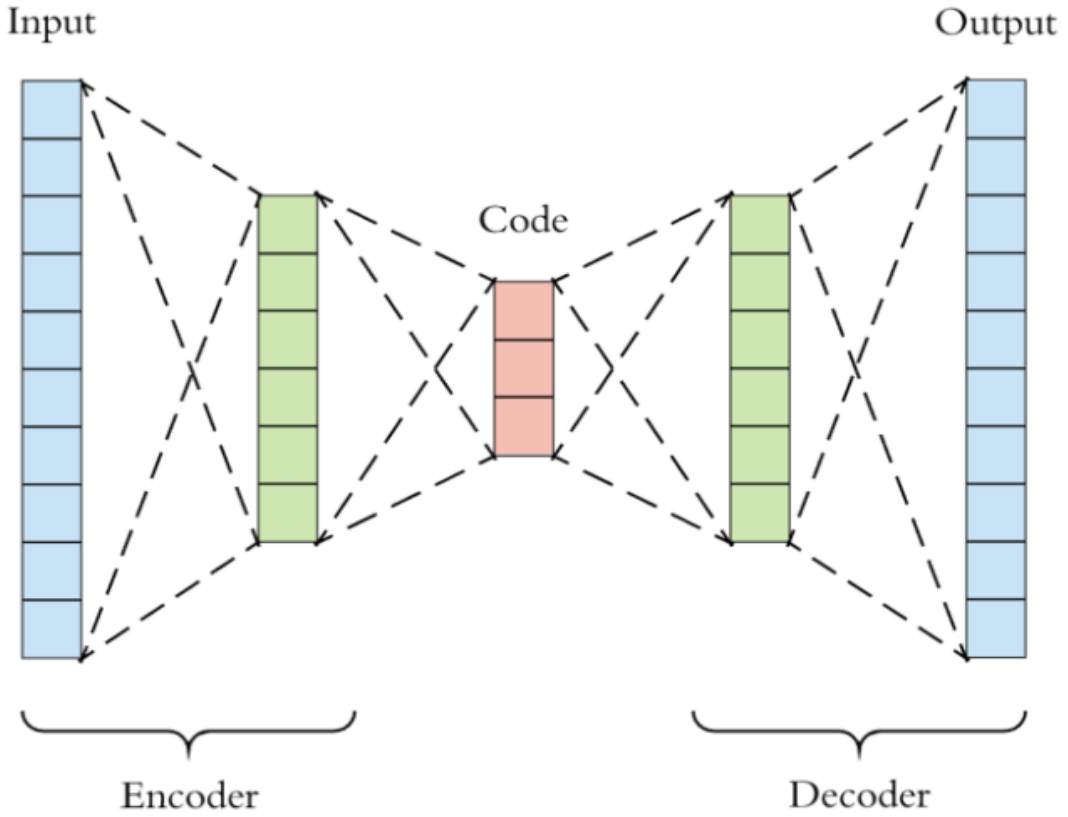
Weekly Meeting

2020.7.1

Topic: Autoencoder & Variational Autoencoder

What are Autoencoders

Autoencoders are surprisingly simple neural architectures. They are basically a form of compression, similar to the way an audio file is compressed using MP3, or an image file is compressed using JPEG.



Features of Autoencoders

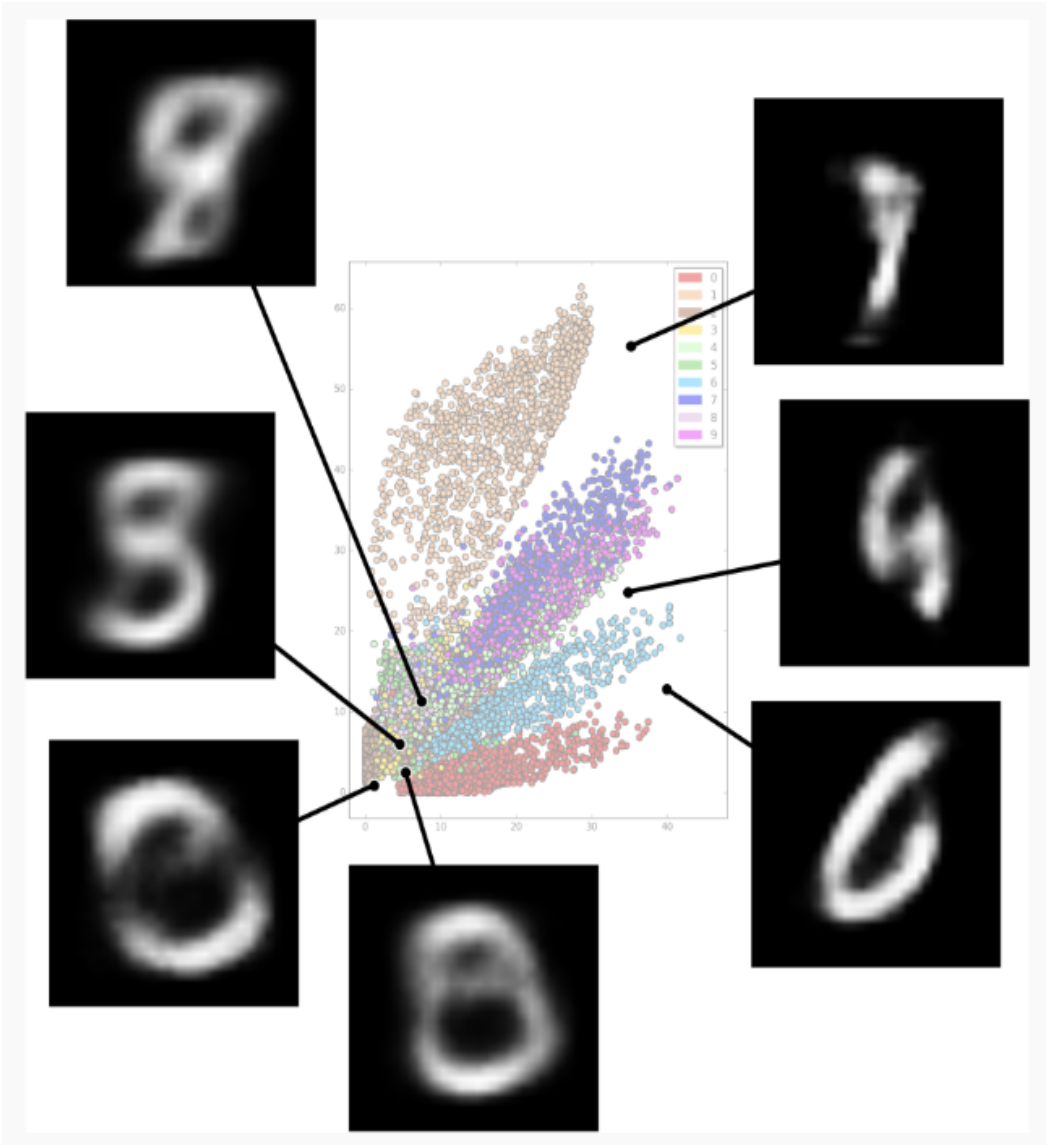
- Autoencoders are closely related to principal component analysis (PCA).
- Generally, the activation function used in autoencoders is non-linear, typical activation functions are ReLU and sigmoid.
- The encoding network can be represented as $\mathbf{z} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$
- The decoding network can be represented as $\mathbf{x}' = \sigma'(\mathbf{W}'\mathbf{z} + \mathbf{b}')$
- The loss function can then be written as

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')\|^2$$

- Since the input and output are the same images, this is not really supervised or unsupervised learning, so we typically call this self-supervised learning.

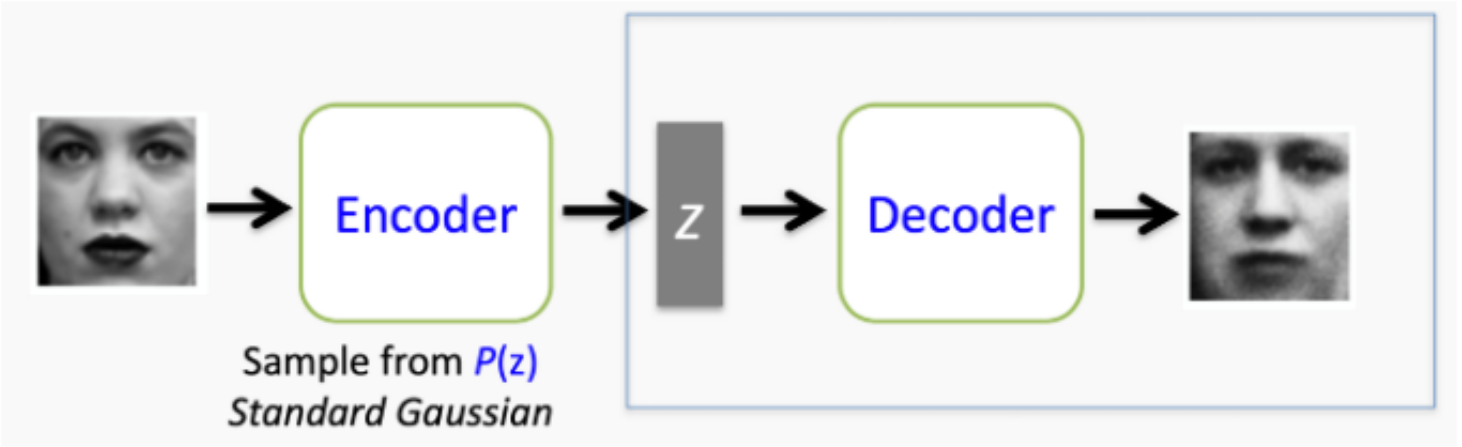
Weakness of Autoencoders

- Gaps in the latent space
- Separability in the latent space
- Discrete latent space



Variational Autoencoders

VAEs inherit the architecture of traditional autoencoders and use this to learn a data generating distribution, which allows us to take random samples from the latent space.

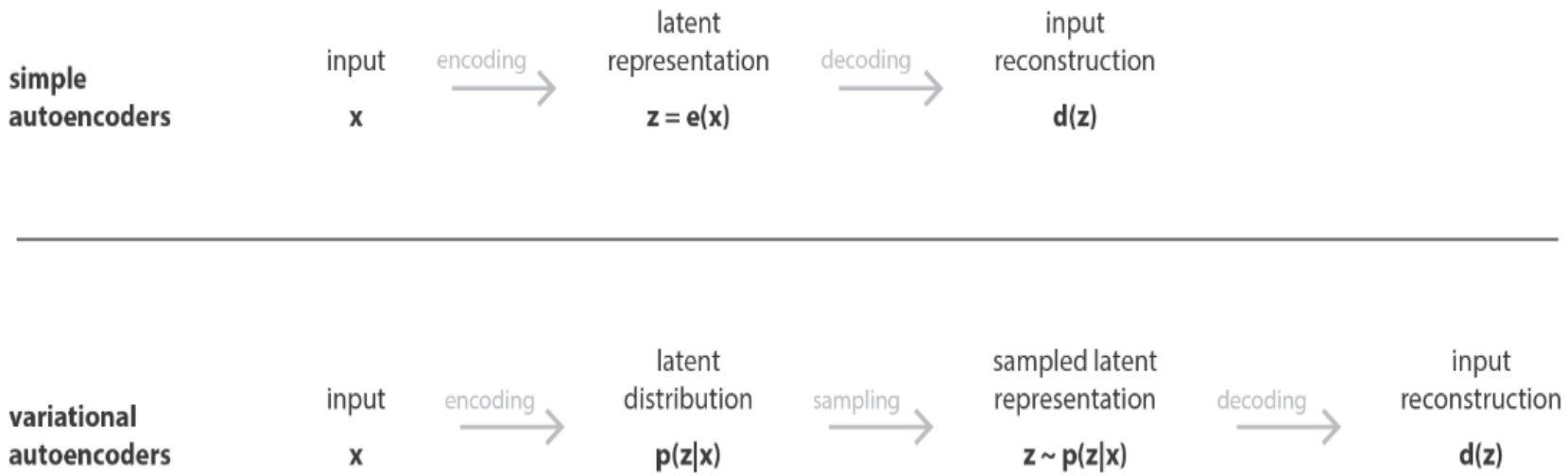


Features of VAE

- AE encodes an input as a single point, VAE encode it as a distribution over the latent space.
- The training steps of VAE are as follows:
  - first, the input is encoded as distribution over the latent space
  - second, a point from the latent space is sampled from that distribution
  - third, the sampled point is decoded and the reconstruction error can be computed
  - finally, the reconstruction error is backpropagated through the network

Features of VAE

- Comparison of AE & VAE



Features of VAE

- Loss function of a datapoint in VAE:

$$l_i(\theta, \phi) = -\mathbb{E}_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i \mid z)] + \mathbb{KL}(q_\theta(z \mid x_i) \,||\, p(z))$$

Reconstruction error

KL divergence

$$\begin{aligned}
\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_{\mathbf{z}\sim q} \log q(\mathbf{z}) - \mathbb{E}_{\mathbf{z}\sim q} \log p(\mathbf{z}|\mathbf{x}) \\
&= \underbrace{\mathbb{E}_{\mathbf{z}\sim q} \log q(\mathbf{z}) - \mathbb{E}_{\mathbf{z}\sim q} \log p(\mathbf{z}, \mathbf{x})}_{\text{(a) — -1*ELBO}} + \underbrace{\log p(\mathbf{x})}_{\text{(b)}} \\
&= -\mathrm{ELBO}(q) + \log p(\mathbf{x})
\end{aligned}$$

Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection

——SOTA on ECG5000

Background & Motivation

Background: The amount of time series data generated in Healthcare is growing very fast and so is the need for methods that can analyse these data, detect anomalies and provide meaningful insights. However, most of the data available is unlabelled and, therefore, anomaly detection in this scenario has been a great challenge for researchers and practitioners.

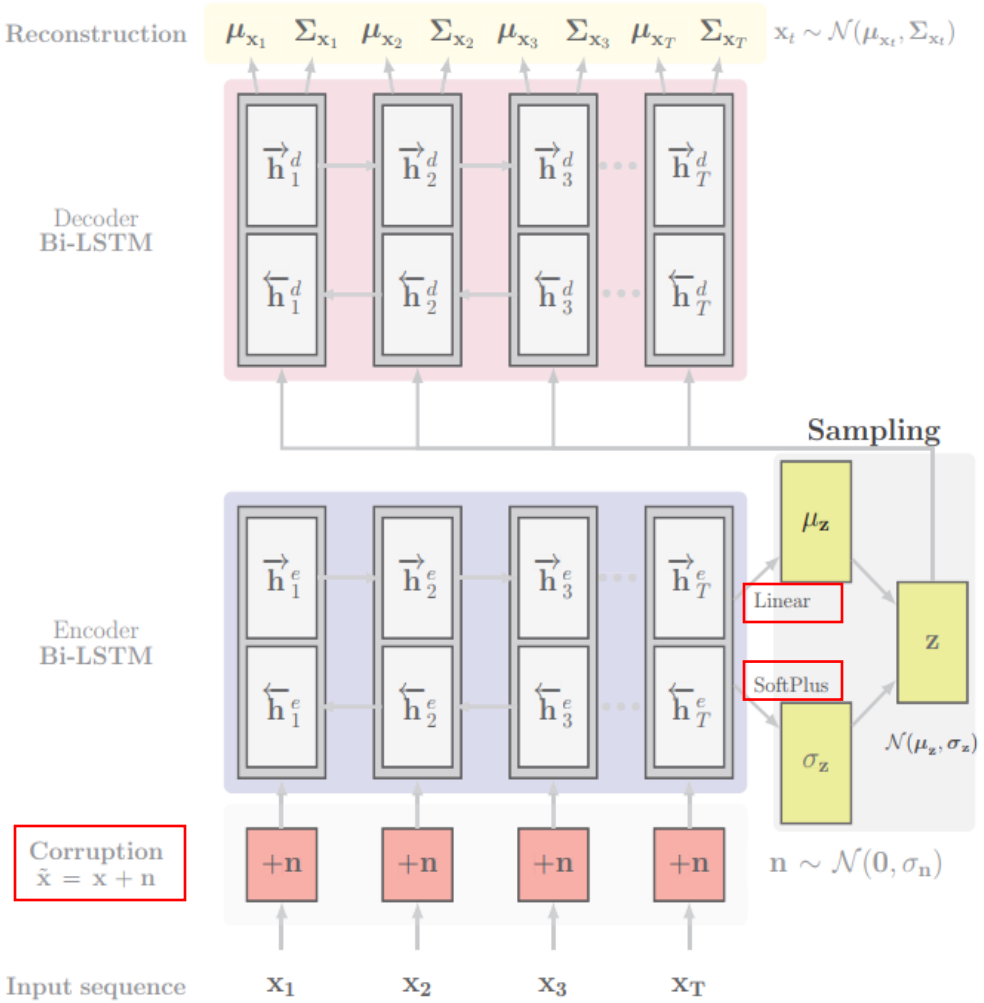
Motivation: Recently, unsupervised representation learning with deep generative models has been applied to find representations of data, without the need for big labelled datasets.

Contributions:

- Both representation learning and anomaly detection are fully unsupervised;
- Unsupervised representation learning of time series data through a Variational Recurrent Autoencoder;
- Latent space-based detection using Clustering and the Wasserstein distance.

Methodology

Representation Learning



- Variational Recurrent Autoencoder;
- The encoder and decoder are both Bi-LSTM structure;
- The traning objective is to minimize

$$\mathcal{L}(\theta, \phi; \mathbf{x}^{(n)}) = -\mathbb{E}_{\tilde{q}_\phi(\mathbf{z}^{(n)}|\mathbf{x}^{(n)})}\left[\log p_\theta(\mathbf{x}^{(n)}|\mathbf{z}^{(n)})\right] + \lambda_{\mathrm{KL}} \mathcal{D}_{\mathrm{KL}}\left(\tilde{q}_\phi(\mathbf{z}^{(n)}|\mathbf{x}^{(n)}) \| p_\theta(\mathbf{z}^{(n)})\right)$$

Methodology

Anomaly Detection

- *Clustering:* apply 3 clustering algorithms in the representation space: hierarchical clustering, spectral clustering, *k*-means++;

- *Wasserstein Distance:* compute the median *Wasserstein* distance between a test sample $\mathbf{z}^{\text{test}}$ and N_W other samples within the test set of latent representations

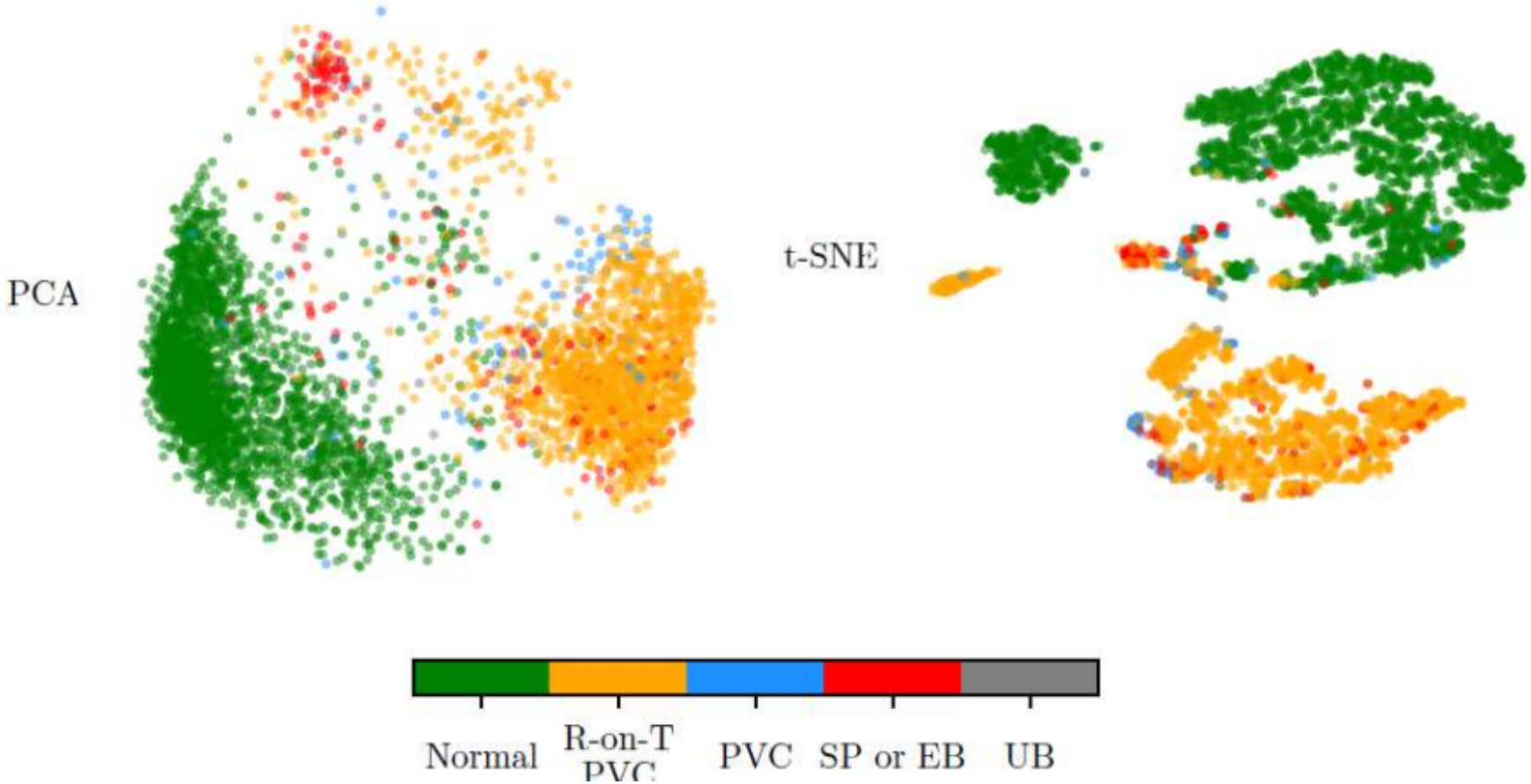
$$W(\mathbf{z}^{\text{test}}, \mathbf{z}^i)^2 = \|\boldsymbol{\mu}_{\mathbf{z}^{\text{test}}} - \boldsymbol{\mu}_{\mathbf{z}^i}\|_2^2 + \|\boldsymbol{\Sigma}_{\mathbf{z}^{\text{test}}}^{1/2} - \boldsymbol{\Sigma}_{\mathbf{z}^i}^{1/2}\|_F^2$$

$$\text{score}(\mathbf{z}^{\text{test}}) = \text{median}\{W(\mathbf{z}^{\text{test}}, \mathbf{z}^i)^2\}_{i=1}^{N_W}$$

- Both methods works under the assumption that most data are normal.

Results

Latent Space Analysis



Results

Anomaly Detection

Metric	Hierarchical	Spectral	*k*-Means	Wasserstein	SVM
AUC	0.9569	0.9591	0.9591	**0.9819**	0.9836
Accuracy	0.9554	0.9581	**0.9596**	0.9510	0.9843
Precision	**0.9585**	0.9470	0.9544	0.9469	0.9847
Recall	0.9463	0.9516	**0.9538**	0.9465	0.9843
F1-score	0.9465	0.9474	**0.9522**	0.9461	0.9844

Source	S/U[a]	Model	AUC	Acc	F_1
Ours	S	VRAE+SVM	**0.9836**	**0.9843**	**0.9844**
	U	VRAE+Clust/W	**0.9819**	**0.9596**	**0.9522**
Lei *et al.* [17]	S	SPIRAL-XGB	0.9100	–	–
Karim *et al.* [16]	S	F-t ALSTM-FCN	–	0.9496	–
Malhotra *et al.* [33]	S	SAE-C	–	0.9340	–
Liu *et al.* [34]	U	oFCMdd	–	–	0.8084

Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications

——WWW 2018

Background & Motivation

Background: To ensure undisrupted business, large Internet companies need to closely monitor various KPIs (*e.g.*, Page Views, number of online users, and number of orders) of its Web applications, to accurately detect anomalies and trigger timely troubleshooting/mitigation

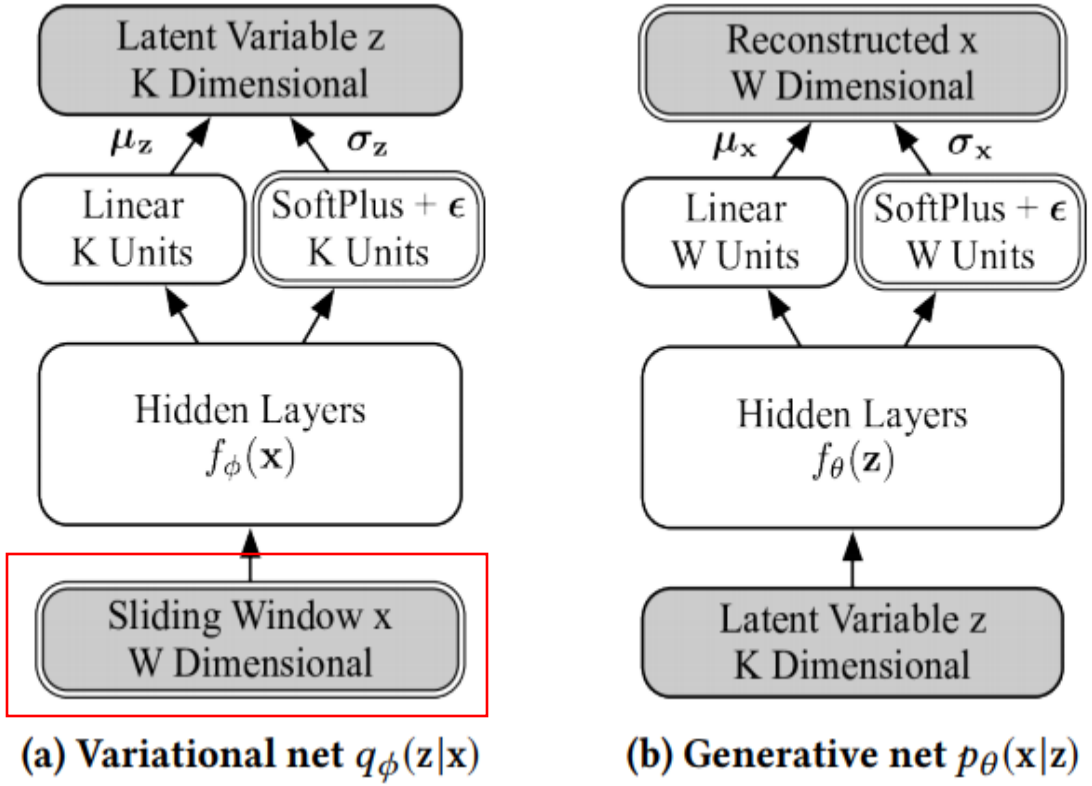
Motivation: However, anomaly detection for these seasonal KPIs with various patterns and data quality has been a great challenge, especially without labels.

Contributions:

- The three techniques in *Donut*, Modified ELBO and Missing Data Injection for training, and MCMC Imputation for detection;
- For the first time in the literature, we discover that adopting VAE (or generative models in general) for anomaly detection requires training on both normal data *and abnormal data*, contrary to common intuition;
- Propose a novel KDE interpretation in z-space for *Donut*, making it the first VAE-based anomaly detection algorithm with solid theoretical explanation.

Methodology

Donut Architecture



(a) Variational net $q_\phi(\mathbf{z}|\mathbf{x})$ **(b) Generative net $p_\theta(\mathbf{x}|\mathbf{z})$**

Methodology

Modified ELBO

- In order to exclude the contribution of anomalies and missing points;

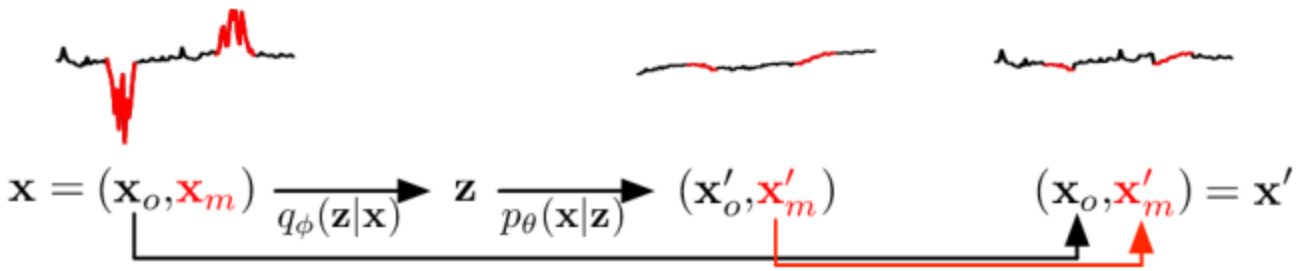
$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]$$

$$\widetilde{\mathcal{L}}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\sum_{w=1}^{W} \alpha_w \log p_\theta(x_w|\mathbf{z}) + \beta \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]$$

Methodology

MCMC Imputation for Detection

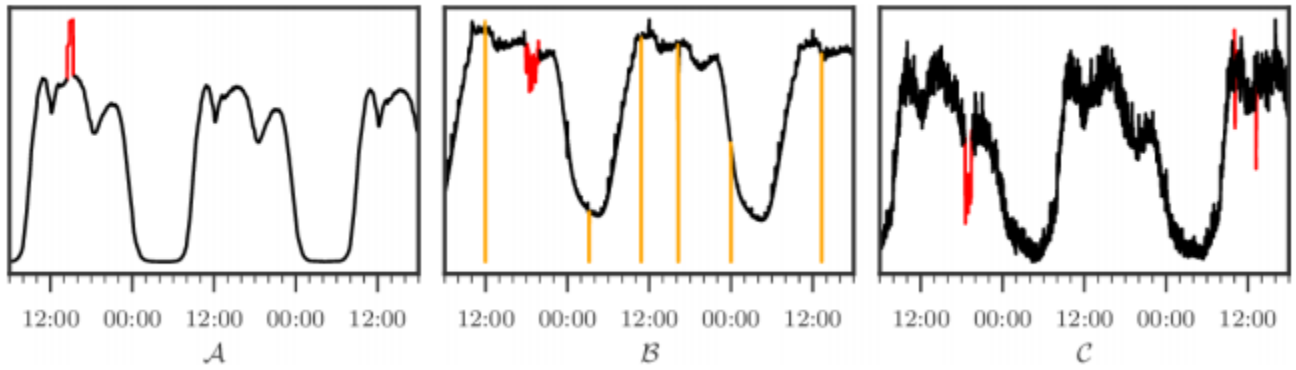
- The detection is to compute “reconstruction probability” $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]$
- To eliminate the biases introduced by missing points, we choose to adopt the MCMC-based missing data imputation.



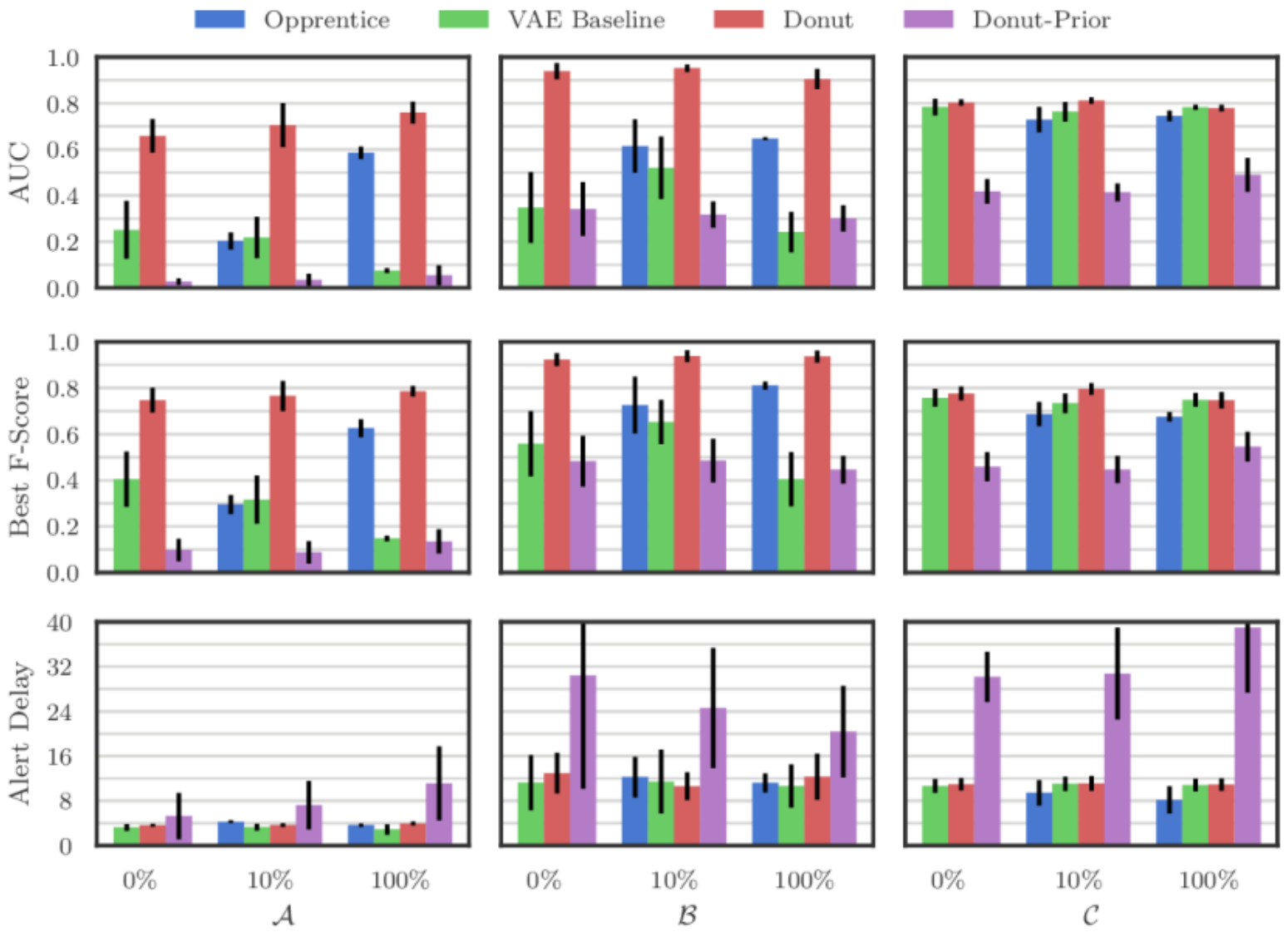
Datasets

Datasets Statistics

DataSet	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$
Total points	296460	317522	285120
Missing points	1222/0.41%	1117/0.35%	304/0.11%
Anomaly points	1213/0.41%	1883/0.59%	4394/1.54%
Total windows*	296341	317403	285001
Abnormal windows**	20460/6.90%	20747/6.54%	17288/6.07%



Results


Opprentice
VAE Baseline
Donut
Donut-Prior
AUC
Best F-Score
Alert Delay
1.0
0.8
0.6
0.4
0.2
0.0
40
32
24
16
8
0
0%
10%
100%
$\mathcal{A}$
$\mathcal{B}$
$\mathcal{C}$