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

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

\textsuperscript{1}Tsinghua University
\textsuperscript{2}Alibaba Group

April 26, 2018
Outline

1. Background
2. Architecture
3. Evaluation
4. Analysis
5. Conclusion
Outline

1. Background
2. Architecture
3. Evaluation
4. Analysis
5. Conclusion
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 Artificial Intelligence for IT Operations (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

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 $x_{t-W+1}, \ldots, x_{t-1}$, 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., $p(y_t = 1|x_{t-W+1}, \ldots, x_t)$, leaving the final decision of triggering alerts to the operators.
Outline

1. Background
2. Architecture
3. Evaluation
4. Analysis
5. Conclusion
Overall Architecture of Donut

Data Preparation
- Standardization
- Fill Missing with Zero
- Sliding Window

Training
- Modified ELBO
- Missing Data Injection

Testing $x$

Detection
- MCMC Imputation
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:**  \( \hat{x}_t = \frac{x_t - \mu_x}{\sigma_x} \).
  
  \( x_t \) are the original KPI values, \( \mu_x \) and \( \sigma_x \) are the mean and std of \( x_t \).

- **Sliding Window:**
  
  We split the KPIs into fixed-length *sliding windows* \( x_t \), which are assumed to be *i.i.d.*, and are used as the input \( x \) of VAE at every time \( t \). For simplicity, we shall omit the subscript \( t \), using \( x \) to denote the window of “current time”, and \( x_1, \ldots, x_W \) to denote each point in \( x \) afterwards.
Network Structure

(a) VAE General Structure

(b) $q_\phi(z|x)$ of Donut

(c) $p_\theta(x|z)$ of Donut

- Variational net: $q_\phi(z|x) = \mathcal{N}(\mu_z, \sigma_z^2 I)$.
- Generative net: $p_\theta(z) = \mathcal{N}(0, I)$, $p_\theta(x|z) = \mathcal{N}(\mu_x, \sigma_x^2 I)$.
- SoftPlus Trick: $\sigma_z = \text{SoftPlus}[W_{\sigma z}^T f_\phi(x) + b_{\sigma z}] + \epsilon$, $\text{SoftPlus}[a] = \log[\exp(a) + 1]$. Similar for $\sigma_x$. 
We use three techniques to handle such “historical anomalies”.

1. **M-ELBO**: We modify the ELBO (objective function) of VAE into **M-ELBO** $\tilde{\mathcal{L}}(x)$:

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

   where $\alpha_w$ and $\beta$ are hyperparameters.

2. **Missing Data Injection**: We randomly set 1% points to be missing at **every epoch** in training, to compensate for having no anomaly labels in the unsupervised scenario.

3. **MCMC Imputation** (Rezende et al., 2014): In detection, we adopt this technique on known missing points.

**Figure**: The anomaly at $t_3$ shall affect $t_4$ and $t_5$, potentially causing trouble in training and detection.
**Illustration of MCMC Imputation**

Figure: Illustration of one iteration in MCMC imputation. MCMC imputation works by using a trained deep generative model to iteratively approach the marginal distribution $p(x_{\text{missing}}|x_{\text{observed}})$.

$$x = (x_o, x_m) \xrightarrow{q_\phi(z|x)} z \xrightarrow{p_\theta(x|z)} (x'_o, x'_m) \rightarrow (x_o, x'_m) = x'$$
The Anomaly Score

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

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

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** (2) for the last point \(x_{W}\) in \(x\), as the anomaly score for the time being:

\[
s(y_{W} = 1) = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x_{W}|z)] \approx \frac{1}{L} \sum_{l=1}^{L} \log p_{\theta}(x_{W}|z^{(l)}), \ z^{(l)} \sim q_{\phi}(z|x)
\]

\(^1\)An and Cho (2015) uses vanilla VAE, without developing techniques like ours to improve performance. We shall compare Donut against their vanilla VAE in evaluation.
Outline

1. Background

2. Architecture

3. Evaluation

4. Analysis

5. Conclusion
Overall Performance

![Graphs showing AUC, Best F-Score, and Alert Delay for different settings and models.](image)

**A**
- AUC
- Best F-Score
- Alert Delay

**B**
- AUC
- Best F-Score
- Alert Delay

**C**
- AUC
- Best F-Score
- Alert Delay

*Graphs represent the performance of different models (Opprentice, VAE Baseline, Donut, Donut-Prior) across different settings (0%, 10%, 100%).*
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

The best performance could be achieved with fairly small $K$.

1. The performance does not drop too heavily for $K$ up to 21.

2. Smoother KPIs seem to demand larger $K$.

The essential of **Dimension reduction**: $W$ (the dimension of $x$) is 120, while the best $K$ (the dimension of $z$) is no larger than 10.

It should be quite easy to empirically choose $K$.

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

1. Background
2. Architecture
3. Evaluation
4. Analysis
5. Conclusion
Reconstruction Probability isn’t a Well-Defined Probability

The reconstruction probability, defined by:

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

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

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

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

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

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

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

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

April 26, 2018 19 / 28
The KDE Interpretation

Figure: Illustration of the KDE interpretation. For a given \( x \) potentially with anomalies, Donut tries to recognize what normal pattern it follows, encoded as \( q_\phi(z|x) \). The black ellipse in the middle figure denotes the 3-\( \sigma_z \) region of \( q_\phi(z|x) \). \( L \) samples of \( z \) are then taken from \( q_\phi(z|x) \), denoted as the crosses in the middle figure. Each \( z \) is associated with a density estimator kernel \( \log p_\theta(x|z) \). The blue curves in the right two figures are \( \mu_x \) of each kernel, while the surrounding stripes are \( \sigma_x \). Finally, the values of \( \log p_\theta(x|z) \) are computed from each kernel, and further averaged together as the reconstruction probability.
All the following techniques work by improving the ability of *Donut* to find “good” posteriors\(^2\) for abnormal \(x\):

- **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 \(x\).
- **Missing Data Injection**: Amplifies the effect of *M-ELBO*.
- **MCMC Imputation**: Alleviate the biases brought by missing points, helping *Donut* to find good posteriors.

\(^2\)“good” implies beneficial for the anomaly detection task.
Figure: MCMC visualization. A normal $x$ is chosen, whose posterior $q_\phi(z|x)$ is plotted at right: the cross denotes $\mu_z$ and the ellipse denotes its $3-\sigma_z$ region. We randomly set 15% $x$ points as missing, to obtain the abnormal $x'$. We run MCMC over $x'$ with 10 iterations. At first, the $z$ sample of $x'$ is far from $q_\phi(z|x)$. After that, $z$ samples quickly approach $q_\phi(z|x)$, and begin to move around $q_\phi(z|x)$ after only 3 iterations.
The Causes of Time Gradient

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

\[ = \mathbb{E} \left[ \log p_{\theta}(\mathbf{x}|\mathbf{z}) \right] + \mathbb{E} \left[ \log p_{\theta}(\mathbf{z}) \right] + H[\mathbf{z}|\mathbf{x}] \]

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

**Figure:** Evolution of $q_\phi(z|x)$ 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

1. Background
2. Architecture
3. Evaluation
4. Analysis
5. Conclusion
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.
Full slide: https://github.com/korepwx/donut/tree/slide.
Thank you

Q & A

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Unsupervised Anomaly Detection via Variational