Jump-Starting Multivariate Time Series
Anomaly Detection for Online Service Systems

Minghua Ma, Shenglin Zhang, Junjie Chen, Jun Xu, Dan Pei, et. al.
Service Reliability is Important

Users

Operators

Companies
Real-World Revenue Loss

A study of 584 U.S. based data center professionals found that 91% of data centers have experienced an unplanned data center outage in the past 24 months.²

[Evolven: GAD COHEN]
Univariate Time Series (UTS) Anomaly Detection

Building anomaly detectors for a univariate time series
Univariate Time Series (UTS)  
Anomaly Detection

Building anomaly detectors for a single time series

Not feasible for thousands of monitoring time series
Univariate Time Series (UTS) Anomaly Detection

Building anomaly detectors for a single time series

Not feasible for thousands of monitoring time series

May lead to alert storms [SEIP20]
Multivariate Time Series (MTS) Anomaly Detection

Capture status of the overall service system

Intuitive & effective & efficient

[KDD18, KDD19, KDD20, KDD21, AAAI19, AAAI21, NeurIPS20]
Multivariate Time Series (MTS) Anomaly Detection

Capture status of the overall service system

Intuitive & effective & efficient

[KDD18, KDD19, KDD20, KDD21, AAAI19, AAAI21, NeurIPS20]

Deep learning based approaches (LSTM, LSTM-VAE, ConvLSTM...)

Initialization Time

Software change (concept drift) -> Anomaly detection -> Initialize

A service is deployed or updated, and anomaly detection approach is launched

The anomaly detection approach becomes effective

Initialization Time
Deep Learning Based Approaches: Long Initialization Time

A service is deployed or updated, and anomaly detection approach is launched

The anomaly detection approach becomes effective

Initialization Time

Offline Training

⚠ Accumulating training data
⚠ Training process

Service Systems

Multivariate Time Series

Anomaly Detection

Time

\[ t_1 \]

\[ t_2 \]
Deep Learning Based Approaches: Long Initialization Time

<table>
<thead>
<tr>
<th>Approach</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>Avg.</th>
<th>Days!</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSCRED [AAAI19]</td>
<td>7</td>
<td>13</td>
<td>-</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>OmniAnomaly [KDD19]</td>
<td>17</td>
<td>15</td>
<td>17</td>
<td>16.3</td>
<td></td>
</tr>
<tr>
<td>LSTM-NDT [KDD18]</td>
<td>69</td>
<td>36</td>
<td>-</td>
<td>52.5</td>
<td></td>
</tr>
<tr>
<td>Donut* [WWW18]</td>
<td>102</td>
<td>110</td>
<td>99</td>
<td>103.6</td>
<td></td>
</tr>
</tbody>
</table>

* denotes UTS anomaly detector, which can be used for MTS by combining it with majority vote

Inappropriate for newly deployed or updated systems
Incremental Retraining

Offline

Original

Training

Test

Time

Incremental Retraining

[IMC15]

Training

Test

Time

Training

Test

Time

Training

Test

Time

(For a fair comparison)
Incremental Retraining Cannot Ensure Satisfactory Performance

Non-robustness and considerable training cost
Outline

The drawback of deep learning based approaches

⇒ Long initialization time

Our key idea of compressed sensing and its challenges

JumpStarter approach

Evaluation
Key Idea: Compressed Sensing (CS)

• CS can reconstruct time series with low energy components.
• Anomalies are always high energy components.
• CS uses a fixed-length window to initialize.

First attempt to use CS for multivariate time series anomaly detection
Two Strawman Solutions Using CS

Examples of CS-based anomaly detection when the MTS is reconstructed as a whole matrix (a) or as separate UTS (b)

(a) Inaccurate reconstruction leads to many false alarms

(b) Low efficiency, cannot capture the complex relationships
Problem of Random Gaussian Sampling

• The sampled matrix: guarantee Restricted Isometry Property (RIP)

Sampling from anomalies can degrade the detection performance
Jump-Starter

Jump-Starting Multivariate Time Series

Anomaly Detection

for Online Service Systems
JumpStarter Overview

Online Processing

- Multivariate Time Series
- Anomalies

Multivariate Processing

- Offline Processing
- Shape-Based Clustering

- Online Processing
- Sliding Window
- Anomaly Score

Anomalies Threshold

- For Each Group
- Concatenate Groups

- Compressed Sensing Reconstruction
- Outlier-Resistant Sampling
- EVT Threshold
JumpStarter Overview
JumpStarter Overview

- **Multivariate Time Series**
- **EVT Threshold**
- **Anomalies**
- **Online Processing**
- **Offline Processing**
- **Shape-Based Clustering**
- **Sliding Window**
- **Concatenate Groups**
- **For Each Group**
- **Outlier-Resistant Sampling**
- **Anomaly Score**
- **Compressed Sensing Reconstruction**
- **Anomalies**
JumpStarter Overview

Multivariate Time Series

Offline Processing

Shape-Based Clustering

For Each Group

Sliding Window

Outlier-Resistant Sampling

Anomalies

EV T Threshold

Anomaly Score

Concatenate Groups

Compressed Sensing Reconstruction

Multivariate Time Series

Online Processing

Shape-Based Clustering

For Each Group

Sliding Window

Outlier-Resistant Sampling

Anomalies

EV T Threshold

Anomaly Score

Concatenate Groups

Compressed Sensing Reconstruction
JumpStarter Offline Processing

Multivariate Time Series

Offline Processing

Shape-Based Clustering

Online Processing

Sliding Window

For Each Group

Outlier-Resistant Sampling

Anomalies

EVT Threshold

Anomaly Score

Concatenate Groups

Compressed Sensing Reconstruction
Shape-Based Clustering

- Strawman (a) cannot deal with different shapes of time series
- Shape-based distance [sigmod15] + hierarchical clustering

![Diagram showing time series and reconstructed shapes](image-url)
Shape-Based Clustering

• Strawman (a) cannot deal with different shapes of time series
• Shape-based distance \([\text{sigmod15}]\) + hierarchical clustering

An example of clustering the MTS into three clusters

<table>
<thead>
<tr>
<th>#</th>
<th>Cluster of Univariate Time Series</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>rx-pkts-eth0, rx-bytes-eth0</td>
<td># received packets/bytes</td>
</tr>
<tr>
<td>2</td>
<td>tcp-insegs, tcp-outsegs, tx-pkts-eth0</td>
<td>TCP network metrics</td>
</tr>
<tr>
<td>3</td>
<td>cpu-ctxt, cpu-user, cpu-system, cpu-nice</td>
<td>CPU utilization metrics</td>
</tr>
</tbody>
</table>
JumpStarter Online Processing

Multivariate Time Series

Offline Processing

Shape-Based Clustering

Multivariate Time Series

Online Processing

Sliding Window

For Each Group

Outlier-Resistant Sampling

Anomalies

EVT Threshold

Anomaly Score

Concatenate Groups

Compressed Sensing Reconstruction
Outlier-Resistant Sampling

Domain-specific insights:
• Anomalies are usually outliers in an observation window.
• The value of time series has time locality.
Outlier-Resistant Sampling
Outlier-Resistant Sampling

![Graph showing Time Series and Sampling Confidence with points \( \phi_1 \) and \( \phi_2 \).]
Outlier-Resistant Sampling

(a) Initialize

(b) Sampling

(c) Results
JumpStarter Online Processing

- Multivariate Time Series
- Offline Processing
- Shape-Based Clustering
- Online Processing
- Sliding Window
- For Each Group
- Outlier-Resistant Sampling
- EVT Threshold
- Anomaly Score
- Concatenate Groups
- Compressed Sensing Reconstruction
- Anomalies

- Offline Processing
- Shape-Based Clustering
- Online Processing
- Sliding Window
- For Each Group
- Outlier-Resistant Sampling
- EVT Threshold
- Anomaly Score
- Concatenate Groups
- Compressed Sensing Reconstruction
- Anomalies
Compressed Sensing Reconstruction

• Multivariate time series: \( x_t = [x_t^1, x_t^2, ..., x_t^n]^T \)

• Compressed sensing reconstruction: \( AX'_t = B \), calculating \( x'_t \)
  • \( A \) is calculated as: \( A = \phi(D \otimes D^T) \), \( D \) is the transform of \( x_t \)
  • \( B \) is the sampling result

• Calculation: CVXPY (convex optimization tool) \cite{JMLR16}

• Anomaly score: measuring the differences between \( x_t \) and \( x'_t \)

• Choosing threshold: Extreme Value Theory (EVT) \cite{KDD17}
A learning-based approach has to *explicitly* learn the probability distribution of a multivariate time series.

Our JumpStarter: the reconstructed multivariate time series *implicitly* inherits the normal behavior.
Outline

The drawback of deep learning based approaches
  ➔ Long initialization time

Our key idea of compressed sensing and its challenges
  ➔ Reconstruction & Sampling

JumpStarter approach
  ➔ Shape-Based Clustering & Outlier-Resistant Sampling

Evaluation
  ➔ Company A (28 service systems) & Company B (30 service systems)
Evaluation: Accuracy

Average F1 Score of JumpStarter and baseline approaches
Evaluation: Efficiency

The initialization time (IT) and detection time (DT) comparison

<table>
<thead>
<tr>
<th>Approach</th>
<th>RRCF</th>
<th>LESINN</th>
<th>MSCRED</th>
<th>Omni-Anomaly</th>
<th>JumpStarter</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT (min)</td>
<td>20</td>
<td>20</td>
<td>&gt;86400</td>
<td>&gt;86400</td>
<td>20</td>
</tr>
<tr>
<td>DT (ms)</td>
<td>41.24</td>
<td>118.63</td>
<td>122.82</td>
<td>191.86</td>
<td>127.13</td>
</tr>
</tbody>
</table>
Case Study

(a) Network Issue

(b) Software Change
Conclusion

To adapt to frequent changes in online service systems, multivariate time series, anomaly detection should be robust and can be quickly initialized.

JumpStarter adopts the Compressed Sensing technique
  • Reconstruction challenge ➔ Shape-based clustering
  • Sampling challenge ➔ Outlier-resistant sampling

Evaluation
  • Real-world online service systems of two Internet companies
  • Achieving an average F1 score of 94.1%, initialization time 20 minutes
  • https://github.com/NetManAIOps/JumpStarter
Thanks

mmh16@mails.tsinghua.edu.cn