Robust and Unsupervised KPI Anomaly Detection Based on Conditional Variational Autoencoder

Zeyan Li, Wenxiao Chen, Dan Pei

Department of Computer Science and Technology
Tsinghua University

November 18, 2018

Table of Contents

Problem Formulation (1/4)

KPI: key performance indicator, *e.g.*, pages views, search response time, number of transactions per minute.

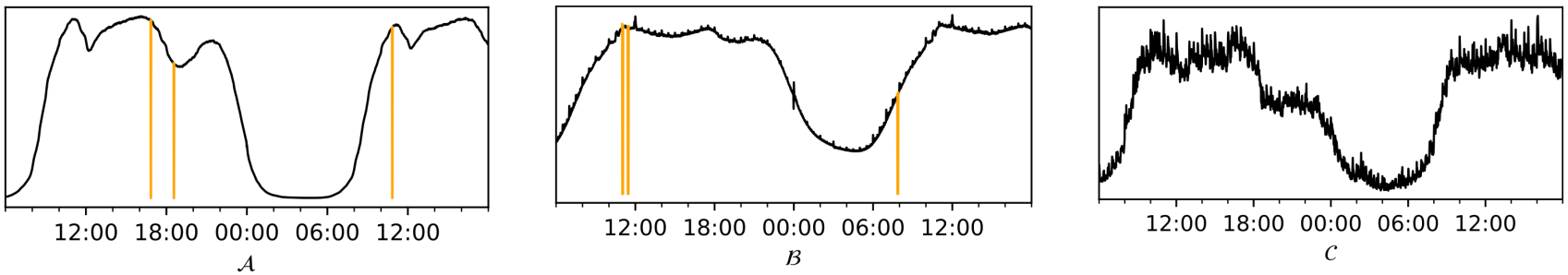


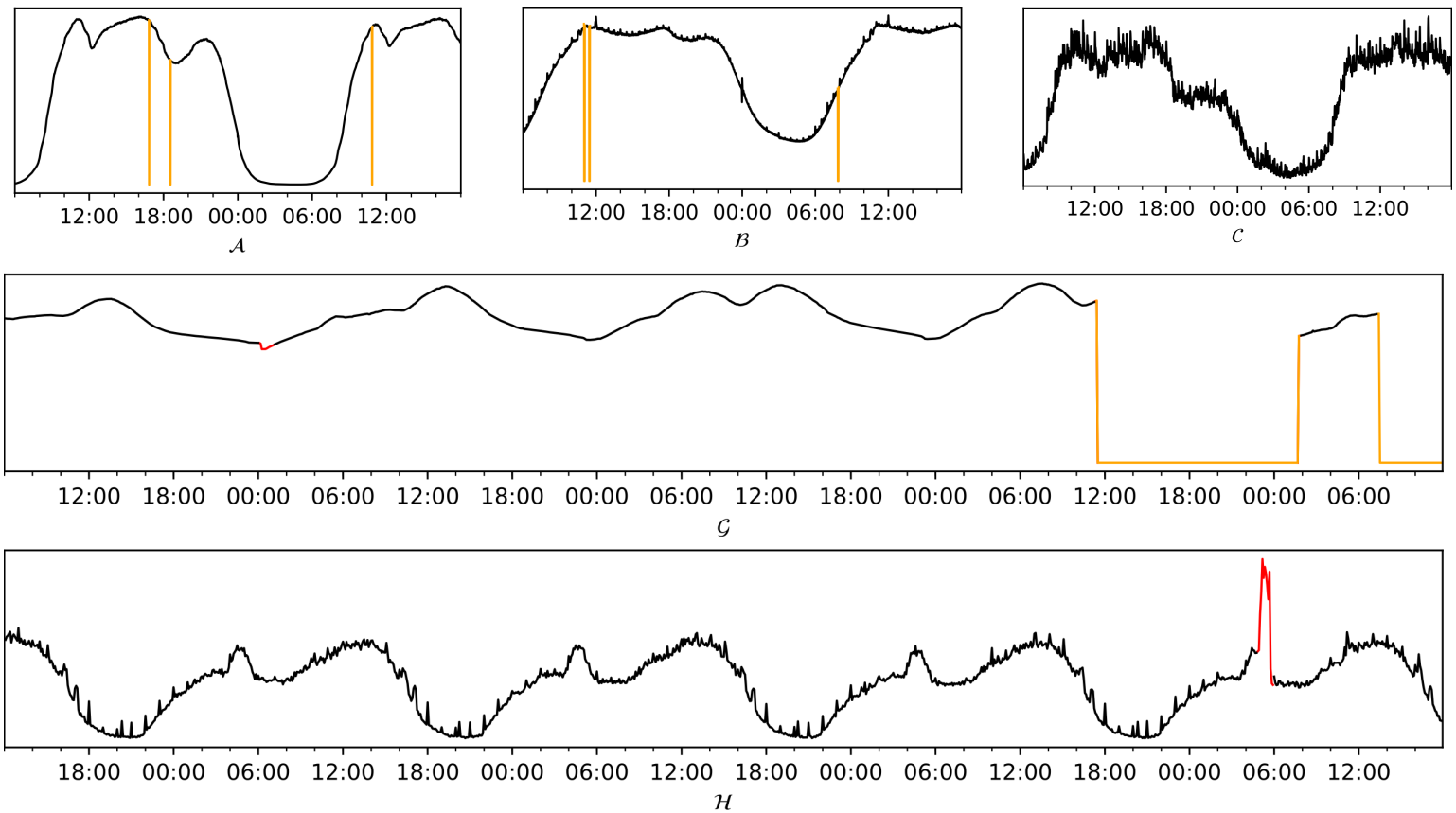
Figure: KPI examples.

To ensure undisrupted web-based services, operators need to closely monitor various KPIs, detect anomalies in them, and trigger timely troubleshooting or mitigation.

In our work, we focus on **business-related KPIs**. These KPIs consist of two parts:

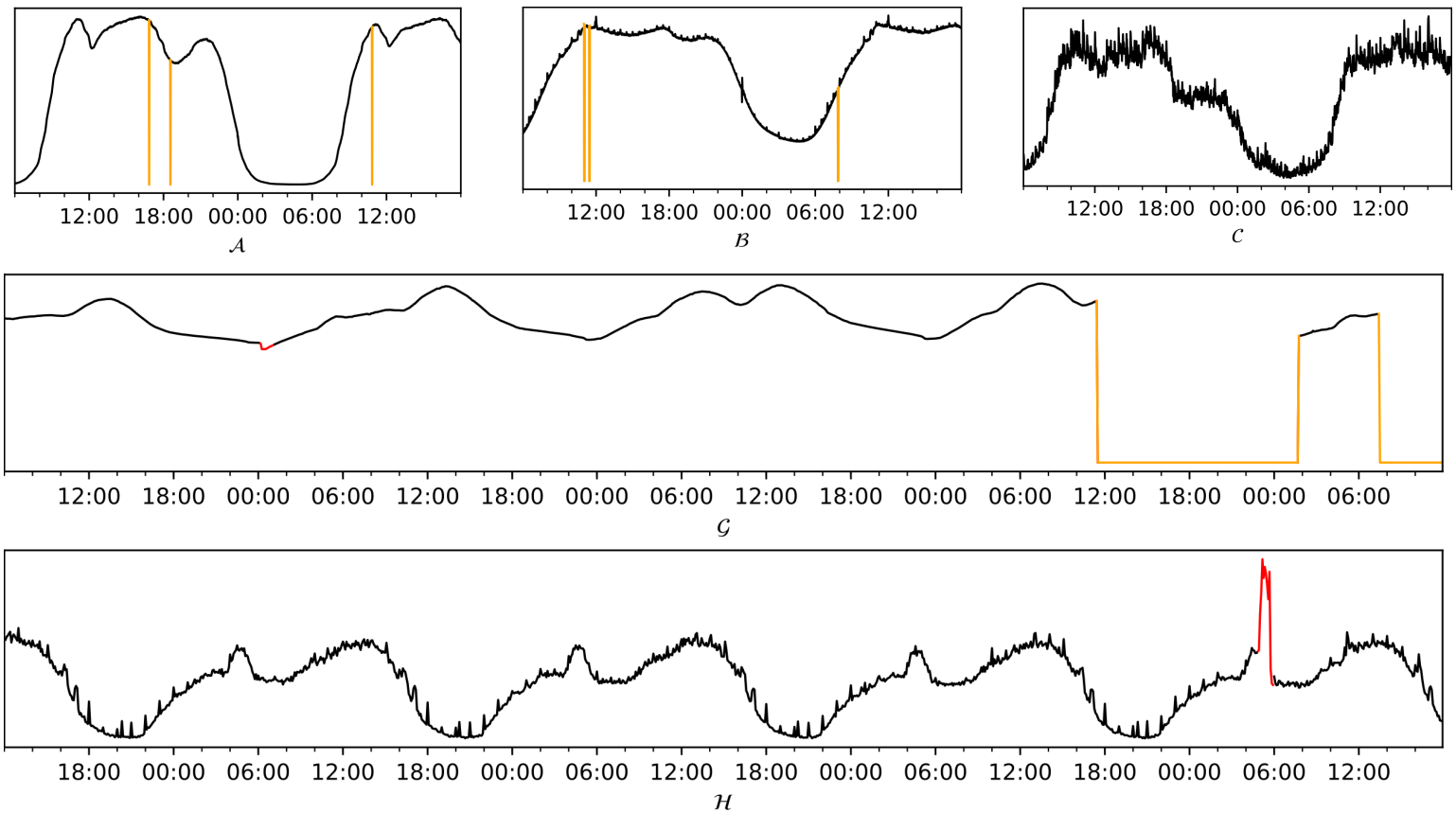
Problem Formulation (2/4)

1. *Seasonal* patterns. Business-related KPIs have it because of the influence from user behavior and schedule



Problem Formulation (3/4)

2. Noises. We assume that the noises follow independent, zero-mean Gaussian distribution.



Problem Formulation (4/4)

- Anomalies: points that do not follow normal patterns.
- Abnormal points: missing points and anomalies.

Sometimes the KPI values are not collected. These data points are called missing points. Missing points are also some kind of anomalies, but it is easy to distinguish them from normal points.

KPI anomaly detection formulation

for any time t, given historical KPI observations $v_{t-W+1:t}$ with length W, determine whether anomaly happens at time t (denoted by $\gamma_t = 1$).

Previous Works (1/1)

Table: Comparison among anomaly detection methodologies

Suffers from	1	2	3	4	5	*Bagel*
Selecting algorithm	Yes	No	Some	No	No	No
Tuning parameters	Yes	No	Some	Some	Some	No
Relying on labels	No	Yes	No	No	No	No
Poor Capacity	Yes	No	Some	No	No	No
Hard to train	No	No	Some	Some	Some	No
Time consuming	Some	Yes	Some	No	No	No

1: traditional statistical method, e.g., time series decomposition [1]
2: supervised ensemble method, e.g., Opprentice [2]
3: traditional unsupervised method, e.g., one-class SVM [3]
4: sequential deep generative model, e.g., VRNN [4]
5: non-sequential deep generative model, e.g. VAE [5], *Donut* [6]

Donut

Donut (Xu et.al. WWW 2018) is a state-of-art unsupervised anomaly detection algorithm for KPI. It is based on variational autoencoder (VAE). They also proposed a theoretical interpretation for *Donut*.

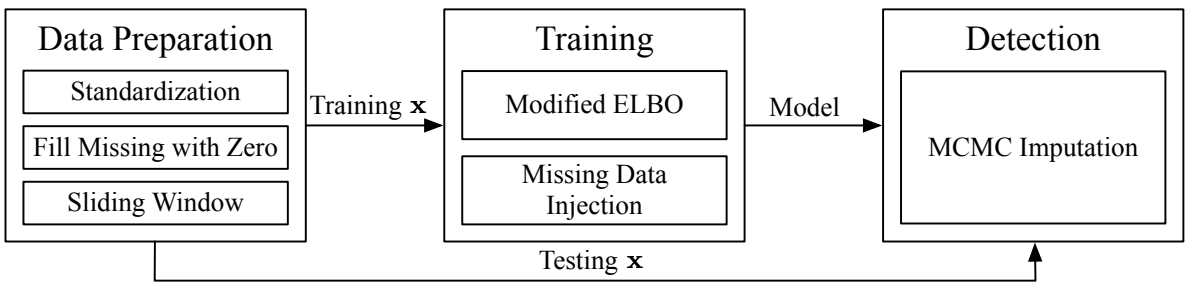


Figure: Overall architecture of *Donut*.

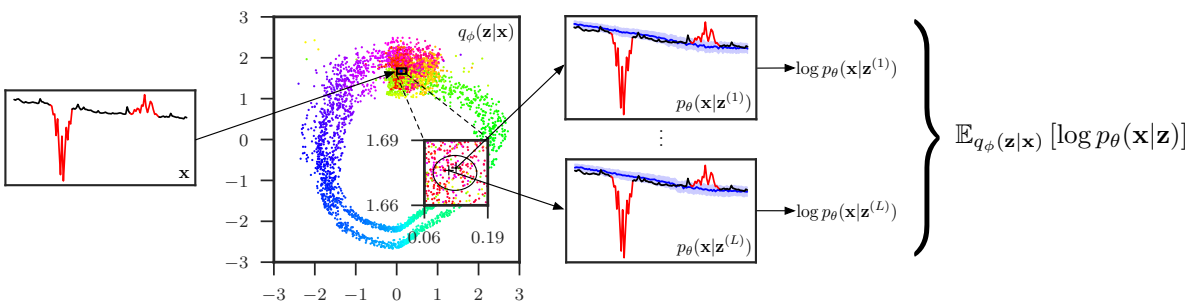


Figure: KDE interpretation for *Donut*.

Drawbacks of *Donut* (1/4)

Donut uses sliding windows, so the time information of a window is totally ignored. It may cause some problems.

For example, patterns occurs frequently may not be normal pattern when considering time.

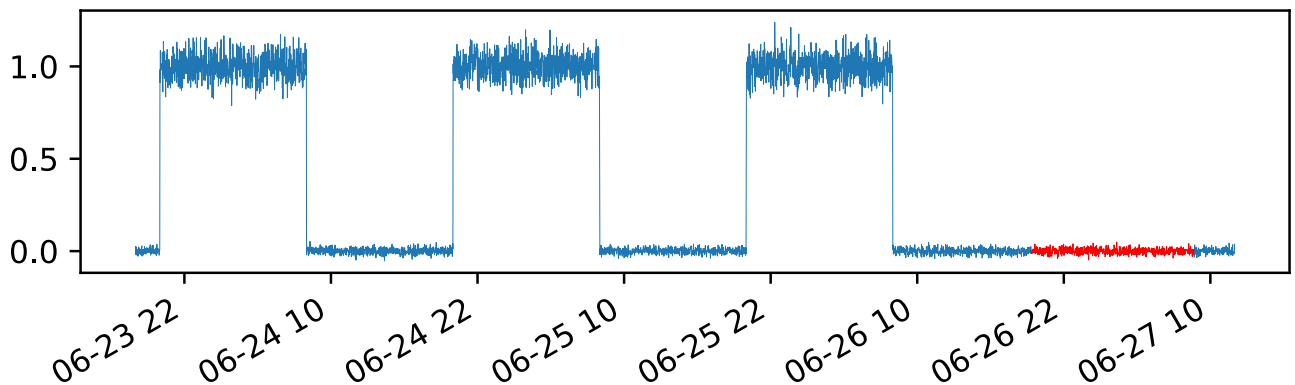


Figure: The KPI value should be around 1 in every night, so the red part is abnormal.

Drawbacks of *Donut* (2/4)

Then we found more problems in real data.

Donut

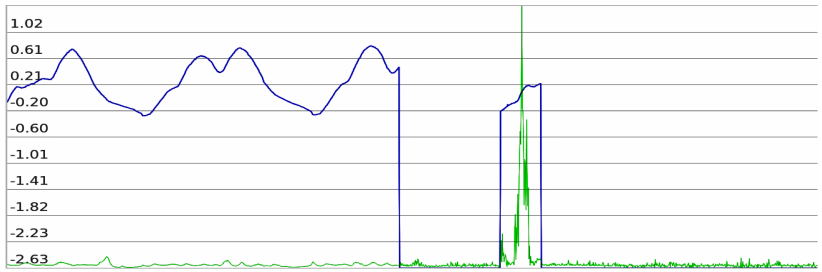


Figure: Anomaly scores of $\mathcal{G}$ given by *Donut*. The blue lines are KPI values. The green lines are the anomaly scores for each point. *Donut* gives too high anomaly scores for the normal fragment surrounded by missing points.

The small normal pieces surrounded by missing fragments is hard to reconstruct for *Donut*, because too many points are missing and *Donut* does not have enough information to reconstruct the normal pattern.

Drawbacks of *Donut* (3/4)

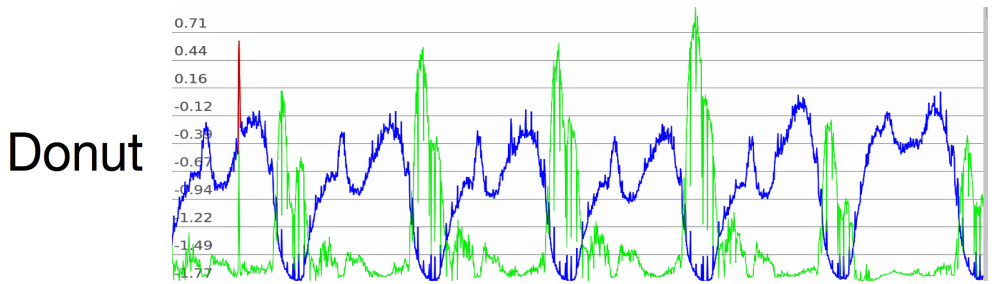


Figure: *Donut* gives too high anomaly scores at many normal valleys, which are mostly smooth but have many periodic spikes.

Since $\mathcal{H}$ is very smooth at most points, the $\mathbf{x}$'s standard deviation will be quite small (nearly zero). Small bias may also cause big impact on likelihood since the standard deviation is too small on a mostly smooth KPI.

Drawbacks of *Donut* (4/4)

Summary:

1. The correct **normal pattern** can not be determined only by a KPI window.
2. Model may be confused because of the abnormal points or noises.
3. The biases brought by noises in KPI can be amplified in the final anomaly detector, likelihood.

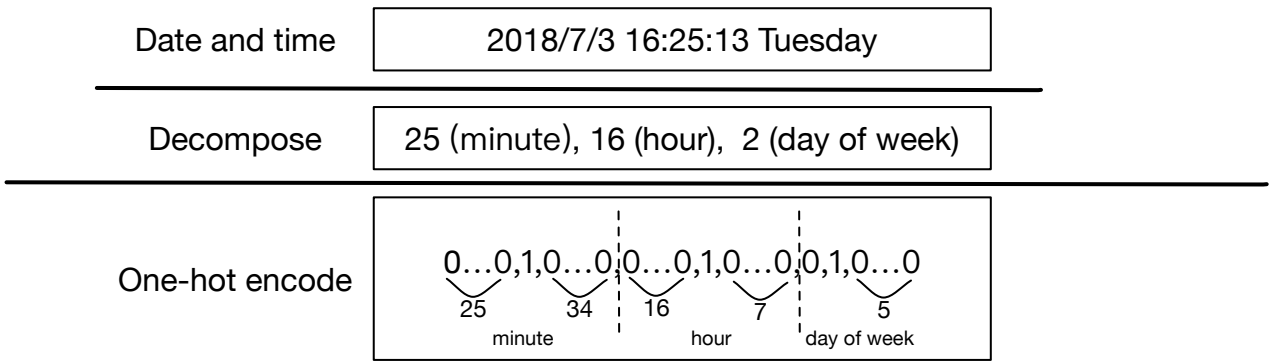
More robust algorithm is needed

Figure: Donut

Figure: Bagel, more healthy

Core Idea

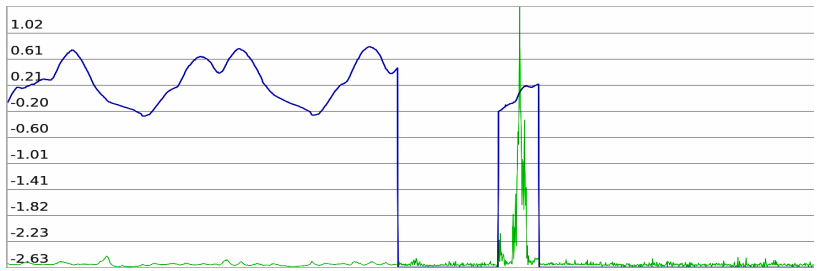
1. use additional time information to help reconstruct normal patterns.
2. encode time information appropriately



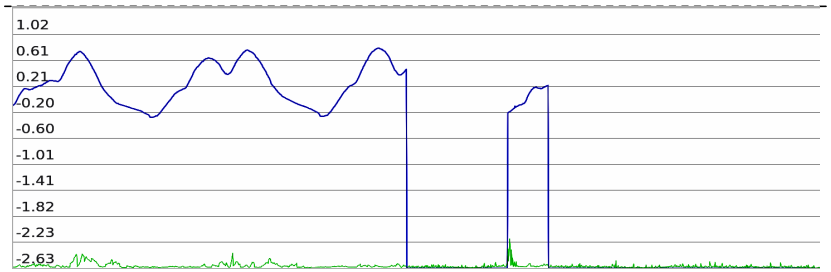
3. make sure that both window shape and time information work well.
   $\Rightarrow$ use dropout layer to avoid overfitting

Effect of the improvements

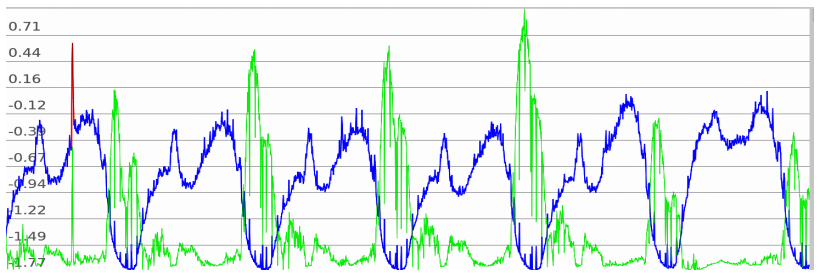
Donut



Bagel



Donut



Bagel

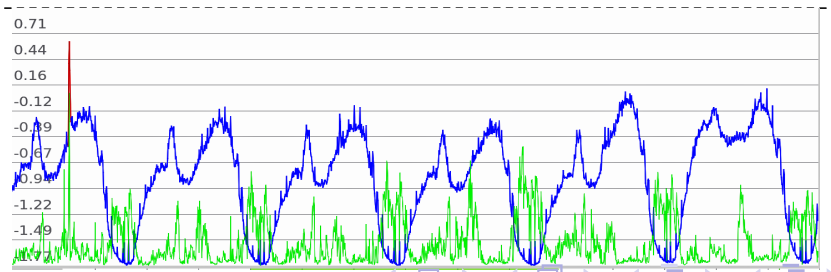


Table of Contents

Overall architecture

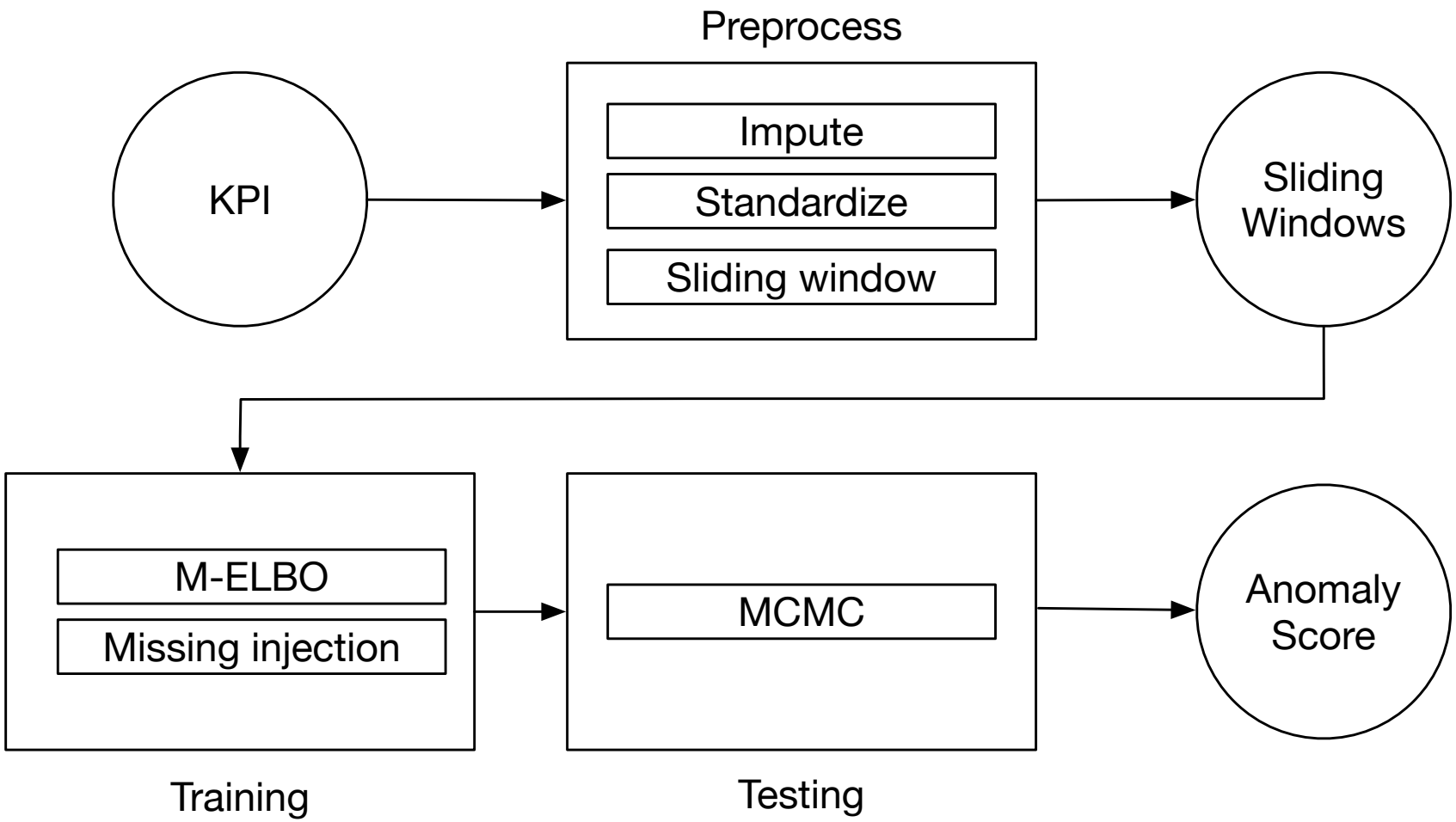


Figure: Overall architecture

Training (1/4)

Preprocessing:

1. Imputing missing points.
2. Standardization for points in each KPI.
3. Sliding window with window length W.

Network structure:

conditional variational autoencoder [7], as shown in Fig. 10.

Training (2/4)

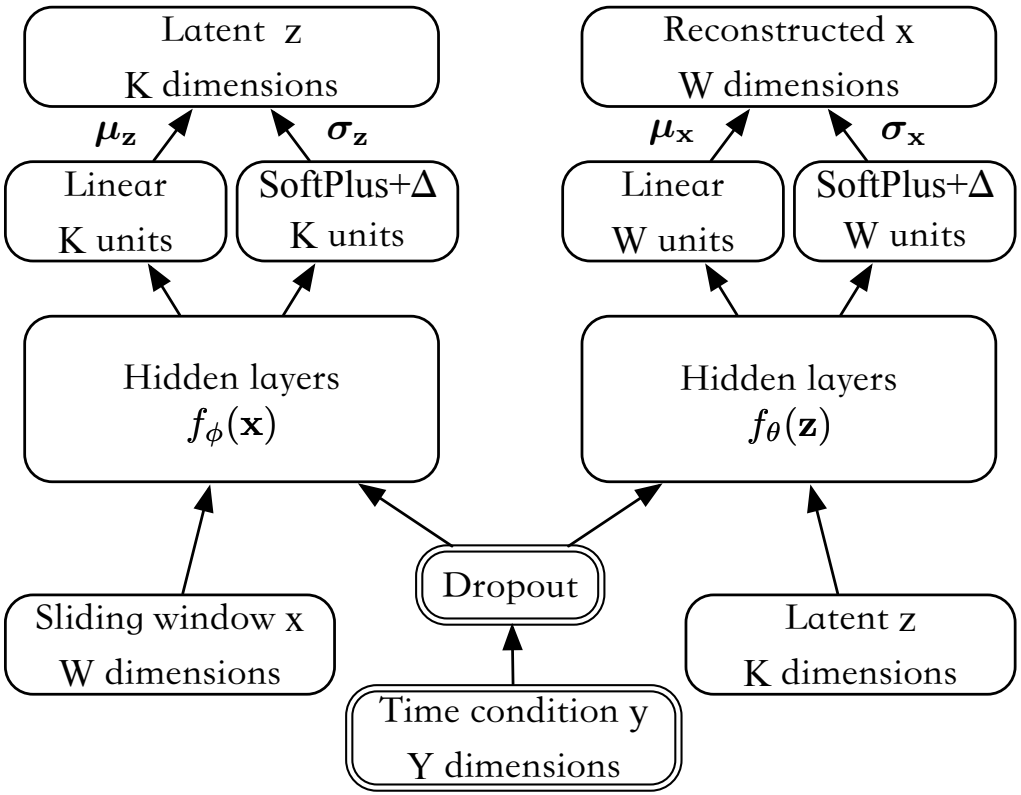
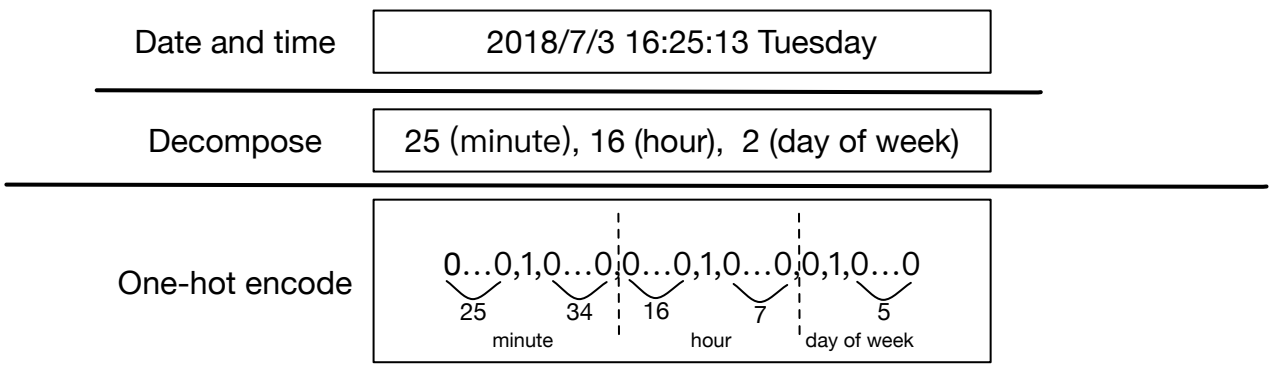


Figure: The overall neural network architecture. The double-lines highlight the major difference with *Donut* [6] in network architecture.

Training (3/4)

Encoding time information (y in Fig. 10):

1. Get the date and time of each window $\mathbf{X}$.
2. Decompose it into useful components.
3. One-hot encode and concatenate.



Training (4/4)

Training objective (M-ELBO [6]):

$$\tilde{\mathcal{L}}(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\sum_{i=1}^{W} \boldsymbol{\alpha}_i \cdot \log p(\mathbf{x}_i|\mathbf{z}, \mathbf{y}) + \beta \cdot \log p(\mathbf{z}|\mathbf{y}) - \log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}))] \tag{1}$$

- $\boldsymbol{\alpha}$: a binary vector, denotes the corresponding anomaly labels of a window $\mathbf{x}$.
- β: the proportion of normal points in a window $\mathbf{x}$

Detection (1/1)

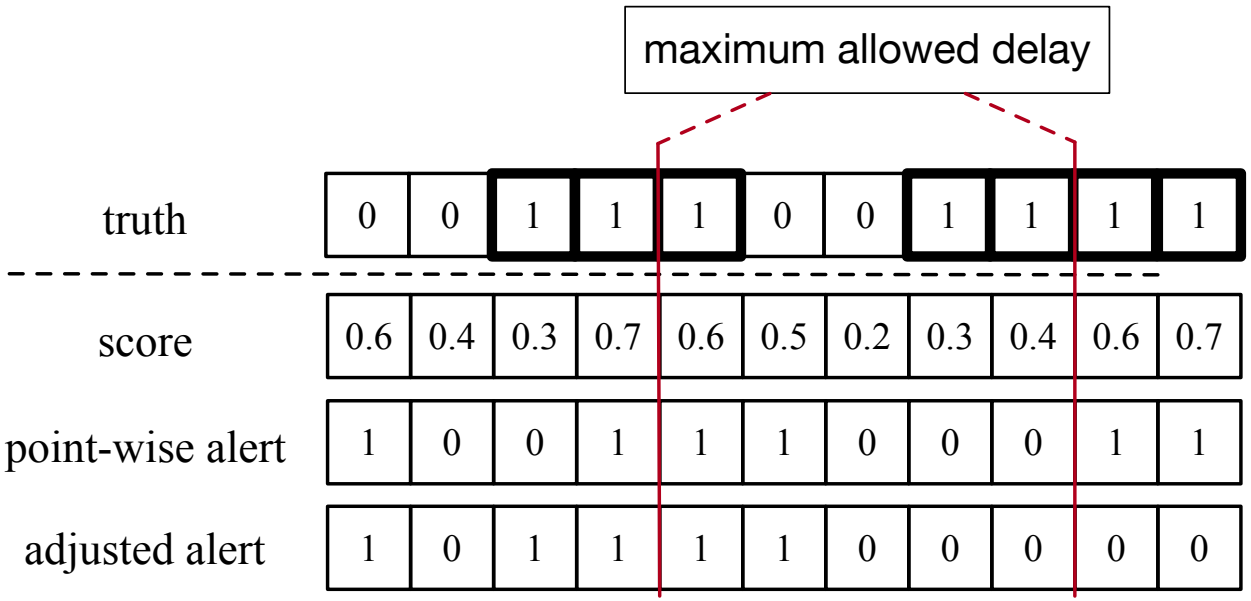
We use **negative reconstruction probability** as the anomaly detector.

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

[6] gives a KDE (kernel density estimation) for it and explain why it is suitable for anomaly detection problem.

Table of Contents

Evaluation Metric



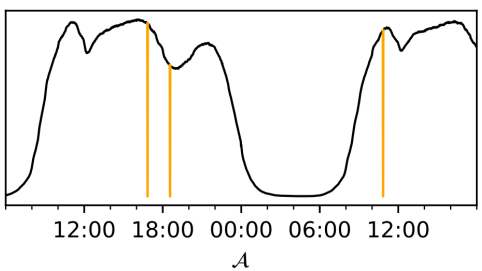
We use F1-score based on the adjusted alerts as the evaluation metric.

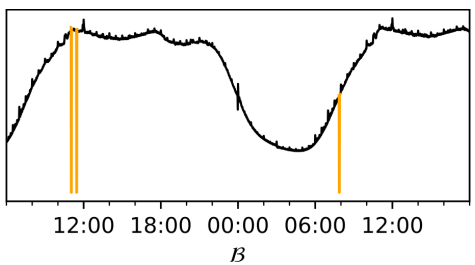
Datasets (1/2)

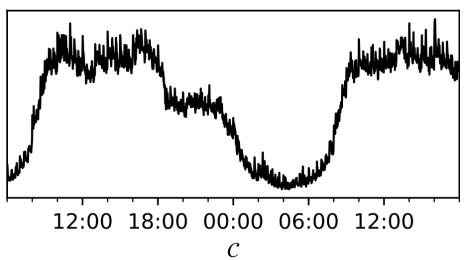
We obtain several well-maintained KPIs from several large Internet companies.

All the anomaly labels are manually confirmed by operators.

- $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are similar to those in [6], so they can demonstrate *Bagel*'s performance on those KPIs that *Donut* claims to handle well. Bagel should have similar performance with *Donut* on them.

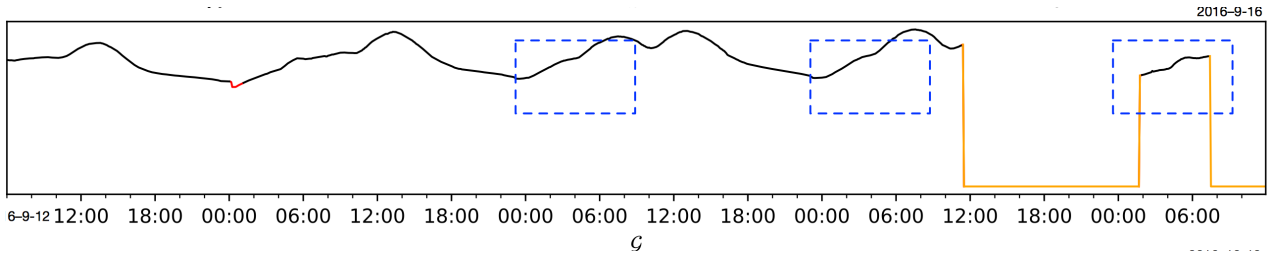




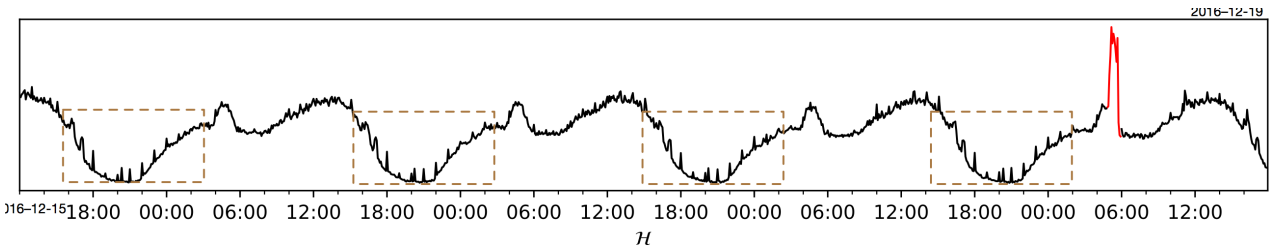


Datasets (2/2)

- $\mathcal{G}$ has many missing points and several long missing fragments (like that shown in item 2, and there are several similar long missing fragments), such that many normal fragments are just small pieces surrounded by missing points.



- $\mathcal{H}$ is quite smooth, but has many periodic spikes every day.



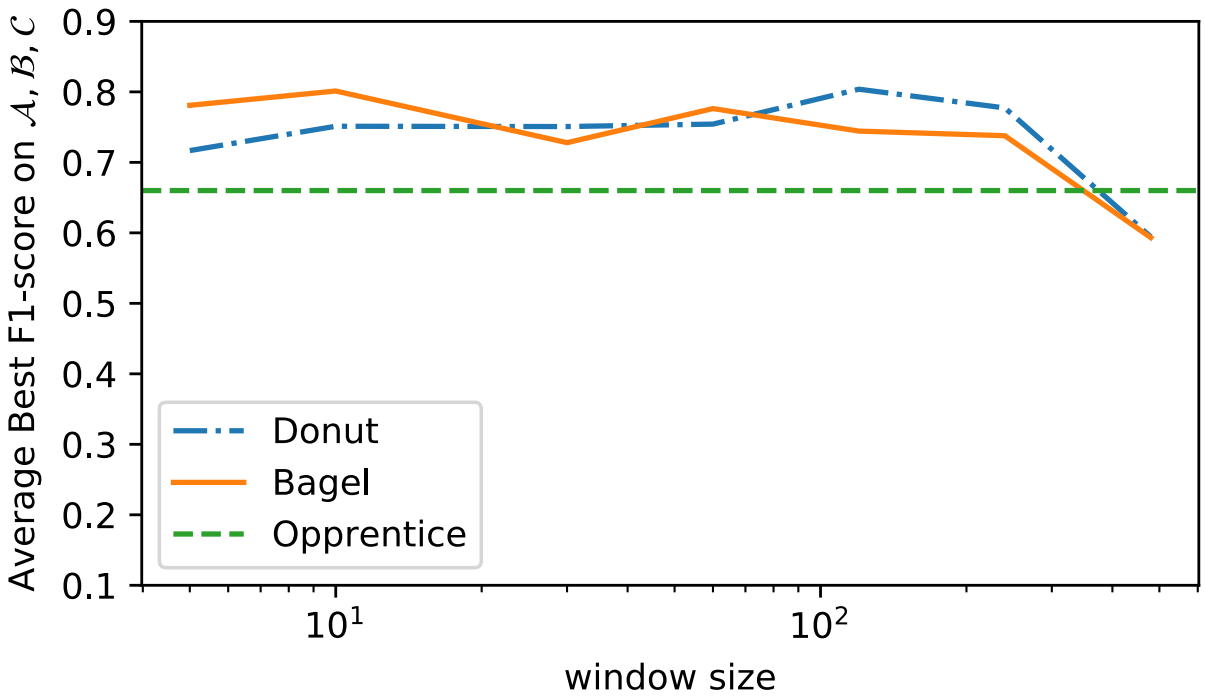
Bagel should significantly outperform *Donut* on them.

Overall Performance on $\mathcal{A}, \mathcal{B}, \mathcal{C}$ (1/2)

We compare *Bagel*'s performance with that of *Donut* and Opprentice.

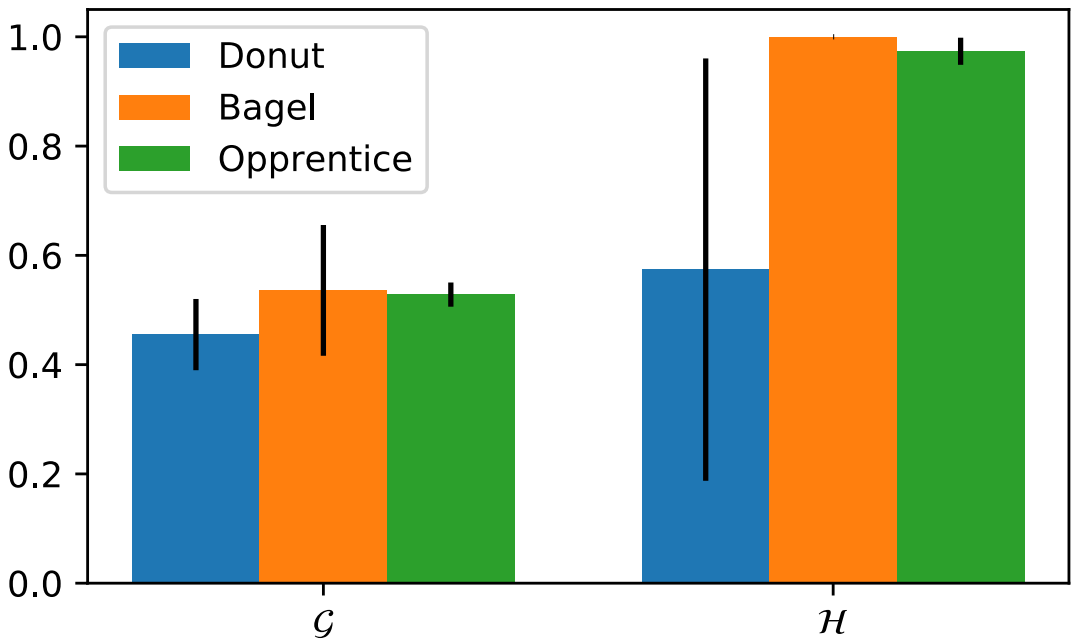
- *Donut*: a state-of-art unsupervised KPI anomaly detection algorithm based on VAE [6].
- Opprentice: a state-of-art supervised ensemble KPI anomaly detection algorithms [2].

Overall Performance on $\mathcal{A}, \mathcal{B}, \mathcal{C}$ (2/2)



On datasets $\mathcal{A}, \mathcal{B}, \mathcal{C}$, *Bagel*'s performance is similar to that of *Donut*'s, which means *Bagel* is also able to handle those KPIs that *Donut* is able to handle.

Overall Performance on $\mathcal{G}, \mathcal{H}$



Bagel significantly outperforms *Donut*, and also outperform Opprentice.

Table of Contents

Conditional KDE explanation (1/2)

Two questions:

1. We use negative reconstruction probability ($-\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{y})]$) as the anomaly detector, but why can it be an effective anomaly detector?

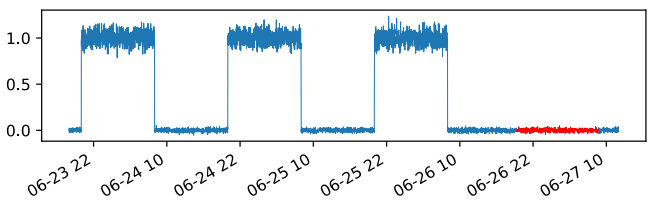
   The answer is almost the same as that of [6].
   1) M-ELBO and the dimension reduction in CVAE makes it able to reconstruct normal patterns from a potential abnormal window.
   2) Reconstruction probability can be considered as a KDE (kernel density estimation). $\log q_\theta(\mathbf{x}|\mathbf{z},\mathbf{y})$ is kernel, and $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ is the weight of kernel.

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{y})] = \sum_{\mathbf{z}^{(i)}} q_\phi(\mathbf{z}^{(i)}|\mathbf{x},\mathbf{y}) \log p_\theta(\mathbf{x}|\mathbf{z}^{(i)},\mathbf{y})$$

Conditional KDE explanation (2/2)

2 Why does time information help?

1) Given a KPI window, its corresponding normal patterns is multimodal.



2) Time information also helps when $\mathbf{x}$ is confusing.

e.g.: in $\mathcal{G}$, there is a normal fragment surrounded by missing points.

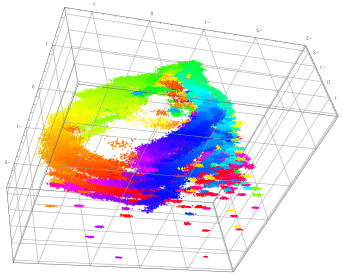
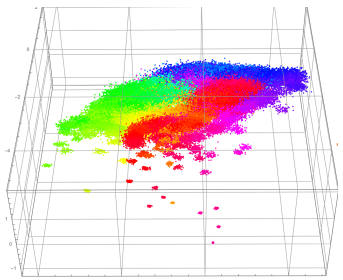
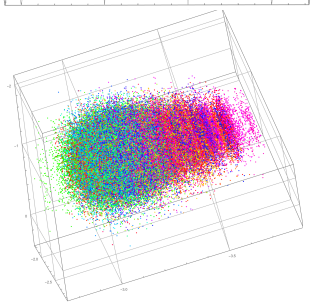
As this normal fragment is much shorter than the training windows used to train the model, *Donut* cannot determine its normal pattern and, therefore, gives wrong anomaly scores.

Dropout for avoiding overfitting on time information (1/3)

Modeling the relationship between latent variables ($\mathbf{z}$) and encoded timestamps ($\mathbf{y}$) is easier than that between latent variables ($\mathbf{z}$) and sliding windows ($\mathbf{x}$), because the KPIs are mostly seasonal and the local variation is not so influential compared to the periodicity.
Therefore CVAE model may be **overfitted on time information easily**.

Time gradient effect: "$\mathbf{z}$ samples drawn from approximated $\mathbf{z}$ posterior $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ with more different $\mathbf{y}$ should be far away from each other".
It is important to find a good $\mathbf{z}$ posterior according to the analysis in [6].

Dropout for avoiding overfitting on time information (2/3)

	Latent Space	Best F1-score
Bagel		0.686
Bagel without Dropout		0.605
Time Only		0.074

Dropout for avoiding overfitting on time information (3/3)

Since the latent spaces of *Bagel* have significant time gradient, similar to that in *Donut* [6], *Bagel* has **similar ability with Donut to reconstruct normal patterns from** $\mathbf{x}$, but *Bagel* has timing information successfully incorporated without overfitting.

Table of Contents

Conclusion

- For the first time in the literature, we **identify the importance of time information for non-sequential deep generative models**, such as *Donut*, in KPI anomaly detection problem.
- To the best of our knowledge, *Bagel* is the first to apply **conditional variational autoencoder** (CVAE) to KPI anomaly detection and use dropout technique to successfully avoid overfitting.
- Our experiments using real data from Internet companies show that, compared to *Donut*, *Bagel* improves the anomaly detection best F1-score by **0.08 to 0.43** for KPIs $\mathcal{G}$ and $\mathcal{H}$, greatly improving *Donut*'s robustness against time information related anomalies.


Y. Chen, R. Mahajan, B. Sridharan, and Z.-L. Zhang, "A provider-side view of web search response time," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 243–254.


D. Liu, Y. Zhao, H. Xu, Y. Sun, D. Pei, J. Luo, X. Jing, and M. Feng, "Opprentice: Towards practical and automatic anomaly detection through machine learning," in *Proceedings of the 2015 Internet Measurement Conference*. ACM, 2015, pp. 211–224.


M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*. ACM, 2013, pp. 8–15.


M. Sölch, J. Bayer, M. Ludersdorfer, and P. van der Smagt, "Variational inference for on-line anomaly detection in high-dimensional time series," *stat*, vol. 1050, p. 23, 2016.


J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, pp. 1–18, 2015.


H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 187–196.


K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in

Advances in Neural Information Processing Systems, 2015, pp. 3483–3491.

Thank You