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

Haowen Xu[1] Wenxiao Chen[1] Nengwen Zhao[1] Zeyan Li[1]
Jiahao Bu[1] Zhihan Li[1] Ying Liu[1] Youjian Zhao[1] Dan Pei[1]
Yang Feng[2] Jie Chen[2] Zhaogang Wang[2] Honglin Qiao[2]

[1]Tsinghua University

[2]Alibaba Group

April 25, 2018

Outline

Outline

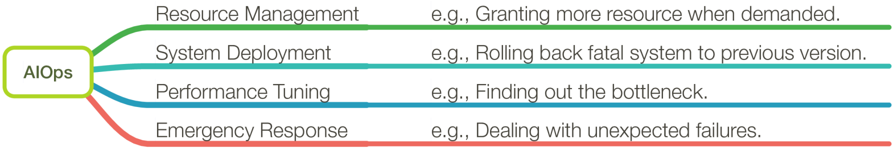
Infrastructures for the Information Era is Complicated

Figure: Data centers of AliCloud at 18 global locations. AliCloud ranks the third in the global public cloud market share.

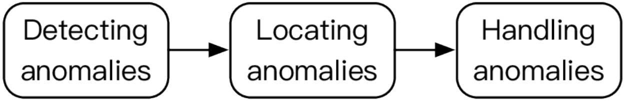
To maintain such huge and complicated system, **A**rtificial **I**ntelligence for IT **Op**eration**s** **(AIOps)** is greatly demanded.

Anomaly Detection is the First Step for Many AIOps Tasks

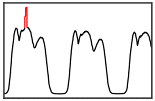
AIOps consists of many tasks.


AIOps
Resource Management
e.g., Granting more resource when demanded.
System Deployment
e.g., Rolling back fatal system to previous version.
Performance Tuning
e.g., Finding out the bottleneck.
Emergency Response
e.g., Dealing with unexpected failures.

For most tasks, **anomaly detection** is the first step to solve problems. In this work, we focus on detecting anomalies on **K**ey **P**erformance **I**ndicators **(KPIs)**, which are commonly used as system monitoring.



(a) Anomaly detection is the first step to solve problems.

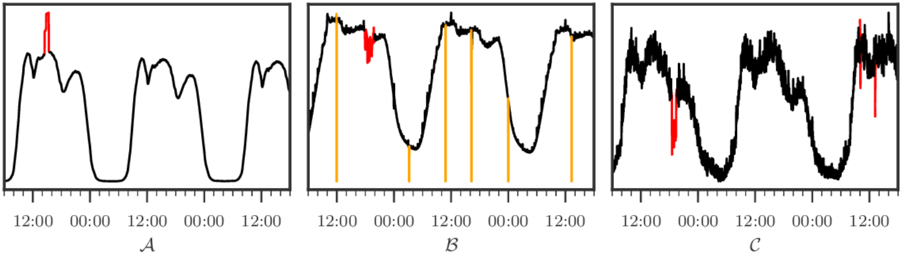


(b) KPI and anomalies.

Problem Scenario: Anomaly Detection for Seasonal KPIs

KPIs are time sequences, yet one of the most fundamental system monitoring indicators. A failure usually causes more or less anomalies on at least one KPI. Thus anomaly detection for KPIs are very useful in **A**rtificial **I**ntelligence for IT **Op**eration**s** **(AIOps)**.

For web applications, the user activities are usually seasonal, so are the KPIs, including high level KPIs like the trading volumes, and low level KPIs like the CPU consumptions. We thus focus on **anomaly detection for seasonal KPIs in this work**.



Problem Formulation: Detection of “Abnormal” Patterns

The problem of **anomaly detection** can be generally formulated as:

Anomaly Detection

Detecting whether or not “abnormal” patterns occur in the testing data.

Since KPIs are time sequences, and since in most real cases human operators are willing to see a detection output every time a new observation arrives, the anomaly detection for KPIs can be formulated as:

Anomaly Detection for KPIs

For each time t, given the on-time KPI observation x_t and historical observations $x_{t-W+1}, \dots, x_{t-1}$, determine whether an “abnormal” pattern has occurred (denoted by $y_t = 1$).

Detection algorithms are often designed to compute a **real-valued score** $s(y_t = 1)$ (“**anomaly score**” *hereafter*), *e.g.*, $p(y_t = 1 | x_{t-W+1}, \dots, x_t)$, leaving the final decision of triggering alerts to the operators.

Previous Work

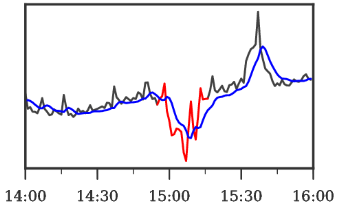
- Anomaly detectors:
  - Traditional statistical detectors are not discriminative enough. *e.g.*, Holt-Winters, TSD, etc.
  - Unsupervised learning based detectors not good enough in practice. *e.g.*, one-class SVM, clustering method, K-Means, GMM.
- Ensemble learning approaches:
  - Supervised ensemble learning demands too heavily on good labeling quality. *e.g.*, Opprentice (Liu et al., 2015).
  - Unsupervised ensemble learning too sensitive to hyper-parameters. *e.g.*, EGADS (Laptev et al., 2015).
- Deep generative models:
  - VRNN (Sölch et al., 2016), uses more advanced model than VAE, but is sensitive to hyper-parameters and extremely slow. Besides, it lacks **dimension reduction**, shown to be essential in our experiments.

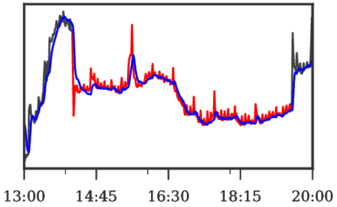
BTW, Opprentice uses **14 anomaly detectors** with **133 configurations of hyper-parameters**, and is claimed to have better performance than all its components, on datasets very similar to ours. We thus use it as a proxy to the traditional detectors in our evaluation.

Outline

The Essential of Stochasticity and Dimension Reduction

We carry out some **LSTM prediction experiments** at an early stage.



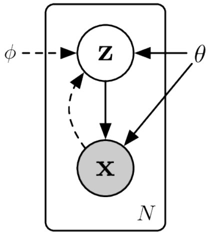


The LSTM prediction model turns out to be almost a "**lagged copy model**", hardly valuable for our task. We thus summarized **two essential factors** for anomaly detection via deep neural networks:

1. Stochasticity: The network should be able to capture the stochasticity of "noises" on the KPIs internally, to avoid over-fitting.
2. Dimension Reduction: Dimension reduction should be applied, to enforce the network focusing only on the normal patterns of the KPIs.

We finally choose the **variational auto-encoder (VAE)** to develop our method, according to the above conclusions.

Background of Variational Auto-Encoder



VAE models the relationship between two random variables, **latent variable** $\mathbf{z}$ and **visible variable** $\mathbf{x}$. The generative net of VAE consists of:

- $\mathbf{z} \sim p_\theta(\mathbf{z})$, a chosen prior.
- $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$, derived from a neural network with parameter θ, modeling the data $\mathbf{x}$.

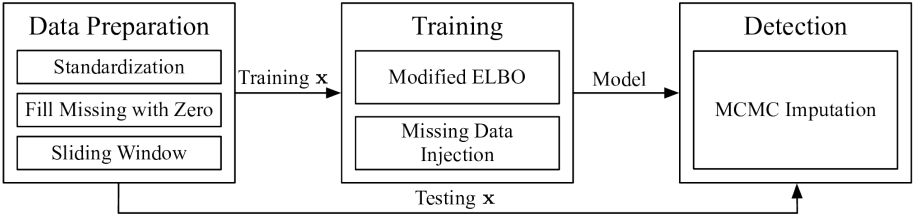
The variational net $q_\phi(\mathbf{z}|\mathbf{x})$ is adopted to approximate the **intractable** $p_\theta(\mathbf{z}|\mathbf{x})$. SGVB (Kingma and Welling, 2014) is often used to jointly train $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x},\mathbf{z})$, by maximizing the **evidence lower-bound (ELBO)** (1):

$$\log p(\mathbf{x}) \geq \mathcal{L}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \quad (1)$$

Monte Carlo integration can be used to approximate the expectation:

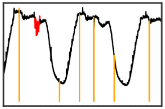
$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})] \approx \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{z}^{(l)}), \quad \mathbf{z}^{(l)} \sim q_\phi(\mathbf{z}|\mathbf{x})$$

Overall Architecture of *Donut*



Data Preparation

- Fill Missing with Zero:

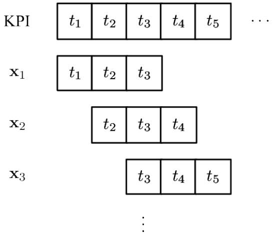


"Missing" are special anomalies, *always* known beforehand. We fill missing points with zeros (orange points in the left figure), and let our model to handle them afterwards.

- Standardization: $\hat{x}_t = (x_t - \mu_x)/\sigma_x$.

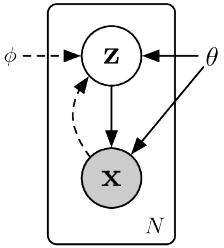
x_t are the original KPI values, μ_x and σ_x are the mean and std of x_t. We shall use x_t to denote $\hat{x}_t$ and neglect the original values x_t *hereafter*.

- Sliding Window:

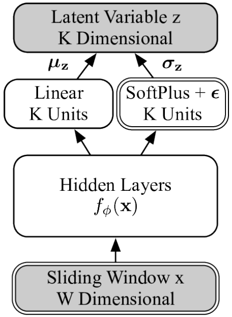


We split the KPIs into fixed-length **sliding windows $\mathbf{x}_t$**, which are assumed to be *i.i.d.*, and are used as **the input $\mathbf{x}$ of VAE** at every time t. For simplicity, we shall omit the subscript t, using $\mathbf{x}$ to denote **the window of "current time"**, and $x_1, \dots, x_W$ to denote **each point in $\mathbf{x}$** afterwards.

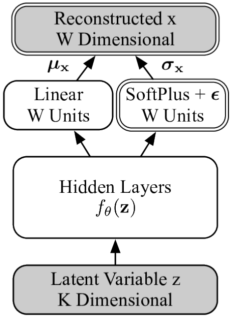
Network Structure



(a) VAE General Structure



(b) $q_\phi(\mathbf{z}|\mathbf{x})$ of *Donut*



(c) $p_\theta(\mathbf{x}|\mathbf{z})$ of *Donut*

- Variational net: $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}}, {\boldsymbol{\sigma}_{\mathbf{z}}}^2\mathbf{I})$.
- Generative net: $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, {\boldsymbol{\sigma}_{\mathbf{x}}}^2\mathbf{I})$.
- SoftPlus Trick: $\boldsymbol{\sigma}_{\mathbf{z}} = \mathrm{SoftPlus}[\mathbf{W}_{\boldsymbol{\sigma}_{\mathbf{z}}}^\top f_\phi(\mathbf{x}) + \mathbf{b}_{\boldsymbol{\sigma}_{\mathbf{z}}}] + \boldsymbol{\epsilon}$, $\mathrm{SoftPlus}[a] = \log[\exp(a) + 1]$. Similar for $\boldsymbol{\sigma}_{\mathbf{x}}$.

Dealing with Missing and Anomaly: Training

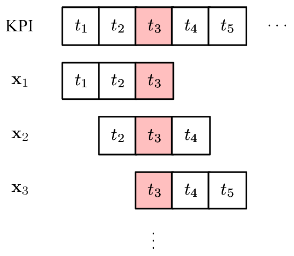


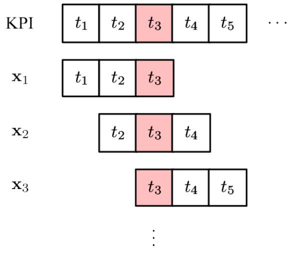
Figure: The anomaly at t_3 shall affect t_4 and t_5, potentially causing trouble in training and detection.

We developed two techniques to handle such "historical anomalies" in training:

1. M-ELBO: We modify the ELBO (1) of VAE into **M-ELBO $\widetilde{\mathcal{L}}(\mathbf{x})$** (2). $\alpha_w = 1$ indicates x_w being "normal", $\alpha_w = 0$ otherwise. $\beta = (\sum_w \alpha_w)/W$. We do not exclude "abnormal" windows from training data, thus the **M-ELBO effectively trains VAE to recover "normal" points in $\mathbf{x}$ even if some "abnormal" points exist**.
2. Missing Data Injection: To further amplify the effect of M-ELBO, we randomly set 1% points to be missing at **every epoch**.

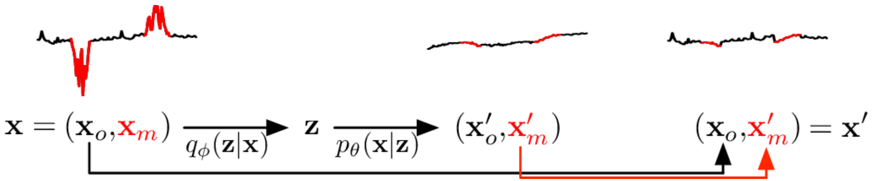
$$\widetilde{\mathcal{L}}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\sum_{w=1}^{W} \alpha_w \log p_\theta(x_w|\mathbf{z}) + \beta \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \quad (2)$$

Dealing with Missing and Anomaly: Detection



In detection, we adopted MCMC imputation to impute the already known **missing points**, which is proposed by Rezende et al. (2014), using a trained deep generative model to iteratively approach the marginal distribution $p(\mathbf{x}_{\text{missing}}|\mathbf{x}_{\text{observed}})$.

The **anomalies** other than missing points are to be detected, thus MCMC cannot be applied. We rely on the effect of **dimension reduction** and **M-ELBO** to resist such points.



$\mathbf{x} = (\mathbf{x}_o, \mathbf{x}_m) \xrightarrow[q_\phi(\mathbf{z}|\mathbf{x})]{} \mathbf{z} \xrightarrow[p_\theta(\mathbf{x}|\mathbf{z})]{} (\mathbf{x}'_o, \mathbf{x}'_m)$
$(\mathbf{x}_o, \mathbf{x}'_m) = \mathbf{x}'$

Figure: Illustration of one iteration in MCMC imputation.

The Anomaly Score

An and Cho (2015) has already adopted VAE in anomaly detection tasks of other domain[1]. They use the **reconstruction probability** (3) of truly *i.i.d.* samples $\mathbf{x}$ (*e.g.*, image pixel vectors) as the anomaly score:

$$s(y=1) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x}|\mathbf{z}^{(l)}),\ \mathbf{z}^{(l)} \sim q_\phi(\mathbf{z}|\mathbf{x}) \tag{3}$$

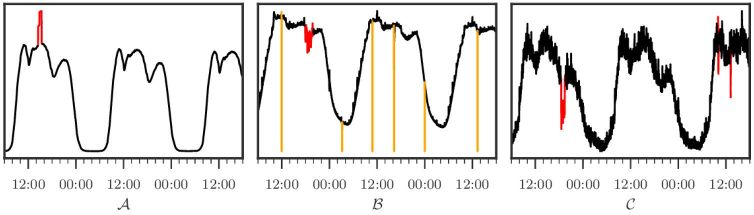
Since the KPIs are time sequences, and the operators are willing to see on-time detection outputs each time a new point arrives, we compute the **element-wise reconstruction probability** (4) for the last point x_W in $\mathbf{x}$, as the anomaly score for the time being:

$$s(y_W=1) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(x_W|\mathbf{z})] \approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x_W|\mathbf{z}^{(l)}),\ \mathbf{z}^{(l)} \sim q_\phi(\mathbf{z}|\mathbf{x}) \tag{4}$$

[1] An and Cho (2015) uses vanilla VAE, without developing techniques like ours to improve performance. We shall compare *Donut* againt their vanilla VAE in evaluation.

Outline

Datasets for Evaluation



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%

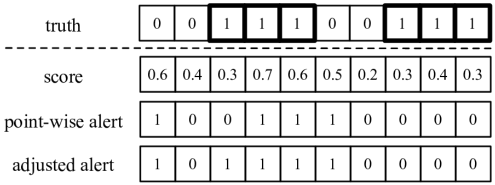
We obtain 18 well-maintained KPIs from Alibaba Group, and choose 3 of them, denoted as $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, with relatively small, medium and large noises among the 18 datasets, so we can **evaluate Donut for noises at different levels**.

Performance Metrics

A **threshold** is needed to turn anomaly scores into **alerts**. In practice, the total number of alerts are often controlled by adjusting the thresholds, we thus report the following metrics, as evaluation of the overall performance of a model:

1. AUC: The **average precision over recalls**, given all possible thresholds.
2. Best F-score: The **largest F-score** (F-score is the harmonic mean of precision and recall), given all possible thresholds.

In real applications, it is acceptable for an algorithm to **trigger an alert for any point in a contiguous anomaly segment**, if the delay is not too long. We thus modify the two metrics to reflect such preference:



truth	0	0	1	1	1	0	0	1	1	1
score	0.6	0.4	0.3	0.7	0.6	0.5	0.2	0.3	0.4	0.3
point-wise alert	1	0	0	1	1	1	0	0	0	0
adjusted alert	1	0	1	1	1	1	0	0	0	0

1st row: ground truth; 2nd row: detection scores; 3rd row: point-wise alerts outputs with a threshold of 0.5; 4th row: adjusted point-wise alerts. We use the adjusted alerts to compute the metrics.

Experiment Setup

Major hyper-parameters[2] of *Donut*

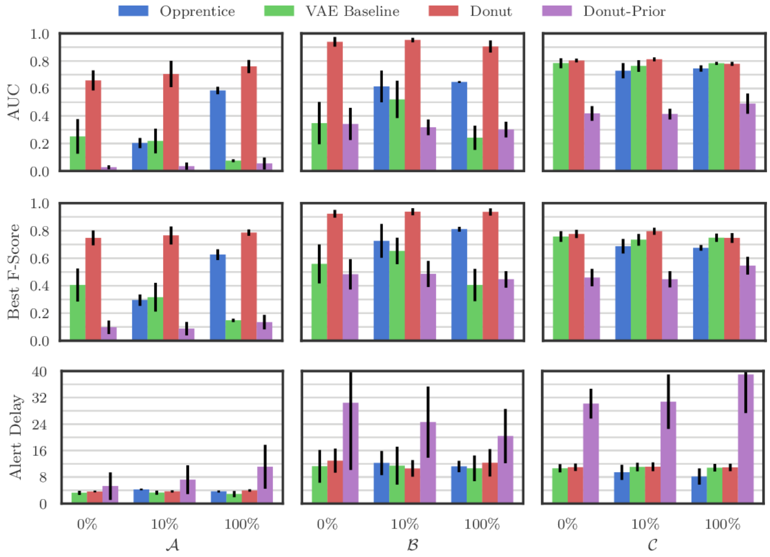
Parameter	Choice	Description
W	120	Window size of $\mathbf{x}$.
K	8 ($\mathcal{A}$), 3 ($\mathcal{B}$ and $\mathcal{C}$)	Number of $\mathbf{z}$ dimensions.
$f_\phi(\mathbf{x})$	2 Layers of 100 ReLU	Hidden layers of $q_\phi(\mathbf{z}\|\mathbf{x})$.
$f_\theta(\mathbf{z})$	2 Layers of 100 ReLU	Hidden layers of $p_\theta(\mathbf{x}\|\mathbf{z})$.
ϵ	10^{-4}	Minimum output value for the std layers.
λ	0.01	Missing data injection ratio.
M	10	MCMC iteration count.
L	1024	Monte Carlo integration sampling number.

Algorithms for comparison

1. Opprentice (Liu et al., 2015): supervised ensemble learning, based on 14 traditional detectors with 133 hyper-parameter configurations.
2. VAE Baseline: vanilla VAE for anomaly detection on *i.i.d.* samples (An and Cho, 2015). Hyper-parameters are chosen to be identical with *Donut*.
3. *Donut*-Prior: identical with *Donut*, expect for using $\mathbb{E}_{p_\theta(\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})]$ instead of the reconstruction probability $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]$.

[2] The rest can be found in the paper.

Overall Performance



Effects of *Donut* Techniques

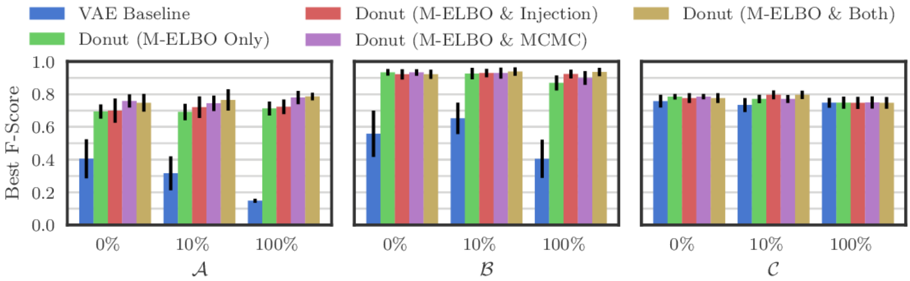


Figure: Best F-score of (1) VAE Baseline, (2) *Donut* with M-ELBO, (3) M-ELBO + missing data injection, (4) M-ELBO + MCMC, and (5) M-ELBO + both MCMC and injection.

The **M-ELBO** alone contributes most of the improvement over VAE Baseline, while the **missing data injection** and the **MCMC imputation** can further benefit the performance.

Impact of Z Dimension Number K

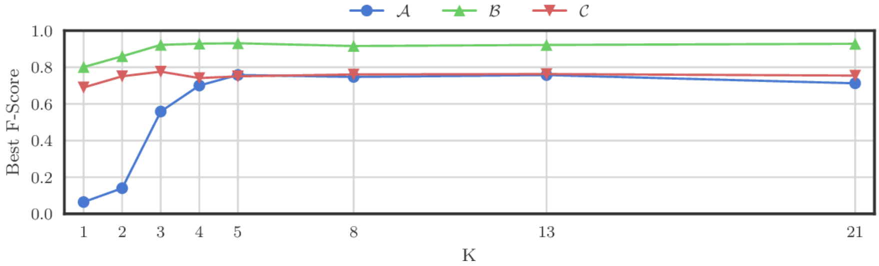


Figure: The best F-score of unsupervised *Donut* with different K on testing set.

- The essential of **Dimension reduction**: W (the dimension of $\mathbf{x}$) is 120, while the best K (the dimension of $\mathbf{z}$) is no larger than 10.
- It should be quite easy to empirically choose K.
  1. The best performance could be achieved with fairly small K.
  2. The performance does not drop too heavily for K up to 21.
- Smoother KPIs seem to demand larger K.

Outline

Reconstruction Probability isn't a Well-Defined Probability

The reconstruction probability, defined by:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] = \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{z}$$

is quite absurd, since we can easily notice that, the following equation is not well-defined under the probability framework:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[p_\theta(\mathbf{x}|\mathbf{z})] = \int q_\phi(\mathbf{z}|\mathbf{x}) \, p_\theta(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{z}$$

An and Cho (2015) just uses the reconstruction probability as the anomaly score, without solid theoretical explanation. Meanwhile, the prior counterpart, *i.e.*, $\mathbb{E}_{p_\theta(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})]$, which is more reasonable under the probability framework, since $p(\mathbf{x}) = \mathbb{E}_{p_\theta(\mathbf{z})}[p(\mathbf{x}|\mathbf{z})]$ is well-defined, but actually shows **much worse performance** in evaluation.

The Time Gradient

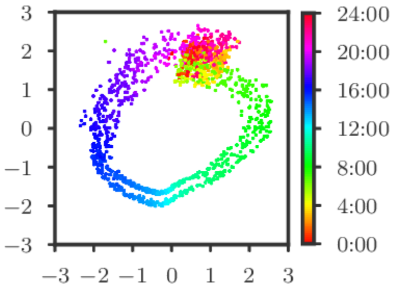
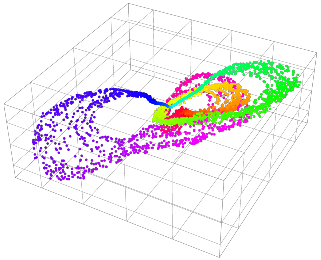
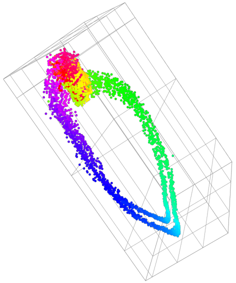


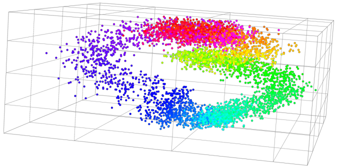
Figure: The $\mathbf{z}$ layout of dataset $\mathcal{B}$. Figure is plotted by sampling $\mathbf{z}$ from $q_\phi(\mathbf{z}|\mathbf{x})$, corresponding to normal $\mathbf{x}$ randomly chosen from the testing set. K is chosen as 2, so the x- and y-axis are the two dimensions of $\mathbf{z}$ samples. The color of a $\mathbf{z}$ sample denotes its time of the day.

- Time gradient: $q_\phi(\mathbf{z}|\mathbf{x})$ are organized in smooth transition: $\mathbf{x}$ at contiguous time are mapped to nearby $q_\phi(\mathbf{z}|\mathbf{x})$.
- Contiguous $\mathbf{x}$ are highly similar in the KPIs of our interest, since they are smooth in general.
- Transition of $q_\phi(\mathbf{z}|\mathbf{x})$ in the shape of $\mathbf{x}$ rather than time is the cause of time gradient, since *Donut* consumes no time information.
- *Donut* encodes the "shape" or "normal patterns" of $\mathbf{x}$ by $\mathbf{z}$, as shown by the time gradient.
- The time gradient can benefit generalization.

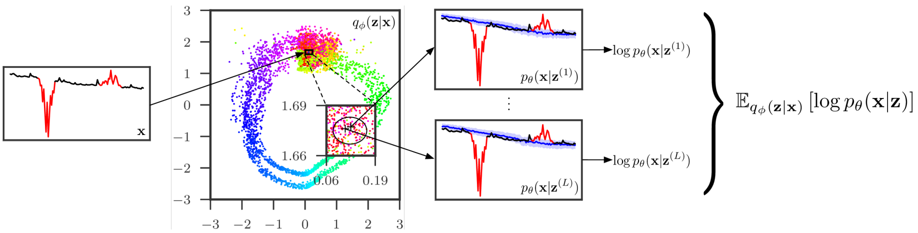
Layouts of $\mathbf{z}$ for All Three Datasets when $K = 3$


$\mathcal{A}$


$\mathcal{B}$


$\mathcal{C}$

The KDE Interpretation



$\mathbf{x}$
$q_\phi(\mathbf{z}|\mathbf{x})$
1.69
1.66
0.06
0.19
$p_\theta(\mathbf{x}|\mathbf{z}^{(1)})$
$\log p_\theta(\mathbf{x}|\mathbf{z}^{(1)})$
$p_\theta(\mathbf{x}|\mathbf{z}^{(L)})$
$\log p_\theta(\mathbf{x}|\mathbf{z}^{(L)})$
$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]$

Figure: Illustration of the KDE interpretation. **For a given x potentially with anomalies, Donut tries to recognize what normal pattern it follows, encoded as $q_\phi(\mathrm{z}|\mathrm{x})$**. The black ellipse in the middle figure denotes the 3-$\boldsymbol{\sigma}_\mathbf{z}$ region of $q_\phi(\mathbf{z}|\mathbf{x})$. **$L$ samples of z are then taken from $q_\phi(\mathrm{z}|\mathrm{x})$**, denoted as the crosses in the middle figure. **Each z is associated with a density estimator kernel $\log p_\theta(\mathrm{x}|\mathrm{z})$**. The blue curves in the right two figures are $\boldsymbol{\mu}_\mathbf{x}$ of each kernel, while the surrounding stripes are $\boldsymbol{\sigma}_\mathbf{x}$. Finally, the **values of $\log p_\theta(\mathrm{x}|\mathrm{z})$ are computed from each kernel, and further averaged together as the reconstruction probability**.

Find Good Posteriors for Abnormal $\mathbf{x}$

All the following techniques work by improving the ability of *Donut* to find "good" posteriors[3] for abnormal $\mathbf{x}$:

- Dimension Reduction: Force *Donut* to focus only on normal patterns.
- M-ELBO: Explicitly trains *Donut* to recover normal points even when abnormal points exist in $\mathbf{x}$.
- Missing Data Injection: Amplifies the effect of *M-ELBO*.
- MCMC Imputation: Alleviate the biases brought by missing points, helping *Donut* to find good posteriors.

[3] "good" implies beneficial for the anomaly detection task.

Visualization of MCMC Imputation

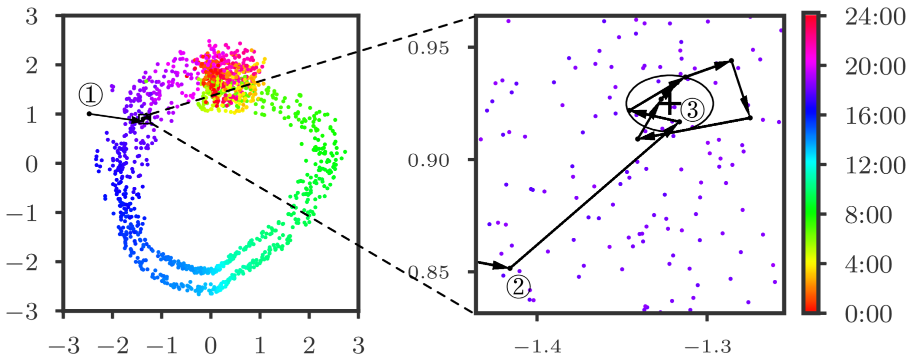


Figure: MCMC visualization. A normal $\mathbf{x}$ is chosen, whose posterior $q_\phi(\mathbf{z}|\mathbf{x})$ is plotted at right: the cross denotes $\boldsymbol{\mu}_\mathbf{z}$ and the ellipse denotes its 3-$\boldsymbol{\sigma}_\mathbf{z}$ region. We randomly set 15% $\mathbf{x}$ points as missing, to obtain the abnormal $\mathbf{x}'$. We run MCMC over $\mathbf{x}'$ with 10 iterations. At first, the $\mathbf{z}$ sample is far from $q_\phi(\mathbf{z}|\mathbf{x})$. After that, $\mathbf{z}$ samples quickly approach $q_\phi(\mathbf{z}|\mathbf{x})$, and begin to move around $q_\phi(\mathbf{z}|\mathbf{x})$ after only 3 iterations.

Revisit of the Causes of Time Gradient

$$\begin{aligned}\mathcal{L}(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \\ &= \mathbb{E}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] + \mathbb{E}\left[\log p_\theta(\mathbf{z})\right] + \mathrm{H}\left[\mathbf{z}|\mathbf{x}\right]\end{aligned}$$

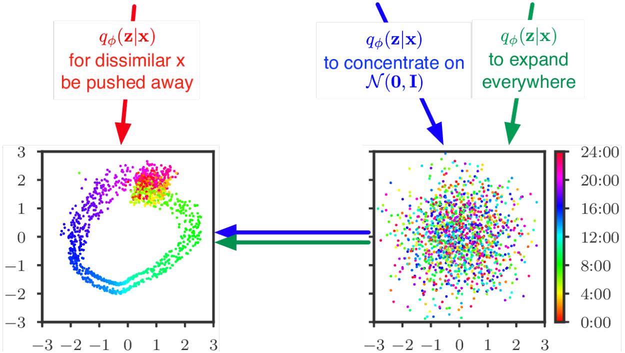


Figure: Causes of the time gradient. Surprisingly, we find no term in ELBO directly pulling $q_\phi(\mathbf{z}|\mathbf{x})$ for similar $\mathbf{x}$ together. The time gradient is likely to be caused mainly by **expansion** ($H(\mathbf{z}|\mathbf{x})$), **squeezing** ($\mathbb{E}[\log p_\theta(\mathbf{z})]$), **pushing** ($\mathbb{E}[\log p_\theta(\mathbf{x}|\mathbf{z})]$), and the **training dynamics** (random initialization and SGVB).

Sub-Optimal Equilibrium

The training dynamics may cause sub-optimal equilibrium. Having larger K (number of $\mathbf{z}$ dimensions) might help to avoid such problems.

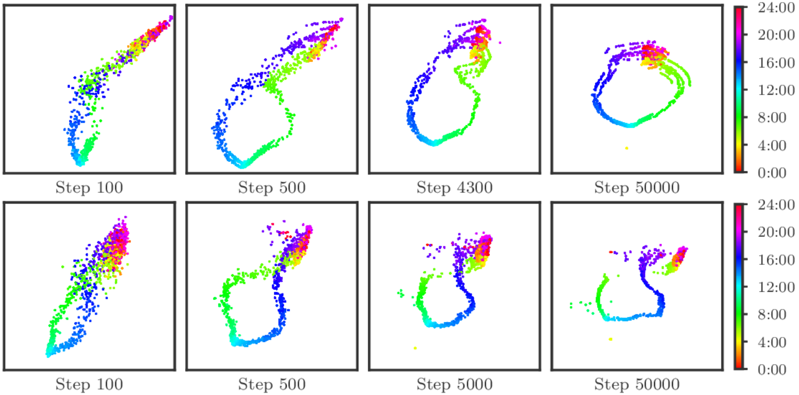


Figure: Evolution of $q_\phi(\mathbf{z}|\mathbf{x})$ of dataset $\mathcal{B}$ during training. Above: a successful training (final F-score 0.871). Below: a pathological training, converges to a sub-optimal equilibrium (final F-score 0.826).

Outline

Conclusion

Our unsupervised anomaly detection algorithm *Donut* for seasonal KPIs, based on VAE, greatly outperforms state-of-art supervised and vanilla VAE anomaly detection algorithms. The best F-scores range from 0.75 to 0.90 for the studied KPIs. The key factors of *Donut* to be successful are:

- Dimension Reduction: forces *Donut* to focus on the overall shape of normal patterns, and gain the ability of resisting abnormal points.
- M-ELBO, Missing Data Injection and MCMC Imputation: further improves *Donut*'s ability to resist abnormal points.

Furthermore, we made the **KDE Interpretation**, which provides a new perspective of VAE-based KPI anomaly detection. All of the above factors can be verified by such interpretation. The KDE Interpretation potentially has more theoretical value in the further development of deep generative models for KPI anomaly detection.

Donut source code published at: `https://github.com/korepwx/donut`.

Thank you

Q & A

References I

An, J. and Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. Technical report, SNU Data Mining Center.

Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *Proceedings of the International Conference on Learning Representations*.

Laptev, N., Amizadeh, S., and Flint, I. (2015). Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1939–1947. ACM.

Liu, D., Zhao, Y., Xu, H., Sun, Y., Pei, D., Luo, J., Jing, X., and Feng, M. (2015). Opprentice: Towards practical and automatic anomaly detection through machine learning. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, IMC '15, pages 211–224, New York, NY, USA. ACM.

References II

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1278–II–1286, Beijing, China. JMLR.org.

Sölch, M., Bayer, J., Ludersdorfer, M., and van der Smagt, P. (2016). Variational inference for on-line anomaly detection in high-dimensional time series. *International Conference on Machine Laerning Anomaly detection Workshop*.